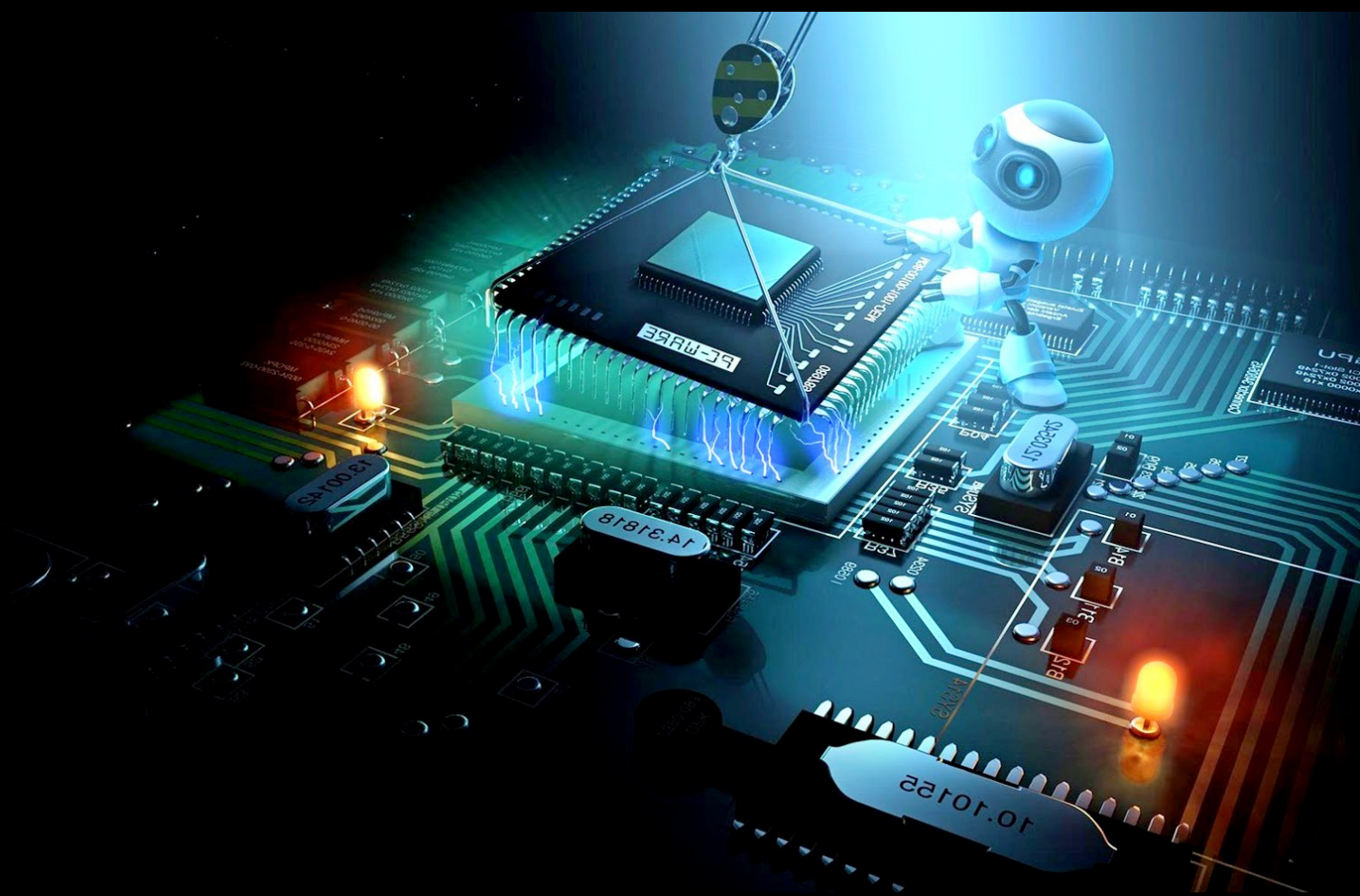


JAROSŁAW MACIEJEWSKI

TECHNICZNE ASPEKTY TWORZENIA BOTA
WIRTUALNYM ŚWIECIE
SECOND LIFE



2021

Wersja 1

Tytuł oryginalny: Techniczne aspekty tworzenia bota w wirtualnym świecie Second Life

Tytuł angielski: Technical aspects of creating a bot in the virtual world of Second Life

Oryginalna wersja znajduje się w repozytorium opracowań naukowych Zenodo.org - <https://zenodo.org/record/4729023>.

DOI 10.5281/zenodo.4729023

E-book wydany na licencji: CC BY-NC-ND 4.0

Szanujemy cudzą własność i prawo!

Zastrzeżonych nazw i znaków firm użyto w ebooku wyłącznie w celu ich identyfikacji.

Jeżeli uważasz, że praca autora jest wartościowa, wspomóż go poprzez wysłanie parę groszy na <https://ko-fi.com/nitropl>

Copyright ©Jarosław Maciejewski

Wersja 1
Polska, 2021

Spis treści

| | |
|--|-----|
| 1. Wstęp..... | 5 |
| 2. Second Life, boty..... | 6 |
| 3. Porównanie oprogramowania do tworzenia botów w SL..... | 9 |
| 4. Corrade..... | 11 |
| 4.1. Wstęp..... | 11 |
| 4.2. Licencja..... | 13 |
| 4.3. Tworzenie konta na Second Life..... | 13 |
| 4.4. Instalacja i konfiguracja Corrade..... | 17 |
| 4.4.1. Instalacja .NET Core..... | 17 |
| 4.4.2. System z rodziny Windows..... | 18 |
| 4.4.3. Systemy z rodziny Linux x64 (rodzina Debian)..... | 30 |
| 4.4.4. Mac OS..... | 31 |
| 4.5. Pliki konfiguracyjne Corrade..... | 33 |
| 4.5.1. Configuration.xml..... | 33 |
| 4.5.2. Nucleus.xml..... | 39 |
| 4.6. Instalowanie i usuwanie Corrade jako usługi systemowej..... | 40 |
| 4.6.1. Windows..... | 40 |
| 4.6.2. Linux..... | 42 |
| 4.7. Corrade.log i Openmetaverse.log..... | 44 |
| 4.8. Dodanie bota do grupy w SL, nadania mu uprawnień, wyłączenie opcji otrzymywania wiadomości grupowych..... | 45 |
| 4.9. System notacyjny uprawnień..... | 47 |
| 4.10. Powiadomienia..... | 48 |
| 4.11. Kody błędów..... | 61 |
| 4.12. Ograniczenia w Second Life..... | 79 |
| 4.13. Komunikacja z botem..... | 79 |
| 4.14. Praktyczne użycie bota na przykładzie wysłania ogłoszenia grupowego..... | 82 |
| 4.15. Integracja Corrade z CasperTech..... | 105 |
| 4.16. Odsiewanie danych..... | 109 |
| 4.17. Słaba i mocna referencja przez nazwę i UUID obiektu..... | 111 |
| 4.18. Potwierdzanie otrzymywania danych od bota w skrypcie LSL..... | 112 |
| 4.19. Rozmiar szafy bota..... | 112 |
| 4.20. Zabezpieczenie Corrade i obrona przed atakami..... | 112 |
| 4.21. Uwagi..... | 113 |
| 5. Leonardo De ArtBot..... | 114 |
| 5.1. Parametry komputera ('dom bota')..... | 114 |
| 5.2. Budowa mózgu bota i komunikacja na linii LSL - skrypt zewnętrzny..... | 115 |
| 5.3. Analiza działania bota..... | 122 |
| 5.4. Podstawy budowy bota..... | 129 |
| 5.4.1. Wstęp..... | 129 |
| 5.4.2. Proponowane funkcje dla bota..... | 139 |
| 5.4.3. Pliki wsadowe..... | 156 |
| 5.4.4. Wielozadaniowość w PHP..... | 158 |
| 5.4.5. Second Life – tworzenie primu i skryptów..... | 161 |

| | |
|---|-----|
| 6. Logiczna siatka do ustawiania obiektów (PosGrid)..... | 194 |
| 7. Najczęściej zadawane pytania..... | 207 |
| 8. Słownik..... | 215 |
| 9. Użyte obrazy..... | 217 |
| 10. Materiały dodatkowe..... | 218 |
| 10.1. Instalator bota na VPS (skrypty Powershell, Windows)..... | 218 |
| 10.2. Aktualizacja programów w bot_installer..... | 227 |
| 10.3. mcorrade.ps1 – zarządzanie botem (Powershell, Windows)..... | 228 |
| 10.4. Opis plików w bot_installer\corrade\bot_ai..... | 229 |
| 10.5. Harmonogram Zadań..... | 235 |

1. Wstęp

Second Life jest wirtualnym światem, dostępnym przez Internet, gdzie wspólnie, różni ludzie spotykają się. Pomimo małej liczby aktywnych użytkowników w stosunku do zarejestrowanych kont, projekt nieprzerwanie żyje. Wśród użytkowników jak wszędzie w Internecie są boty – programy komputerowe, które mają określone zadania.

W tym ebooku znajdziesz informacje na temat Second Life, stworzenia własnego bota, ciekawe projekty i rozwiązania. Do ebooka mogą być dołączone dodatkowe materiały, co ułatwi Tobie pracę.

Ebook jest obecnie przetłumaczony na język polski i angielski, dlatego też dopasowałem niektóre rzeczy w obu dokumentach:

- domyślną walutą jest dolar amerykański (USD),
- liczby są zapisane po angielsku.

W przyszłości planowane jest najpierw wydawanie luźnych rozszerzeń do tego ebooka, a potem w kolejnej wersji połączenie ich z ebookiem w jedną całość.

Umiejętności jakie musisz mieć przy pracy z tym ebookiem:

- znajomość LSL,
- znajomość PHP,
- podstawy poruszania się po Second Life, zakładanie grupy, zarządzanie grupą, itd.

2. Second Life, boty

Second Life jest to projekt internetowy rozwijany od 2003 r. przez firmę Linden Research Inc., który pozwala na wspólne tworzenie wirtualnego świata. Może on przypominać trochę grę The Sims, jednakże różni się tym od tej gry, że to my kierujemy naszą postacią (awatara), która nie posiada żadnych umiejętności, statystyk, profesji. Second Life jest zbudowane na zasadzie siatki serwerów, na którym z każdy z nich, jest uruchomionych kilka regionów (wirtualnych ziem). Co do samych wirtualnych ziem w Second Life to według danych pobranych z portalu gridsurvey.com z dnia 2021-03-11⁰¹, całkowita ich powierzchnia wynosi ok. 1,700 km², większość ziem jest pod kontrolą prywatną i oznaczeniu 'Umiarkowany'. O ile zarejestrowanych kont w Second Life jest ok. 65 mln, to tylko aktywnych jest ok. 0.1% z nich.

| | | | | | | |
|-----------------------------|----------------|-----------------|--------------|---------------------|--------------|------------------------------------|
| Last updated: | 2021-03-11 | | | | | |
| Ownership | General | Moderate | Adult | Offline | TOTAL | Total Area (km²) |
| Linden | 1,569 | 6,463 | 406 | | 8,438 | 553 |
| Private | 1,045 | 9,547 | 6,797 | 11 | 17,400 | 1,140 |
| Total | 2,614 | 16,010 | 7,203 | 11 | 25,838 | 1,693 |
| Linden [%] | 60.02% | 40.37% | 5.64% | 0.00% | 32.66% | 32.66% |
| Private [%] | 39.98% | 59.63% | 94.36% | 100.00% | 67.34% | 67.34% |
| Total residents | 64,687,961 | | | General [%] | 10.12% | |
| Residents online | 42,328 | | | Moderate [%] | 61.96% | |
| Residents online [%] | 0.07% | | | Adult [%] | 27.88% | |

Dane na temat ziem i rezydentów z Second Life

Aby dostać się do Second Life, najpierw trzeba założyć konto na stronie⁰² oraz ściągnąć i zainstalować program komputerowy. Można tutaj zainstalować oficjalny program⁰³ albo któryś wspierany firm trzecich, np. Firestorm Viewer⁰⁴.

Second Life nie ma narzuconego z góry klimatu świata, pomimo, że większość lokacji przypomina swoim wyglądem te znane z prawdziwego życia, to można jednak znaleźć lokalizacje utrzymane w klimacie średniowiecza, fantasy, starożytności czy sci-fi.

01 Strona: <http://www.gridsurvey.com/>

02 Strona: <https://join.secondlife.com/?lang=en-US>

03 Instalator Second Life: <https://secondlife.com/support/downloads/>

04 Instalator Firestorm Viewer: <https://www.firestormviewer.org/>

Po zalogowaniu się do Second Life, najlepiej najpierw rozpocząć od konfiguracji samego programu (skrót klawiszowy CTRL + P), aby ustawić sobie program pod siebie, a potem zapoznać się z interfejsem programu, np. jak chodzić, latać, budować, rozmawiać na czacie i głosowo, itd.

Jako wirtualna postać możemy wchodzić w interakcję z innymi awatarami (np. poprzez rozmowę głosową lub pisaną na czacie) oraz obiektami.

Sam program zawiera w sobie proste narzędzia do budowania świata. Podstawowym materiałem jest prim (nazywany potocznie 'klockiem'), natomiast bardziej zaawansowane struktury mogą być tworzone i zaimportowane z Blendera (tzw. Mesh). Prim w połączeniu z innymi primami i mesh'ami tworzy obiekt. Takich połączeń z jednym primem lub mesh'em może być maksymalnie 255. Aby ożywić prim, mesh czy obiekt, należy podłączyć pod niego skrypt LSL, który jest kombinacją języków typu C++, C#, Java i który jest kompilowany przez mono.

Oczywiście mając do dyspozycji narzędzie do budowy świata, podstawową znajomość języka LSL nie oznacza, że Second Life daje nieograniczone możliwości. Każdy region ma ograniczoną liczbę przechowywania primów, zazwyczaj jest to maksymalnie 15,000 lub 30,000 primów. Do tego można dołączyć maksymalny rozmiar jednego skryptu, które wynosi 64 KB oraz maksymalną liczbę awatarów w całym regionie w ilości 100 osób.

Bot (inne nazwy to na przykład: web robot, robot internetowy, agent, oskryptowany agent, NPC (ang. Non-Player Character - pol. postać nie będąca graczem), softbot) jest programem komputerowym, który wykonuje zadania zazwyczaj szybciej i prościej niż człowiek, nie posiada fizycznego ciała, może mieć wirtualne ciało, emuluje aktywność człowieka w Internecie, np. konwersację z drugim człowiekiem.

W Second Life boty można podzielić na dwa typy:

- primowo-meshowe - bot jest stworzony przy pomocy primów, obiektów i meshy oraz oskryptowany tylko przy użyciu LSL,
- awatary - bot jest uruchamiany jako zwykły awatar z zewnętrznego komputera, komunikacja odbywa się przez interfejs programistyczny, który może być mieszany z LSL.

Polityka Second Life odnośnie botów jest przejrzysta. Bot w zasadzie może robić wszystko to co robią normalni ludzie pod postacią awatarów, jednakże nie może generować zbyt dużego, sztucznego ruchu na działce, kopiować rzeczy bez patrzenia się na prawa autorskie, wysyłania więcej niż 5000 wiadomości na dzień, grupowe wiadomości są liczone jako 1 wiadomość na 1

odbiorcę, kupować region MainLand, innych rzeczy wymienionych w Warunkach Usługi, więc w zasadzie bot może robić wszystko w myśl zasady: 'co nie jest zabronione jest dozwolone'.

Boty w Second Life zajmują się różnymi, przypisanymi do nich zadaniami, np. mogą symulować ruch powietrzny (ATC – Air Traffic Control), zarządzać regionem (manger, ochroniarz), zarządzać grupą, pracować jako hostessy, model na wybiegu pokazujący ubrania do sprzedania, grać jako aktor w minigrze, itd.

3. Porównanie oprogramowania do tworzenia botów w SL

W Internecie znajdziemy kilka projektów, które pozwalają uruchomić własnego bota w Second Life. Należy się zastanowić, w jaki sposób uruchomimy naszego bota:

- Czy będzie on działał przez całą dobę, czy może przez parę godzin w określone dni,
- Czy będzie działał na naszym domowym sprzęcie czy wynajmiemy serwer,
- Czy chcemy skorzystać z darmowego lub płatnego oprogramowania.

Uruchomienie bota na własnym sprzęcie ma taką zaletę, że cały czas mamy bota i komputer pod naszą kontrolą, pod ręką w naszym domu. Na pewno musimy być gotowi na opłaty za prąd i Internet oraz na naprawy lub wymiany uszkodzonych lub zużytych podzespołów komputera w wyniku eksploatacji. Na komputer możemy przeznaczyć dowolny jakiś stary komputer o ile spełnia minimalne wymagania, można kupić używany komputer typu SFF⁰⁵, laptop, netbook.

Decydując się na uruchomienie bota na serwerze zdalnym, np. VPS, mamy zdalny dostęp do serwera i bota, musimy być przygotowani na opłatę cykliczną za wynajem serwera.

Przy VPS musimy poszukać sobie dobrej oferty firmy świadczącej wynajem serwera VPS, dlatego ja nie chcę tutaj robić analiz ofert, ponieważ bardzo szybko mogą być nieaktualne. Wystarczy wpisać w wyszukiwarce internetowej frazę 'vps linux'⁰⁶ lub 'vps windows'⁰⁷ w zależności z jakim systemem szukamy VPS.

Poniżej lista oprogramowania do uruchomienia własnego bota w Second Life:

- **SmartBots** - bot jest hostowany wyłącznie na serwerze firmy, koszt wynajęcia bota to: \$0.32/tydzień za wersję podstawową (z możliwością rozszerzenia o dodatkowe komponenty) i \$1.92/tydzień za wersję pełną. SmartBot pozwala na zapraszanie odwiedzających do grupy, wysyłanie o określonej godzinie ogłoszenia do wielu grup, monitorowania czatu grupy przez przeglądarkę internetową, dodawanie spamerów do czarnej listy, tworzenie punktów kontrolnych przez które bot będzie przechodził (przydatne np. podczas wycieczek), itd. Komunikacja z botem odbywa się przez: LSL, HTTP. SmartBots współpracuje z dość bogatą listą oprogramowania firm trzecich, np. odnośnie

05 SFF – Small Form Factory

06 Google Search z frazą 'vps linux' - <https://www.google.com/search?q=vps+linux>

07 Google Search z frazą 'vps windows' - <https://www.google.com/search?q=vps+windows>

dzierżawy (np. CasperLet, HippoRent), sprzedaży (np. CasperVend), reklamy (np. IntelliAd). Adres WWW: <https://www.mysmartbots.com/>.

- **Corrade** - oprogramowanie bota należy ściągnąć ze strony dewelopera. Bot może być hostowany na własnym komputerze lub na VPS, komunikacja z botem odbywa się przez LSL, HTTP, MQTT, WebSockets, TCP. Współpracuje z oprogramowaniem firm trzecich odnośnie wynajmu i sprzedaży, np. CasperLet, CaspetVend. Adres WWW: <https://corrade.grimore.org/>.
- **QubicBot** - oprogramowanie działające tylko pod system Windows. Pełna wersja kosztuje \$12, można też hostować na serwerze producenta, wówczas zapłacimy parę centów za tydzień użytkowania. Program pozwala na uruchamianie nielimitowanej ilości botów, wysyłanie grupowych i prywatnych wiadomości, integrację z SmartBots i PikkuBot (aby rozszerzyć swoje możliwości). Adres WWW: <http://qubicbot.com/>.

Jak widać, mamy kilka propozycji, od nas zależy, które oprogramowanie nam się spodoba.



W dalszej części książki będą używane przykłady zgodne z oprogramowaniem Corrade.

4. Corrade

4.1. Wstęp

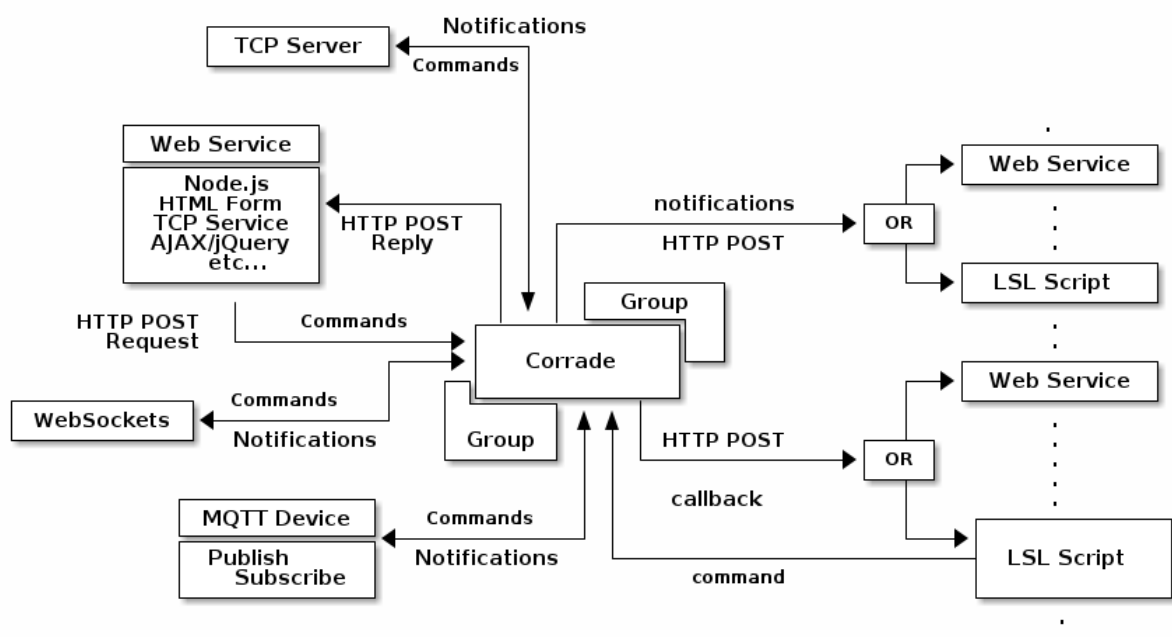


Rysunek 4.1 - Logo Corrade

Corrade:

- wielofunkcyjne, wieloplatformowe oprogramowanie do uruchomienia własnego bota w wirtualnym świecie Second Life,
- jest mostem pomiędzy użytkownikiem a Second Life,
- wspiera systemy: Linux ARM, Linux 64-bit, Mac OSX 64-bit, Windows 7 i nowsze 64-bitowe,
- kontrola przez skrypty wewnętrzne (LSL) lub/i zewnętrzne (dowolny język programowania),
- integracja z zewnętrznymi usługami poprzez wbudowane serwery:
 - HTTP,
 - WebSockets,
 - MQTT,
 - TCP.
- konfiguracja bota odbywa się w przeglądarce internetowej przez adres: <http://127.0.0.1:54377/>,
 - zaawansowana konfiguracja poprzez edycję pliku Configuration.xml i Nucleus.xml
- wymagania:

- zainstalowany jeden z wcześniej wspomnianych systemów operacyjnych,
- .NET Core 3.1+,
- min. 2 GB pamięci (zalecane 4 GB),
- stabilne i szybkie połączenie internetowe.
- strona WWW: <https://corrade.grimore.org/>



Rysunek 4.2 - Sposób przepływu informacji w Corrade

Corrade łączy się z Second Life jako normalny awatar, gdzie jak każdy awatar należy do jakiejś istniejącej grupy w tym świecie. W tej grupie ma przypisane role (jedną lub więcej) z odpowiednimi uprawnieniami.

Domyślnie polecenia można wysyłać przez skrypt LSL, który może także odebrać odpowiedzi z bota (callback) albo też ta odpowiedź może zostać przekierowana do usługi zewnętrznej, np. do skryptu na serwerze zewnętrznym.

To samo dotyczy powiadomień, które odbierają w czasie rzeczywistym odpowiedzi od bota i mogą być przetwarzane w skrypcie LSL lub przez usługę zewnętrzną.

Polecenia czy powiadomienia także mogą być wysyłane przez: serwer TCP, serwer WebSockets, serwer MQTT lub skrypt napisany w dowolnym języku programowania (np. PHP, Perl, Python, Ruby, itd.), gdzie tą samą drogą odbywa się przesłanie odpowiedzi.

Wysłanie polecenia do Corrade należy powiązać z grupą i hasłem jako danymi autoryzacyjnymi.

Na stronie projektu znajdziemy dość obszerne API do bota, a przy większości poleceń są podane przykłady użycia. Deweloper zapewnia także przykłady użycia skryptów LSL na swojej stronie sklepu w Second Life Marketplace⁰⁸ (wszystkie są za darmo do użycia i modyfikacji) oraz szablony HTTP (np. mapa 2D i 3D, czat grupowy, itd.).

4.2. Licencja

Oprogramowanie Corrade oraz powiązane z nim materiały są wydane na licencji 'Wizardry and Steamworks Project-Closed and Open-Derivatives License 1.0' (WAS PC & OD 1.0)⁰⁹, która w zasadzie nie zezwala na dokonywanie inżynierii wstecznej na programie i wymaga oznaczania Corrade w materiałach w których się pojawia.

Licencja pozwala na modyfikowanie kodów źródłowych, użycie w celach prywatnych i komercyjnych, kopiowanie, dystrybucję, sublicencję, relicencję na każdy materiał stworzoną przez tą samą firmę lub organizację w ramach licencji WAS PC & OD 1.0.

Producent zapewnia oprogramowania 'jest jak jest' bez gwarancji poprawnego działania oprogramowania.

Wsparcie techniczne jest na dobrym poziomie, w razie błędów, wystarczy napisać odpowiednie zgłoszenie na stronie WWW producenta.

Możesz wykorzystać oprogramowanie Corrade w celach komercyjnych, jednakże musisz w widoczny sposób oznaczyć producenta Corrade.

Funkcje pomocnicze, np. `wasURLEscape`, `wasCSVToList`, itd. są licencjowane według WAS PC & OD 1.0.

4.3. Tworzenie konta na Second Life

Stworzenie konta w Second Life jest bardzo proste. Wchodzimy na stronę Second Life¹⁰ (Rysunek 4.3 - Strona WWW Second Life), następnie albo klikamy na pomarańczowy przycisk

08 Second Life Marketplace - <https://marketplace.secondlife.com/stores/165275>

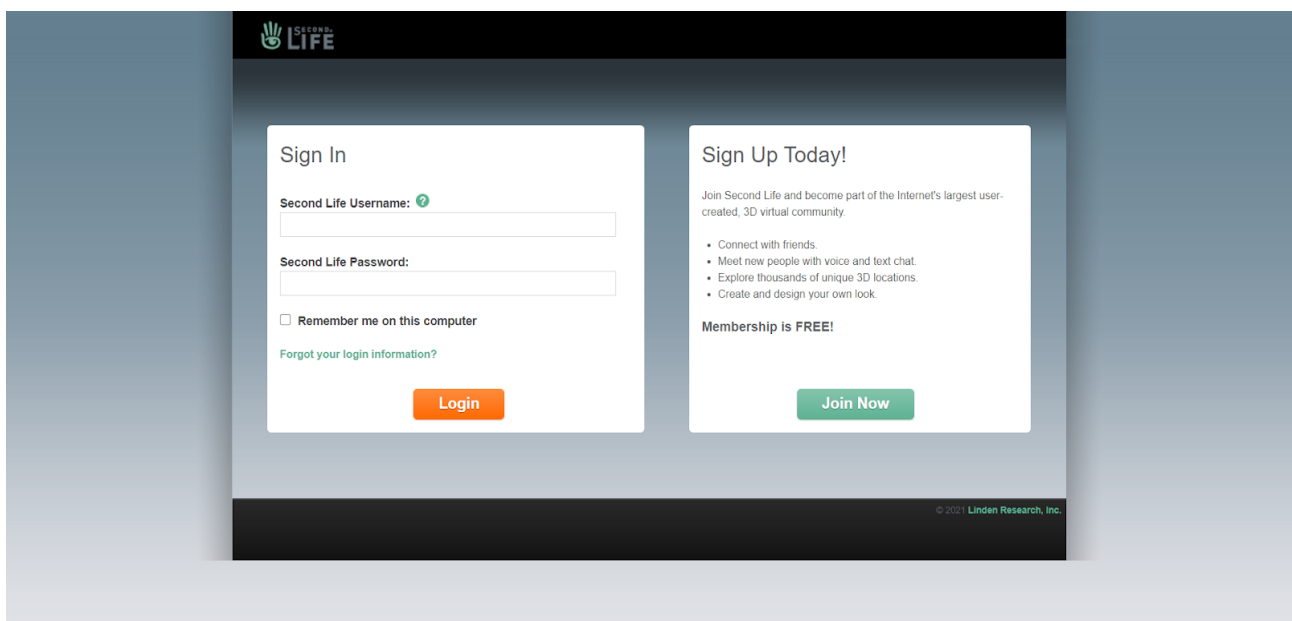
09 Licencja <https://grimore.org/licenses/was-pc-od>

10 Strona Second Life - <https://secondlife.com/>

'Join free' albo 'Sign up' po prawej górnej stronie ekranu (Rysunek 4.4 - Second Life - część do zalogowania się lub rejestracji).

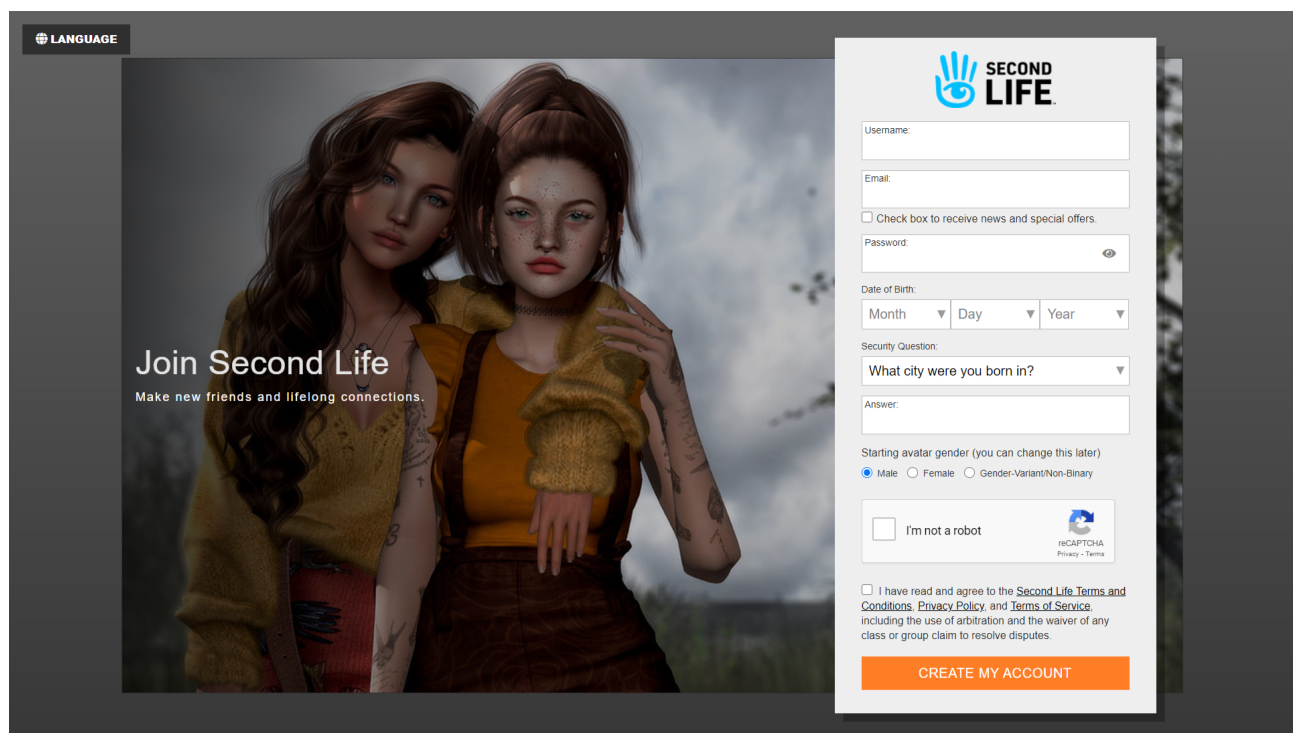


Rysunek 4.3 - Strona WWW Second Life



Rysunek 4.4 - Second Life - część do zalogowania się lub rejestracji

Jeżeli po prawej górnej stronie ekranu klikniesz na 'Login' to będziesz miał możliwość zalogowania się do swojego konta i zarządzania tym kontem.



Join Second Life
Make new friends and lifelong connections.

Username:

Email:

☐ Check box to receive news and special offers.

Password:

Date of Birth:

Month Day Year

Security Question:

What city were you born in?

Answer:

Starting avatar gender (you can change this later)

☒ Male ☐ Female ☐ Gender-Variant/Non-Binary

☐ I'm not a robot

☐ I have read and agree to the [Second Life Terms and Conditions](#), [Privacy Policy](#), and [Terms of Service](#), including the use of arbitration and the waiver of any class or group claim to resolve disputes.

CREATE MY ACCOUNT

Rysunek 4.5 - Strona do rejestracji konta

Wracając, po kliknięciu zobaczysz formularz do wypełnienia:

- Username – nazwa awatara w Second Life, może zawierać litery i cyfry, nie ma możliwości utworzenia awatara ze swoim własnym imieniem i nazwiskiem – wszystkie otrzymują nazwisko 'Resident',
- Email – podaj swój email na który zostanie zarejestrowany awatar, na tej email również będziesz otrzymywać powiadomienia z Second Life, np. zapisane wiadomości offline,
- Check box to receive news and special offers – zaznacz jeżeli chcesz otrzymywać wiadomości i specjalne oferty z Second Life,
- Password – wpisz hasło do konta swojego awatara,
- Date of Birth – twoja data urodzenia,
- Security Question – wybierz typ pytania zabezpieczającego konto,
- Answer – wpisz swoją odpowiedź na wybrane pytanie zabezpieczające,

- Starting avatar gender – wybierz jaką płeć ma mieć twój awatar: męską, żeńską, albo nieokreśloną, np. pod postacią fikcyjną,
- I'm not robot – zaznacz i w razie czego rozwiąż test, że nie zakładasz konto jako robot internetowy,
- I have read and agree (...) – zaznacz opcję jeżeli zgadzasz się z warunkami i regulaminem Second Life.

Zapewne po rejestracji należy potwierdzić chęć utworzenia konta poprzez kliknięcie w link, który zostanie przesłany na wcześniej podany email.

Po rejestracji i zatwierdzeniu możemy zalogować się do konta na stronie Second Life oraz do świata Second Life.

Zaleca się, aby na koncie zmienić status awatara (Rysunek 4.6 - Zmiana statusu awatara na stronie Second Life), jednakże nie daje to żadnych korzyści, raczej zwiększa statystykę.

The screenshot shows the 'Scripted Agent Status' page in Second Life. The left sidebar contains a list of account management options. The main content area explains the 'Scripted Agent' status and provides a form to change it. Red boxes and numbers 1-5 highlight the following elements:

- 1. Home link in the sidebar.
- 2. Account link in the sidebar.
- 3. Change button in the main content area.
- 4. Radio button for 'Yes, identify this account as a scripted agent or Bot'.
- 5. Save Changes button.

Rysunek 4.6 - Zmiana statusu awatara na stronie Second Life

4.4. Instalacja i konfiguracja Corrade

4.4.1. Instalacja .NET Core

Do uruchomienia programu Corrade potrzebujesz programu .NET Core, który ujednolica framework .NET dla wszystkich znanych systemów operacyjnych.

Na stronie pobierania .NET¹¹ (Rysunek 4.7 - Strona pobierania .NET) znajdziemy aktualnie: .NET (to mamy zainstalować), .NET Core (jest to wersja starsza z przedłużonym wsparciem programu) i .NET Framework (jeszcze starsza platforma, używana przez starsze wersje systemów Windows, np. Windows 7¹²). Nas interesuje wersja, która jest po lewej stronie ekranu - .NET. Przy tej pozycji klikamy na link 'All .NET downloads', następnie klikamy na '.NET 5.0 (recommended)' [ta wersja może być inna] jako wersję oznaczoną jako polecaną. W górnej części tabelki masz najnowszą wersję możliwą do ściągnięcia. Gdy zlokalizowałeś ostatnią wersję, przejdź do prawej kolumny tabelki o nazwie 'Run apps – Runtime'. Ściągnij program '.NET Desktop Runtime' wersji x64 dla systemu Windows lub '.NET Runtime' w wersji instalacyjnej (Installers) w wersji x64. Tutaj też mamy do dyspozycji możliwość instalacji programu na inne systemy operacyjne, tj. Linux czy MacOS na inne architektury procesora.

Ściągnięty wcześniej program dla systemu Windows lub Mac OS uruchamiamy, przechodzimy przez parę etapów instalacji i zamykamy program. Jego obsługa jest bezproblemowa.

Jeżeli w przyszłości będziemy aktualizować .NET do nowej wersji to nie zapomnijmy odinstalować wcześniej starsze wersje przez 'Dodaj lub usuń programy'.

11 Strona pobierania .NET - <https://dotnet.microsoft.com/download>

12 Ale nie musisz być instalowany, można zainstalować .NET



Rysunek 4.7 - Strona pobierania .NET

Jeżeli chodzi o system oparty o Linux, to warto najpierw sprawdzić czy możemy zainstalować .NET przez program snap, zatem warto najpierw jego zainstalować, a potem w terminalu wpisać:

```
sudo snap install dotnet-sdk --classic
```

Co spowoduje zainstalowanie środowiska SDK i Runtime dla .NET

Potem możemy normalnie zaktualizować .NET przez snap:

```
sudo snap refresh dotnet-sdk
```

Jeżeli nasz system nie pozwala na instalację programu snap, to należy wykonać standardowe kroki opisane w instrukcji na wyżej wspomnianej stronie.

4.4.2. System z rodziny Windows



W materiałach dodatkowych znajdziesz folder ze skrypcem w PowerShell i instrukcją szybkiej instalacji bota.

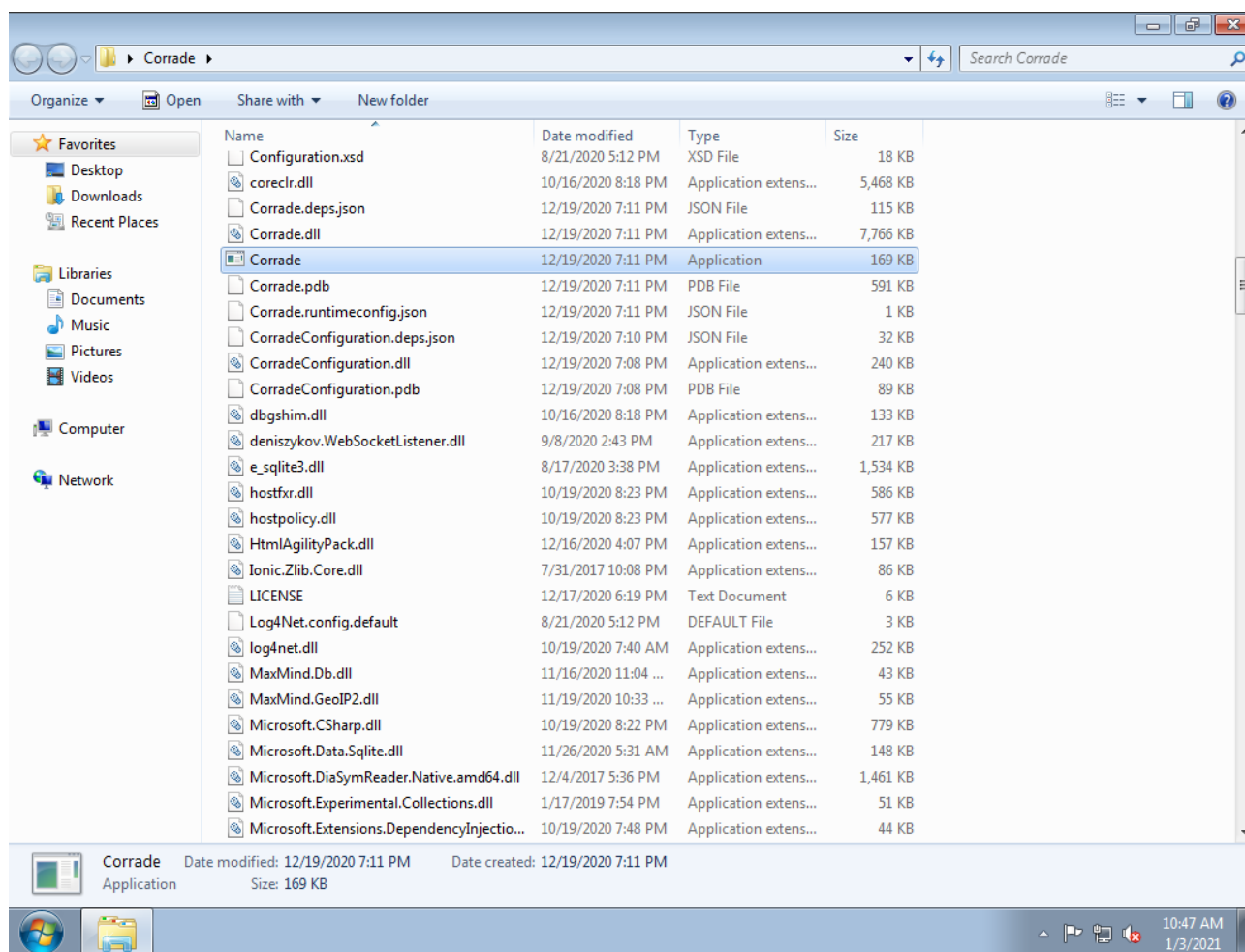
1. Wejdź na stronę <https://corrade.grimore.org/>.
2. Kliknij na przycisk DOWNLOAD.
3. Kliknij na 'corrade'.
4. Kliknij na 'win-x64'.



Szybciej się dostaniesz poprzez kliknięcie na link:

<https://corrade.grimore.org/download/corrade/win-x64/>.

5. Kliknij 2x na 'Last modified', aby lista plików była wyświetlona według czasu od najmłodszego do najstarszego.
6. Kliknij w pierwszy ZIP, który widzisz na liście: Corrade-[WERSJA]-win-x64.zip.
7. Zaczekaj aż się pobierze.
8. Rozpakuj archiwum ZIP do folderu.
9. Otwórz katalog, do którego rozpakowały się pliki.
10. Znajdź i uruchom program Corrade.exe.

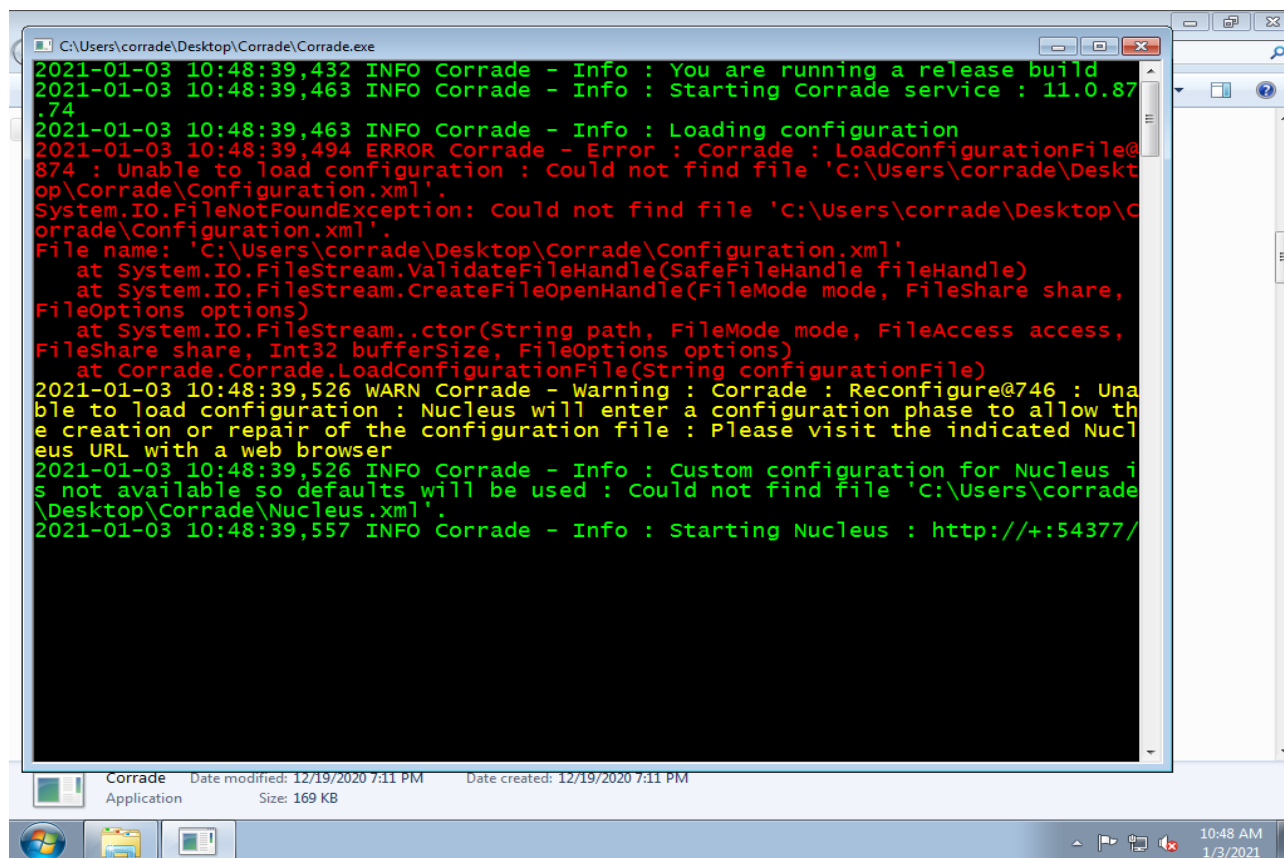


Rysunek 4.8 - Folder z plikami Corrade oraz zaznaczonym programem

Zostanie uruchomiony program w wersji konsolowej.

Będziesz widzieć na bieżąco komunikaty odnośnie działania oprogramowania Corrade.

Domyślnie jest to informacja, że skoro nie znaleziono pliku konfiguracyjnego, to program uruchomił się z domyślnymi ustawieniami oraz jest dostępny Nucleus na porcie 54377, gdzie można skonfigurować bota.



```
C:\Users\corrade\Desktop\Corrade\Corrade.exe
2021-01-03 10:48:39,432 INFO Corrade - Info : You are running a release build
2021-01-03 10:48:39,463 INFO Corrade - Info : Starting Corrade service : 11.0.87.74
2021-01-03 10:48:39,463 INFO Corrade - Info : Loading configuration
2021-01-03 10:48:39,494 ERROR Corrade - Error : Corrade : LoadConfigurationFile@874 : Unable to load configuration : Could not find file 'C:\Users\corrade\Desktop\Corrade\Configuration.xml'.
System.IO.FileNotFoundException: Could not find file 'C:\Users\corrade\Desktop\Corrade\Configuration.xml'.
File name: 'C:\Users\corrade\Desktop\Corrade\Configuration.xml'
   at System.IO.FileStream.ValidateFileHandle(SafeFileHandle fileHandle)
   at System.IO.FileStream.CreateFileOpenHandle(FileMode mode, FileShare share, FileOptions options)
   at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, FileOptions options)
   at Corrade.Corrade.LoadConfigurationFile(String configurationFile)
2021-01-03 10:48:39,526 WARN Corrade - Warning : Corrade : Reconfigure@746 : Unable to load configuration : Nucleus will enter a configuration phase to allow the creation or repair of the configuration file : Please visit the indicated Nucleus URL with a web browser
2021-01-03 10:48:39,526 INFO Corrade - Info : Custom configuration for Nucleus is not available so defaults will be used : Could not find file 'C:\Users\corrade\Desktop\Corrade\Nucleus.xml'.
2021-01-03 10:48:39,557 INFO Corrade - Info : Starting Nucleus : http://+:54377/
```

Rysunek 4.9 - Corrade w wersji konsolowej

11. Uruchamiamy przeglądarkę internetową bez dodatków blokujących reklamy.

12. Wpisujemy adres: <http://127.0.0.1:54377>.

Przed naszymi oczami powinna się pojawić ekran:



Rysunek 4.10 - Autoryzacja dostępu do Nucleus

13. W pole 'Codeword' wpisujemy domyślne hasło dostępu: 'nucleus' (bez cudzysłówów) i klikamy na przycisk 'Login'.
Pojawi się nam kolejny ekran:



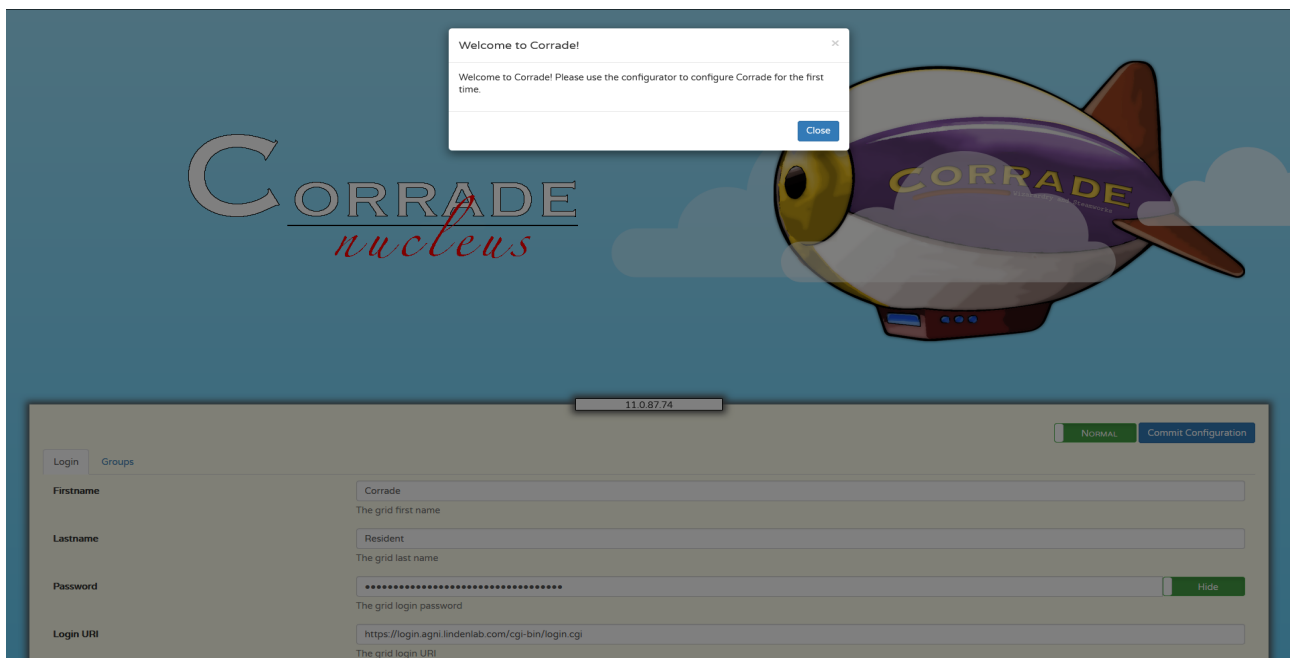
Rysunek 4.11 - Nucleus - opcje do wybrania

Tutaj możemy przejść do:

- 'Configure' - konfiguracji bota,
- 'Heartbeat' - zużycie RAM i CPU bota,
- 'View Logs' - proste przeglądanie logów bota,
- 'Logout' - wylogowanie się z Nucleus.

14. My przejdziemy do skonfigurowania naszego bota, więc klikamy na 'Configure'.

15. Wpisujemy hasło dostępu do Nucleus.



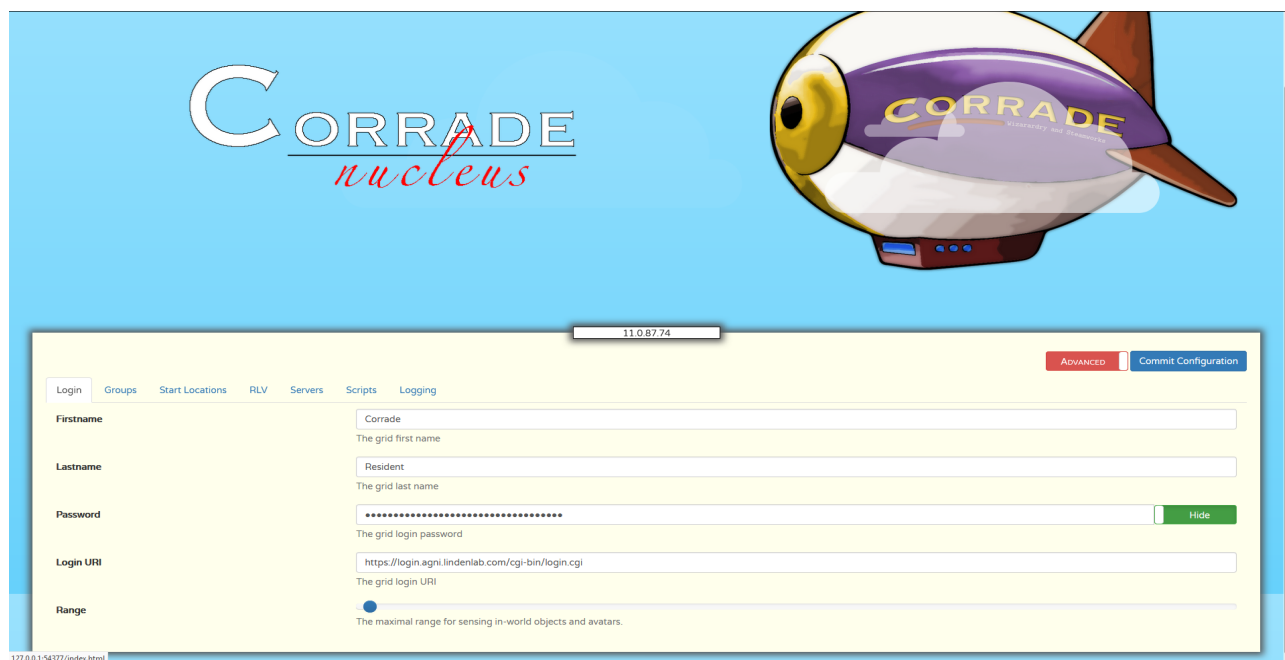
Rysunek 4.12 - Nucleus - informacja o pierwszej konfiguracji bota

Ponieważ program nie wykrywa pliku konfiguracyjnego Configuration.xml, wyświetli nam miły komunikat o możliwości skonfigurowania bota.

16. Klikamy na Close.

Obok przycisku 'Confirm Configuration' (Zatwierdź konfigurację) mamy możliwość przełączania się pomiędzy prostą (Normal) a zaawansowaną (Advanced) konfiguracją bota.

Ja omówię zaawansowaną wersję.



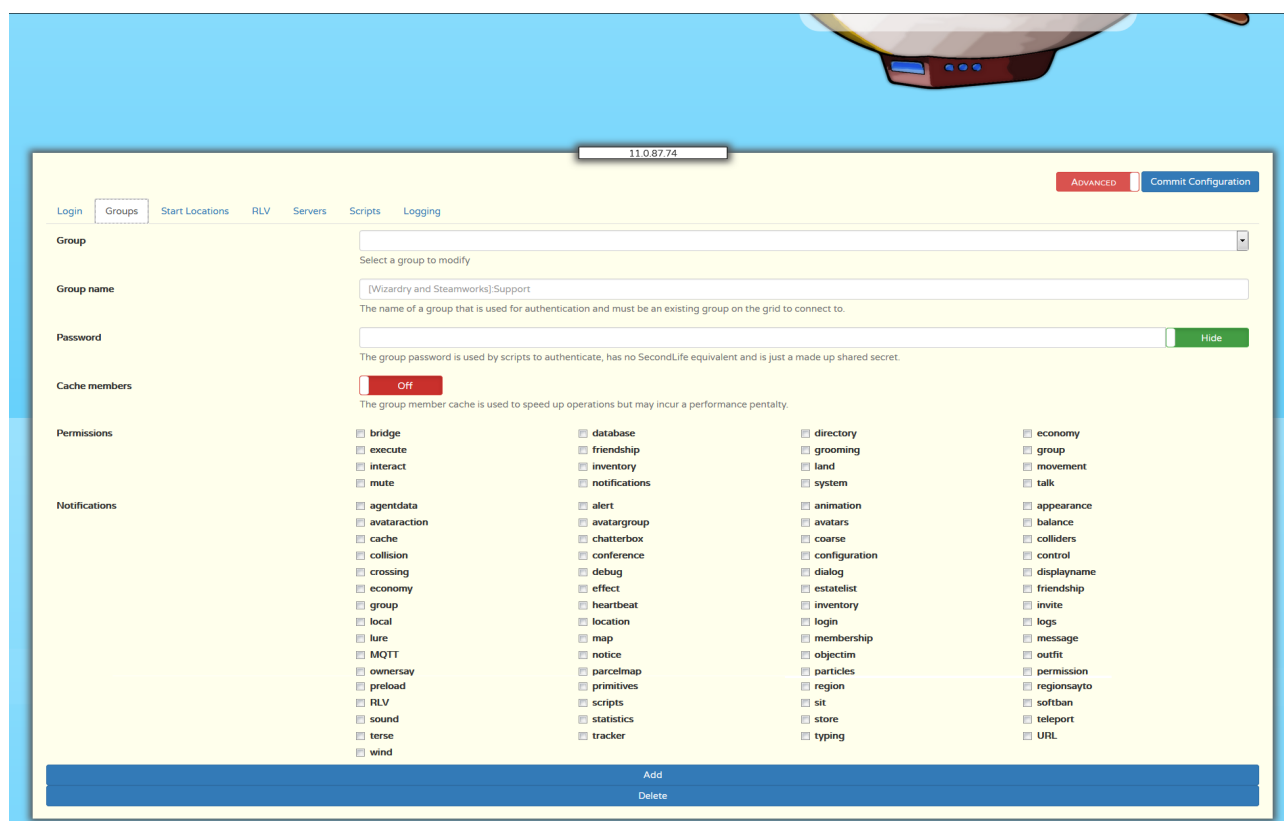
The screenshot shows the 'Login' tab of the CORRADE nucleus configuration interface. The form contains the following fields and values:

| Field | Value |
|-----------|--|
| Firstname | Corrade |
| Lastname | Resident |
| Password | ***** |
| Login URI | https://login.agni.lindenlab.com/cgi-bin/login.cgi |
| Range | 110.87.74 |

Rysunek 4.13 - Zakładka 'Login'

Na karcie 'Login' (Rysunek 4.13 - Zakładka 'Login'):

17. podajemy imię i nazwisko naszego bota (domyślne nazwisko to: Resident).
18. hasło do konta bota.
19. URL logowania do siatki (zostawiamy domyślnie dla Second Life jako <https://login.agni.lindenlab.com/cgi-bin/login.cgi>).
20. maksymalny zasięg wykrywania obiektów i awatarów w świecie (można zostawić domyślną wartość).



Rysunek 4.14 - Zakładka 'Groups'

Na karcie 'Groups' zarządzamy grupami.

21. Grupa musi istnieć w świecie Second Life i bot musi być członkiem tej grupy (przed konfiguracją najlepiej zalogować się jako bot, dołączyć do grupy, odznaczyć powiadomienia grupowe).
22. Aby dodać grupę, najpierw na dole klikamy na przycisk 'Add'.
23. Zostanie dodana grupa 'new group' do listy.
24. Z listy rozwijalnej 'Group' wybieramy właśnie nową dodaną grupę 'new group'.
25. Zostaną wczytane ustawienia i uprawnienia dla tej grupy. Możemy je swobodnie zmieniać.
26. Przede wszystkim zmieniamy nazwę grupy na właściwą.
27. Potem wpisujemy wymyślone przez nas hasło (najlepiej, aby było ono długie od 10 znaków w górę, składające się z małych liter, dużych liter, cyfr, ze znakami specjalnymi najlepiej sobie darować - Corrade może mieć problem z ich zakodowaniem).



W Second Life nie ma czegoś takiego jak hasło grupowe.

Hasło ustalane dla grupy w Corrade jest dodatkową ochroną przed nieautoryzowanym

dostępem do bota.

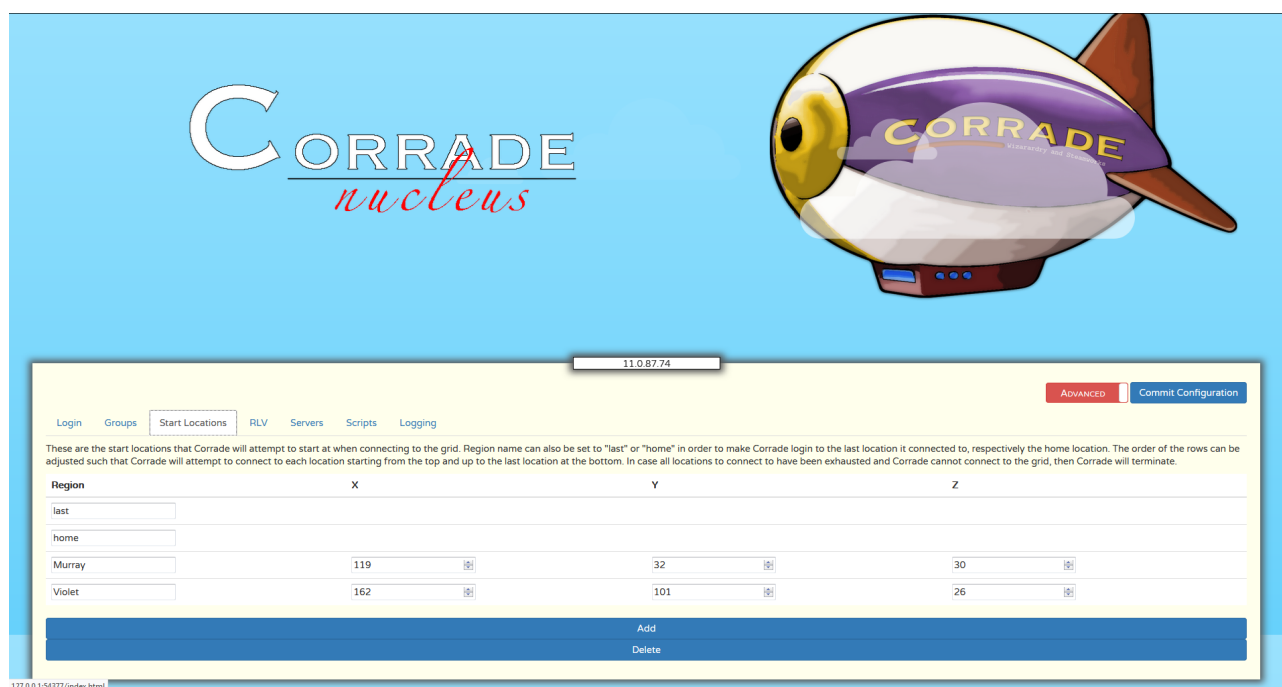
28. Następnie przy 'Cache members' decydujemy, czy ta opcja ma być włączona czy wyłączona - powoduje ona, że dane awatarów z grupy zostaną zapisane na dysku, aby później można było na nich szybciej wykonać operacje. Czasami ta opcja potrafi zawieść i u mnie jest wyłączona.

29. Dalej mamy listę uprawnień.

Zaznaczamy potrzebne, odznaczamy niepotrzebne uprawnienia.

Na pewno powinny być zaznaczone te uprawnienia, z których korzystają komendy, z których my korzystamy.

30. Jeżeli jest wszystko w porządku, przechodzimy do dalszej karty, ale tutaj jeszcze na tej karcie zalecam skasowanie domyślnej grupy o nazwie '[Wizardry and Steamworks]:Support'. Po prostu z listy 'Group' wybieramy tę grupę i dajemy na przycisk 'Delete'.



Rysunek 4.15 - Karta 'Start Locations'

31. Na karcie 'Start Locations' wskazujemy miejsce lub miejsca startowe podczas logowania się bota do Second Life. Jeżeli będzie tych miejsc startowych więcej, to jeżeli podczas logowania się bota do Second Life jedno miejsce nie będzie dostępne, to bot będzie

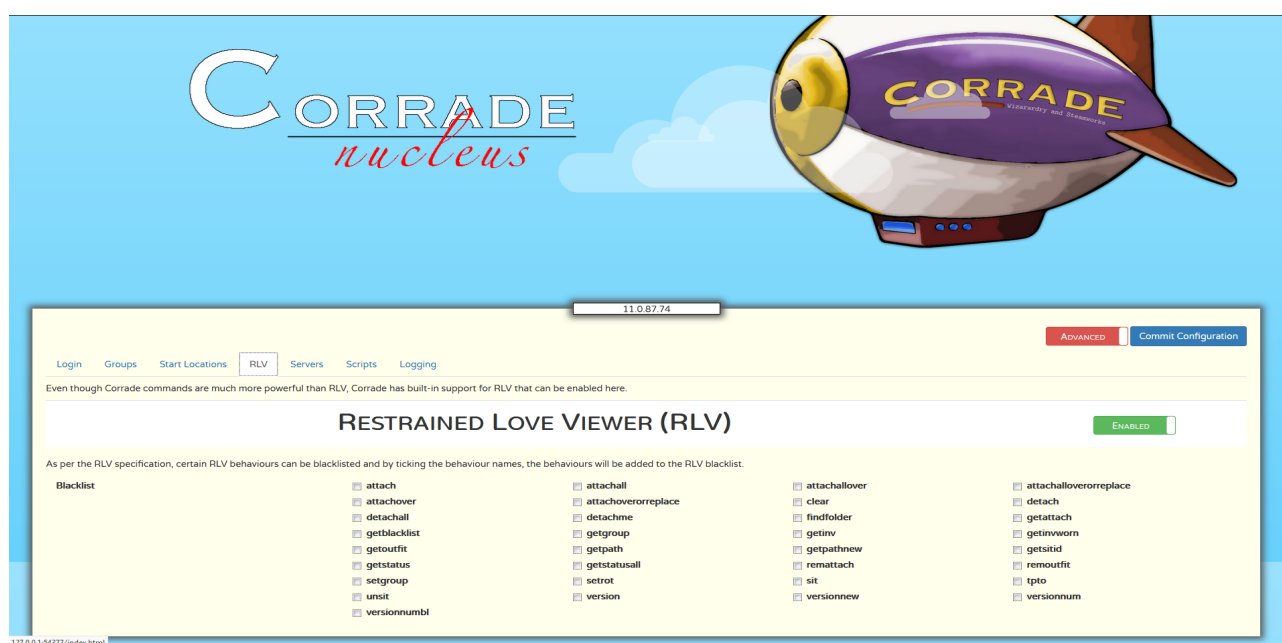
próbował się pojawić w następnym miejscu. Jeżeli natomiast wszystkie miejsca będą niedostępne, to bot nie zaloguje się do Second Life, a w logach i konsoli pojawi się komunikat, że miejsca są nieosiągalne.

32. Aby dodać nowy punkt startowy, dajemy na przycisk 'Add'.

Następnie do ostatniego wiersza możemy wpisać:

- last - bot loguje się do ostatniego zapisanego miejsca podczas wylogowania się z Second Life,
- home - bot loguje się do miejsca startowego,
- konkretną pozycję w Second Life, poprzez wpisanie nazwy regionu, pozycji X,Y,Z.

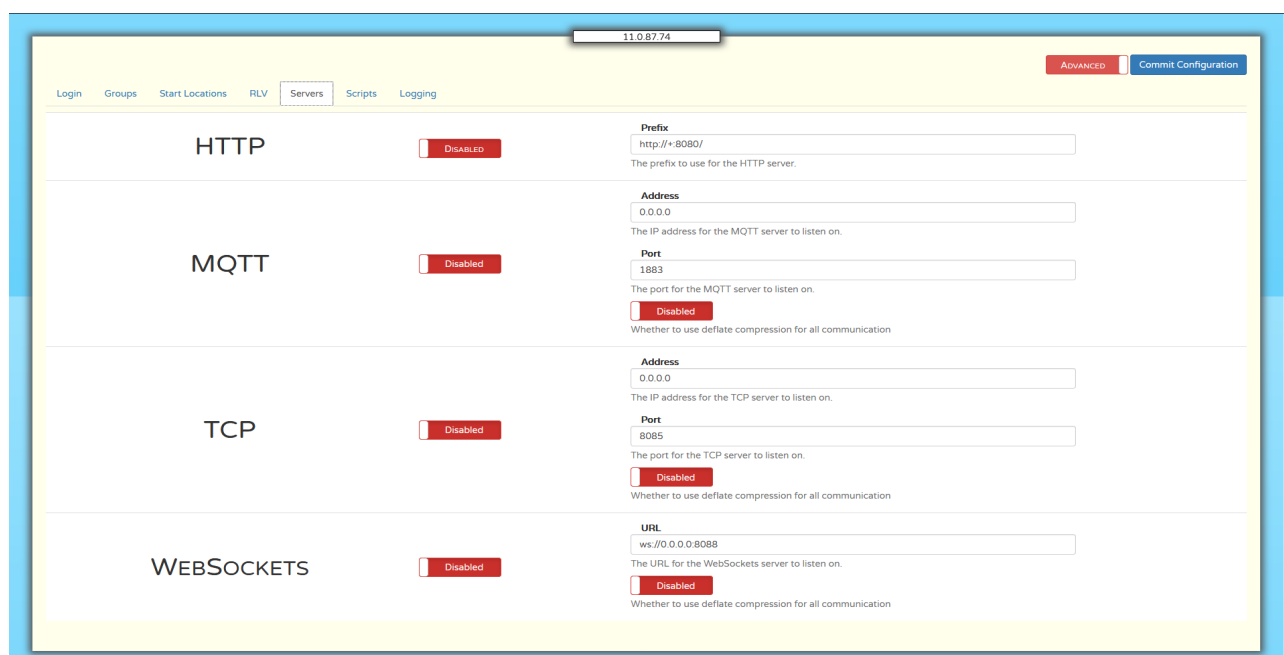
33. Aby skasować dane miejsce startowe klikamy na pole region i klikamy następnie na przycisk 'Delete'.



Rysunek 4.16 - Karta 'RLV'

34. Na karcie 'RLV' można wyklikać te opcje, które mają być zakazane.

Domyślnie pomijam tę kartę.



Rysunek 4.17 - Karta 'Servers'

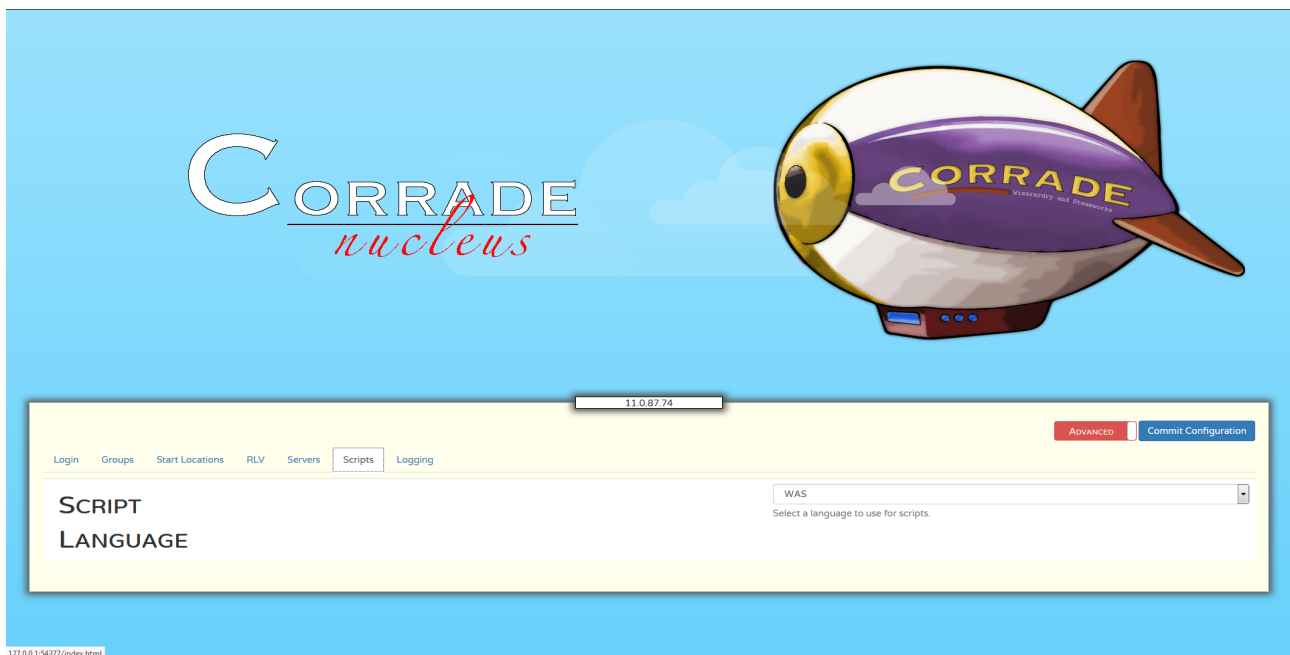
35. Na karcie 'Servers' określamy, które serwery mają odbierać żądania i wysyłać odpowiedzi do bota. Jeżeli wszystkie są wyłączone, to komendy do bota możesz wysyłać tylko przez LSL w Second Life.

Aby włączyć dany serwer, obok niego klikamy 'Disabled' na 'Enabled'.

Przy serwerze możemy określić adres IP i port (a także czy dane mają być kompresowane podczas komunikacji), na którym ma nasłuchiwać.

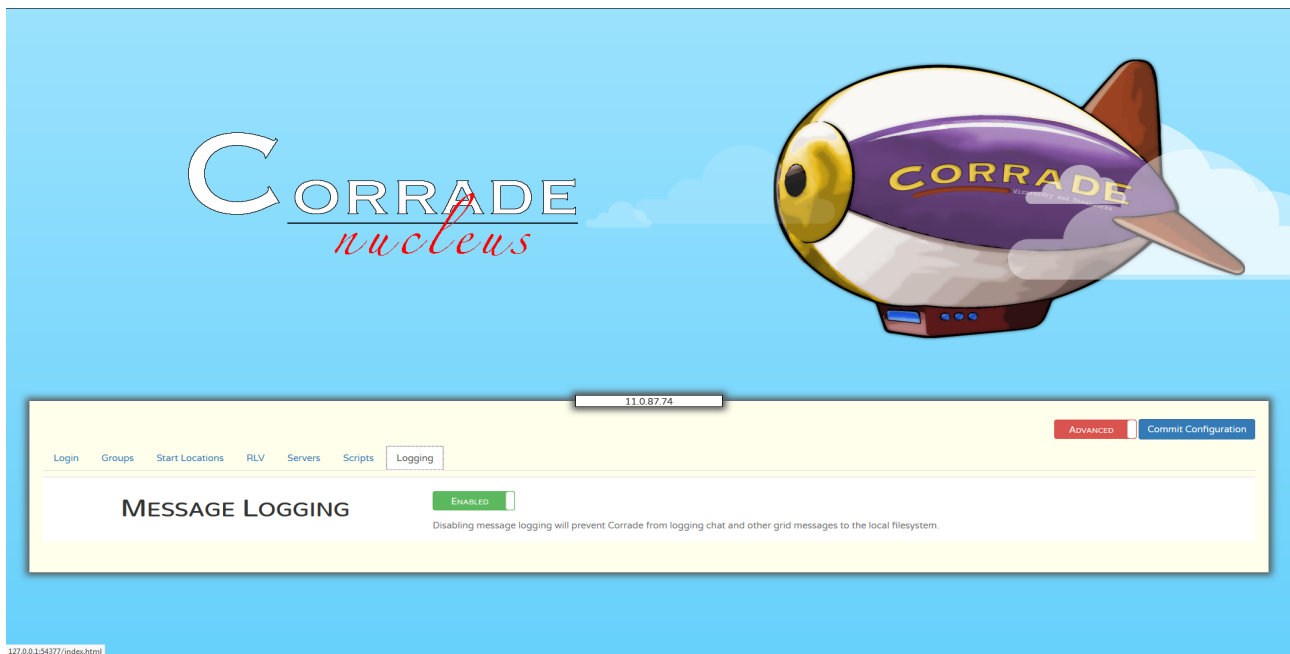
Adresem IP może być:

- 127.0.0.1 - serwer będzie nasłuchiwał lokalnie, a więc inne programy i skrypty muszą być umieszczone na tym samym komputerze co bot,
- + (HTTP) lub 0.0.0.0 - serwer będzie nasłuchiwał na każdym interfejsie karty sieciowej,
- konkretny adres IP - serwer będzie nasłuchiwał na określonym adresie IP.



Rysunek 4.18 - Karta 'Scripts'

36. Na karcie 'Scripts' określamy, czy bot ma przetwarzać skrypty według WAS czy JSON.




Rysunek 4.19 - Karta 'Logging'

37. Na karcie 'Logging' określamy czy Corrade ma logować wszystkie wiadomości pochodzące z: czatu, IM, itd.

38. Jeżeli wszystko gotowe, wystarczy kliknąć na przycisk 'Commit Configuration', wpisujemy hasło uwierzytelniające i klikamy na 'Submit'.

Jeżeli wszystko przebiegło pomyślnie, to zobaczymy okienko, że zmiany zostały zastosowane, bot powinien zalogować się do Second Life, a w konsoli zobaczymy dalsze komunikaty bez żadnych błędów.

4.4.3. Systemy z rodziny Linux x64 (rodzina Debian)

1. Uruchom terminal,
2. Pobierz archiwum zip ze strony corrade.grimore.org/download/corrade/linux-x64,
3. Rozpakuj je do folderu,
4. Przejdź do folderu Corrade,
5. Uruchom Corrade, poprzez wpisanie: `./Corrade`,
6. Dalsza konfiguracja odbywa się przez Nucleus, zobacz do rozdziału:  **System z rodziny Windows.**

Poniżej przedstawiano komendy, które należy wpisać do terminala.

```
cd /tmp/

zip=$( wget -O- "https://corrade.grimore.org/download/corrade/linux-x64/?C=M;O=D" |
grep -o "Corrade-.*\.zip\">" | sed 's/Corrade-//' | sed 's/\">/' | sort -hr | head -n 1
)

wget --no-check-certificate -O corrade.zip
"https://corrade.grimore.org/download/corrade/linux-x64/Corrade-$zip"
unzip -o corrade.zip -d "$HOME/corrade"

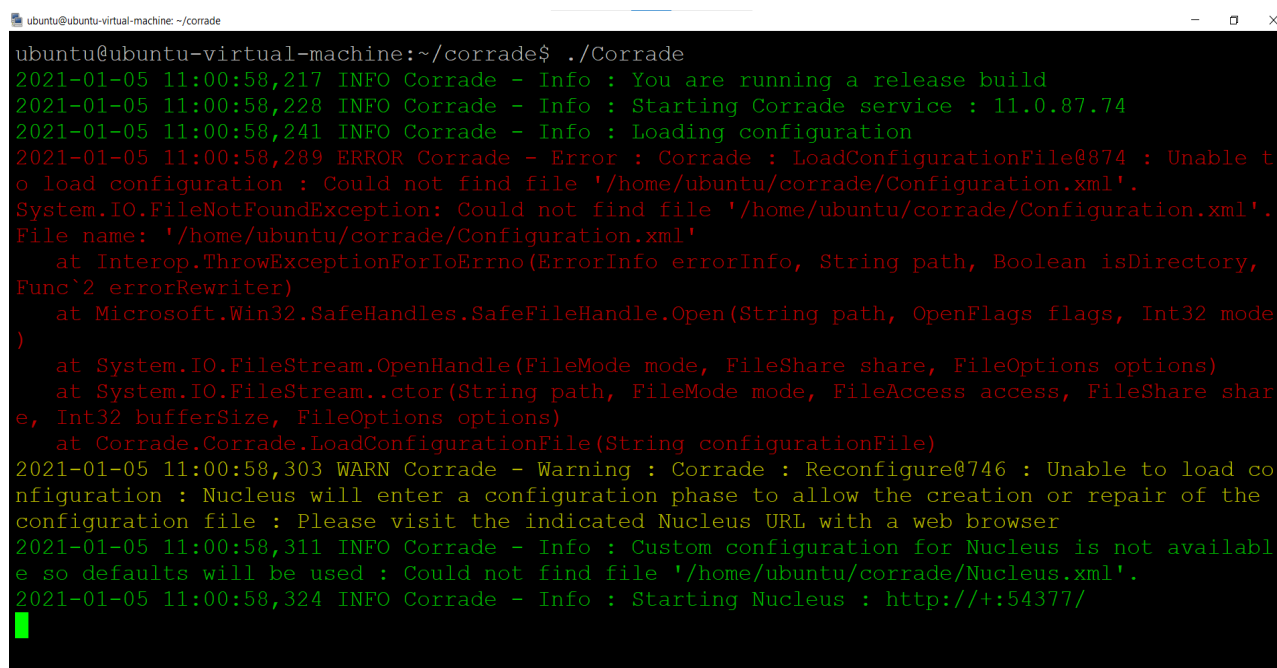
cd $HOME/corrade

./Corrade
```



W przypadku problemu, nadaj uprawnienia folderowi, gdzie Corrade się znajduje.
`chmod -R 0757 corrade`

Po uruchomieniu pojawiają się komunikaty z programu:



```
ubuntu@ubuntu-virtual-machine: ~/corrade$ ./Corrade
2021-01-05 11:00:58,217 INFO Corrade - Info : You are running a release build
2021-01-05 11:00:58,228 INFO Corrade - Info : Starting Corrade service : 11.0.87.74
2021-01-05 11:00:58,241 INFO Corrade - Info : Loading configuration
2021-01-05 11:00:58,289 ERROR Corrade - Error : Corrade : LoadConfigurationFile@874 : Unable to load configuration : Could not find file '/home/ubuntu/corrade/Configuration.xml'.
System.IO.FileNotFoundException: Could not find file '/home/ubuntu/corrade/Configuration.xml'.
File name: '/home/ubuntu/corrade/Configuration.xml'
   at Interop.ThrowExceptionForIoErrno(ErrorInfo errorInfo, String path, Boolean isDirectory, Func`2 errorRewriter)
   at Microsoft.Win32.SafeHandles.SafeFileHandle.Open(String path, OpenFlags flags, Int32 mode)
   at System.IO.FileStream.OpenHandle(FileMode mode, FileShare share, FileOptions options)
   at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, FileOptions options)
   at Corrade.Corrade.LoadConfigurationFile(String configurationFile)
2021-01-05 11:00:58,303 WARN Corrade - Warning : Corrade : Reconfigure@746 : Unable to load configuration : Nucleus will enter a configuration phase to allow the creation or repair of the
configuration file : Please visit the indicated Nucleus URL with a web browser
2021-01-05 11:00:58,311 INFO Corrade - Info : Custom configuration for Nucleus is not available so defaults will be used : Could not find file '/home/ubuntu/corrade/Nucleus.xml'.
2021-01-05 11:00:58,324 INFO Corrade - Info : Starting Nucleus : http://+:54377/
```

Rysunek 4.20 - Uruchomiony program Corrade w Linux

4.4.4. Mac OS

Na początku musimy wyłączyć sprawdzanie podpisu programu, ponieważ Corrade jest niepodpisany i zostanie rozpoznany jako utworzony przez nieznanego programistę.

W tym celu:

1. Kliknij na dole okna na ikonkę 'Launchpad' (domyślnie jest to druga ikonka od lewej strony).
2. Wpisz w wyszukiwarce słowo 'terminal'.
3. Wybierz program 'Terminal' z listy.
4. W programie Terminal wpisz polecenie:

```
sudo spctl --master-disable
```

Wpisz hasło do konta.


Jeżeli wszystko się udało, to sprawdzanie zostało wyłączone. Najlepiej przy Corrade nie włączać z powrotem.

5. Uruchamiamy przeglądarkę Safari.
6. Przechodzimy do strony <https://corrade.grimore.org/download/corrade/osx-x64/?C=M;O=D> i pobieramy ostatni świeży plik ZIP.
7. Czekamy na ściągnięcie pliku.
8. Safari będzie tak miłe dla nas, że rozpakuje dla nas pobrany plik ZIP.
9. Uruchamiamy Terminal lub aktywujemy wcześniej uruchomiony.
10. Przechodzimy do ~/Downloads (przy okazji potwierdzamy okno, że chcemy dać Terminalowi dostęp do tego katalogu).

```
cd ~/Downloads
```

1. Wykonaj listowanie katalogu (z użyciem polecenia ls), aby zobaczyć, jak nazywa się folder Corrade, do którego wypakowano najnowszą wersję. Zamień jego nazwę na Corrade z użyciem polecenia mv.
11. Przejdź do katalogu Corrade.
12. Uruchom Corrade:

```
./Corrade
```

13. Dalsza konfiguracja odbywa się przez Nucleus, zobacz do rozdziału:  **System z rodziny Windows.**

4.5. Pliki konfiguracyjne Corrade

Pliki konfiguracyjne domyślnie nazywają się:

- Configuration.xml.default,
- Nucleus.xml.default.

Wystarczy zamienić ich nazwy na:

- Configuration.xml,
- Nucleus.xml.

Przy czym Configuration.xml jest automatycznie tworzony podczas zapisywania zmian w Nucleus.

4.5.1. Configuration.xml

Plik Configuration.xml zawiera konfigurację bota. Jest on zapisany w formacie XML.

| Klucz | Opis |
|---|---|
| FirstName | imię awatara |
| LastName | nazwisko awatara (domyślnie) |
| Password | zakodowane hasło do konta awatara, składa się z prefiksu \$1\$ a następnie hasło zakodowane w MD5, hasło powinno mieć długość 6-16 znaków |
| LoginURI | adres URL do siatki, domyślnie będzie to Second Life |
| Servers | definiuje ustawienia serwerów wbudowanych |
| Servers => HTTPServer | ustawienia dla serwera HTTP |
| Servers => HTTPServer => Enable | serwer jest włączony (1) lub wyłączony (0) |
| Servers => HTTPServer => Prefixes => Prefix | określa adres HTTP, pod którym będzie dostępny serwer HTTP (adres IP:port) |

| Klucz | Opis |
|--|--|
| MQTTServer | ustawienia dla serwera MQTT |
| MQTTServer => Enable | serwer jest włączony (1) lub wyłączony (0) |
| MQTTServer => IPAddress | adres IP, pod którym będzie dostępny serwer |
| MQTTServer => Port | port, pod którym będzie dostępny serwer |
| MQTTServer => MQTTCertificate => Path | ścieżka do certyfikatu dla MQTT |
| MQTTServer => Compression | włącza (1) lub wyłącza kompresję (0) |
| TCPServer | ustawienia dla serwera TCP |
| TCPServer => Enable | serwer jest włączony (1) lub wyłączony (0) |
| TCPServer => IPAddress | adres IP pod którym będzie dostępny serwer |
| TCPServer => Port | port, pod którym będzie dostępny serwer |
| TCPServer => TCPCertificate => Path | ścieżka do certyfikatu dla TCP |
| TCPServer => TCPCertificate => Protocol | używany protokół |
| TCPServer => TCPCertificate => Password | hasło do certyfikatu |
| WebSocketsServer | ustawienia dla serwera WebSockets |
| WebSocketsServer => Enable | serwer jest włączony (1) lub wyłączony (0) |
| WebSocketsServer => WebSocketsCertificate => Path | ścieżka do certyfikatu |
| WebSocketsServer => WebSocketsCertificate => Password | hasło dla certyfikatu |
| WebSocketsServer => URL | URL pod którym będzie dostępny serwer WebSockets |
| WebSocketsServer => Compression | włącza (1) lub wyłącza kompresję (0) |
| StartLocations | definiuje miejsca startowe |
| StartLocations => StartLocation => Region | przyjmuje jedną z wartości: <ul style="list-style-type: none"> • home - teleport do miejsca oznaczonego jako 'dom', • last - teleport do miejsca, w którym bot |

| Klucz | Opis |
|---|--|
| | ostatnio się wylogował, • konkretna nazwa regionu. |
| StartLocations => StartLocation => Position | pozycja X, Y, Z, na której powinien się zalogować bot |
| Groups | definiuje grupy i uprawnienia bota do tych grup |
| Groups => Notifications | definiuje dostępne powiadomienia |
| Groups => Password | definiuje hasło do grupy, hasło jest zapisane jako SHA1 |
| Groups => Name | nazwa grupy (istniejącej w Second Life) |
| Groups => Permissions | definiuje dostępne uprawnienia do grupy |
| Groups => CacheMembers | określa, czy zapis podręczny członków grupy jest włączony (1) lub wyłączony (0) |
| RLV => Enable | włącza RLV |
| RLV => Blacklist | definiuje, które zachowania są wyłączone |
| Range | określa zasięg wykrywania awatarów i obiektów |
| IgnoreOfflineMessages | włącza (10) lub wyłącza (1) możliwość otrzymywania wiadomości offline |
| MessageLogging | zapisuj wiadomości do pliku |
| Feedback | określa, czy informacje zwrotne diagnostyczne mają być wysyłane na serwer producenta Corrade |
| ScriptLanguage | określa, jaki format danych jest używany przez Corrade do komunikacji (WAS lub JSON) |

Tabela 1 - Tabela z opisem poszczególnych zmiennych

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Configuration [<!ATTLIST Configuration xmlns:xsi CDATA #IMPLIED
xsi:noNamespaceSchemaLocation CDATA #IMPLIED>]>
<Configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="urn:corrade-configuration-
```

```
schema">
  <FirstName>MyBot</FirstName>
  <LastName>Resident</LastName>
  <Password>$1$842e4818f8ede223c9b920d4f7425c9b</Password>
  <LoginURI>https://login.agni.lindenlab.com/cgi-bin/login.cgi</LoginURI>
  <Servers>
    <HTTPServer>
      <Enable>1</Enable>
      <Prefixes>
        <Prefix>http://127.0.0.1:9199/</Prefix>
      </Prefixes>
    </HTTPServer>
    <MQTTServer>
      <Enable>0</Enable>
      <IPAddress>0.0.0.0</IPAddress>
      <Port>1883</Port>
      <MQTTCertificate>
        <Path>mqtt.pfx</Path>
      </MQTTCertificate>
      <Compression>0</Compression>
    </MQTTServer>
    <TCPServer>
      <Enable>0</Enable>
      <IPAddress>0.0.0.0</IPAddress>
      <Port>8085</Port>
      <TCPCertificate>
        <Path>tcp.pfx</Path>
        <Password>corrade</Password>
        <Protocol>Tls12</Protocol>
      </TCPCertificate>
      <Compression>0</Compression>
    </TCPServer>
    <WebSocketsServer>
      <Enable>0</Enable>
      <WebSocketsCertificate>
        <Path>ws.pfx</Path>
        <Password>corrade</Password>
      </WebSocketsCertificate>
      <URL>ws://0.0.0.0:8088</URL>
      <Compression>0</Compression>
    </WebSocketsServer>
  </Servers>
  <StartLocations>
    <StartLocation>
      <Region>Lorena Pink</Region>
      <Position>
        <X>9</X>
        <Y>127</Y>
        <Z>23</Z>
```



```
        </Position>
        </StartLocation>
    </StartLocations>
    <Groups>
        <Group>
            <Notifications>
                <Notification>group</Notification>
                <Notification>message</Notification>
                <Notification>notice</Notification>
                <Notification>local</Notification>
                <Notification>dialog</Notification>
                <Notification>permission</Notification>
                <Notification>invite</Notification>
                <Notification>sit</Notification>
                <Notification>teleport</Notification>
                <Notification>inventory</Notification>
                <Notification>wind</Notification>
                <Notification>sound</Notification>
                <Notification>terse</Notification>
                <Notification>RLV</Notification>
                <Notification>ownersay</Notification>
                <Notification>preload</Notification>
                <Notification>MQTT</Notification>
                <Notification>lure</Notification>
                <Notification>economy</Notification>
                <Notification>crossing</Notification>
                <Notification>collision</Notification>
                <Notification>cache</Notification>
                <Notification>avataraction</Notification>
                <Notification>agentdata</Notification>
                <Notification>tracker</Notification>
                <Notification>statistics</Notification>
                <Notification>scripts</Notification>
                <Notification>primitives</Notification>
                <Notification>parcelmap</Notification>
                <Notification>map</Notification>
                <Notification>location</Notification>
                <Notification>heartbeat</Notification>
                <Notification>effect</Notification>
                <Notification>debug</Notification>
                <Notification>conference</Notification>
                <Notification>alert</Notification>
                <Notification>avatargroup</Notification>
                <Notification>chatterbox</Notification>
                <Notification>typing</Notification>
                <Notification>store</Notification>
                <Notification>region</Notification>
                <Notification>particles</Notification>
                <Notification>objectim</Notification>
```

```

        <Notification>membership</Notification>
        <Notification>login</Notification>
        <Notification>estatelist</Notification>
        <Notification>configuration</Notification>
        <Notification>coarse</Notification>
        <Notification>avatars</Notification>
        <Notification>animation</Notification>
        <Notification>URL</Notification>
        <Notification>softban</Notification>
        <Notification>regionsayto</Notification>
        <Notification>outfit</Notification>
        <Notification>logs</Notification>
        <Notification>friendship</Notification>
        <Notification>displayname</Notification>
        <Notification>control</Notification>
        <Notification>colliders</Notification>
        <Notification>appearance</Notification>
        <Notification>balance</Notification>
    </Notifications>
    <Password>4e000b85758746ec818d53513c3d3e791822fdb6</Password>
    <Name>My Group</Name>
    <Permissions>
        <Permission>inventory</Permission>
        <Permission>movement</Permission>
        <Permission>grooming</Permission>
        <Permission>interact</Permission>
        <Permission>notifications</Permission>
        <Permission>talk</Permission>
        <Permission>group</Permission>
        <Permission>land</Permission>
        <Permission>mute</Permission>
        <Permission>execute</Permission>
        <Permission>bridge</Permission>
        <Permission>friendship</Permission>
        <Permission>database</Permission>
        <Permission>system</Permission>
        <Permission>directory</Permission>
        <Permission>economy</Permission>
    </Permissions>
    <Cookies />
    <CacheMembers>1</CacheMembers>
</Group>
</Groups>
<RLV>
    <Enable>0</Enable>
    <Blacklist>
        <Behaviour>sendim</Behaviour>
        <Behaviour>tplm</Behaviour>
    </Blacklist>

```

```

    </RLV>
    <Range>64</Range>
    <Language />
    <PublishLanguage>1</PublishLanguage>
    <IgnoreOfflineMessages>1</IgnoreOfflineMessages>
    <MessageLogging>1</MessageLogging>
    <MultipleSimulatorConnections>0</MultipleSimulatorConnections>
    <Feedback>0</Feedback>
    <NotificationSettings>
        <AutoPrune>
            <Enable>1</Enable>
            <Conditions>
                <Condition>
                    <StatusCode>404</StatusCode>
                    <Expression>sim.+?agni.lindenlab.com:[0-9]+?\cap/[0-
9a-fA-F\ - ]+?</Expression>
                </Condition>
            </Conditions>
        </AutoPrune>
        <Scripts>
            <Update>1000</Update>
        </Scripts>
        <Colliders>
            <Update>1000</Update>
        </Colliders>
        <Statistics>
            <Update>1000</Update>
        </Statistics>
        <Wind>
            <Update>1000</Update>
        </Wind>
        <Location>
            <Update>1000</Update>
        </Location>
        <Terse>
            <Update>1000</Update>
        </Terse>
    </NotificationSettings>
    <ScriptLanguage>JSON</ScriptLanguage>
</Configuration>

```

Tekst 1 - Plik konfiguracyjny bota

4.5.2. Nucleus.xml

Konfiguracja Nucleus.

- Prefix - określa adres URL, pod którym będzie dostępny Nucleus, + oznacza pod każdym adresem IP.
- Enabled - określa, czy Nucleus jest włączony (1) lub wyłączony (0).
- CodeWord - określa hasło dostępowe do Nucleus.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Configuration [
<!ATTLIST Configuration xmlns:xsi CDATA #IMPLIED xsi:noNamespaceSchemaLocation CDATA
#IMPLIED>
]>
<Configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="urn:corrade-nucleus-schema">
    <Prefix>http://+:54377/</Prefix>
    <Enabled>1</Enabled>
    <CodeWord>nucleus</CodeWord>
</Configuration>
```

Tekst 2 - Plik konfiguracyjny Nucleus

4.6. Instalowanie i usuwanie Corrade jako usługi systemowej

Corrade standardowo nie posiada możliwości zainstalowania programu jako usługi systemowej dlatego też poniżej przedstawiam sposób zainstalowania Corrade jako usługi w popularnych systemach operacyjnych.

4.6.1. Windows

W folderze contrib\windows znajdziemy następujące pliki:

- install-corrade-service.bat - instaluje Corrade jako usługę (o nazwie Corrade Resident),

- nssm.exe - program, który umożliwia dodanie uchwytu do programu, którego standardowo nie można obsłużyć jako usługi; program instaluje się jako usługa i przechwyci wszelkie żądania systemowe (uruchomienie, restart, zatrzymanie) dla konkretnego programu,
- nssm-GUI.bat - powyższy program, ale możliwość dodania i edycji za pomocą graficznego interfejsu użytkownika,
- uninstall-corrade-service.bat - odinstalowanie usługi.

Można też samemu zainstalować i odinstalować usługę:

1. Uruchom wiersz poleceń z uprawnieniami administratora w głównym folderze bota.
2. Wpisz i zatwierdź polecenie: powershell.
3. Poniżej znajdują się polecenia, które skopiuj: linjka po linijce.

Uwaga! Poniższe linijki instalują Corrade jako usługę o nazwie: Corrade.

Możesz zmienić nazwę usługi na swoją, jednakże dostosuj wszystkie linijki aby zawierały nową nazwę twojej usługi.

```
$currentDir = $(Get-Location).Path
$binPath = "$currentDir\Corrade.exe"
.\contrib\windows\nssm.exe install Corrade "$binPath"
.\contrib\windows\nssm.exe set Corrade AppDirectory "$currentDir"
.\contrib\windows\nssm.exe set Corrade AppExit Default Restart
.\contrib\windows\nssm.exe set Corrade DisplayName Corrade
.\contrib\windows\nssm.exe set Corrade ObjectName LocalSystem
.\contrib\windows\nssm.exe set Corrade Start SERVICE_AUTO_START
.\contrib\windows\nssm.exe set Corrade Type SERVICE_WIN32_OWN_PROCESS
.\contrib\windows\nssm.exe set Corrade AppThrottle 1500
.\contrib\windows\nssm.exe set Corrade AppRestartDelay 1000
```

4. zamknij okno wiersza polecenia.

Aby zdeinstalować program jako usługę, w wierszu poleceń należy wpisać i zatwierdzić:

```
sc stop corrade  
sc delete corrade
```

4.6.2. Linux

Dla Linux znajdziemy wstępnie przygotowany plik corrade.service w folderze contrib\linux, który jest obsługiwany przez demona SystemD. Wystarczy go skopiować, skonfigurować i włączyć do demona uruchamiania usług systemowych.

Otwórz wiersz poleceń (terminal), wpisz po kolei polecenia i je pozatwierdzaj:

Dodanie nowego użytkownika o nazwie 'corrade'.

```
sudo useradd -m corrade
```

Ustawianie nowego hasła dla użytkownika 'corrade'.

Wpisz nowe hasło dwa razy dla tego użytkownika.

```
sudo passwd corrade
```

Dodanie 'corrade' do grupy umożliwiającej uruchamianie polecenia sudo.

```
sudo usermod -aG sudo corrade
```

Logowanie się jako 'corrade'.

Wpisz ustanowione wcześniej hasło.

```
su corrade
```

Przejdź do katalogu domowego użytkownika.

```
cd ~
```

Do tego folderu ściągnij lub przekopiuj corrade.

Nadaj uprawnienia do tego folderu nowemu użytkownikowi i grupie: corrade.

```
chown -R corrade:corrade corrade
```

Przejdź do folderu z corrade.

Następnie przejdź do folderu z corrade.service.

```
cd contrib/linux/
```

przekopiuj plik do folderu systemd.

```
sudo cp -af corrade.service /etc/systemd/system/
```

zedytuj ten plik:

```
sudo nano /etc/systemd/system/corrade.service
```

Tutaj:

- w WorkingDirectory popraw ścieżkę do folderu corrade,
- w ExecStart popraw ścieżkę do pliku corrade (to samo co w WorkingDirectory tylko dodaj /Corrade) na /home/corrade/corrade.

Jeżeli wszystko gra, zapisz zmiany i wróć do konsoli.

```
sudo systemctl enable corrade.service  
sudo systemctl start corrade.service
```

Wejdź przez przeglądarkę na adres: http://ADRES_IP_KOMPUTERA:54377 i skonfiguruj bota.

Jeżeli coś jest nie tak, sprawdź logi bota, systemu, może uprawnienia do folderu są nie właściwe...

4.7. Corrade.log i Openmetaverse.log

- **Logs**

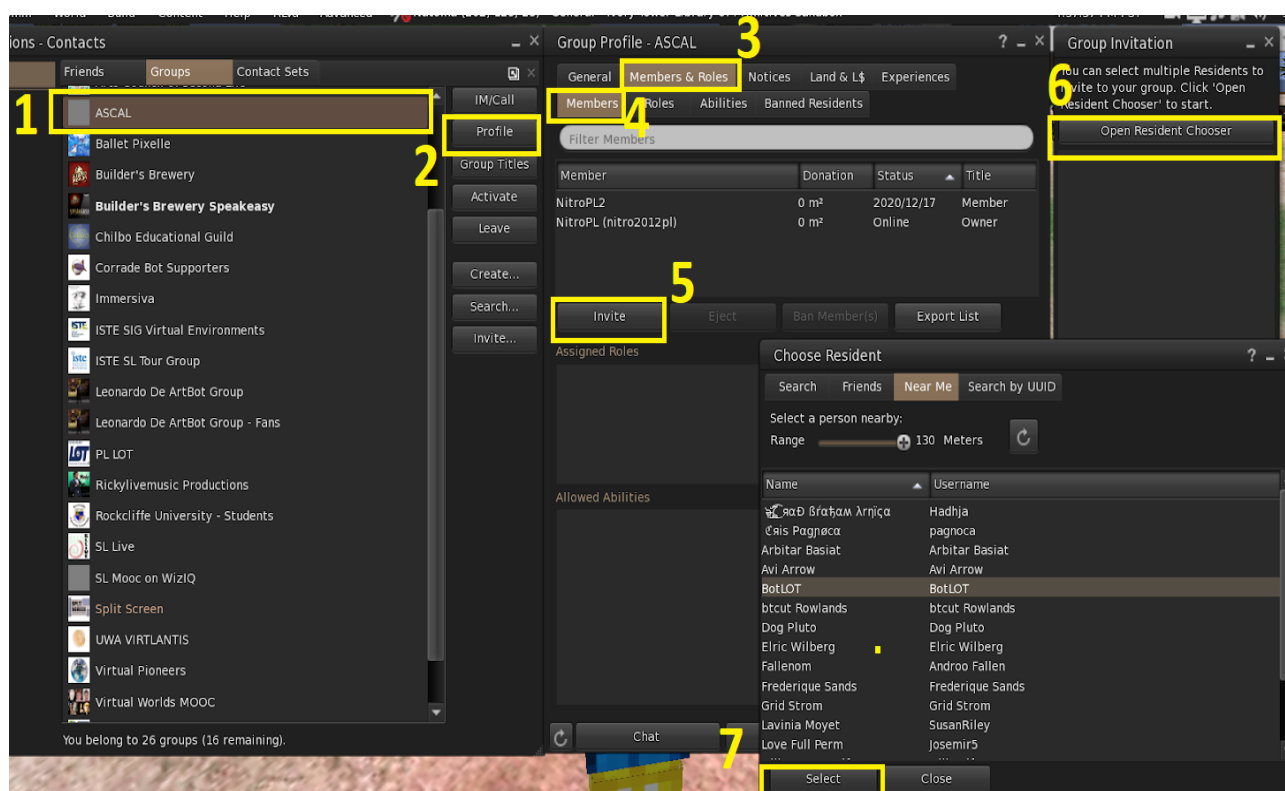
W folderze 'Logs' trzymane są wszelkie pliki tekstowe do których Corrade zarejestrował zdarzenia.

- **Corrade.log** odnosi się do programu Corrade,
- **OpenMetaverse.log** odnosi się do świata Second Life,
- pliki z rozszerzeniem .log odnoszą się do bieżących rejestrowanych plików,
- pliki z rozszerzeniem .logNUMER są plikami zarchiwizowanymi z danego dnia, np. Corrade.log20200912 odnosi się do wydarzeń zarejestrowanych z dnia 2020-09-12,
- folder **Chat**
 - **IM** - zapisane rozmowy prywatne z botem, nazwa pliku to UUID awatara z którym bot rozmawiał,
 - **Local** - zapisane rozmowy z czatu publicznego, nazwa pliku to nazwa regionu w którym znajdował się bot,
 - **OwnerSay** - zapisane rozmowy poprzez rejestrację z funkcji llOwnerSay() (wysłane przez właściciela primu/obiektu), nazwa pliku to UUID primu, właściciela, itp.

4.8. Dodanie bota do grupy w SL, nadania mu uprawnień, wyłączenie opcji otrzymywania wiadomości grupowych

Dodanie bota do grupy i przypisanie mu roli z uprawnieniami jest bardzo łatwe.

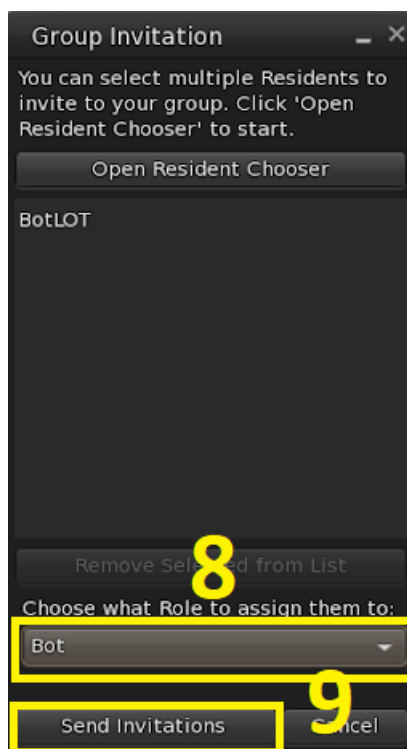
Krótki rozdział jak tego dokonać z wykorzystaniem przeglądarki Firestorm Viewer.



Rysunek 4.21 - Wybór awataru do zaproszenia grupowego

1. Wybierz grupę z listy.
2. Wejdź w szczegóły grupy.
3. Przejdź do zakładki 'Członkowie i role'.
4. Przejdź do zakładki 'członkowie'.
5. Kliknij na przycisk 'Zaproś'.
6. Kliknij na przycisk, który pozwala na wybranie awatarów do zaproszenia.

7. W tym oknie za pomocą odpowiedniej zakładki znajdź swojego bota, którego dodasz do grupy. W odpowiednim polu wpisz jego nazwę i daj na przycisk wyszukiwania, a następnie dodaj go do listy do zaproszenia. Jeżeli wszystko będzie ok, wciśnij przycisk 'Wybierz'.



Rysunek 4.22 - Wybranie roli i wysłanie zaproszenia awatarowi

8. Wybieramy rolę, do której przypiszemy bota. Ta rola musi być wcześniej utworzona, najlepiej też wcześniej przypisać w tej roli uprawnienia grupowe botowi.
9. Wysyłamy zaproszenie.

Teraz wystarczy zalogować się jako bot i przyjąć zaproszenie do grupy.

4.9. System notacyjny uprawnień

W Second Life znany jest system uprawnień, który jest widoczny zwłaszcza przy edycji obiektów. Corrade również obsługuje ten system zapisując go w specyficzny dla siebie sposób, np.

cdemvt-----cde-vtcdemvt

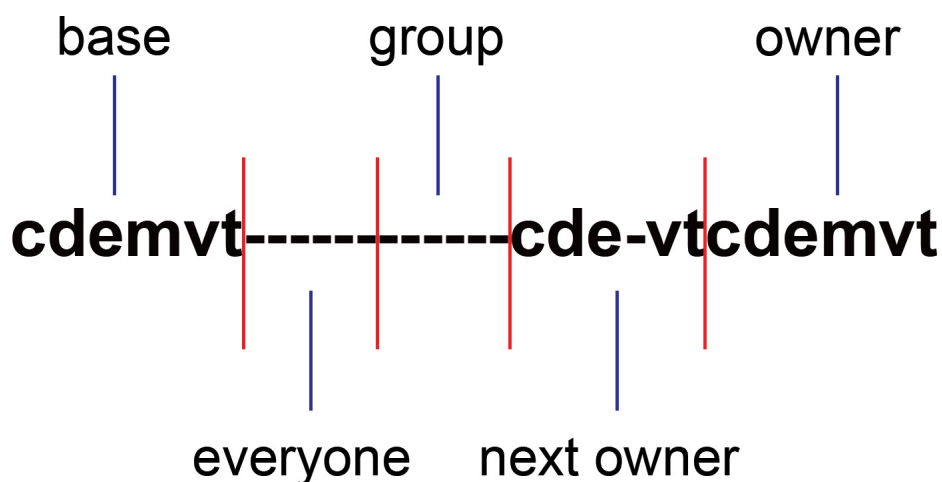
Ten zapis składa się z 5 segmentów, który każdy z nich składa się po 6 znaków, które określają rodzaj uprawnień.

Kolejne segmenty uprawnień to: podstawowy, każdy, grupowy, następny właściciel, właściciel.

Każdy znak oznacza:

- c (copy) – kopiowanie,
- d (damage) - uszkodzenie (używane w symulatorze OpenSimulator),
- e (export) - eksport (używane w symulatorze OpenSimulator),
- m (modify) – modyfikacja,
- v (move) – przenoszenie,
- t (transfer) – transferowanie,
- - (dash) - brak uprawnień.

Przyjrzymy się głębiej wcześniej zdefiniowanemu przykładowi.



Rysunek 4.23 - Podział uprawnień obiektu

- pierwsze 6 znaków - cdemvt - oznaczają pełne uprawnienia dla maski 'podstawowa',
- następne 6 znaków - ----- - oznaczają brak uprawnień dla maski 'każdy',
- następne 6 znaków - ----- - oznaczają brak uprawnień dla maski 'grupa',
- następne 6 znaków - cde-vt - oznaczają, że dla maski 'następny właściciel' ustawiono uprawnienia: kopiowania, uszkodzania, eksportu, przenoszenia, transportu, natomiast następny właściciel nie ma możliwości modyfikacji obiektu,
- ostatnie 6 znaków - cdemvt - oznaczają pełne uprawnienia dla maski 'właściciel'.

4.10. Powiadomienia

Powiadomienie to funkcja bota, które odbiera dane z Second Life w czasie rzeczywistym i na podstawie których bot może wywoływać określoną akcję.

Na przykład jest powiadomienie 'alert', które odbiera wiadomości odnośnie regionu. Najczęściej awatary otrzymują informacje o restarcie regionu. Jeżeli bot otrzyma taką informację to może, np. teleportować się tymczasowo do innej lokacji.

Mamy inne przykładowe powiadomienie 'balance', które informuje na bieżąco o stanie konta bota. Na przykład bot może pełnić funkcję sprzedawcy towarów lub menedżera wynajmu działek w regionie, co w związku z tym przyjmuje płatności i zwraca nadwyżki (np. można wynająć ziemię maksymalnie na miesiąc, a ktoś zapłacił za 2 miesiące, więc bot musi oddać pieniądze za dodatkowy miesiąc). Jeżeli bot otrzyma powiadomienie, że na jego koncie nie ma pieniędzy (saldo jest zerowe a nawet ujemne) może wysłać do swojego administratora w postaci IM lub emaila z prośbą o uzupełnienie pieniędzy w kasie.

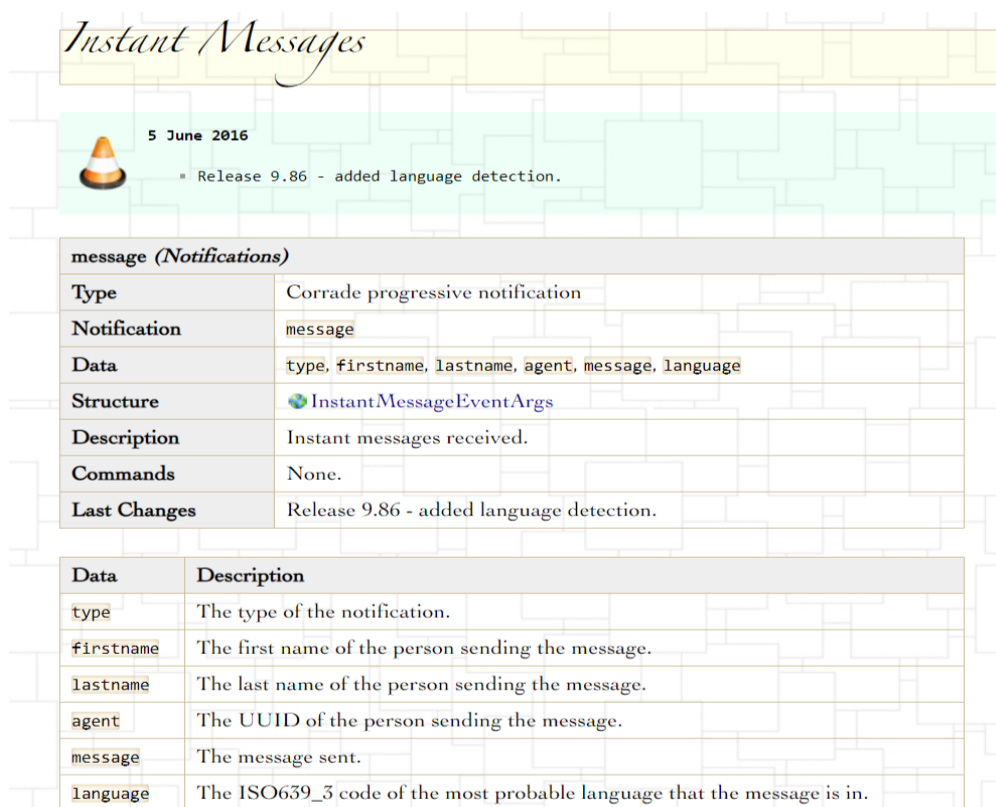
Ostatnim przykładem jaki chciałbym podać jest powiadomienie 'terse', który zwraca informację o nowej pozycji i obrocie awatara czy obiektu. Można użyć tego powiadomienia, np. do śledzenia awatarów w regionie i sprawdzenia po ich pozycji, czy nie naruszają jakiegoś obszaru dostępnego tylko dla uwierzytelnionych osób. Jeżeli byłaby taka sytuacja, to można najpierw

wysłać ostrzeżenie do takiej osoby wiadomością prywatną, a po np. 30 sekundach zbanować ją tymczasowo na 1 godzinę.

Powyżej tylko przedstawiłem możliwe scenariusze z użyciem niektórych powiadomień. Wszystkie powiadomienia znajdziesz na stronie oprogramowania Corrade¹³.

Należy pamiętać, aby czyścić bota ze starych URL powiadomień. Można tego dokonać za pomocą polecenia 'notify' z opcją 'action' ustawioną na 'purge' lub przed uruchomieniem bota skasować plik State\Notifications.xml, w tym celu najlepiej utworzyć sobie plik wsadowy, gdzie najpierw kasujemy plik State\Notifications.xml a następnie uruchamiamy bota. Jeżeli masz materiały dodatkowe, zobacz plik bot_installer\corrade\bot_ai\start.bat.

Przedstawię teraz kod w LSL w jaki sposób zainstalować powiadomienie, tutaj 'message'¹⁴ zostanie on wykorzystany do odbioru wiadomości prywatnych od awatarów, które do niego napiszą.



| message (Notifications) | |
|-------------------------|---|
| Type | Corrade progressive notification |
| Notification | message |
| Data | type, firstname, lastname, agent, message, language |
| Structure | InstantMessageEventArgs |
| Description | Instant messages received. |
| Commands | None. |
| Last Changes | Release 9.86 - added language detection. |

| Data | Description |
|-----------|---|
| type | The type of the notification. |
| firstname | The first name of the person sending the message. |
| lastname | The last name of the person sending the message. |
| agent | The UUID of the person sending the message. |
| message | The message sent. |
| language | The ISO639_3 code of the most probable language that the message is in. |

Rysunek 4.24 - Dokumentacja dla powiadomienia 'message'

13 Powiadomienia - https://grimore.org/secondlife/scripted_agents/corrade/api/notifications

14 https://grimore.org/secondlife/scripted_agents/corrade/api/notifications/message

W tabelce na stronie mamy rozpisane, jakie dane są zwracane z poszczególnych pól: typ powiadomienia; imię, nazwisko, UUID nadawcy; wiadomość przesłana od nadawcy; wykryty język wiadomości.

Poniżej pełny kod LSL na zainstalowanie i używanie tego powiadomienia.

Definiujemy zmienne dla UUID bota, hasła grupowego, nazwy grupy.

Pod zmienną URL będziemy przechowywać przypisany URL od Second Life, z którego będą przychodziły powiadomienia.

Zmienna TAG będzie przechowywała nazwę skryptu.

Funkcja:

- `setDebug()` - wysyła wiadomość do właściciela primu/obiektu (u mnie jest to bot),
- `strReplace()` - zastępuje znaki w tekście,
- `wasCSVToList()` - funkcja WAS do przekształcenia CSV na listę,
- `wasKeyValueGet()` - funkcja WAS do pobrania wartości na podstawie klucza z danych,
- `wasKeyValueEncode()` - funkcja WAS do kodowania danych na podstawie RFC 4180,
- `wasURLUnescape()` - funkcja WAS do rozkodowania danych na podstawie RFC 4180,
- `wasKeyValueToJSON()` - funkcja WAS, która przekształca dane połączone '&' do postaci JSON,
- `wasJSONToKeyValueData()` - funkcja WAS, która przekształca JSON do postaci danych połączonych '&'.

W stanie 'default' następuje przypisanie nazwy skryptu do zmiennej, pokazuje liczbę wolnej pamięci dla skryptu, następnie wywołuje następny stan 'ReadConfigurationNotecard', gdzie ten stan czyta notatkę o nazwie 'configuration', która jest umieszczona w tym samym primie co skrypt.

W notatce muszą się znaleźć następujące linijki:

- `BOTID=UUID_BOTA,`
- `GROUP=NAZWA_GRUPA,`
- `PASSWORD=HASŁO_DO_GRUPY.`

Po pobraniu zmiennych z notatki i przypisania ich do zmiennych w skrypcie jest wywoływany kolejny stan o nazwie 'url', w którym następuje próba pobrania URL z Second Life. Jeżeli wszystko się tutaj udało, to następuje przejście do kolejnego stanu o nazwie 'NotifyInstall'. W tym stanie instalujemy oczekiwane powiadomienie. W parametrze URL podajemy zmienną URL, do której wcześniej pobraliśmy URL przypisany przez Second Life. Oczywiście to może być dobrze URL do skryptu na zewnętrznym serwerze.

Całą tablicę parametrów wysyłamy do bota w formacie JSON.

W `http_request()` oczekujemy na odbiór danych od bota, przetwarzamy je w ten sposób, że po prostu sprawdzamy, czy nasze polecenie zostało wykonane, jeżeli jest sukces, to idziemy do następnego stanu o nazwie 'sense', jeżeli wystąpił błąd, to wykonywany jest stan 'preNotifyInstall', który w zasadzie powraca do bieżącego stanu, ponieważ nie da rady wywołać ponownie bezpośrednio bieżący stan.

W stanie sense 'nasłuchujemy' na bieżąco nadchodzących wiadomości od powiadomienia 'message', tutaj też przypisywaliśmy sobie do zmiennych wartości omawiane wcześniej z tabelki ze strony powiadomienia.

Dalej od nas zależy co z tym zrobimy.

```
key CORRADE;
string PASSWORD;
string GROUP;
string URL = "";
string TAG;
string val;
list lmessage;
string sjump = "";
string body;

integer line;
list tuples = [];

list lprepare = [];

setDebug(string msg)
{
    llOwnerSay "["+ TAG +" " + msg);
}

string strReplace(string str, string search, string replace) {
    return llDumpList2String(llParseStringKeepNulls((str = "") + str, [search], []),
replace);
}

list wasCSVToList(string csv)
{
    list l = [];
```

```

list s = [];
string m = "";
do
{
    string a = llGetSubString(csv, 0, 0);
    csv = llDeleteSubString(csv, 0, 0);
    if(a == ",")
    {
        if(llList2String(s, -1) != "\\")
        {
            l += m;
            m = "";
            jump continue;
        }
        m += a;
        jump continue;
    }
    if(a == "\\\" && llGetSubString(csv, 0, 0) == a)
    {
        m += a;
        csv = llDeleteSubString(csv, 0, 0);
        jump continue;
    }
    if(a == "\\")
    {
        if(llList2String(s, -1) != a)
        {
            s += a;
            jump continue;
        }
        s = llDeleteSubList(s, -1, -1);
        jump continue;
    }
    m += a;
    @continue;
} while(csv != "");
return l + m;
}

string wasKeyValueGet(string k, string data)
{

```



```

if(llStringLength(data) == 0) return "";
if(llStringLength(k) == 0) return "";
list a = llParseStringKeepNulls(data, ["&", "="], []);
integer i = llListFindList(llList2ListStrided(a, 0, -1, 2), [ k ]);
if(i != -1){
    string ret = llList2String(wasCSVToList(wasURLUnescape(llList2String(a, 2*i+1))),
0);
    ret = strReplace(ret, "\\r", "");
    ret = strReplace(ret, "\\n", "");
    ret = llStringTrim(ret, STRING_TRIM);

    return ret;
}
return "";
}

string wasKeyValueEncode(list data)
{
    list k = llList2ListStrided(data, 0, -1, 2);
    list v = llList2ListStrided(llDeleteSubList(data, 0, 0), 0, -1, 2);
    data = [];
    do
    {
        data += llList2String(k, 0) + "=" + llList2String(v, 0);
        k = llDeleteSubList(k, 0, 0);
        v = llDeleteSubList(v, 0, 0);
    } while(llGetListLength(k) != 0);
    return llDumpList2String(data, "&");
}

string wasURLEscape(string i)
{
    string o = "";
    do
    {
        string c = llGetSubString(i, 0, 0);
        i = llDeleteSubString(i, 0, 0);
        if(c == "") jump continue;
        if(c == " ")
        {
            o += "+";

```

```
        jump continue;
    }
    if(c == "\\n")
    {
        o += "%0D" + llEscapeURL(c);
        jump continue;
    }
    o += llEscapeURL(c);
    @continue;
} while(i != "");
return o;
}

string wasURLUnescape(string i)
{
    return llUnescapeURL( llDumpList2String( llParseStringKeepNulls( llDumpList2String(
llParseStringKeepNulls( i, ["+"], [] ), " " ), ["%0D%0A"], [] ), "\\n" ) );
}

default
{
    state_entry()
    {
        TAG = llGetScriptName();
        setDebug("Free memory: " + (string)llGetFreeMemory());
        state ReadConfigurationNotecard;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}
```

```
state ReadConfigurationNotecard
{
    state_entry()
    {
        if(llGetInventoryType("configuration") != INVENTORY_NOTECARD)
        {
            setDebug("Sorry, could not find a configuration inventory notecard.");
            return;
        }
        setDebug("Reading configuration file...");
        line = 0;
        llGetNotecardLine("configuration", line);
    }

    dataserver(key id, string data)
    {
        if(data == EOF)
        {
            if(llGetListLength(tuples) % 2 != 0)
            {
                setDebug("Error in configuration notecard.");
                return;
            }

            CORRADE = llList2Key(tuples, llListFindList(tuples, ["BOTID"])+1);

            if(CORRADE == NULL_KEY)
            {
                setDebug("Error in configuration notecard: BOT ID KEY");
                return;
            }

            GROUP = llList2String(tuples, llListFindList(tuples, ["GROUP"])+1);

            if(GROUP == "")
            {
                setDebug("Error in configuration notecard: GROUP");
                return;
            }
        }
    }
}
```

```

        PASSWORD = llList2String(tuples, llListFindList(tuples, ["PASSWORD"])+1);

        if(PASSWORD == "")
        {
            setDebug("Error in configuration notecard: PASSWORD");
            return;
        }

        state url;
    }
    if(data == "") jump continue;
    integer i = llSubStringIndex(data, "#");
    if(i != -1) data = llDeleteSubString(data, i, -1);
    list o = llParseStringKeepNulls(data, ["="], []);
    string k = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String(o,0), STRING_TRIM), ["\""], []), "\"");
    string v = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String( o, 1 ), STRING_TRIM), ["\""], [] ), "\"");
    if(k == "" || v == "") jump continue;
    tuples += k;
    tuples += v;
    @continue;
    llGetNotecardLine("configuration", ++line);
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

state url
{

```

```
state_entry()
{
    if ((GROUP == "BOTID") || (GROUP == "GROUP") || (GROUP == "PASSWORD")){
        llResetScript();
        return;
    }

    if ((PASSWORD == "BOTID") || (PASSWORD == "GROUP") || (PASSWORD == "PASSWORD"))
{
        llResetScript();
        return;
    }

    setDebug("Requesting URL...");
    llSetTimerEvent(0.5);
}

timer(){
    llSetTimerEvent(60);
    llRequestURL();
}

http_request(key id, string method, string body)
{
    llHTTPResponse(id, 200, "OK");
    if(method != URL_REQUEST_GRANTED) return;
    URL = body;
    setDebug("Got URL...");
    state NotifyInstall;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
```

```
        llResetScript();
    }
}

state NotifyInstall {
    state_entry() {
        llOwnerSay("Binding to the Message Alert notification...");
        llSetTimerEvent(0.5);
    }

    timer(){
        llSetTimerEvent(60);
        lprepare = [
            "group", wasURLEscape(GROUP),
            "password", wasURLEscape(PASSWORD),
            "callback", wasURLEscape(URL),
            "URL", wasURLEscape(URL),
            "command", "notify",
            "action", "set",
            "type", "message",
            "_script", TAG
        ];

        llOwnerSay(wasKeyValueEncode(lprepare));
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");
        if(wasKeyValueGet("success", body) != "True") {
            llOwnerSay("Failed to bind to the Message Alert notification...");
            state preNotifyInstall;
        }
        // DEBUG
        llOwnerSay("IM Alert notification installed...");
        state sense;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
```

```
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state preNotifyInstall
{
    state_entry()
    {
        state NotifyInstall;
    }

    changed(integer change)
    {
        {
            if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
            {
                llResetScript();
            }
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state sense {
    state_entry() {
        setDebug("Listen Region Restart...");
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");
        string firstname = wasKeyValueGet("firstname", body);
    }
}
```

```
string lastname = wasKeyValueGet("lastname", body);
string agent = wasKeyValueGet("agent", body);
string message = wasKeyValueGet("message", body);

//WHAT YOU DO NEXT?

}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}
```

Tekst 3 - Pełny skrypt w LSL do zainstalowania powiadomienia 'message' z odbiorem danych

4.11. Kody błędów

Kiedy Corrade nie może przetworzyć żądania, odsyła odpowiedź, gdzie znajduje się kod błędu i wiadomość.

Poniżej przedstawiam obszerną tabelę z kodami błędów.

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|---|---|
| 48 | text too long | tekst zbyt długi |
| 303 | unable to post proposal | nie można wysłać propozycji |
| 337 | no map items found | nie znaleziono pozycji mapy |
| 458 | group members are by default in the everyone role | członkowie grupy są domyślnie zdefiniowani w roli "Każdy" |
| 616 | timeout ejecting agent | przekroczono czas wyrzucenia agenta |
| 1253 | cannot remove owner role | nie można usunąć roli właściciela |
| 1382 | unknown sound requested | zażądano nieznanego dźwięku |
| 1458 | unknown update type | nieznany typ aktualizacji |
| 1488 | no dialog specified | nie określono okna dialogowego |
| 1536 | primitive not for sale | prim nie na sprzedaż |
| 1691 | unable to post event | nie można wysłać zdarzenia |
| 2021 | invalid version provided | dostarczono nieprawidłowej wersji |
| 2087 | no avatars provided | Nie podano awatarów |
| 2169 | insufficient funds | niewystarczające środki |
| 2188 | could not get parcel info data | nie można uzyskać danych o działce |
| 2193 | no avatars to ban or unban | brak awatarów do zbanowania lub odbanowania |
| 2380 | unknown image format provided | podano nieznany format obrazu |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|------------------------------|---|
| 2432 | invalid grab position | nieprawidłowa pozycja uchwytu |
| 2707 | could not find title | nie można znaleźć tytułu |
| 3098 | invalid proposal quorum | nieprawidłowe kworum propozycji |
| 3475 | invalid scale | nieprawidłowa skala |
| 3638 | no data provided | nie podano danych |
| 4541 | group invite not found | nie znaleziono zaproszenia do grupy |
| 4797 | friendship offer not found | nie znaleziono oferty przyjaźni |
| 4994 | unable to decode asset data | nie może zdekodować danych zasobów |
| 5034 | unable to create item | nie można utworzyć przedmiotu |
| 5762 | primitives already linked | prim już połączony |
| 6097 | unknown inventory type | nieznany typ szafy |
| 6330 | no notice found | nie znaleziono powiadomienia |
| 6617 | dialog button not found | nie znaleziono przycisku dialogowego |
| 7140 | no SIML response received | nie otrzymałem odpowiedzi SIML |
| 7168 | timeout getting profile pick | przekroczono czas na pobranie zdjęcia z profilu |
| 7255 | timeout transferring asset | przekroczono czas na transfer zasobu |
| 7457 | no permissions for item | brak uprawnień dla zasobu |
| 7628 | invalid proposal duration | nieprawidłowa propozycja trwania |
| 7703 | general error | błąd ogólny |
| 8169 | SQL execution failed | Błąd wykonania SQL |
| 8241 | could not find classified | Nie można znaleźć kwalifikacji |
| 8339 | no event identifier provided | brak identyfikatora wydarzenia |
| 8411 | could not start process | nie można uruchomić procesu |
| 8653 | command not found | komenda nieznaleziona |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|------------------------------------|--|
| 8842 | unknown tree type | nieznany typ drzewa |
| 8846 | no host provided | nie dostarczono hosta |
| 9111 | timeout getting land users | przekroczono czas na pobranie użytkowników regionu |
| 9348 | unknown animation action | nieznana akcja animacji |
| 9541 | timeout uploading item | przekroczony czas na upload pozycji |
| 9703 | group not configured | grupa nie skonfigurowana |
| 9924 | first life text too large | |
| 9935 | maximum group list length reached | osiągnięto maks. rozmiar grupowej listy |
| 10348 | primitives already delinked | primy już zostały rozłączone |
| 10374 | timeout retrieving estate covenant | przekroczono czas na pobranie tekstu umowy majątku |
| 10522 | failed to read log file | błąd przy odczycie pliku logu |
| 10691 | timeout mapping friend | przekroczono czas na mapowanie przyjaciela |
| 10776 | group schedules exceeded | harmonogramy grup zostały przekroczone |
| 10945 | invalid surface coordinates | nieprawidłowe współrzędne powierzchni |
| 11050 | timeout getting group roles | przekroczono czas na pobranie roli grupy |
| 11229 | timeout getting group titles | przekroczono czas tytułów grupy |
| 11287 | could not create role | nie można utworzyć roli |
| 11502 | not in group | nie w grupie |
| 11910 | no executable file provided | nie dostarczono pliku wykonywalnego |
| 11979 | could not retrieve pick | nie można pobrać obrazka |
| 12320 | unable to upload item data | nie można załadować danych pozycji |
| 12408 | group synchronization failed | błąd przy synchronizacji grupy |
| 12758 | no role name specified | nie podano nazwy roli |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--|--|
| 13030 | no chatlog path provided | nie podano ścieżki do logu czatu |
| 13053 | too many characters for group name | zbyt wiele znaków w nazwie grupy |
| 13296 | invalid binormal vector | nieprawidłowy wektor binormalny |
| 13399 | no access token provided | nie podano tokenu dostępu |
| 13491 | ban would exceed maximum ban list length | ban przekroczyłby maksymalną długość listy banów |
| 13712 | timeout retrieving estate list | przekroczono czas podczas pobierania listy majątku |
| 13764 | status not found | status nie znaleziony |
| 13857 | unexpected item in path | nieoczekiwana pozycja w ścieżce |
| 14337 | unknown restart action | nieznana akcja restartu |
| 14634 | could not eject agent | nie można wyrzucić agenta |
| 14989 | no avatars found | nie znaleziono awatarów |
| 15345 | already in group | już w grupie |
| 15517 | eject needs demote | wyrzucany potrzebuje degradacji |
| 15719 | timeout retrieving group ban list | przekroczono czas na pobranie grupowej listy banów |
| 15964 | transfer would exceed maximum count | transfer przekroczyłby maksymalną liczbę |
| 16233 | invalid asset | nieprawidłowy zasób |
| 16263 | could not retrieve attachments | nie można pobrać załączników |
| 16450 | could not retrieve mute list | nie można pobrać listy wyciszonych |
| 16572 | invalid viewer effect | nieprawidłowy efekt przeglądarki |
| 16667 | empty terrain data | puste dane terenu |
| 16927 | no estate powers for | brak uprawnień majątku dla komendy |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--|--|
| | command | |
| 17019 | unable to set home | nie można ustawić pozycji startowej |
| 17894 | description would exceed maximum size | opis przekroczyłby maksymalny rozmiar |
| 18463 | could not retrieve object media | nie można otrzymać mediów obiektu |
| 18490 | invalid interval | nieprawidłowy interwał |
| 18680 | no pattern provided | nie dostarczono wzoru |
| 18737 | unable to process data | nie można uzyskać danych procesu |
| 19011 | ambiguous path | dwuznaczna ścieżka |
| 19059 | no land rights | brak uprawnień regionu |
| 19143 | timeout meshmerizing object | |
| 19325 | too many characters for group title | zbyt wiele znaków dla tytułu grupy |
| 19343 | unable to retrieve data | nie można pobrać danych |
| 19862 | agent not in group | agent nie znajduje się w grupie |
| 20048 | timeout getting avatar data | przekroczono czas na pobranie danych awatara |
| 20303 | notice does not contain attachment | ogłoszenie nie zawiera załącznika |
| 20541 | timeout getting top scripts | przekroczono czas na pobranie skryptów top |
| 20547 | invalid width | nieprawidłowa szerokość |
| 20822 | too many characters for notice subject | zbyt wiele znaków dla tematu ogłoszenia |
| 20900 | could not leave group | nie można opuścić grupy |
| 20900 | invalid xml path | nieprawidłowa ścieżka xml |
| 21106 | message may not contain HTML | wiadomość nie może zawierać HTML |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|------------------------------------|---|
| 21160 | unable to write file | nie można zapisać pliku |
| 21296 | no topic provided | nie dostarczono tematu |
| 21718 | no location provided | nie dostarczono lokalizacji |
| 21743 | event posting rejected | odrzucono wysłanie wydarzenia |
| 21833 | invalid date | nieprawidłowa data |
| 21894 | AI feature not enabled | Cecha AI nie jest włączona |
| 22119 | no proposal to reject | brak propozycji do wyrzucenia |
| 22576 | timeout requesting to set home | przekroczono czas na żądanie ustawienia punktu startowego |
| 22655 | expected folder as target | oczekiwano folderu jako cel |
| 22693 | primitive not found | prim nie znaleziony |
| 22733 | unable to join group chat | nie można połączyć się z czatem grupowym |
| 22737 | object not found | obiekt nie znaleziony |
| 22786 | conference member not found | członek konferencji nie znaleziony |
| 22970 | timeout getting profile classified | przekroczono czas na pobranie klasyfikacji grupy |
| 23114 | timeout waiting for sensor | przekroczono czas na oczekiwanie na sensor |
| 23123 | nucleus server error | błąd serwera Nucleus |
| 23309 | friend offline | przyjaciół jest offline |
| 23364 | timeout getting script state | przekroczono czas na pobranie stanu skryptu |
| 23570 | invalid schedules provided | dostarczono nieprawidłowe harmonogramy |
| 23716 | classified not found | klasyfikacja nieznaleziona |
| 23805 | no session specified | nie dostarczono sesji |
| 23926 | unable to delete event | nie można skasować wydarzenia |
| 23932 | no position provided | nie dostarczono pozycji |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--|---|
| 24939 | group not open | grupa nie jest publiczna |
| 25003 | no category provided | nie dostarczono kategorii |
| 25119 | unable to retrieve transactions | nie można pobrać transakcji |
| 25252 | no SQL string provided | nie podano zapytania SQL |
| 25329 | failed rezzing child primitive | błąd podczas rezzowania dziecka primu |
| 25420 | could not retrieve classified | nie można pobrać klasyfikacji |
| 25426 | timeout getting group role members | przekroczono czas na pobranie członków roli grupy |
| 25897 | unknown estate list action | nieznana akcja listy majątku |
| 25984 | inventory item not found | pozycja szafy nie znaleziona |
| 26178 | too many characters for notice message | zbyt wiele znaków dla wiadomości ogłoszenia |
| 26356 | timeout retrieving group notices | przekroczono czas na otrzymanie ogłoszeń grupowych |
| 26393 | timeout updating mute list | przekroczono czas na aktualizację listy wyciszonych |
| 26623 | unable to create folder | nie można utworzyć folderu |
| 26715 | mute entry not found | nie znaleziono wpisu wyciszonego |
| 26749 | timeout modifying group ban list | przekroczono czas na modyfikację listy grupowego banu |
| 27605 | cannot eject owners | nie można wyrzucić właścicieli |
| 27910 | timeout getting parcel list | przekroczono czas na pobranie listy działki |
| 28002 | group not found | grupa nie została znaleziona |
| 28087 | no database file configured | plik bazy danych nie został skonfigurowany |
| 28126 | auto return time outside limit range | czas autozwracania jest poza zakresem limitu |
| 28247 | maximum amount of | osiągnięto maksymalną ilość klasyfikatorów |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|---|---|
| | classifieds reached | |
| 28353 | invalid feed provided | dostarczono nieprawidłową kanału |
| 28429 | unknown move action | nieznana akcja przesuwania |
| 28487 | invalid normal vector | nieprawidłowy normalny wektor |
| 28613 | could not sit | nie można usiąść |
| 28625 | maximum manager list length reached | osiągnięto maksymalną długość listy zarządców |
| 28866 | no permissions provided | nie dostarczono uprawnień |
| 28891 | invalid terraform action | nieprawidłowa akcja deformowania |
| 29345 | empty attachments | puste załączniki |
| 29438 | no task specified | nie określono zadania |
| 29512 | empty wearables | puste akcesoria do noszenia |
| 29530 | could not read XML file | nie można odczytać pliku XML |
| 29745 | attachments would exceed maximum attachment limit | załączniki przekroczyłyby maksymalny limit liczby załączników |
| 29947 | unknown agent access | nieznany dostęp agenta |
| 30129 | could not get current groups | nie można pobrać bieżących grup |
| 30207 | friend not found | przyjaciół nie został znaleziony |
| 30293 | name too large | nazwa jest zbyt długa |
| 30384 | timeout downloading terrain | przekroczono czas na ściągnięcie terenu |
| 30473 | no group power for command | brak uprawnień grupowych dla komendy |
| 30556 | no description for status | brak opisu dla statusu |
| 31049 | no button index specified | nie określono indeksu przycisku |
| 31126 | too many characters for proposal message | zbyt wiele znaków dla propozycji wiadomości |
| 31237 | timeout leaving group | przekroczono czas na opuszczenie grupy |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--|--|
| 31267 | agent has been banned | agent nie został zbanowany |
| 31381 | timeout waiting for execution | przekroczono czas na wykonanie |
| 31417 | too many characters for group role description | zbyt wiele znaków dla opisu roli grupy |
| 31493 | unable to go home | nie można teleportować się na punkt startowy |
| 31868 | no database path provided | nie dostarczono ścieżki do bazy danych |
| 32157 | no consumer key provided | nie podano klucza konsumenta |
| 32164 | teleport failed | błąd podczas teleportacji |
| 32355 | execution returned no result | wykonanie nie zwróciło żadnych wyników |
| 32362 | timeout getting parcels | przekroczono czas na pobranie działek |
| 32366 | the agent already is a friend | agent jest już oznaczony jako przyjaciel |
| 32404 | timeout creating group | przekroczono czas na tworzenie grupy |
| 32453 | could not get parcel info data | nie można pobrać informacji o działce |
| 32472 | role not found | rola nie została znaleziona |
| 32528 | agent is soft banned | agent został zbanowany (softban) |
| 32709 | unknown wearable type | nieznany typ akcesorium do noszenia |
| 32923 | timeout getting profile | przekroczono czas na pobranie profilu |
| 33047 | failed rezzing root primitive | błąd podczas rezzowania korzenia primu |
| 33381 | unable to reach events page | nie można pobrać strony wydarzeń |
| 33413 | cannot delete the everyone role | nie można skasować roli "Każdy" |
| 33564 | unable to save configuration | Nie można zapisać konfiguracji |
| 33714 | could not retrieve wearables | Nie można pobrać akcesoriów do noszenia |
| 33717 | no flags provided | Nie podano flag |
| 33821 | no notice provided | Nie podano ogłoszenia |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--|--|
| 33994 | unable to retrieve form parameters | Nie można otrzymać parametrów formularza |
| 34084 | timeout getting group roles members | Przekroczono czas na pobranie członków ról grupy |
| 34379 | proposal rejected | Odrzucono propozycję |
| 34749 | the specified folder contains no equipable items | Określony folder nie zawiera wyposażenia |
| 34869 | already subscribed to feed | już subskrybowano kanał |
| 34964 | timeout getting region | Przekroczono czas na pobranie regionu |
| 35198 | could not set display name | nie można ustawić nazwy wyświetlanej |
| 35277 | notecard message body too large | wiadomość notatki jest zbyt duża |
| 35316 | parcel not for sale | działka nie jest na sprzedaż |
| 35392 | could not join group | nie można dołączyć do grupy |
| 35447 | region not found | region nie został znaleziony |
| 36068 | type can only be voice or text | typ może być tylko: głosowy lub tekstowy |
| 36121 | invalid offset | nieprawidłowy offset |
| 36123 | teleport throttled | osiągnięto teleport |
| 36616 | invalid item type | nieprawidłowy typ pozycji |
| 36675 | no server provided | nie podano serwera |
| 36684 | unable to upload item | nie można załadować pozycji |
| 36896 | no index provided | nie podano indeksu |
| 37211 | timeout creating item | przekroczono czas na utworzenie pozycji |
| 37470 | timeout getting group members | przekroczono czas na pobranie członków grupy |
| 37559 | destination too close | miejsce docelowe jest zbyt blisko |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|------------------------------------|---|
| 37841 | could not get primitive properties | nie można pobrać właściwości primu |
| 38125 | unable to obtain money balance | nie można pobrać salda konta |
| 38184 | invalid price | nieprawidłowa cena |
| 38271 | script compilation failed | błąd podczas kompilacji skryptu |
| 38278 | unable to read file | nie można odczytać pliku |
| 38289 | timeout waiting for display name | przekroczono czas na nazwę wyświetlaną |
| 38504 | invalid permissions | nieprawidłowe uprawnienia |
| 38609 | timeout changing links | przekroczono czas na zmianę linkowania |
| 38624 | invalid secret provided | podano nieprawidłowy sekret |
| 38798 | invalid face specified | podano nieprawidłową stronę |
| 38858 | effect not found | efekt nie został znaleziony |
| 38931 | no label or index specified | nie podano etykiety lub indeksu |
| 38945 | no source specified | nie podano źródła |
| 39016 | timeout getting role powers | przekroczono czas na pobranie uprawnień roli |
| 39359 | unknown type | nieznany typ |
| 39391 | expected item as source | oczekiwano pozycji jako źródła |
| 39613 | maximum number of groups reached | osiągnięto maksymalną liczbę grup |
| 39647 | mute entry already exists | wyciszony wpis już istnieje |
| 39787 | timeout reaching destination | przekroczono czas na osiągnięcie miejsca docelowego |
| 39921 | could not set agent access | nie można ustawić dostępu agentowi |
| 40491 | no transactions found | nie znaleziono transakcji |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|---|---|
| 40665 | no name provided | nie podano nazwy |
| 40762 | no consumer secret provided | nie podano sekret konsumenta |
| 40773 | invalid number of items specified | podano nieprawidłową liczbę pozycji |
| 40901 | SIML not enabled | SIML nie jest włączony |
| 40908 | cannot delete a group member from the everyone role | nie można skasować członka grupy z roli "Każdy" |
| 41007 | no secret provided | nie podano sekretu" |
| 41190 | invalid terraform brush | nie prawidłowe malowanie terenu |
| 41211 | no terraform brush specified | nie określono malowania terenu |
| 41256 | maximum number of roles exceeded | osiągnięto maksymalną liczbę roli |
| 41257 | unable to post divorce | nie można wysłać rozwodu |
| 41574 | timeout rezzing primitive | przekroczono czas na zrezzowanie primu |
| 41612 | unknown date time stamp | nieznany stempel czasu i daty |
| 41676 | unknown top type | nieznany typ szczytu |
| 41810 | invalid proposal majority | nieprawidłowa większość propozycji |
| 41969 | unable to start conference | nie można uruchomić konferencji |
| 42051 | unknown access list type | nieznany typ listy dostępu |
| 42140 | group already configured | grupa już skonfigurowana |
| 42216 | no channel specified | nie określono kanału |
| 42248 | unable to save Corrade configuration | nie można zapisać konfiguracji Corrade |
| 42249 | inventory offer not found | nie można znaleźć oferty szafy |
| 42351 | timeout starting conference | przekroczono czas na uruchomienie konferencji |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--------------------------------------|---|
| 42536 | maximum user list length reached | osiągnięto maksymalną długość listy użytkownika |
| 42798 | timeout retrieving notice | przekroczono czas na pobranie ogłoszenia |
| 42903 | unknown image format requested | nieznane żądanie formatu obrazu |
| 43003 | unknown syntax type | nieznany typ składni |
| 43156 | no item specified | nie określono pozycji |
| 43615 | setting permissions failed | błąd podczas ustawiania uprawnień |
| 43671 | unable to revoke proposal | nie można odwołać propozycji |
| 43683 | description too large | opis zbyt duży |
| 43713 | fly action can only be start or stop | akcja lotu może być tylko: start lub stop |
| 43767 | proposal already sent | propozycja już została wysłana |
| 43780 | timeout during teleport | przekroczenie czasu podczas teleportowania |
| 43898 | session not found | sesja nie została znaleziona |
| 43982 | invalid amount | nieprawidłowa ilość |
| 43982 | no folder specified | nie określono folderu |
| 44059 | unable to get event identifier | nie można pobrać identyfikatora wydarzenia |
| 44397 | timeout uploading item data | przekroczono czas na załadowanie danych pozycji |
| 44447 | unknown directory search type | nieznany typ wyszukiwania w katalogu |
| 44537 | unknown horde balancer | |
| 44554 | primitives not in same region | prymy nie są w tym samym regionie |
| 45074 | link would exceed maximum link limit | łączenia przekroczyłyby maksymalny limit łączenia |
| 45173 | invalid angle provided | podano nieprawidłowy kąt |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|---|---|
| 45364 | could not set script state | nie można ustawić stanu skryptu |
| 45568 | could not retrieve group ban list | nie można pobrać grupowej listy banów |
| 46316 | timeout requesting sit | przekroczono czas żądania siadania |
| 46612 | no partner found | nie znaleziono partnera |
| 46804 | could not send message | nie można wysłać wiadomości |
| 46858 | timeout receiving SIML response | przekroczono czas na otrzymanie wiadomości SIML |
| 46942 | empty classified name | pusta nazwa klasyfikatora |
| 46990 | unknown estate list | nieznana lista majątku |
| 47101 | no message provided | nie podano wiadomości |
| 47172 | timeout getting top colliders | |
| 47350 | no type provided | nie podano typu |
| 47469 | timeout getting picks | przekroczono czas na pobranie obrazu |
| 47571 | too many or too few characters for display name | zbyt mało znaków dla nazwy wyświetlanej |
| 47624 | invalid pay target | nieprawidłowy cel płatności |
| 47808 | cannot remove user from owner role | nie można usunąć agenta z roli "Właściciel" |
| 48110 | no history found | nie znaleziono historii |
| 48467 | no Corrade permissions | brak uprawnień Corrade |
| 48775 | no effect UUID provided | nie podano efektu UUID |
| 48899 | empty pick name | pusta nazwa obrazu |
| 49113 | pick not found | obraz nie znaleziono |
| 49640 | notification not allowed | powiadomienie nie jest dozwolone |
| 49722 | failed to get display name | błąd podczas pobierania nazwy wyświetlanej |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|---|---|
| 50003 | unable to reject proposal | nie można odrzucić propozycji |
| 50203 | no permission for parcel | brak uprawnień dla działki |
| 50218 | item is not a script | pozycja nie jest skryptem |
| 50405 | maximum amount of picks reached | maksymalna liczba obrazów została osiągnięta |
| 50593 | maximum ban list length reached | osiągnięto maksymalną |
| 51050 | script permission request not found | żądanie uprawnień skryptu nie został znaleziony |
| 51086 | no name or UUID provided | nie podano nazwy lub UUID |
| 52299 | too many or too few characters in message | zbyt mało znaków dla wiadomości |
| 52751 | timeout requesting price | przekroczono czas żądania ceny |
| 53059 | could not update parcel list | nie można zaktualizować listy działki |
| 53066 | path not found | nie znaleziono ścieżki |
| 53274 | unknown material type | nieznany typ materiału |
| 53494 | too many characters for event description | zbyt dużo znaków dla opisu wydarzenia |
| 53549 | no description provided | nie podano opisu |
| 53829 | timeout getting group account summary | przekroczono czas dla pobierania podsumowania konta grupy |
| 53947 | invalid days | nieprawidłowe dni |
| 54084 | no peers matching context | |
| 54154 | scale would exceed building constraints | skala może przekroczyć granice budowania |
| 54214 | could not create group | nie można utworzyć grupy |
| 54450 | unable to agree to ToS | nie można wyrazić zgody ToS |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|----------------------------------|---|
| 54456 | unknown effect | nieznany efekt |
| 54528 | description may not contain HTML | opis nie może zawierać HTML |
| 54668 | unable to authenticate | nie można uwierzytelnić |
| 54854 | avatar not in range | awatar nie jest |
| 54956 | timeout retrieving estate info | przekroczono czas dla zasięgu |
| 55051 | message too long | wiadomość jest zbyt długa |
| 55091 | no access token secret provided | nie podano sekretu tokenu dostępowego |
| 55110 | unknown mute type | nieznany typ wyciszenia |
| 55394 | no matching dialog found | nie znaleziono pasującego dialogu |
| 55755 | MQTT publish failed | błąd podczas publikacji MQTT |
| 55979 | unknown direction | nieznany kierunek |
| 56094 | no schedule found | nie znaleziono harmonogramu |
| 56345 | unable to reach partnership page | nie można pobrać strony partnerskiej |
| 56379 | second life text too large | |
| 56901 | no terraform action specified | nie określono akcji terraformu terenu |
| 57005 | timeout uploading terrain | przekroczono czas na załadowanie tematu |
| 57085 | failed to download asset | błąd podczas ściągania zasobu |
| 57196 | no duration provided | nie podano czasu trwania |
| 57961 | agent not found | agent nie znaleziony |
| 58183 | invalid rotation | nieprawidłowa rotacja |
| 58212 | platform not supported | platforma nie jest obsługiwana |
| 58478 | could not get parcel resources | nie można pobrać zasobów działki |
| 58493 | default folder not found | domyślny folder nie został znaleziony |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|--|---|
| 58619 | could not terraform | nie można terraformować terenu |
| 58751 | name may not contain HTML | nazwa nie może zawierać HTML |
| 58870 | unable to divorce | nie można się rozwieść |
| 59048 | invalid asset data | nieprawidłowe dane zasobu |
| 59103 | timeout getting classifieds | przekroczenie czasu podczas pobierania klasyfikacji |
| 59282 | no path provided | nie podano ścieżki |
| 59524 | invalid position | nieprawidłowa pozycja |
| 59526 | unknown action | nieznana akcja |
| 60025 | unknown asset type | nieznany typ zasobu |
| 60073 | teleport lure not found | nie znaleziono zaproszenia do teleportacji |
| 60269 | asset upload failed | błąd podczas załadowania zasobu |
| 60427 | could not get current outfit folder | nie można pobrać bieżącego folderu ubrania |
| 60515 | position would exceed maximum rez altitude | pozycja przekroczyłaby max. wysokość rezzowania |
| 61018 | folder not found | folder nie znaleziony |
| 61067 | unable to convert to requested format | nie można przekonwertować do żadanego formatu |
| 61113 | unknown entity | nieznana jednostka |
| 61317 | invalid workers provided | |
| 61473 | invalid status supplied | podano nieprawidłowy status |
| 61492 | unknown language | nieznany status |
| 61983 | unable to accept proposal | nie można zaakceptować propozycji |
| 62130 | invalid texture coordinates | nieprawidłowe koordynaty tekstury |
| 62531 | could not get land resources | nie można pobrać zasobów regionu |

| Error Code | Message (English) | Wiadomość (Polski) |
|------------|---|---|
| 62646 | effect UUID belongs to different effect | efekt UUID należy do innego efektu |
| 62753 | could not rebake | nie można odświeżyć |
| 63024 | unable to load configuration | nie można załadować konfiguracji |
| 63486 | invalid height | nieprawidłowa wysokość |
| 63597 | no date provided | nie podano daty |
| 63713 | timeout joining group | przekroczono czas na dołączenie do grupy |
| 63915 | no time provided | nie podano czasu |
| 64123 | invalid mute target | nieprawidłowy cel wyciszenia |
| 64179 | unable to serialize primitive | nie można serializować danych |
| 64368 | unknown grass type | nieznany typ trawy |
| 64390 | could not find parcel | nie można znaleźć działki |
| 64420 | primitives are children of object | primy są dziećmi obiektu |
| 64423 | timeout getting folder contents | przekroczony czas pobierania zawartości folderu |
| 64450 | unknown sift | nieznany przesiewu |
| 64698 | could not compile regular expression | nie można skompilować wyrażenia regularnego |
| 64868 | invalid proposal text | nieprawidłowy tekst propozycji |
| 65003 | friend does not allow mapping | przyjaciół nie pozwala na mapowanie |
| 65101 | no search text provided | nie podano tekstu wyszukiwania |
| 65241 | no title provided | nie podano tytułu |
| 65303 | invalid url provided | podano nieprawidłowy url |
| 65327 | invalid notification types | nieprawidłowe typy powiadomienia |

Tabela 2 - Kody błędów

4.12. Ograniczenia w Second Life

Świat Second Life ma swoje ograniczenia, które zostały narzucone przez zespół rozwijający ten projekt, aby przebywanie w tym świecie było płynne i nie dokuczały pewne problemy.

Te ograniczenia należy wziąć pod uwagę podczas budowania własnego bota:

- Liczba primów
Jeden region może udostępniać od 15,000 do 30,000 primów dla wszystkich. Musimy pomyśleć, ile primów chcemy (włącznie z rezerwą) oraz minimalny i maksymalny rozmiar działki w regionie. Rozmiar działki oraz liczba primów jest ze sobą powiązane – im większa działka tym więcej primów jest do dyspozycji, jednakże ta liczba nie może przekroczyć maksymalnej liczby primów regionu. Średnio za 1024 m² działki z 469 primami zapłacimy ok. 2.00 USD, wszystko zależy u jakiego dzierżawcy będziemy wynajmować działkę.
- Maksymalna długość wiadomości to 1024 znaki, dłuższy tekst zostanie obcięty do tego rozmiaru,
- `llSendMessage()` - max. liczba wiadomości wysłanych przez wszystkie obiekty to: 2500 wiadomości / 30 min. co daje ok. 83 wiadomości/1 s,
- Maksymalny rozmiar skryptu LSL w trybie mono: 64KB,
- Maksymalne współrzędne:
 - X – 256 m,
 - Y – 256 m,
 - Z – 4,096 m.

4.13. Komunikacja z botem

Na początku zajmijmy się komunikacją z botem przez skrypt LSL, który tworzymy wewnątrz primu. Wszystko też zależy od tego czy ten prim będzie częścią ubrania bota czy też będzie stał gdzieś na działce.

Jeżeli prim będzie jako część ubrania bota, to ten prim musi być własnością bota (tworzymy prim jako bot – najlepiej zalogować się jako bot). Ten prim zawsze będzie przy bocie. Jedynym minusem tego rozwiązania jest to, że kiedy bot zostanie teleportowany podczas restartu regionu do innej lokacji, gdzie jest zakaz używania skryptów przez inne awatary, wówczas skrypt z primu będącego częścią ubrania awatara się nie uruchomi. Aby poradzić sobie z tym problemem najlepiej stworzyć skrypt np. w PHP, który będzie uruchamiany np. co 5-15 minut przez systemowy harmonogram zadań, który będzie sprawdzał, czy bot jest na właściwej działce [skrypt jest dołączony w materiałach dodatkowych]. Przy tym rozwiązaniu najlepiej skorzystać z polecenia `llownerSay()`¹⁵, które przesyła wiadomość bezpośrednio do właściciela primu i nie posiada w zasadzie żadnych limitów w ilości przesłanych wiadomości.

Jeżeli prim będzie stał gdzieś na działce, to najlepiej w skrypcie skorzystać z funkcji `llinstantmessage()`¹⁶, która wysyła wiadomość prywatną do bota, bez względu na to gdzie ten się w Second Life. Ta funkcja ma nałożony limit liczby wiadomości, która wynosi 2500/30 min. i dotyczy wszystkich obiektów z danego regionu. Jeżeli region zostanie zresetowany to może wysłać info do bota, aby ten teleportował się z powrotem na działkę. Minusem tego rozwiązania jest to, że w momencie kiedy region jest restartowany nie mamy dostępu do tego primu z tym skrypcem. Warto tutaj jeszcze wspomnieć, że jeżeli wybierzemy to rozwiązanie to w skrypcie najlepiej sprawdzać co jakiś czas, czy bot rzeczywiście jest obecny w Second Life.

Jakie rozwiązanie tutaj wybrać? To w zasadzie zależy od nas. Ja prim ze skryptami podłączyłem jako niewidoczne ubranie bota, nie sprawia mi żaden trud ich edycji, po prostu loguję się ręcznie jako bot, edytuję, zapisuję, wyłączam przeglądarkę, włączam Corrade i sprawdzam czy działa zgodnie z moimi oczekiwaniami, jeżeli nie, to wyłączam program i powtarzam czynności. Natomiast oprócz tego są również primy, które korzystają z tego drugiego rozwiązania.

Kolejną sprawą do omówienia jest budowa serca i mózgu naszego bota w wirtualnym świecie. Oskryptowanie wszystkiego w LSL w Second Life jest nieopłacalne i szybko na nas się zemści, ponieważ programowanie w LSL jest bardzo ograniczone, a pojedynczy skrypt nie może przekroczyć 64KB w trybie kompilowania Mono, gdzie 16 KB jest przeznaczane na: bytecode, heap, stack.

15 <http://wiki.secondlife.com/wiki/LlownerSay>

16 <http://wiki.secondlife.com/wiki/Llinstantmessage>

Zaletą takiego ograniczenia skryptów jest ograniczenie zasobochłonności skryptów, ponieważ na jednym serwerze jest uruchomionych kilka regionów, a na każdym z nich działa niezliczona liczba skryptów. To powoduje, że wszystko w miarę płynnie działa.

Wadą jest jednak brak możliwości tworzenia skryptów, które wykonywałyby zaawansowane rzeczy w wirtualnym świecie. Możesz albo rozbić duży skrypt na kolejne mniejsze skrypty i spróbować połączyć je ze sobą, co w Second Life byłoby pracochłonnym zadaniem lub przenieść dużą część funkcjonalności bota poza Second Life i tylko w wirtualnym świecie otrzymywać stan zrealizowanych części funkcjonalności. Ma to również zalety mniejszych plików LSL w primie i mniejsze ryzyko pojawienia się błędu związanego z przekroczeniem limitu pamięci dla skompilowanego kodu (np. 'Błąd wykonania skryptu: Kolizja stosu', 'BŁĄD: zestaw kodu bajtowego, niepowodzenie - brak pamięci'). Niezależnie od tego, którą opcję wybierzemy, pamiętajmy, że prim nie może zawierać wielu skryptów LSL - spowodują one bardzo duże obciążenie serwera, a gdy manager regionu lub obsługa Second Life to wykryje, wtedy może nam nakazać odchudzanie skryptów w primie lub zlecić opuszczenie regionu.

Dlatego najlepszym pomysłem jest przeniesienie całego procesu przetwarzania przez bota poza środowisko Second Life, a w samym Second Life w skryptach LSL tylko pobierać status wykonania przez środowisko zewnętrzne. W takiej sytuacji przychodzi z pomocą komenda 'execute' z API bota, które pozwala na wykonanie dowolnego polecenia z parametrami w środowisku systemu, na którym jest uruchomiony bot. Dodatkowo bot obsługuje żądania wywoływania poleceń API bota przez HTTP, MQTT, TCP czy WebSockets, jednakże HTTP jest najczęściej używany. Co to nam daje? Poprzez kombinację tych dwóch właściwości możemy uruchamiać powłokę konsoli dowolnego języka programowania, np. Python, Java, C#, C++, PHP, itp., który będzie pozwalał na wywoływanie żądań HTTP do bota i odbieranie od niego odpowiedzi. Ma to też swój kolejny plus w postaci możliwości podłączenia przetwarzania danych w chmurze, itd.

4.14. Praktyczne użycie bota na przykładzie wysłania ogłoszenia grupowego.

Czas, aby nasz bot zrobił coś konkretnego oprócz podawania tylko suchych informacji w tej książce. Wszystkie polecenia znajdziemy na stronie grimore.org¹⁷.

Pomimo, że wybierzemy jedno polecenie, to omówimy wszystkie przypadki użycia, z wykorzystaniem dodatkowych funkcji przygotowanymi przez WAS oraz moimi na przykładowych skryptach napisanych w LSL i PHP, wysyłających dane do bota przy pomocy WAS i JSON.

Naszym przykładowym poleceniem będzie 'notice'¹⁸, które pozwala na wysłanie ogłoszenia grupowego w Second Life. Przechodzimy zatem do strony szczegółowego opisu polecenia, która jest prosta do czytania i zrozumienia. Mamy do dyspozycji: listę zmian, przykłady użycia, tabelę z parametrami, itd.

Na początku przyjrzyjmy się poniższej tabelce.

| Command | Required Parameters | Required Corrade Permissions | Required Group Abilities | Example |
|---------|-------------------------|------------------------------|--------------------------|---|
| notice | group, password, action | group | Notices+Send Notices | <pre>llInstantMessage(CORRADE, wasKeyValueEncode(["command", "notice", "group", wasURLEscape(GROUP), "password", wasURLEscape(PASSWORD), "action", "send", "subject", "My store", "message", "Store is updated!"]));</pre> |

Rysunek 4.25 - Polecenie 'notice' - wymagane parametry

Widzimy tutaj wymagane parametry: grupa, hasło (one będą stale się powtarzać przy innych poleceniach), action (jego szczegółowy opis jest w kolejnej tabelce znajdującej się na dole strony). Kolejna kolumna to 'Wymagane uprawnienia Corrade'. Te uprawnienia muszą być włączone w grupie, w której znajduje się bot w konfiguracji Corrade, aby móc operować danym poleceniem. Najłatwiej jest otworzenie pliku Configuration.xml i sprawdzenie czy dla danej grupy jest odpowiedni wpis. Jeżeli nie ma takiego wpisu, to możemy sami dopisać dane uprawnienie do pliku, po czym zapisać plik i zrestartować bota.

¹⁷ https://grimore.org/secondlife/scripted_agents/corrade/api/commands

¹⁸ https://grimore.org/secondlife/scripted_agents/corrade/api/commands/notice

```

<Password>63</Password>
<Name>Group</Name>
<Permissions>
  <Permission>group</Permission>
  <Permission>inventory</Permission>
  <Permission>movement</Permission>
  <Permission>grooming</Permission>
  <Permission>interact</Permission>
  <Permission>notifications</Permission>
  <Permission>talk</Permission>
  <Permission>group</Permission>
  <Permission>land</Permission>
  <Permission>mute</Permission>
  <Permission>execute</Permission>
  <Permission>bridge</Permission>
  <Permission>friendship</Permission>
  <Permission>database</Permission>
  <Permission>system</Permission>
  <Permission>directory</Permission>
  <Permission>economy</Permission>
</Permissions>

```

Rysunek 4.26 - Uprawnienia w pliku konfiguracyjnym

Drugim sposobem jest konfiguracja przez stronę Nucleus, gdzie w danej grupie zaznaczamy brakujące uprawnienia, przy czym po zapisaniu zmian, Nucleus sam przeładuje konfigurację bota, więc nie trzeba samemu manualnie restartować bota.

Kolejna kolumna, która może się pojawić to 'Wymagane zdolności grupowe'. Informują one, jakie uprawnienia powinien mieć bot po stronie Second Life do wykonania danej akcji. Zazwyczaj dotyczy to grupy czy roli grupy do której bot został przypisany.

Grupowe uprawnienia będą również wymagane przy poleceniach:

- ban,
- setgroupdata,
- softban,
- batcheject,
- moderate,
- notice,
- eject,
- setrolepowers,
- createrole.

U nas musimy przypisać botowi możliwość wysłania ogłoszeń grupowych. Najlepiej stworzyć nową rolę, do której przypisujemy wyłącznie bota i w tej roli włączamy dla niego wymagane uprawnienia. Ostatnia kolumna zawiera przykład użycia komendy.

Spójrzmy teraz na kolejną tabelkę.

Mamy tutaj podane wszystkie parametry i podparametry dla danego polecenia. Czytamy tabelę od lewej strony.

W pierwszej kolumnie mamy parametry. Dla naszego polecenia jest tylko jeden, o nazwie 'action', który jest wymagany podczas przesyłania do bota.

W drugiej kolumnie mamy możliwe wartości, które zostały przypisane do 'action', dla naszego polecenia są to: send (wysyłanie ogłoszenia), list (wyświetla wszystkie ogłoszenia grupowe), accept (przyjmuje załącznik z ogłoszenia), decline (odrzuca załącznik z ogłoszenia).

W naszym przypadku wybieramy 'send', bo będziemy wysyłać ogłoszenie.

Dalej mamy podane kolejne parametry możliwe do ustawienia - message (wiadomość ogłoszenia), subject (temat ogłoszenia), item (załącznik do ogłoszenia), permissions (uprawnienia do załącznika ogłoszenia). Możemy wybrać te, które nam pasują. My wybieramy 'message' i 'subject'.

Nasz kod LSL z wysyłaniem danych z użyciem WAS będzie wyglądał tak:


```
key ownerID = "YOUROWNERID"
key CORRADE;
string PASSWORD;
string GROUP;
string URL = "";
string TAG;
string val;
list lmessage;
string sjump = "";
string body;
key query;

integer line;
list tuples = [];

setDebug(string msg)
{
    llInstantMessage(ownerID, "["+ TAG +" ] " + msg);
}

integer StringMatch(list lwhatsearch, string sinwhere){
    sinwhere = llToLower(sinwhere);
    sinwhere = strReplace(sinwhere, "!", "");
    sinwhere = strReplace(sinwhere, "?", "");

    list lsinwhere = llParseString2List(sinwhere,[" "],[]);

    integer imatch = llGetListLength(lwhatsearch);
    integer imatched = 0;

    integer i=0;
    for(i=0; i<imatch; i++){

        list lwords = llParseString2List(llList2String(lwhatsearch,i),["|"],[]);

        integer j=0;
        for(j=0; j<llGetListLength(lwords); j++){
            if(llListFindList(lsinwhere, [llList2String(lwords,j)]) >= 0){
                imatched++;
            }
        }
    }
    if(imatch == imatched){
        return 1;
    }else{
        return 0;
    }
}
```

```

string strReplace(string str, string search, string replace) {
    return llDumpList2String(llParseStringKeepNulls((str = "") + str, [search], []),
replace);
}

list wasCSVToList(string csv)
{
    list l = [];
    list s = [];
    string m = "";
    do
    {
        string a = llGetSubString(csv, 0, 0);
        csv = llDeleteSubString(csv, 0, 0);
        if(a == ",")
        {
            if(llList2String(s, -1) != "\\")
            {
                l += m;
                m = "";
                jump continue;
            }
            m += a;
            jump continue;
        }
        if(a == "\"" && llGetSubString(csv, 0, 0) == a)
        {
            m += a;
            csv = llDeleteSubString(csv, 0, 0);
            jump continue;
        }
        if(a == "\\")
        {
            if(llList2String(s, -1) != a)
            {
                s += a;
                jump continue;
            }
            s = llDeleteSubList(s, -1, -1);
            jump continue;
        }
        m += a;
        @continue;
    } while(csv != "");
    return l + m;
}

string wasKeyValueGet(string k, string data)
{

```

```

if(llStringLength(data) == 0) return "";
if(llStringLength(k) == 0) return "";
list a = llParseStringKeepNulls(data, ["&", "="], []);
integer i = llListFindList(llList2ListStrided(a, 0, -1, 2), [ k ]);
if(i != -1){
    string ret = llList2String(wasCSVToList(wasURLUnescape(llList2String(a, 2*i+1))),
0);
    ret = strReplace(ret, "\\r", "");
    ret = strReplace(ret, "\\n", "");
    ret = llStringTrim(ret, STRING_TRIM);

    return ret;
}
return "";
}

string wasKeyValueEncode(list data)
{
    list k = llList2ListStrided(data, 0, -1, 2);
    list v = llList2ListStrided(llDeleteSubList(data, 0, 0), 0, -1, 2);
    data = [];
    do
    {
        data += llList2String(k, 0) + "=" + llList2String(v, 0);
        k = llDeleteSubList(k, 0, 0);
        v = llDeleteSubList(v, 0, 0);
    } while(llGetListLength(k) != 0);
    return llDumpList2String(data, "&");
}

string wasURLEscape(string i)
{
    string o = "";
    do
    {
        string c = llGetSubString(i, 0, 0);
        i = llDeleteSubString(i, 0, 0);
        if(c == "") jump continue;
        if(c == " ")
        {
            o += "+";
            jump continue;
        }
        if(c == "\n")
        {
            o += "%0D" + llEscapeURL(c);
            jump continue;
        }
        o += llEscapeURL(c);
    }
}

```

```
@continue;
} while(i != "");
return o;
}

string wasURLUnescape(string i)
{
    return llUnescapeURL( llDumpList2String( llParseStringKeepNulls(
llDumpList2String( llParseStringKeepNulls( i, ["+"], [] ), " " ), ["%0D%0A"], [] ),
"\n" ) );
}

default
{
    state_entry()
    {
        state ReadConfigurationNotecard;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state ReadConfigurationNotecard
{
    state_entry()
    {
        if(llGetInventoryType("configuration") != INVENTORY_NOTECARD)
        {
            setDebug("Sorry, could not find a configuration inventory notecard.");
            return;
        }
        line = 0;
        llGetNotecardLine("configuration", line);
    }

    dataserver(key id, string data)
    {
        if(data == EOF)
```

```

{
    if(llGetListLength(tuples) % 2 != 0)
    {
        setDebug("Error in configuration notecard.");
        return;
    }

    CORRADE = llList2Key(tuples, llListFindList(tuples, ["BOTID"])+1);

    if(CORRADE == NULL_KEY)
    {
        setDebug("Error in configuration notecard: BOT ID KEY");
        return;
    }

    GROUP = llList2String(tuples, llListFindList(tuples, ["GROUP"])+1);

    if(GROUP == "")
    {
        setDebug("Error in configuration notecard: GROUP");
        return;
    }

    PASSWORD = llList2String(tuples, llListFindList(tuples, ["PASSWORD"])+1);

    if(PASSWORD == "")
    {
        setDebug("Error in configuration notecard: PASSWORD");
        return;
    }

    state url;
}
if(data == "") jump continue;
integer i = llSubStringIndex(data, "#");
if(i != -1) data = llDeleteSubString(data, i, -1);
list o = llParseStringKeepNulls(data, ["="], []);
string k = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String(o,0), STRING_TRIM), ["\""], []), "\"");
string v = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String( o, 1 ), STRING_TRIM), ["\""], [] ), "\"");
if(k == "" || v == "") jump continue;
tuples += k;
tuples += v;
@continue;
llGetNotecardLine("configuration", ++line);
}

changed(integer change)

```

```
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}

state url
{
    state_entry()
    {
        if ((GROUP == "BOTID") || (GROUP == "GROUP") || (GROUP == "PASSWORD")){
            llResetScript();
            return;
        }

        if ((PASSWORD == "BOTID") || (PASSWORD == "GROUP") || (PASSWORD ==
"PASSWORD")){
            llResetScript();
            return;
        }

        llSetTimerEvent(0.5);
    }

    timer(){
        llSetTimerEvent(60);
        llRequestURL();
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");
        if(method != URL_REQUEST_GRANTED) return;
        URL = body;
        state CheckBotAvailable;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }
}
```

```
    }
}

on_rez(integer start_param)
{
    llResetScript();
}

}

state CheckBotAvailable
{
    state_entry()
    {
        llSetTimerEvent(1);
    }

    timer(){
        llSetTimerEvent(10);
        query = llRequestAgentData(ownerID, DATA_ONLINE);
    }

    dataserver(key queryid, string data)
    {
        if (query == queryid)
        {
            if((integer)data == 1){
                state SendNotice;
            }
        }
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state SendNotice {
    state_entry() {
        llInstantMessage(CORRADE,
            wasKeyValueEncode(
```

```
        [
            "command", "notice",
            "group", wasURLEscape(GROUP),
            "password", wasURLEscape(PASSWORD),
            "action", "send",
            "subject", "test subject",
            "message", "test message",
            "callback", wasURLEscape(URL)
        ]
    )
);

}

http_request(key id, string method, string body)
{
    llHTTPResponse(id, 200, "OK");
    if(wasKeyValueGet("success", body) != "True") {
        setDebug("Failed to send notice");
    }
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}
```

Tekst 4 - Pełny skrypt do wysłania ogłoszenia grupowego z użyciem WAS

Nasz kod LSL z wysyłaniem danych z użyciem JSON będzie wyglądał tak:


```
key ownerID = "YOUROWNERID"
key CORRADE;
string PASSWORD;
string GROUP;
string URL = "";
string TAG;
string val;
list lmessage;
string sjump = "";
string body;
key query;

integer line;
list tuples = [];

setDebug(string msg)
{
    llInstantMessage(ownerID, "["+ TAG +" ] " + msg);
}

integer StringMatch(list lwhatsearch, string sinwhere){
    sinwhere = llToLower(sinwhere);
    sinwhere = strReplace(sinwhere, "!", "");
    sinwhere = strReplace(sinwhere, "?", "");

    list lsinwhere = llParseString2List(sinwhere,[" "],[]);

    integer imatch = llGetListLength(lwhatsearch);
    integer imatched = 0;

    integer i=0;
    for(i=0; i<imatch; i++){

        list lwords = llParseString2List(llList2String(lwhatsearch,i),["|"],[]);

        integer j=0;
        for(j=0; j<llGetListLength(lwords); j++){
            if(llListFindList(lsinwhere, [llList2String(lwords,j)]) >= 0){
                imatched++;
            }
        }
    }
    if(imatch == imatched){
        return 1;
    }else{
        return 0;
    }
}
```

```

string strReplace(string str, string search, string replace) {
    return llDumpList2String(llParseStringKeepNulls((str = "") + str, [search], []),
replace);
}

list wasCSVToList(string csv)
{
    list l = [];
    list s = [];
    string m = "";
    do
    {
        string a = llGetSubString(csv, 0, 0);
        csv = llDeleteSubString(csv, 0, 0);
        if(a == ",")
        {
            if(llList2String(s, -1) != "\\")
            {
                l += m;
                m = "";
                jump continue;
            }
            m += a;
            jump continue;
        }
        if(a == "\"" && llGetSubString(csv, 0, 0) == a)
        {
            m += a;
            csv = llDeleteSubString(csv, 0, 0);
            jump continue;
        }
        if(a == "\\")
        {
            if(llList2String(s, -1) != a)
            {
                s += a;
                jump continue;
            }
            s = llDeleteSubList(s, -1, -1);
            jump continue;
        }
        m += a;
        @continue;
    } while(csv != "");
    return l + m;
}

string wasKeyValueGet(string k, string data)
{

```

```

if(llStringLength(data) == 0) return "";
if(llStringLength(k) == 0) return "";
list a = llParseStringKeepNulls(data, ["&", "="], []);
integer i = llListFindList(llList2ListStrided(a, 0, -1, 2), [ k ]);
if(i != -1){
    string ret = llList2String(wasCSVToList(wasURLUnescape(llList2String(a, 2*i+1))),
0);
    ret = strReplace(ret, "\\r", "");
    ret = strReplace(ret, "\\n", "");
    ret = llStringTrim(ret, STRING_TRIM);

    return ret;
}
return "";
}

string wasKeyValueEncode(list data)
{
    list k = llList2ListStrided(data, 0, -1, 2);
    list v = llList2ListStrided(llDeleteSubList(data, 0, 0), 0, -1, 2);
    data = [];
    do
    {
        data += llList2String(k, 0) + "=" + llList2String(v, 0);
        k = llDeleteSubList(k, 0, 0);
        v = llDeleteSubList(v, 0, 0);
    } while(llGetListLength(k) != 0);
    return llDumpList2String(data, "&");
}

string wasURLUnescape(string i)
{
    return llUnescapeURL( llDumpList2String( llParseStringKeepNulls(
llDumpList2String( llParseStringKeepNulls( i, ["+"], [] ), " " ), ["%0D%0A"], [] ),
"\n" ) );
}

string wasKeyValueToJSON(string kvp, string T) {
    list data = llParseString2List(kvp, ["&", "="], []);
    list temp = [];
    do {
        string k = llList2String(data, 0);
        string v = llList2String(data, 1);
        string o = "";
        o += "\"" + k + "\"";
        if(T == JSON_ARRAY) {
            o += ",";
            jump type;
        }
    }

```

```

    o += ":";
    @type;
    if((float)v != 0 || v == "0") {
        o += v;
        jump dump;
    }
    o += "\"" + v + "\"";
    @dump;
    temp += o;
    data = llDeleteSubList(data, 0, 1);
} while(llGetListLength(data) != 0);
if(T == JSON_ARRAY)
    return "[" + llDumpList2String(temp, ",") + "]";
if(T == JSON_OBJECT)
    return "{" + llDumpList2String(temp, ",") + "}";
return JSON_INVALID;
}

string wasJSONToKeyValueData(string JSON) {
    list output = [];
    list symbols = [];
    string data = "";
    integer level = 1;
    // parse JSON
    do {
        string c = llGetSubString(JSON, 0, 0);
        // we are at an object start
        if(c == "{") {
            if(level > 1) {
                data += c;
            }
            // increment the level
            ++level;
            // add the symbol
            symbols += c;
            // and continue
            jump continue;
        }
        // we are at an object end
        if(c == "}") {
            // decrement the level
            --level;
            if(level > 1) {
                data += c;
                output += data;
                data = "";
            }
            // pop symbols
            string s = llList2String(symbols, -1);

```

```
    symbols = llDeleteSubList(symbols, -1, -1);
    // got an object start
    if(s == "{") {
        // so continue
        jump continue;
    }
    // error
    return JSON_INVALID;
}
// we are at data end or data start
if(c == "\\") {
    if(level > 2) {
        data += "\\";
        jump continue;
    }
    // pop symbols
    string s = llList2String(symbols, -1);
    symbols = llDeleteSubList(symbols, -1, -1);
    // if we are at data end
    if(s == "\\") {
        // add the data to the output
        output += data;
        // flush data
        data = "";
        // and continue
        jump continue;
    }
    // we are not a the end of data
    // add the symbol back
    symbols += s;
    // add the current character
    symbols += c;
    // and continue
    jump continue;
}
if(level > 2) {
    data += c;
    jump continue;
}
// pop symbols
string s = llList2String(symbols, -1);
symbols = llDeleteSubList(symbols, -1, -1);
if(s == "\\") {
    data += c;
    symbols += s;
    jump continue;
}
symbols += s;
@continue;
```

```

    JSON = llDeleteSubString(JSON, 0, 0);
} while(llStringLength(JSON) != 0);
// now encode to key-value data
list k = llList2ListStrided(output, 0, -1, 2);
list v = llList2ListStrided(llDeleteSubList(output, 0, 0), 0, -1, 2);
output = [];
do {
    output += llList2String(k, 0) + "=" + llList2String(v, 0);
    k = llDeleteSubList(k, 0, 0);
    v = llDeleteSubList(v, 0, 0);
} while(llGetListLength(k) != 0);
return llDumpList2String(output, "&");
}

default
{
    state_entry()
    {
        state ReadConfigurationNotecard;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state ReadConfigurationNotecard
{
    state_entry()
    {
        if(llGetInventoryType("configuration") != INVENTORY_NOTECARD)
        {
            setDebug("Sorry, could not find a configuration inventory notecard.");
            return;
        }
        line = 0;
        llGetNotecardLine("configuration", line);
    }

    dataserver(key id, string data)

```

```

{
    if(data == EOF)
    {
        if(llGetListLength(tuples) % 2 != 0)
        {
            setDebug("Error in configuration notecard.");
            return;
        }

        CORRADE = llList2Key(tuples, llListFindList(tuples, ["BOTID"])+1);

        if(CORRADE == NULL_KEY)
        {
            setDebug("Error in configuration notecard: BOT ID KEY");
            return;
        }

        GROUP = llList2String(tuples, llListFindList(tuples, ["GROUP"])+1);

        if(GROUP == "")
        {
            setDebug("Error in configuration notecard: GROUP");
            return;
        }

        PASSWORD = llList2String(tuples, llListFindList(tuples, ["PASSWORD"])+1);

        if(PASSWORD == "")
        {
            setDebug("Error in configuration notecard: PASSWORD");
            return;
        }

        state url;
    }
    if(data == "") jump continue;
    integer i = llSubStringIndex(data, "#");
    if(i != -1) data = llDeleteSubString(data, i, -1);
    list o = llParseStringKeepNulls(data, ["="], []);
    string k = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String(o,0), STRING_TRIM), ["\""], []), "\"");
    string v = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String( o, 1 ), STRING_TRIM), ["\""], [] ), "\"");
    if(k == "" || v == "") jump continue;
    tuples += k;
    tuples += v;
    @continue;
    llGetNotecardLine("configuration", ++line);
}

```

```
changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}

state url
{
    state_entry()
    {
        if ((GROUP == "BOTID") || (GROUP == "GROUP") || (GROUP == "PASSWORD")){
            llResetScript();
            return;
        }

        if ((PASSWORD == "BOTID") || (PASSWORD == "GROUP") || (PASSWORD ==
"PASSWORD")){
            llResetScript();
            return;
        }

        llSetTimerEvent(0.5);
    }

    timer(){
        llSetTimerEvent(60);
        llRequestURL();
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");
        if(method != URL_REQUEST_GRANTED) return;
        URL = body;
        state CheckBotAvailable;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
```



```
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state CheckBotAvailable
{
    state_entry()
    {
        llSetTimerEvent(1);
    }

    timer(){
        llSetTimerEvent(10);
        query = llRequestAgentData(ownerID, DATA_ONLINE);
    }

    dataserver(key queryid, string data)
    {
        if (query == queryid)
        {
            if((integer)data == 1){
                state SendNotice;
            }
        }
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state SendNotice {
    state_entry() {
```

```

        lprepare = [
            "command", "notice",
            "group", GROUP,
            "password", PASSWORD,
            "action", "send",
            "subject", "test subject",
            "message", "test message",
            "callback", URL
        ];

        llInstantMessage(CORRADE, wasKeyValueToJSON(wasKeyValueEncode(lprepare),
JSON_OBJECT));

    }
    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");
        body = wasJSONToKeyValueData(body);
        if(wasKeyValueGet("success", body) != "True") {
            setDebug("Failed to send notice");
        }
    }
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

```

Tekst 5 - Pełny skrypt do wysłania ogłoszenia grupowego z użyciem JSON

Natomiast w PHP, jeżeli używasz mojej funkcji do wysyłki to kod będzie krótszy (nie ma znaczenia czy wysyłka jest z użyciem WAS czy JSON):

```

$params = array(
    "command" => "notice",
    "group" => BOT_GROUP,
    "password" => BOT_PASSWORD,
    "subject" => "test subject",
    "message" => "test message",
);
$ret = SendToBot($params);
print_r($ret);

```

Tekst 6 - Wysłanie ogłoszenia grupowego w PHP

Omówię teraz oba kody.

key ownerID = "YOUROWNERID" - tutaj zamiast YOUROWNERID wstawiamy swoje UUID awatara, które z łatwością możemy uzyskać z Firestorm Viewer.

state CheckBotAvailable - stan, gdzie sprawdzamy, czy bot jest w ogóle dostępny w Second Life, jeżeli nie ma go to po prostu oczekuje, aż się pojawi i wówczas wykonuje dalsze instrukcje. Zalecane jest takie sprawdzanie, jeżeli prim ze skryptem nie jest częścią ubrania i stoi swobodnie gdzieś na działce w Second Life. Sprawdzanie jest wykonywane co 10 sekund przez funkcję timer().

state SendNotice - nasz stan, gdzie wysyłamy ogłoszenie. Sformułowaliśmy nasze polecenie z dodatkowymi parametrami w postaci listy LSL [], gdzie każdy parametr ma swoją wartość.

Można zauważyć, że niektóre są wysyłane do funkcji wasURLEscape(), która koduje specjalne znaki do formatu akceptowanego przez Second Life [zobacz FAQ].

Całość listy jest kodowana za pomocą funkcji wasKeyValueEncode(), która zamienia do postaci URL parametr1=wartość1¶metr2=wartość2&... i jest wysyłana do bota za pomocą polecenia llInstantMessage().

Przy wysyłaniu za pomocą JSON jest dużo prościej, gdyż tylko tworzymy listę, gdzie wartości nie są objęte żadnymi funkcjami, która jest najpierw kodowana do postaci URL, a potem do postaci JSON w postaci obiektowej.

http_request(key id, string method, string body) - ta funkcja oczekuje i zwraca dane od bota, ponieważ wcześniej w parametrach zdefiniowaliśmy callback (tj. informację zwrotną), który zostanie ustawiony na URL przyznany przez Second Life. To tutaj dokonujemy dalszego przetworzenia. Ja wybrałem w tym miejscu tylko sprawdzenie czy komenda została poprawnie wykonana.

changed(integer change) - wykrywa zmiany związane tutaj ze zmianą właściciela, regionu czy restartem regionu i podejmuje odpowiednie akcje, u mnie po prostu jest nią restart skryptu.

on_rez(integer start_param) - wykonuje akcję związaną z rezzowaniem obiektu, u mnie restart skryptu.

Pamiętaj, aby sprawdzać swój kod czy przesyła prawidłowo polecenie z argumentami i odsyła oczekiwane dane.

Tutaj należy jeszcze omówić jeszcze jeden temat.

Wartość parametru zazwyczaj ma jeden element, a co zrobić, jeżeli ma więcej elementów w postaci listy? Z pomocą przychodzi funkcja wasListToCSV(), która konwertuje listę do postaci wartości oddzielonych przecinkami.

```
"avatars", wasListToCSV([
    wasURLEscape("Philip Linden"),
    // Agent by UUID
    "ff19c6b5-4824-4de3-995a-d739d588cf18",
    "Ectogram Resident"
])
```

Tekst 7 - Przekształcenie listy do postaci CSV

Jeżeli natomiast mamy dane zwracane z bota w formacie CSV, to możemy je przekonwertować do listy za pomocą wasCSVToList().

Jeżeli używasz PHP to odpowiednikami tych funkcji będą: wasArrayToCSV() [podajemy elementy typu array()] i wasCSVToArray().

A co z kodowaniem w JSON? W przypadku PHP tutaj nie musimy tworzyć wymyślnych funkcji, ponieważ PHP ma już wbudowaną funkcję do kodowania danych w JSON, a jeżeli korzystasz z mojej funkcji to nie musisz się w ogóle tym przejmować.

W LSL przy kodowaniu JSON tablicę wartości również zamieniamy przy pomocy funkcji wasListToCSV(), jednakże na razie nigdzie nie używamy funkcji wasKeyValueEncode().

Następnie kodujemy wszystko za pomocą wasKeyValueEncode() a następnie wasKeyValueToJSON(). W ten sposób mamy gotowe żądanie do bota w formacie JSON.

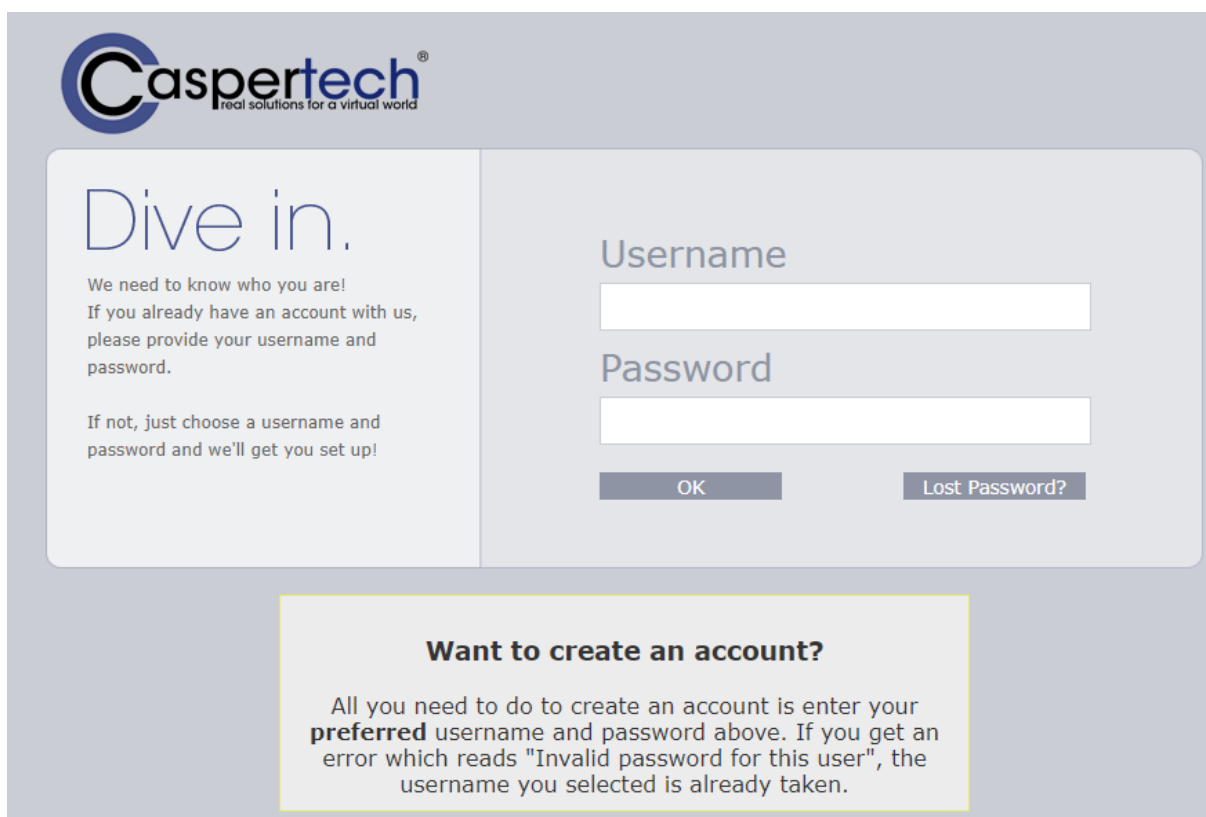
W skrócie może wyglądać to tak:

```
llInstantMessage(CORRADE, wasKeyValueToJSON(wasKeyValueEncode(lprepare), JSON_OBJECT));
```

4.15. Integracja Corrade z CasperTech¹⁹

CasperTech umożliwia proste, zautomatyzowane otrzymywanie pieniędzy za dzierżawienie ziemi czy sprzedaż produktów. Przy dzierżawie ziemi z użyciem systemu CasperTech, system sam będzie odnawiał czas dzierżawy po zapłacie przez najemcę.

1. Zaloguj się na stronie CasperTech – <https://let.casperdns.com/>.



Caspertech
real solutions for a virtual world

Dive in.

We need to know who you are!
If you already have an account with us,
please provide your username and
password.

If not, just choose a username and
password and we'll get you set up!

Username

Password

OK Lost Password?

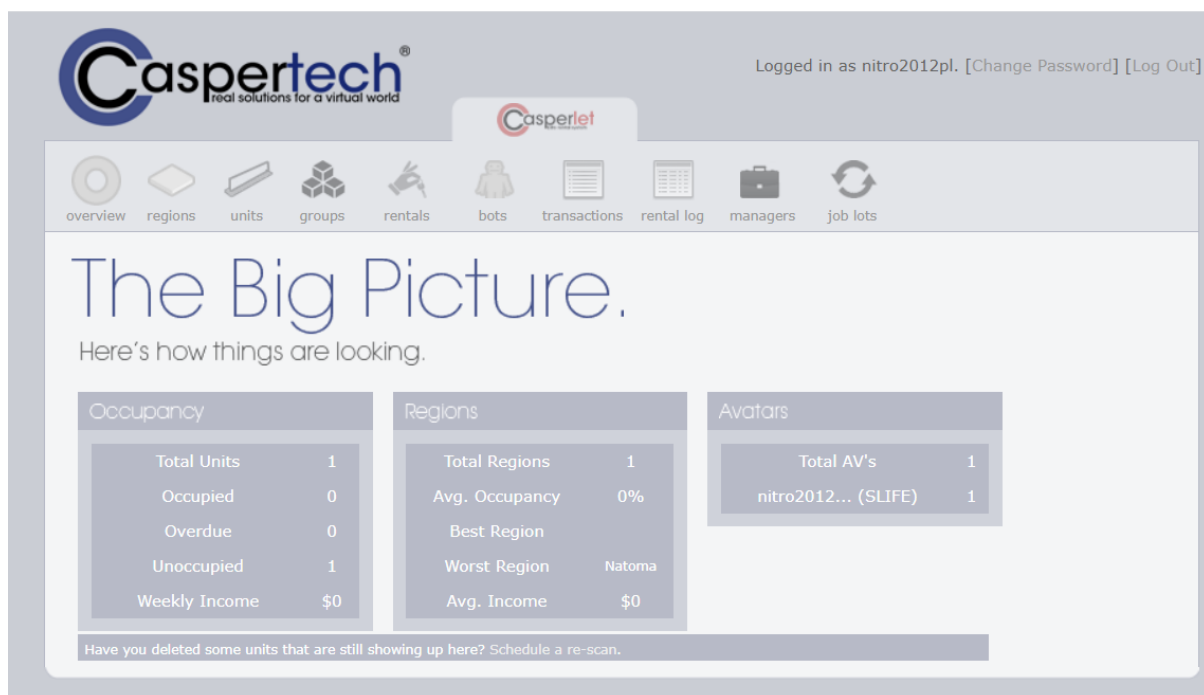
Want to create an account?

All you need to do to create an account is enter your **preferred** username and password above. If you get an error which reads "Invalid password for this user", the username you selected is already taken.

Rysunek 4.27 - Strona logowania CasperTech

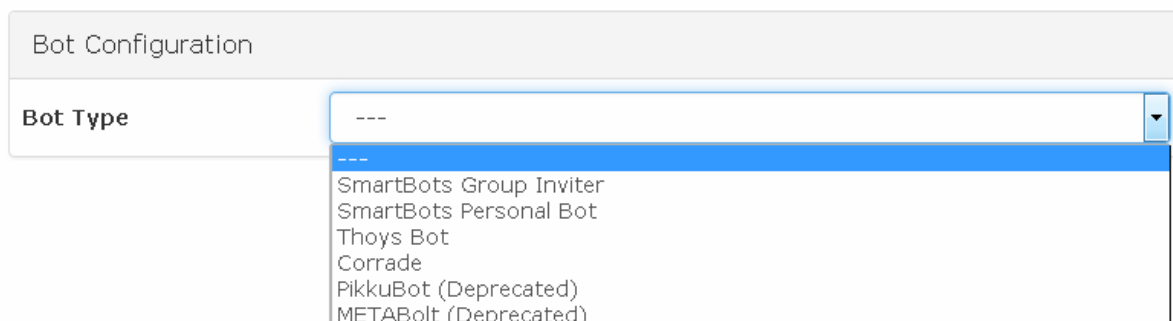
2. Następnie z menu wybierz 'Bots'.
3. Na ekranie kliknij na przycisk 'Add Bot'.

¹⁹ Na podstawie: <https://wiki.casperdns.com/index.php/Corrade>



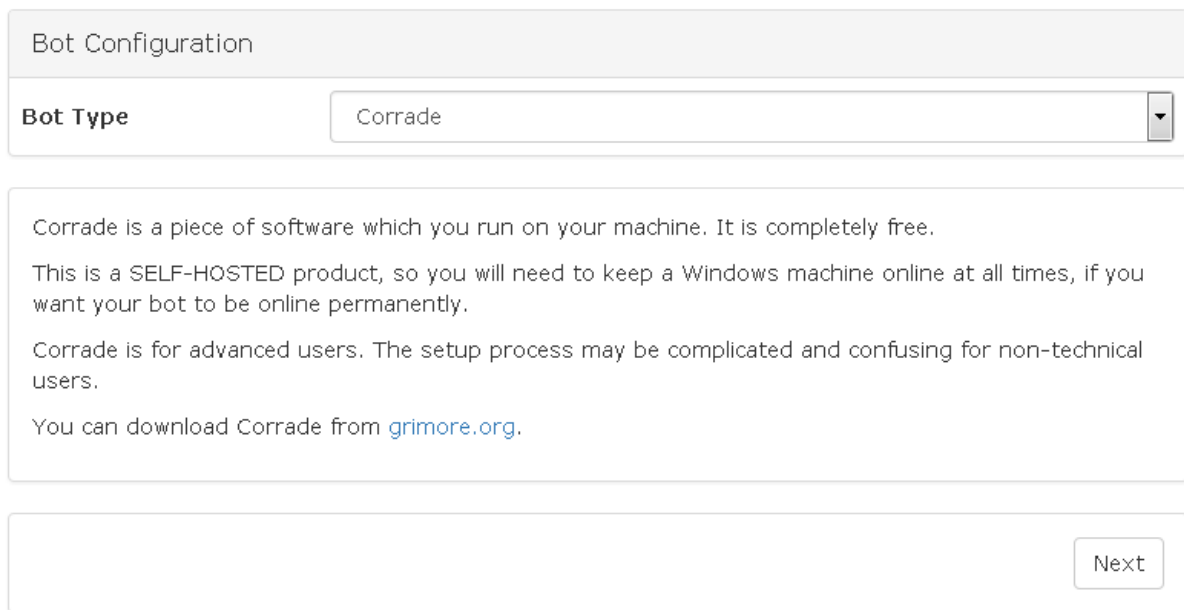
Rysunek 4.28 - Strona CasperTech po zalogowaniu

4. Z listy rozwijalnej wybierz 'Corrade'.



Rysunek 4.29 - Wspierane rodzaje botów przez CasperTech

5. Na ekranie zobaczysz krótką informację na temat Corrade.
6. Jeżeli wszystko się zgadza, kliknij na przycisk 'Next'.



The screenshot shows a 'Bot Configuration' window. At the top, the title 'Bot Configuration' is displayed. Below it, there is a section labeled 'Bot Type' with a dropdown menu currently showing 'Corrade'. Underneath the dropdown, there is a text box containing the following information: 'Corrade is a piece of software which you run on your machine. It is completely free. This is a SELF-HOSTED product, so you will need to keep a Windows machine online at all times, if you want your bot to be online permanently. Corrade is for advanced users. The setup process may be complicated and confusing for non-technical users. You can download Corrade from grimore.org.' At the bottom right of the window, there is a 'Next' button.

Rysunek 4.30 - Opis wyboru bota jako Corrade

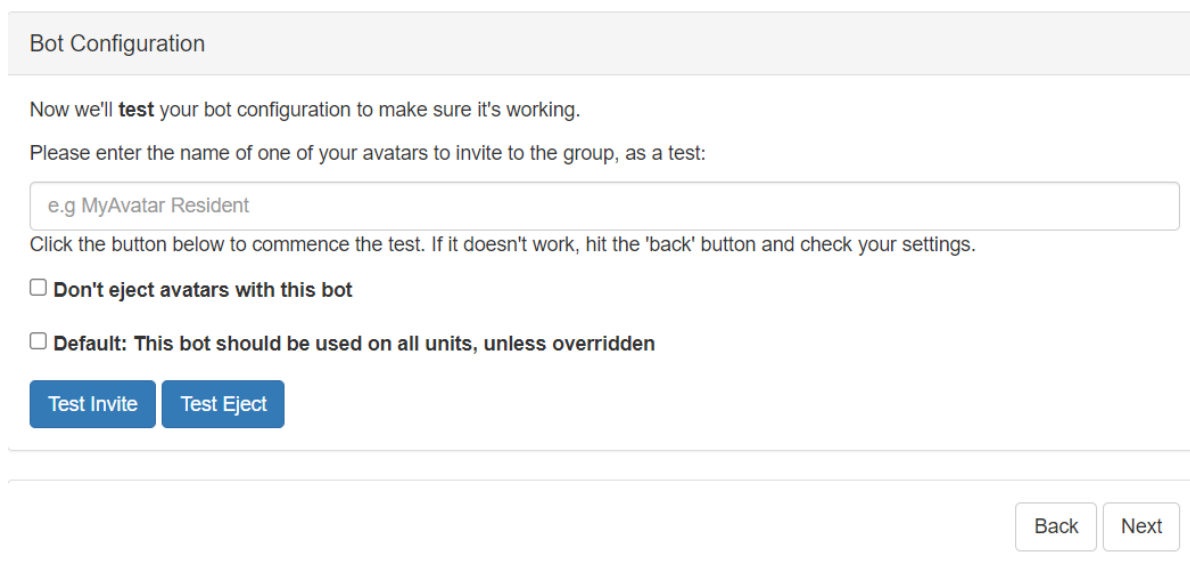
7. Na następnym ekranie uzupełnij niezbędne dane:

- Label - nazwa wyświetlana na liście na stronie CasperTech,
- Bot UUID - UUID bota,
- Group name - nazwa grupy do której jest przypisany bot,
- Secure Code - hasło do grupy, które wcześniej zostało ustalone i wpisane do pliku konfiguracyjnego bota,
- Bot Owner Avatar UUID - UUID twojego awatara jako właściciela bota,
- Role Name - opcja dodatkowa; nazwa roli,
- Role UUID - opcja dodatkowa; UUID roli.

| Bot Configuration | |
|-----------------------|---|
| Label | <input type="text" value="e.g Tenants Group"/> |
| Bot UUID | <input type="text" value="e.g 5a5139e9-5ba9-4d4c-a78a-e6785e03ed89"/> |
| Group Name | <input type="text" value="e.g AwesomeLand Tenants"/> |
| Security Code | <input type="text" value="e.g mysecretpassword"/> |
| Bot Owner Avatar UUID | <input type="text" value="e.g d1cd5b71-6209-4595-9bf0-771bf689ce00"/> |
| Role Name (Optional) | <input type="text" value="e.g Everyone"/> |
| Role UUID (Optional) | <input type="text" value="e.g 0aeb2aa7-5f10-1c49-ddac-48ae397d437f"/> |

Rysunek 4.31 - Ekran z konfiguracją bota

8. Na następnym ekranie masz możliwość przetestowania konfiguracji swojego bota.
9. Wprowadź nazwę awatara do którego bot ma wysłać zaproszenia do grupy lub wyrzucić z grupy.
10. Możemy też pominąć ten krok.
11. Podajemy nazwę awatara, który będzie testował bot.
12. Mamy możliwość zaznaczenia dodatkowych opcji:
 - Don't eject avatars with this bot - nie wyrzucaj awatarów za pomocą tego bota,
 - This bot should be used on all units, unless overridden - czy bot ma obsługiwać wszystkie jednostki dzierżawy, chyba, że inne ustawienia nadpiszą tę opcję.
13. Gdy klikniemy na 'Test invite' strona wyśle polecenie do bota, który zaprosi wybranego wcześniej awatara do grupy.
14. Gdy klikniemy na 'Test eject' strona wyśle polecenie do bota, który wyrzuci awatara z grupy.
15. Jeżeli wszystko ok, to kliknij na przycisk 'Next'.



Rysunek 4.32 - Ekran z możliwością przetestowania bota

16. Jeżeli wszystko jest w porządku to klikamy na OK lub next, który zapisze konfigurację naszego bota i powrócimy do listy botów.

4.16. Odsiewanie danych

Przy odbiorze danych od bota funkcji `http_request()` w LSL stosowany jest limit 2 KB narzucony przez Second Life. Powyżej tego limitu dane są obcinane. Aby uzyskać pełne dane są dwa wyjścia: albo przenieść wywołanie polecenia do PHP albo zastosować tzw. przesiew danych. Odsiew danych pozwala na wybranie danych, które nas interesują i nie obcięcie danych narzuconych przez wspomniany wyżej limit.

Do dyspozycji mamy następujących odsiewacze:

- `count` - podaje liczbę pozycji, które pasują do podanego wcześniej wyrażenia regularnego

Dane wejściowe: a,1,b,2,c,3,d,4,e,5,f,6,g,7,h,8,i,9

```
"sift", wasListToCSV([  
    "count", "[0-9]"  
])
```

Dane wyjściowe: 9

Komentarz: Corrade po przetworzeniu danych wybiera tyle te elementy z listy, które są liczbami, czyli: 1,2,3,4,5,6,7,8,9, a więc długość tej listy wynosi 9 elementów.

- distinct - zwraca tylko unikatowe elementy na liście

Dane wejściowe: a,b,c,a,b,c,a,b,c,d,a,b,d,a,e,a,h,a

```
"sift", wasListToCSV([  
  "distinct", ""  
])
```

Dane wyjściowe: a,b,c,d,e,h

- each - zwraca co określony element z listy

Dane wejściowe: a,1,b,2,c,3,d,4,e,5,f,6,g,7,h,8,i,9

```
"sift", wasListToCSV([  
  "each", 2  
])
```

Dane wyjściowe: a,b,c,d,e,f,g,h,i

Komentarz: Zwraca w przykładzie co drugi element z listy.

- html - zwraca elementy, które pasują do xpath HTML
- jsonpath - zwraca elementy, które pasują do ścieżki JSON
- match - zwraca elementy lub części z elementów, które pasują do wyrażenia regularnego

Dane wejściowe: Item Cost 100,2,Item Cost 500,3,Item Cost 10,4,Item Cost 15

```
"sift", wasListToCSV([  
  "match", "Item Cost ([0-9]+)"  
])
```

Dane wyjściowe: 100,500,10,15

- md5 - generuje hash MD5 dla tekstu wynikowego
- sha1 - generuje hash SHA1 dla tekstu wynikowego
- permute - przedstawia elementy listy na podstawie zadanej rotacji

- random - zwraca losowy element z listy
- reverse - zwraca listę elementów w odwrotnej kolejności
- select - wybiera wartości na podstawie wcześniej zdefiniowanego klucza

Dane wejściowe: Price,10,Item,Book,Price,50,Item,Chair,Price,200,Item,PC

```
"sift", wasListToCSV([  
    "select", "Price"  
]),
```

Dane wyjściowe: 10,50,200

- skip - pomija określoną liczbę początkowych elementów z listy
- take - pobiera określoną liczbę elementów z listy
- xpath - zwraca elementy na podstawie ścieżki XPath

Powyższe odsiewacze danych można ze sobą łączyć.

4.17. Słaba i mocna referencja przez nazwę i UUID obiektu

W Second Life wszystko ma swój identyfikator, tzw. UUID (Universally unique identifier).

W Corrade jeżeli odnosimy się do obiektów to możemy odnosić się przez ich nazwę lub UUID. Jednakże wywoływanie obiektu przez jego nazwę jest niezalecane, ponieważ w tym samym regionie może być więcej obiektów, które posiadają tą samą nazwę co szukany obiekt. Zamiast operować tylko na jednym obiekcie będziemy operować na wszystkich znalezionych. Nazywamy to słabą referencją. Dużo lepszym pomysłem jest odwoływanie się do obiektu przez jego identyfikator UUID, który jest unikalny dla każdego obiektu znajdującego się w regionie. Nazywamy to mocną referencją.

4.18. Potwierdzanie otrzymywania danych od bota w skrypcie LSL.

Przy każdym uzyskaniu danych od bota w skrypcie LSL należy potwierdzić botowi, że owe dane zostały otrzymane bez problemów.

Przykład:

```
http_request(key id, string method, string body){  
    llHTTPResponse(id, 200, "OK"); //potwierdzenie  
    //.....  
}
```

W przypadku pominięcia tego zapisu Corrade przez ok. minutę oczekuje na potwierdzenie zwrotne. Ta jedna linijka skraca ten czas i Corrade może zamknąć jedno połączenie wcześniej.

4.19. Rozmiar szafy bota.

Należy od czasu do czasu zalogować się jako awatar swojego bota i dokonać inspekcji jego szafy. Należy skasować te rzeczy, których on nie używa, ponieważ z jednej strony zaśmiecają one jego szafę a z drugiej strony powodują, że Corrade potrzebuje więcej czasu na ich zapisanie w pamięci podręcznej.

4.20. Zabezpieczenie Corrade i obrona przed atakami

1. Czy plik konfiguracyjny jest właściwie zabezpieczony? (np. odpowiednie uprawnienia pliku, dostęp tylko dla odpowiednich użytkowników)
2. Czy bot ma dostęp do odpowiednich grup?
3. Czy bot ma dostęp do odpowiednich uprawnień i powiadomień w grupach?
4. Czy komputer na którym jest uruchomiony łączy się bezpiecznie z Internetem i zaufanymi serwerami?
5. Czy ustawiłeś limitowanie zasobów do Corrade?

6. Czy używasz w skryptach LSL `llRequestSecureURL()` zamiast `llRequestURL()` [ochrona przed MITM]?
7. Czy komputer jest zabezpieczony przed atakami DDoS?
8. Czy do komputera mają dostęp tylko zaufane osoby?
9. Czy folder bota ma stosowne uprawnienia?
10. Czy używając komendy 'execute' poprawnie zdefiniowałeś polecenie do wywołania? Czy nie dojdzie do wywołania nieoczekiwanych zmian w systemie, np. kasacji ważnych plików?
11. Czy używasz `llSensor` i `llSensorRepeat` do uzyskania UUID obiektów?

Uwagi

- MD5 czy SHA1 są słabymi hasłami do generowania hasła, jeżeli masz podejrzenia, że na konto bota dokonano włamu - zmień natychmiast hasło na silniejsze, które ponownie wpisz do pliku konfiguracyjnego. Przejrzyj powyższą listę czy wszystko zrobiłeś, aby uniknąć przed włamem na konto.
- `llRequestSecureURL()` powoduje uzyskanie bezpiecznych URL z Second Life (HTTPS), jednakże Second Life posługuje się samopodpisanymi certyfikatami SSL, co powoduje konieczność wyłączenia sprawdzania wydawcy certyfikatu. Jeżeli bezpieczne URL od Second Life sprawiają problem (np. przy powiadomieniach) korzystaj z `llRequestURL()`.

4.21. Uwagi

1. Polecenia takie jak `getavatarpositions`, `getmapavatarpositions` zwracają błędną współrzędną Z - maksymalna liczba to 1000, dlatego nie są zalecane przy uzyskiwaniu współrzędnych bota czy innych awatarów. Jeżeli chcesz uzyskać dokładne współrzędne to użyj powiadomienia tracker lub polecenia LSL `llGetObjectDetails`.

5. Leonardo De ArtBot

5.1. Parametry komputera ('dom bota')



Rysunek 5.1 - Fujitsu ESPRIMO e5730 - przód



Rysunek 5.2 - Fujitsu ESPRIMO e5730 - tył

Leonardo jest uruchamiany na komputerze typu SFF, który jest zamknięty w małej podręcznej obudowie. Komputer został wyprodukowany w okolicach roku 2008, ale działa bez problemów.

Parametry komputera to:

- system operacyjny: Windows 7 SP1 64-bit,
- procesor: Intel Core 2 Duo E8600 @ 3.33 Ghz (2 rdzenie, 2 wątki),
- pamięć: 4 GB RAM DDR2-SDRAM, taktowanie 400 Mhz (PC2-6400); w systemie dostępne tylko 3.65 GB,
- karta graficzna: zintegrowana Intel Q45/Q43,
- dysk twardy: Hitachi 160 GB HDD, SATA, szybkość dysku – 7200 rpm,
- sieć:
 - (główna) LAN; Intel 82567LF-3, szybkość: 100 Mbps, podłączony do routera WiFi z Internetem mobilnym (sieć Orange PL),
 - (zapasowa) modem mobilny: Huawei E3372 HiLink, max. 150 Mbps; Orange PL,
- oprogramowanie:
 - Corrade, wersja: 11.0.88.76 (data publikacji: 2021-01-06 07:12),
 - Serwer WWW (oprogramowanie XAMPP):
 - Apache 2.4.41,
 - baza danych: MariaDB 10.4.11,
 - phpMyAdmin 4.9.2,
 - OpenSSL 1.1.0g,
 - PHP 7.4.1 (VC15 64-bitowy, thread safe)

5.2. Budowa mózgu bota i komunikacja na linii LSL - skrypt zewnętrzny

Mózg bota Leonarda został zbudowany w oparciu o język PHP i LSL.

Wybrałem PHP z następujących względów: znam ten język bardzo dobrze, bardzo szybko pisze się kod w tym języku, pozwala na ograniczenie maksymalnej ilości pamięci do użycia.

Sam folder bota ma obecnie rozmiar ok. 2.22 GB, na który składają się pliki i foldery oprogramowania Corrade, pliki logów, pliki z 'mózgiem' bota.

Na skład tego folderu składają się następujące podfoldery i podpliki (Rysunek 5.4 - Struktura folderów i plików w bocie Leonardo):

- Inne pliki i katalogi związane z Corrade,
- Cache – zapisuje dane z Second Life do pamięci podręcznej (np. zasoby, dane awatarów, itd.),
- Contrib (linux, windows, macosx) – zawiera instrukcje i programy jak zainstalować Corrade jako usługę w systemie,
- Logs – logi programów oraz zapisane do plików rozmowy prywatne pomiędzy botem a awatarami, rozmowy publiczne z czatu publicznego, czatu grupowego, itd.,
- mcorrade.ps1 + inne pliki – mój skrypt w Powershell do zarządzania botem wraz z innymi plikami,
- Nucleus – pliki Nucleus,
- State – zapisuje pliki powiadomień z nadanymi URLami z Second Life,
- php – mózg bota napisany w PHP
 - botconfig.ini – plik konfiguracyjny INI bota,
 - cache.ini – plik pamięci podręcznej,
 - composer.phar – plik do zarządzania zależnościami projektu Composer,
 - config.php – plik konfiguracyjny PHP,
 - cron.bat – program wsadowy cron,
 - corrademonitorservice.php – plik PHP, który pozwala monitorować proces bota,
 - functions.php – funkcje PHP,
 - inPolygon.class.php – klasa PHP sprawdzająca czy dany punkt należy do wielokątu,
 - pliki PHP zaczynające się od part* - pliki realizujące różne funkcje przez bota; są podłączane przez główny plik servebot.php,
 - phplint.bat – program wsadowy sprawdzający poprawność składni w folderze z mózgiem bota,
 - servebot.php – główny plik PHP, który realizuje funkcje bota; podłącza inne części zapisane w plikach PHP; jest wywoływany przez bota z Second Life,

- logs – folder z plikami z zapisanymi zdarzeniami przez serverbot.php,
- lsl – pliki LSL,
- vendor – folder z zależnościami Composer.

Gdyby oszacować jaki procent zajmują skrypty PHP a jakie skrypty LSL w postaci ilości linii kodu, to po obliczeniu wychodzi stosunek ok. 53-57%/ 47-43%, co jest dość małą różnicą.

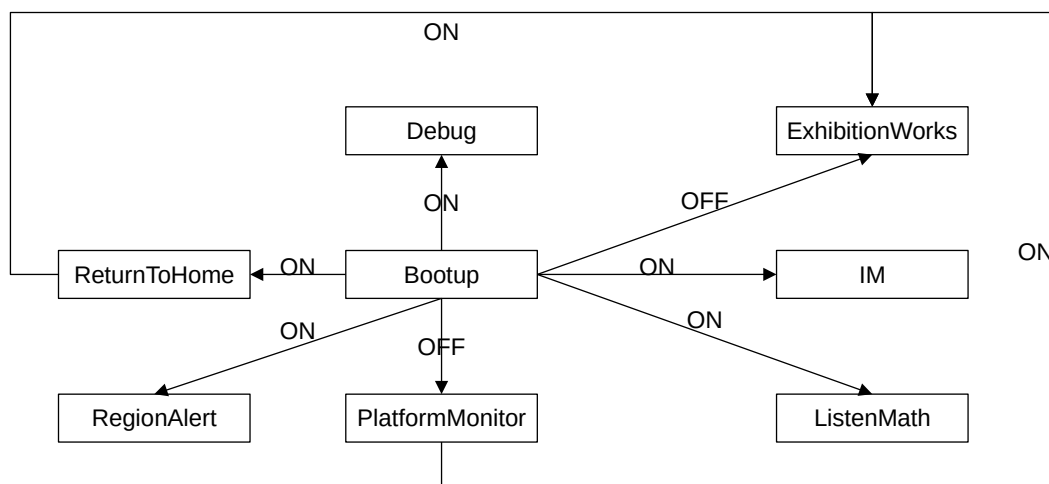
LSL jako standard działa po stronie Second Life, gdzie wywołuje żądania PHP (wykonanie pliku serverbot.php) poprzez komendę 'execute' z API Corrade, które są wykonywane na komputerze, na którym jest uruchomiony bot.

Centrum uruchomieniowe bota znajduje się w jednym primie o nazwie ArtBotStartupV1, który jest noszone jako niewidoczne ubranie przez bota. Bot jest właścicielem tego primu, a wszelkie zmiany w tym primie wymagają ręcznego zalogowania się jako bot i edycji tego primu. Nie mam tutaj z tym problemu, ponieważ bardzo rzadko loguję się jako bot, a nawet jeżeli już do tego dochodzi, nie jest to dla mnie jakoś uciążliwe. Jest to jednak plus, ponieważ bot wszystko ma przy sobie. Skrypty awatara nie uruchomią się, jeżeli ten będzie na działce gdzie jest wyłączona opcja uruchamiania skryptów awatarów, dlatego też w harmonogramie zadań systemu jest uruchamiany okresowo skrypt, który sprawdza, czy bot jest na właściwej działce, a jeżeli nie, to próbuje go teleportować.

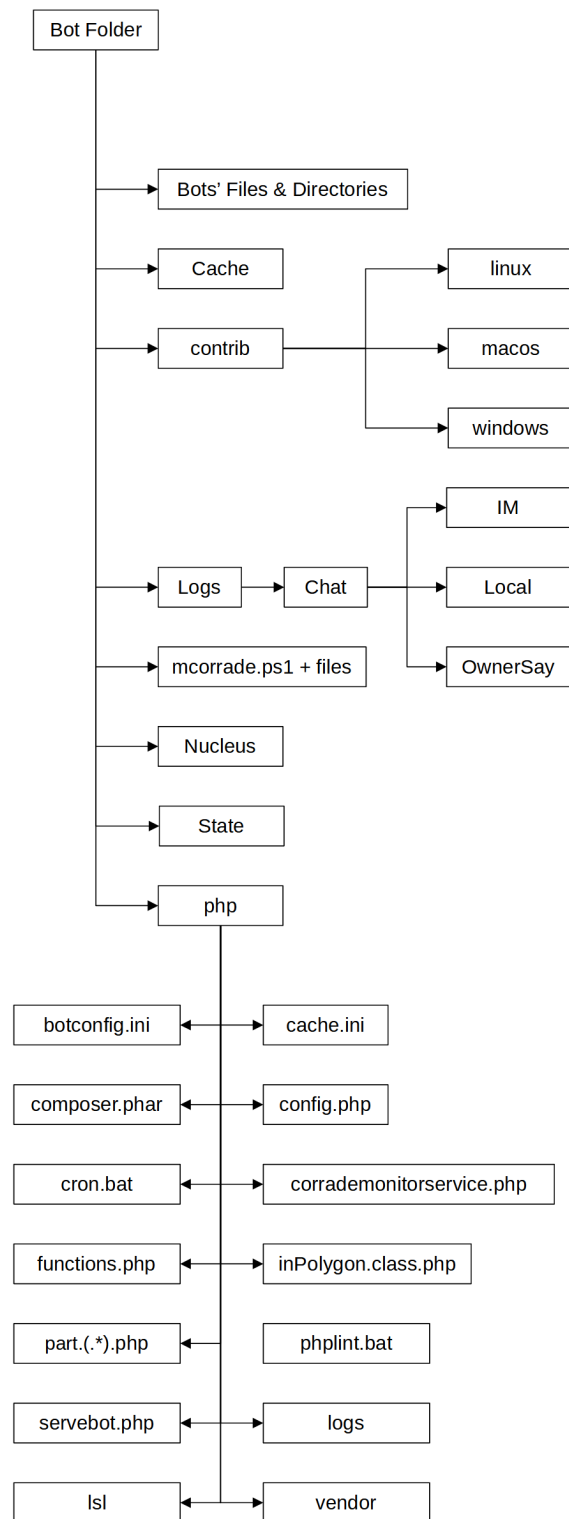
W skład prima wchodzi następujące skrypty:

- **Bootup** - jest pierwszym skryptem, który decyduje, które najpierw skrypty w primie uruchomić (Rysunek 5.3 - Zależność uruchamiania poszczególnych skryptów w primie); jest on uruchamiany, gdy tylko bot zaloguje się w Second Life lub zostanie ponownie podłączony prim jako część ubrania; o kolejności uruchamiania skryptów decyduje autor bota,
- **Debug** - skrypt, który próbuje restartować inny skrypt, który zgłosił błąd (zazwyczaj typu: 'skrypt nie znaleziono'),
- **ExhibitionWorks** - skrypt, który obsługuje otrzymane tekstury od awatarów i umieszcza je na wspólnej wystawie (jest ona dość blisko stanowiska bota),
- **IM** - skrypt, który obsługuje prywatne wiadomości od awatarów,

- **ListenMath** - skrypt, który obsługuje niektóre matematyczne funkcje,
- **PlatformMonitor** - skrypt, który monitoruje stanowisko pracy bota - jeżeli nie ma jakiegoś obiektu - tworzy lub rezzuje go ze szafy bota,
- **RegionAlert** - skrypt, który monitoruje czy zlecono restart regionu i w takim przypadku wysyła informację grupową na Facebook i Twittera,
- **ReturnToHome** - skrypt, który sprawdza czy bot jest na właściwej działce.



Rysunek 5.3 - Zależność uruchamiania poszczególnych skryptów w primie



Rysunek 5.4 - Struktura folderów i plików w bocie Leonardo

Bez względu na skrypt, jego funkcje, gdy ma dojść do przetworzenia danych, to dane wejściowe są przesyłane dalej do zewnętrznych plików PHP, które są w folderze bota, a bot w Second Life może otrzymać status wykonania tego przetwarzania. Wysyłka odbywa się za pomocą polecenia `execute` z parametrami: nazwa grupy; hasło grupy; ścieżka pliku do wykonania; parametry przekazywane do pliku wykonawczego; nazwa skryptu LSL, z którego pochodzi wywołanie w celu ewentualnego dalszego rozpoznania w logach.

Przykładowa funkcja wysyłająca do pliku PHP wygląda tak:

```
sendPHP(list value){
    lprepare = [
        "group", GROUP,
        "password", PASSWORD,
        "command", "execute",
        "file", "php",
        "parameter", "php/servebot.php " + llDumpList2String(value, "|"),
        "_script", TAG
    ];

    llOwnerSay(wasKeyValueToJSON(wasKeyValueEncode(lprepare), JSON_OBJECT));
}
```

Tekst 8 - funkcja do wysyłania danych do zewnętrznego pliku PHP

Plik wykonawczy PHP jest wpisany do globalnej ścieżki systemowej PATH w systemie Windows.

Wywoływany jest zawsze plik `php/servebot.php`, który obsługuje wszystkie żądania PHP od bota.

`llDumpList2String` jest funkcją LSL, która konwertuje listę na łańcuch znakowy, oddzielając pozycje listy przy pomocy odpowiedniego separatora (tutaj `|` - kreską pionową).

`_script` jest dodatkowym parametrem, który nie będzie przetwarzany przez oprogramowanie bota, jednakże przy analizie logów pozwala wychwycić, który skrypt LSL wywołał polecenie. Jego wartość jest przechowywana przez zmienną `TAG`, która przechowuje nazwę skryptu, która jest przypisywana podczas wywoływania stanu `default()` poprzez funkcję `llGetScriptName()`.

Całość jest wysłana bezpośrednio do bota przez polecenie `llOwnerSay`, które nie powoduje opóźnień podczas przekazywania wiadomości albo ignorowania wiadomości ze względu na przepustowość, jaka jest przy innych podobnych funkcjach. Przed wysłaniem wszystko jest kodowane do formatu JSON.

```
default
{
    state_entry()
    {
        TAG = llGetScriptName();
        setDebug("Free memory: " + (string)llGetFreeMemory());
        state ReadConfigurationNotecard;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}
```

Tekst 9 - przykładowy stan default(){} w skrypcie LSL

Przykładowe wykorzystanie gdzieś w kodzie LSL może wyglądać tak:

```
sendPHP(["action", "ACTIONNAME", "param2", varname2]);
```

gdzie jako kolejne wartości listy przyjmuje się:

- action – ACTIONNAME = nazwa akcji, która musi się zgadzać z nazwą akcji w pliku servebot.php,
- param2 – varname2 = wartość parametru (parametrów może być wiele).

Lista musi mieć parzystą liczbę elementów.

Ponieważ wysyłamy parametry do programu wykonywalnego w wersji konsolowej bez graficznego interfejsu użytkownika, należy pamiętać, że maksymalna długość łańcucha znakowego przekazywanego jako parametr do pliku wykonywalnego nie może przekroczyć 2047 znaków²⁰.

5.3. Analiza działania bota

W okresie 2020-10-28 – 2020-11-23 zbierano informacje o bocie za pomocą wbudowanego powiadomienia 'heartbeat', które wysyłało dane do zewnętrznego skryptu, które końcowo zapisywało do bazy danych. Ponieważ powiadomienie 'heartbeat' zwraca wyniki co 1 sekundę, w bazie danych otrzymano ok. 1,7 miliona wierszy (Rysunek 5.5 - Tabela gdzie zostały zapisane dane z powiadomienia 'heartbeat'). Postanowiłem zostawić tylko te wiersze, które zostały zapisane o pełnej godzinie i minucie – w efekcie z 1,7 miliona wierszy zostało ok. 31 tysięcy (Rysunek 5.6 - Odchudzona tabela z zapisanymi danymi z powiadomienia 'heartbeat'). Wygenerowałem poniżej wykresy i tabele dla 3 szukanych danych: średnie użycie procesora, średnie użycie pamięci RAM oraz liczba uruchomionych wątków.

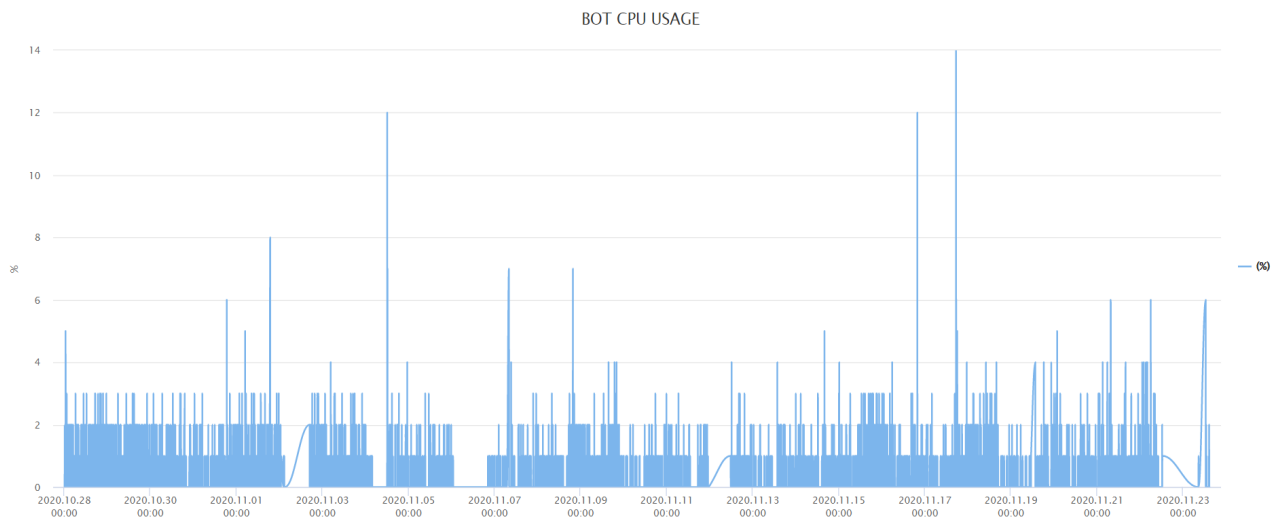
| Tabela | Działanie | Rekordy | Typ | Metoda porównywania napisów | Rozmiar | Nadmiar |
|------------------------------------|---|------------|--------|-----------------------------|----------|---------|
| <input type="checkbox"/> heartbeat | Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń | ~1,778,265 | InnoDB | utf8mb4_unicode_ci | 158.7 MB | - |
| <input type="checkbox"/> live | Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń | 36,205 | InnoDB | latin1_swedish_ci | 7.5 MB | - |
| 2 tabel | Suma | ~1,806,470 | InnoDB | utf8mb4_general_ci | 158.2 MB | 0 B |

Rysunek 5.5 - Tabela gdzie zostały zapisane dane z powiadomienia 'heartbeat'

| Tabela | Działanie | Rekordy | Typ | Metoda porównywania napisów | Rozmiar | Nadmiar |
|------------------------------------|---|---------|--------|-----------------------------|---------|---------|
| <input type="checkbox"/> heartbeat | Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń | 31,286 | InnoDB | utf8mb4_unicode_ci | 3.5 MB | - |
| <input type="checkbox"/> live | Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń | 36,205 | InnoDB | latin1_swedish_ci | 7.5 MB | - |
| 2 tabel | Suma | 67,491 | InnoDB | utf8mb4_general_ci | 11.0 MB | 0 B |

Rysunek 5.6 - Odchudzona tabela z zapisanymi danymi z powiadomienia 'heartbeat'

20 <https://docs.microsoft.com/en-us/troubleshoot/windows-client/shell-experience/command-line-string-limitation>

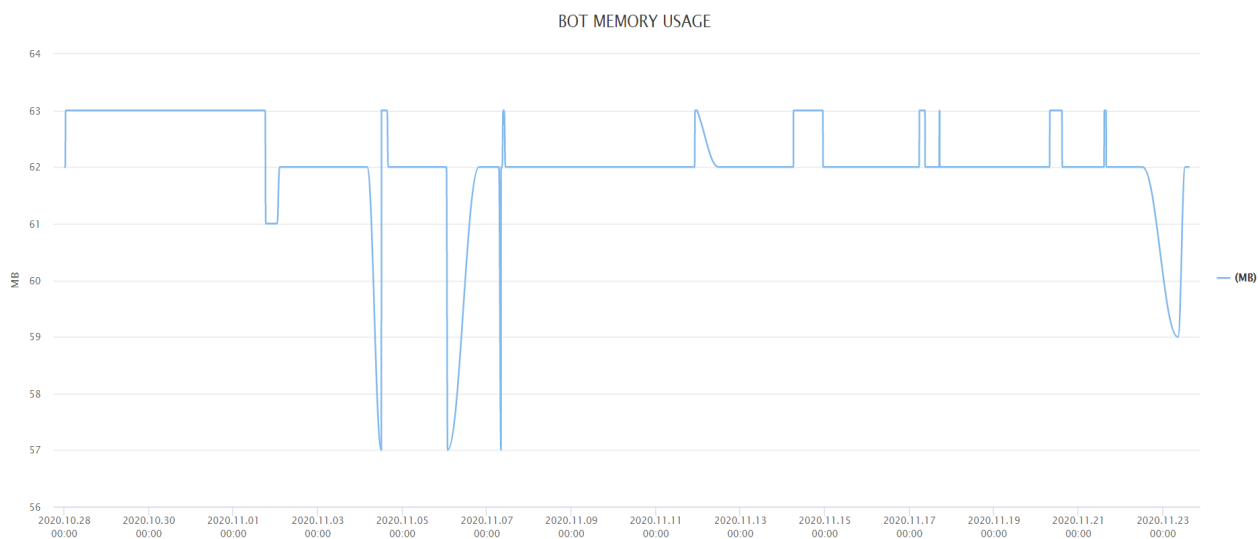


Rysunek 5.7 - Wykres zużycia procesora przez bota

| | % |
|-----|------|
| MIN | 0 |
| MAX | 14 |
| AVG | 0.15 |

Tabela 3 - Tabela z krótką statystyką zużycia procesora

Na podstawie analizy wykresu i tabelki można wywnioskować, że zużycie procesora przez program Corrade jest bardzo niskie – w szczytowym momencie osiągnął wartość 14%.

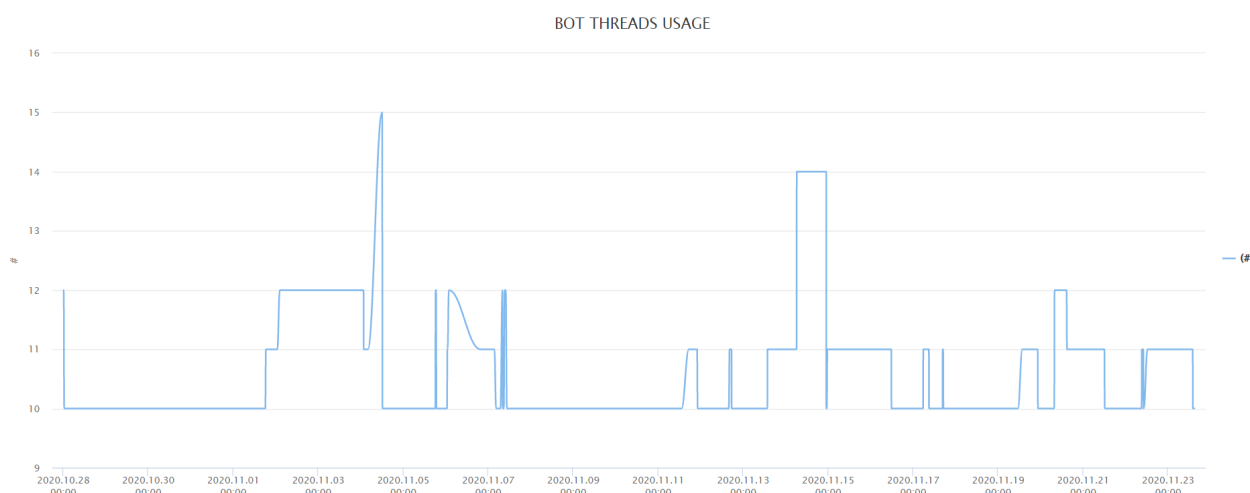


Rysunek 5.8 - Wykres zużycia pamięci RAM przez bota

| | % |
|-----|----|
| MIN | 57 |
| MAX | 63 |
| AVG | 62 |

Tabela 4 - Tabela z krótką statystyką pamięci

Na podstawie analizy wykresu i tabelki można wywnioskować, że zużycie pamięci RAM przez program i jest w granicach 57 – 63 MB. Komputer na którym bot był uruchomiony ma 4 GB pamięci RAM, co oznacza, że program zużywa pamięć w granicach 1.39% - 1.53%.



Rysunek 5.9 - Wykres użycia wątków przez Corrade

| | % |
|-----|----|
| MIN | 10 |
| MAX | 15 |
| AVG | 10 |

Tabela 5 - Tabela z krótką statystyką użycia ilością wątków

Na podstawie analizy wykresu i tabelki można wywnioskować, że program Corrade średnio używa 10 wątków do obsługi bota.

Następnej analizie podlegać będzie dostępność bota. Dostępność była badana w okresie 2020-04-15 – 2020-12-25. Dane zostały pobrane z dziennika zdarzeń systemu Windows za pomocą polecenia powershell, które pobiera wydarzenia zarejestrowane przez program NSSM (ten od rejestracji programu jako usługi), który następnie rejestrował wydarzenia ze statusem 'START control' lub 'STOP control'.

```
powershell -Command "Get-WinEvent -FilterHashtable @{LogName=\"Application\";
ProviderName=\"nssm\";StartTime=\"2020-04-14 00:00:00\";EndTime=\"2020-12-31
00:00:00\"} | Sort-Object -Property TimeCreated | Where-Object -Property Message -Match
'START control|STOP control'| Select-Object -Property TimeCreated,Message | ConvertTo-
Csv"
```

Efektem jest wynik w postaci CSV, gdzie mamy zapisany czas zarejestrowania wydarzenia oraz wiadomość. Wszystko teraz zależy od nas jak przetworzymy te dane (Rysunek 5.10 - Przykład pobrania informacji o uruchomieniu usługi NSSM).

Ja wpisywałem dane ręcznie do arkusza Google, gdzie automatycznie liczyłem sobie różnice pomiędzy datami, sumy i dostępność bota (Rysunek 5.11 - Mój przykładowy arkusz do wpisywania czasów uruchomienia i awarii bota).

```
"2020-11-14 00:32:42","Service Corrade received START control, which will be handled."
"2020-11-14 07:10:31","Service Corrade received STOP control, which will be handled."
"2020-11-14 07:11:06","Service Corrade received START control, which will be handled."
"2020-11-14 23:54:45","Service Corrade received STOP control, which will be handled."
"2020-11-14 23:54:52","Service Corrade received START control, which will be handled."
"2020-11-15 00:28:54","Service Corrade received STOP control, which will be handled."
"2020-11-15 00:29:01","Service Corrade received START control, which will be handled."
"2020-11-16 08:09:46","Service Corrade received STOP control, which will be handled."
"2020-11-16 08:09:53","Service Corrade received START control, which will be handled."
"2020-11-16 12:39:45","Service Corrade received STOP control, which will be handled."
"2020-11-16 12:39:52","Service Corrade received START control, which will be handled."
"2020-11-17 06:39:53","Service Corrade received STOP control, which will be handled."
```

Rysunek 5.10 - Przykład pobrania informacji o uruchomieniu usługi NSSM

| | | | | | | | | | | | | |
|----|---------------------|---------------------|------------|---------------------|---------------------|------------|---------|-------|----------|---------|--------|-------|
| 1 | | SUM | 5032:57:29 | | SUM | 1061:03:00 | MIN | 81,69 | | | | |
| 2 | | MTTF | 10:37:05 | | MTTR | 2:14:02 | AVG | 95,25 | | | | |
| 3 | START | END | DIFFERENCE | START | END | DIFFERENCE | HISTORY | | | | | AVG |
| 4 | 2020-04-15 02:35:47 | 2020-04-15 02:35:47 | 0:00:00 | 2020-04-15 02:35:47 | 2020-04-15 02:35:47 | 0:00:00 | 100,00 | | 0:00:00 | 0:00:00 | 100,00 | 95,25 |
| 5 | 2020-04-15 02:35:47 | 2020-04-15 23:32:19 | 20:56:32 | 2020-04-15 23:32:19 | 2020-04-15 23:32:45 | 0:00:26 | 99,97 | | 10:28:16 | 0:00:13 | 99,97 | 95,25 |
| 6 | 2020-04-15 23:32:45 | 2020-04-15 23:41:57 | 0:09:12 | 2020-04-15 23:41:57 | 2020-04-15 23:42:22 | 0:00:25 | 99,93 | | 7:01:55 | 0:00:17 | 99,93 | 95,25 |
| 7 | 2020-04-15 23:42:22 | 2020-04-15 23:52:38 | 0:10:16 | 2020-04-15 23:52:38 | 2020-04-15 23:52:47 | 0:00:09 | 99,92 | | 5:19:00 | 0:00:15 | 99,92 | 95,25 |
| 8 | 2020-04-15 23:52:47 | 2020-04-16 00:47:45 | 0:54:58 | 2020-04-16 00:47:45 | 2020-04-16 00:53:33 | 0:05:48 | 99,49 | | 4:26:12 | 0:01:22 | 99,49 | 95,25 |
| 9 | 2020-04-16 00:53:33 | 2020-04-16 00:59:15 | 0:05:42 | 2020-04-16 00:59:15 | 2020-04-16 00:59:34 | 0:00:19 | 99,47 | | 3:42:47 | 0:01:11 | 99,47 | 95,25 |
| 10 | 2020-04-16 00:59:34 | 2020-04-16 00:59:50 | 0:00:16 | 2020-04-16 00:59:50 | 2020-04-16 01:00:40 | 0:00:50 | 99,41 | | 3:10:59 | 0:01:08 | 99,41 | 95,25 |
| 11 | 2020-04-16 01:00:40 | 2020-04-16 18:03:06 | 17:02:26 | 2020-04-16 18:03:06 | 2020-04-16 18:03:10 | 0:00:04 | 99,66 | | 4:54:55 | 0:01:00 | 99,66 | 95,25 |
| 12 | 2020-04-16 18:03:10 | 2020-04-16 18:08:12 | 0:05:02 | 2020-04-16 18:08:12 | 2020-04-16 18:29:59 | 0:21:47 | 98,76 | | 4:22:43 | 0:03:19 | 98,76 | 95,25 |
| 13 | 2020-04-16 18:29:59 | 2020-04-16 18:34:15 | 0:04:16 | 2020-04-16 18:34:15 | 2020-04-16 18:41:30 | 0:07:15 | 98,46 | | 3:56:52 | 0:03:42 | 98,46 | 95,25 |
| 14 | 2020-04-16 18:41:30 | 2020-04-16 18:47:53 | 0:06:23 | 2020-04-16 18:47:53 | 2020-04-16 18:48:20 | 0:00:27 | 98,45 | | 3:35:55 | 0:03:25 | 98,45 | 95,25 |
| 15 | 2020-04-16 18:48:20 | 2020-04-16 19:00:03 | 0:11:43 | 2020-04-16 19:00:03 | 2020-04-16 19:00:53 | 0:00:50 | 98,42 | | 3:18:54 | 0:03:12 | 98,42 | 95,25 |
| 16 | 2020-04-16 19:00:53 | 2020-04-16 19:16:54 | 0:16:01 | 2020-04-16 19:16:54 | 2020-04-16 19:21:08 | 0:04:14 | 98,26 | | 3:04:50 | 0:03:16 | 98,26 | 95,25 |
| 17 | 2020-04-16 19:21:08 | 2020-04-21 23:32:56 | 124:11:48 | 2020-04-21 23:32:56 | 2020-04-21 23:35:29 | 0:02:33 | 99,54 | | 11:43:54 | 0:03:13 | 99,54 | 95,25 |
| 18 | 2020-04-21 23:35:29 | 2020-04-21 23:36:02 | 0:00:33 | 2020-04-21 23:36:02 | 2020-04-21 23:37:04 | 0:01:02 | 99,53 | | 10:57:01 | 0:03:05 | 99,53 | 95,25 |
| 19 | 2020-04-21 23:37:04 | 2020-04-21 23:54:15 | 0:17:11 | 2020-04-21 23:54:15 | 2020-04-21 23:54:22 | 0:00:07 | 99,53 | | 10:17:01 | 0:02:54 | 99,53 | 95,25 |

Rysunek 5.11 - Mój przykładowy arkusz do wpisywania czasów uruchomienia i awarii bota

Na podstawie zebranych danych z arkusza stworzyłem sobie tabelkę dostępności mojego bota w poszczególnych miesiącach (Rysunek 5.12 - Zebrane dane o dostępności i awaryjności bota).

Na początku wyjaśnienie kilku oznaczeń:

- 'TOTAL MONTH'S HOURS' - podaje liczbę godzin w danym miesiącu,
- 'RU' – suma wszystkich czasów, przez które bot był dostępny,
- 'RE' – suma wszystkich czasów, przez które bot był niedostępny,
- 'RU [%]' - stosunek dostępności bota do liczby godzin w danym miesiącu wyrażone w procentach,
- 'RE [%]' - stosunek niedostępności bota do liczby godzin w danym miesiącu wyrażone w procentach,
- 'MTTR – F' – liczba awarii w danym miesiącu,
- 'MTTR' – średni czas od momentu wystąpienia awarii do momentu jej naprawy,
- 'MTTF' – średni czas od momentu ostatniej awarii lub uruchomienia bota do momentu następnej awarii,
- 'MTBF' – średni czas pomiędzy awariami – jest to suma MTTR i MTTF,
- 'AVAILABILITY' – dostępność bota wyrażona w procentach.

MTTR jest liczone ze wzoru:

$$MTTR = \sum_{i=1}^N T_i$$

gdzie:

- T_i – czas awarii,
- N – liczba awarii.

MTTF jest liczone ze wzoru:

$$MTTF = \frac{T_D - \sum_{i=1}^N T_i}{N + 1}$$

gdzie:

- T_D – czas dostępności bota,
- T_i – czas awarii bota,
- N – liczba awarii.

Dostępność bota jest liczona ze wzoru:

$$AVAILABILITY = \frac{MTTF}{MTTF + MTTR} \cdot 100$$

Przez awarię rozumiem każde wydarzenie, kiedy mój bot był niedostępny.

Przeanalizujemy teraz obrazek z wynikami.

| DATE | 2020-04-01 | 2020-05-01 | 2020-06-01 | 2020-07-01 | 2020-08-01 | 2020-09-01 | 2020-10-01 | 2020-11-01 | 2020-12-01 |
|---------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| TOTAL MONTH'S HOURS | 720:00:00 | 744:00:00 | 720:00:00 | 744:00:00 | 744:00:00 | 720:00:00 | 744:00:00 | 720:00:00 | 744:00:00 |
| RU | 380:34:42 | 734:01:03 | 669:13:54 | 735:16:31 | 740:00:35 | 688:42:36 | 234:19:11 | 507:14:57 | 460:00:39 |
| RE | 00:49:30 | 09:18:16 | 50:06:05 | 08:43:28 | 03:59:24 | 30:37:23 | 509:40:08 | 212:45:02 | 135:03:02 |
| RU [%] | 52.86% | 98.66% | 92.95% | 98.83% | 99.46% | 95.65% | 31.49% | 70.45% | 61.83% |
| RE [%] | 0.11% | 1.25% | 6.96% | 1.17% | 0.54% | 4.25% | 68.50% | 29.55% | 18.15% |
| MTTR - F | 20 | 38 | 130 | 80 | 51 | 55 | 11 | 73 | 22 |
| MTTR | 00:02:28 | 00:14:41 | 00:23:07 | 00:06:33 | 00:04:42 | 00:33:24 | 46:20:01 | 02:54:52 | 06:08:19 |
| MTTF | 34:14:47 | 18:50:18 | 05:06:49 | 09:04:39 | 14:13:51 | 12:18:37 | 19:31:39 | 06:51:17 | 26:28:34 |
| MTBF | 34:17:16 | 19:05:00 | 05:29:57 | 09:11:12 | 14:18:33 | 12:52:02 | 65:51:40 | 09:46:09 | 32:36:53 |
| AVAILABILITY | 99.88% | 98.72% | 92.99% | 98.81% | 99.45% | 95.67% | 29.65% | 70.17% | 81.18% |

Rysunek 5.12 - Zebrane dane o dostępności i awaryjności bota

Stosunek dostępności bota do liczby godzin w danym miesiącu wyrażone w procentach waha się w przedziale 31.49% - 99.46%. W kwietniu ta wartość była mała, gdyż bot został uruchomiony od połowy kwietnia, następnie widzimy w kolejnych miesiącach utrzymanie się wartości w przedziale 92% - 99%, następnie jest gwałtowny spadek w październiku po czym w listopadzie wzrost i w grudniu następuje zamknięcie bota z dostępnością ok. 62%. Widać to również po stosunku niedostępności bota do liczby godzin w danym miesiącu wyrażone w procentach, gdzie w październiku jest najwyższa, a potem spada. Przed listopadem jest na niskim, akceptowalnym poziomie. Bot średnio działał od ok. 5-34 godzin w miesiącu zanim nie wystąpiła awaria. Awaria pod postacią niedostępności bota trwała od 2 minut do nawet ok. 46 godzin. Na końcu można

wyczytać dostępność bota, która opiera się na czasie dostępności bota i awarii. Można zauważyć jak wcześniej, że w październiku nastąpił dramatyczny spadek dostępności bota przez awarie.

Dlaczego od października liczba awarii była taka wysoka? Ponieważ po aktualizacji programu występowały różne problemy, kiedy bot nie mógł się uruchomić.

26 listopada 2020 r. zaczęły występować błędy związane z poprawnym wyświetleniem awatarów – wygląd awatara przypominał chmurę, która zazwyczaj symbolizuje ładowanie jego wyglądu podczas pobytu w Second Life. Incydent nazwano 'Thanksgiving Bakefail'. Ten błąd do Corrade dotarł gdzieś na początku grudnia, kiedy zaczęto zgłaszać problemy z brakiem dostępności do szafy bota, przez co nie mógł ani zrezzować przedmiotów ani swojego wyglądu. Błąd po stronie Second Life został naprawiony po jakiś 2-3 tygodniach.

Kolejna awaria również występowała po stronie Second Life w listopadzie, gdzie region na którym przebywał bot miał potężne opóźnienia, które powodowały albo brakiem możliwości odpalania się kluczowych skryptów w ubraniu bota albo ich spowolnionego działania. W tej sytuacji zdecydowałem się na przeniesienie do innego regionu, gdzie opóźnienia były minimalne.

5.4. Podstawy budowy bota

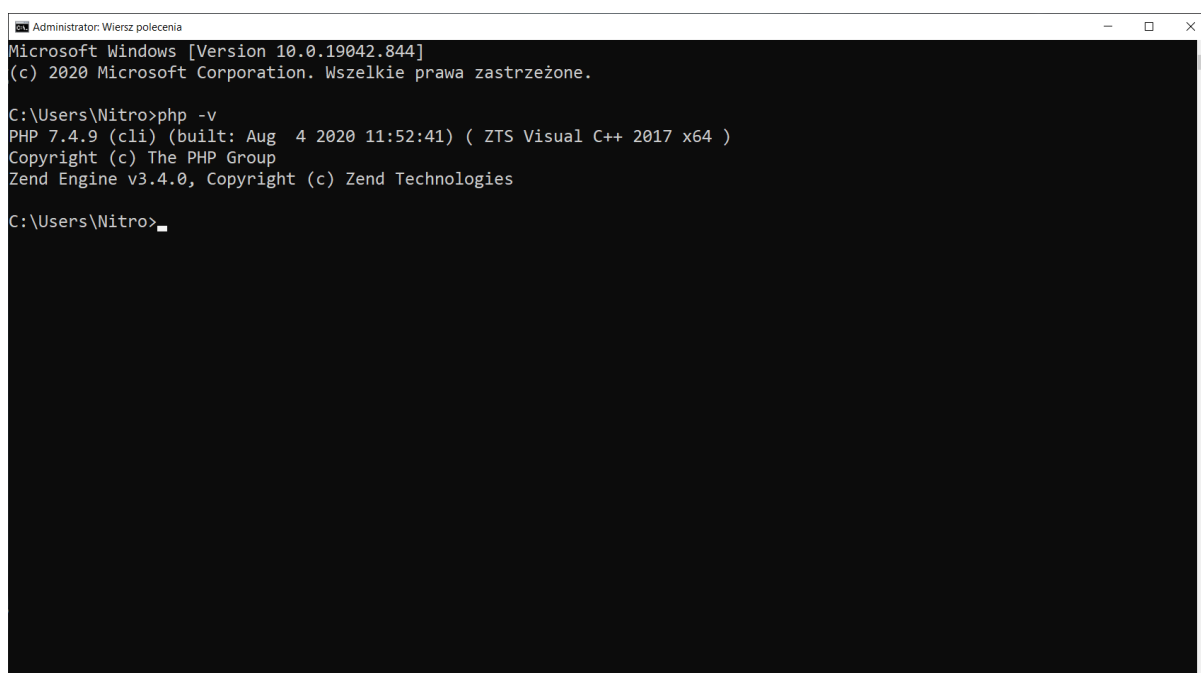
5.4.1. Wstęp

W tym rozdziale pokażę Tobie jak zbudować podstawowego bota.

Nasze założenia:

- całe przetworzenie wykonujemy poza środowiskiem Second Life,
- język programowania, w którym będziemy tworzyć mózg i serce bota poza Second Life jest PHP,
- Serce bota w Second Life będzie w postaci primu, który będzie częścią ubrania bota.

Na początku należy sprawdzić, czy PHP jest dodany do ścieżki systemowej, co pozwoli nam na wywołanie programu PHP z każdego miejsca systemu.



```
Administrator: Wiersz polecenia
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. Wszelkie prawa zastrzeżone.

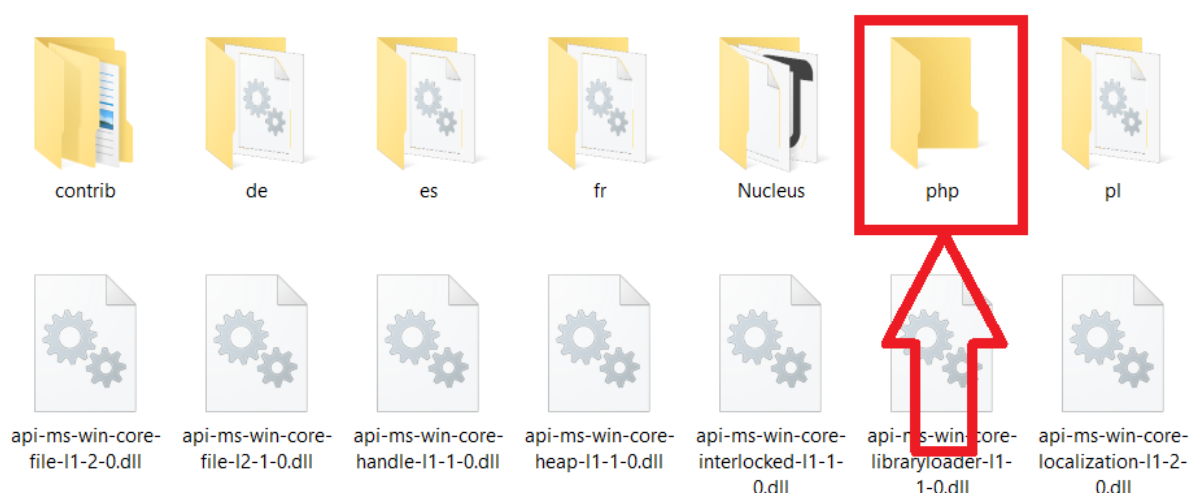
C:\Users\Nitro>php -v
PHP 7.4.9 (cli) (built: Aug  4 2020 11:52:41) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

C:\Users\Nitro>
```

Rysunek 5.13 - Sprawdzenie czy PHP jest dodany do ścieżki systemowej

Jeżeli po zatwierdzeniu polecenia nie pojawia się m.in. wersja PHP a pojawia się informacja, że program nie został znaleziony, to należy ściągnąć PHP **w wersji 7.x** dla systemu Windows ze strony <https://windows.php.net/download/> w postaci archiwum ZIP. Wybieramy wersję x64 Thread Safe. Wypakowujemy archiwum np. do C:\php. Wchodzimy do właściwości systemu (Klikamy prawym przyciskiem myszy na 'Ten komputer' lub 'Mój komputer' następnie wybieramy 'Właściwości'). Następnie wybieramy kolejno 'Zaawansowane ustawienia systemu' => na karcie 'Zaawansowane' klikamy na 'Zmienne środowiskowe'. W części 'Zmienne systemowe' odszukaj 'Path', dodaj do niego ścieżkę do folderu C:\php. Pozatwierdzaj wszystko, zrestartuj komputer.

W folderze swojego bota utwórz folder o nazwie 'php'. W tym folderze będziemy trzymać serce bota. Jest to też łatwy sposób na wywoływanie poszczególnych części napisanych w języku PHP.



Rysunek 5.14 - Folder PHP w folderze Corrade

Przechodzimy do wcześniej utworzonego folderu.

W tym folderze na początku zainstalujemy Composer, który pozwala na łatwe zarządzanie zależnościami projektu. Na początku, uruchamiany wiersz poleceń w folderze PHP. W tym celu najszybciej jest nacisnąć lewy klawisz SHIFT i kliknąć prawym przyciskiem myszy gdzieś na puste pole obszaru folderu, a następnie z menu wybieramy 'Otwórz okno tutaj'.

Uwaga! Jeżeli nie ma takiej opcji to można dodać odpowiednie wpisy do rejestru lub w Menu Start wpisujemy cmd a następnie za pomocą komendy cd przechodzimy do wybranego przez nas folderu. Wchodzimy na stronę Composer (<https://getcomposer.org/download/>), gdzie znajdziemy polecenia do wpisania w wierszu poleceń w celu pobrania Composera.

Kopiujemy wszystkie linie za jednym zamachem i wklejamy do wiersza poleceń (używamy prawego przycisku myszy).

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '756890a4488ce9024fc62c56153228907f1545c228516cbf63f8
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Rysunek 5.15 - Instalacja Composer

Jeżeli wszystko poszło dobrze, to w folderze php znajdziemy plik composer.phar.

Aby wywołać pomoc Composer wpisujemy:

`php composer.phar`

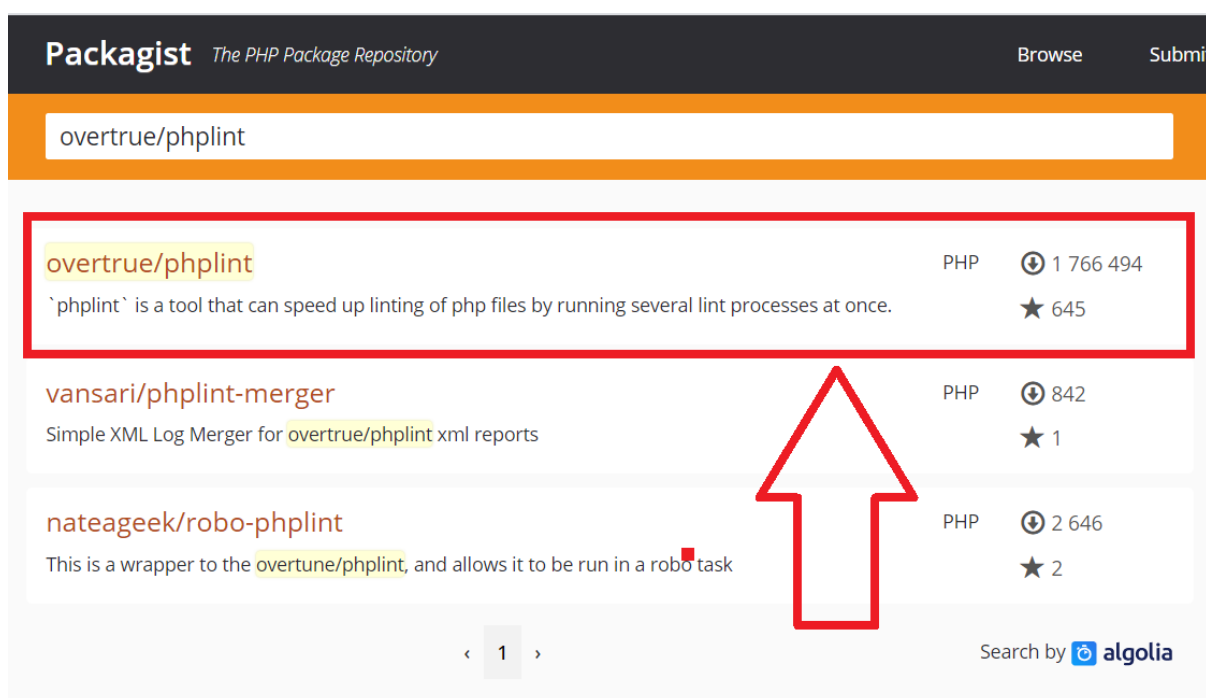
Najważniejsze polecenia to:

- `php composer.phar require nazwaPakietu` - instaluje dany pakiet,
- `php composer.phar remove nazwaPakietu` - usuwa wcześniej zainstalowany pakiet,
- `php composer.phar update` - aktualizacja wszystkich zainstalowanych pakietów,
- `php composer.phar selfupdate` - aktualizuje Composer,
- `php composer.phar outdated` - pokazuje przestarzałe wersje zainstalowanych pakietów.

Pakiety znajdziesz pod adresem <https://packagist.org/>, są to po prostu projekty PHP, które są umieszczone na serwerze GIT. Możesz je samodzielnie ściągnąć bez programu Composer, włączyć do swojego projektu, jednakże będziesz musiał pilnować samodzielnie czy są nowe wersje na stronach projektu, itd., więc Composer po prostu ułatwia ten proces.

Na początku zainstalujemy pakiet o nazwie: `overtrue/phplint`, który pozwala na sprawdzenie składni plików PHP. Będzie on przydatny nam podczas pracy nad sercem bota w wyłapywaniu błędów w składni PHP, które mogą zablokować pracę naszego bota.

Na stronie packagist.org wpisujemy nazwę tego pakietu i klikamy na pierwszą propozycję z listy wyszukiwanych.



Rysunek 5.16 - Wyszukiwanie pakietu na stronie Packagist

Na tej stronie mamy szczegółowe dane odnośnie pakietu. Powinniśmy zwrócić uwagę na wymagania danego pakietu. Owszem nie musimy przy każdym pakiecie sprawdzać jego wymagania, gdyż Composer podczas instalacji sprawdza czy wymagania pakietu są zgodne ze stanem komputera na którym ten pakiet ma być zainstalowany. Więc jeżeli coś będzie nie tak, to anuluje instalację i powiadomi, które części wymagań nie spełniają oczekiwań.

overtrue/phplint

composer require overtrue/phplint

`phplint` is a tool that can speed up linting of php files by running several lint processes at once.

Maintainers

Details

github.com/overtrue/phplint

Source

Issues

Installs: 1 770 684

Dependents: 445

Suggesters: 0

Security: 0

Stars: 641

Watchers: 20

Forks: 64

Open Issues: 1

2.3.5 2020-12-11 17:27 UTC

| requires | requires (dev) | suggests |
|--|--|----------|
| <ul style="list-style-type: none"> php: >=5.5.9 ext-json: * n98/junit-xml: 1.0.0 symfony/console: ^3.2 ^4.0 ^5.0 symfony/finder: ^3.0 ^4.0 ^5.0 symfony/process: ^3.3 ^4.0 ^5.0 symfony/yaml: ^3.0 ^4.0 ^5.0 | <ul style="list-style-type: none"> brainmaestro/composer-git-hooks: ^2.7 friendsofphp/php-cs-fixer: ^2.16 jakub-onderka/php-console-highlighter: ^0.3.2 ^0.4 | None |

| provides | conflicts | replaces |
|----------|-----------|----------|
| None | None | None |

MIT fa2a2ba64f997f268171a5b12d141b2fe94e5ad0

overtrue <anzhengchao@gmail.com>

#syntax #check #lint #phplint

dev-master

2.3.5

2.3.4

2.3.3

2.3.2

2.3.1

2.3.0

2.2.0

2.1.0

2.0.3

2.0.2

2.0.0

This package is auto-updated.

Last update: 2021-02-28 07:02:46 UTC

Rysunek 5.17 - Strona overtrue/phplint

Wracamy do instalacji pakietu.

W tym celu wpisujemy w wierszu poleceń:

```
php composer.phar require overtrue/phplint
```

Polecenie ściągnie pakiet wraz zależnościami.

Uwagi!

- od czasu do czasu (np. co 2 tyg. lub raz na miesiąc) sprawdzamy aktualizacje pakietów poprzez: `php composer.phar update`,
- przestarzałe pakiety – przez `php composer.phar outdated` i w razie konieczności usuwamy je i instalujemy je na nowo,
- również dokonujemy aktualizację samego Composer.

Jeżeli wszystko się udało to w folderze znajdziemy folder `vendor` - tutaj są trzymane nasze pakiety, które będziemy włączać do projektu, `composer.json` - plik z informacjami, jakie pakiety i jakie ich wersje zostały zainstalowane, `composer.lock` - plik z szerszymi informacjami odnośnie zainstalowanych pakietów.

Utwórzmy teraz plik `phplint.bat` do którego kopiujemy poniższą zawartość:

```
SET mypath=%~dp0
SET mypath=%mypath:~0,-1%
vendor\bin\phplint -j 5 --no-cache -n --exclude=vendor -vvv "%mypath%" -n >
phplint.txt
```

Pozwala ona na użycie wspomnianego `phplint` z wykluczeniem folderu `vendor`, z pełną diagnostyką, gdzie całość zapisuje do pliku `phplint.txt`.

Następnie tworzymy jeden z kluczowych plików PHP o nazwie `servebot.php` (może być inna nazwa), który będzie obsługiwał żądania kierowane z Second Life.

Wklejamy do niego następującą zawartość:

```
<?php
error_reporting(0);

chdir(__DIR__);

ini_set('memory_limit', '512M');
set_time_limit(300);
date_default_timezone_set('UTC');

require_once('vendor/autoload.php');
require_once('config.php');
require_once('inPolygon.class.php');
require_once('functions.php');

$args = $argv;
$args = array_slice($args, 1);
```

```

$args = implode(' ', $args);
preg_match('/\&\_script\=([^\&]+)/', $args, $found);
$scriptNameExternal = (isset($found[1])) ? $found[1] : '';
$args = preg_replace('/(\?|\&)([^\&]+\)=([^\&]+)/', '', $args);
$body = explode('|', $args);
$body_t = array();
for($i=0;$i<count($body);$i+=2):
    $body_t[ $body[$i] ] = $body[$i+1];
endfor;
$body = $body_t;

switch($body['action']){

}

```

Wyjaśnienie:

- `error_reporting(0)` - wyłącza jakiekolwiek raportowanie błędów,
- `chdir()` - wykonuje wszystko w bieżącym folderze,
- `ini_set('memory_limit', '512M');` - ustawia maksymalny limit pamięci na 512 MB dla wszystkich skryptów,
- `date_default_timezone_set('UTC');` - ustawia domyślną strefę czasową na uniwersalną,
- `require_once('vendor/autoload.php');` - załącza plik `vendor/autoload.php` od Composer,
- `require_once('config.php');` - załącza plik konfiguracyjny `config.php` (utworzymy go później),
- `require_once('inPolygon.class.php');` - załącza plik `inPolygon.class.php` związany ze sprawdzeniem czy dany punkt należy do wielokąta (utworzymy go później),
- `require_once('functions.php');` - załącza plik `functions.php` związany z funkcjami (utworzymy go później).

Dalsze linie pozwalają na przekształcenie parametrów wysyłanych do tego pliku do postaci tablicy.

Bot z Second Life może wywołać podobną postać do:

```
php\serverbot.php 'parametr1|wartość1|parametr2|wartość2|...'
```

wówczas ta część tego pliku przekształci do tablicy:

```

$body['parametr1'] = 'wartość1';
$body['parametr2'] = 'wartość2';

```

Znakiem rozdzielającym jest pionowa linia '|'.

Ta część skryptu również na wszelki wypadek wyłuskuje nazwę skryptu LSL, który wcześniej wywołał plik servebot.php (o ile ta nazwa została przesłana).

switch(\$body['action']) - zaimplementowaliśmy wstępnie przełącznik po wartościach argumentu 'action', który będzie odpowiedzialny za wywoływanie akcji.

```
php\serverbot.php 'action|nazwa akcji|parametr2|wartość2|...'
```

Utwórz plik functions.php w zasadzie na razie pustą zawartością:

```
<?php
```

Utwórz plik inPolygon.class.php z następującą zawartością:

```
<?php
/*
Description: The point-in-polygon algorithm allows you to check if a point is
inside a polygon or outside of it.
Author: Michaël Niessen (2009)
Website: http://AssemblySys.com

If you find this script useful, you can show your
appreciation by getting Michaël a cup of coffee ;)
donation to Michaël

As long as this notice (including author name and details) is included and
UNALTERED, this code is licensed under the GNU General Public License version 3:
http://www.gnu.org/licenses/gpl.html
*/

class pointLocation {
    var $pointOnVertex = true; // Check if the point sits exactly on one of the
vertices?

    function pointLocation() {
    }

    function pointInPolygon($point, $polygon, $pointOnVertex = true) {
        $this->pointOnVertex = $pointOnVertex;

        // Transform string coordinates into arrays with x and y values
        $point = $this->pointStringToCoordinates($point);
        $vertices = array();
        foreach ($polygon as $vertex) {
            $vertices[] = $this->pointStringToCoordinates($vertex);
        }
    }
}
```

```

    // Check if the point sits exactly on a vertex
    if ($this->pointOnVertex == true and $this->pointOnVertex($point, $vertices)
== true) {
        return "vertex";
    }

    // Check if the point is inside the polygon or on the boundary
    $intersections = 0;
    $vertices_count = count($vertices);

    for ($i=1; $i < $vertices_count; $i++) {
        $vertex1 = $vertices[$i-1];
        $vertex2 = $vertices[$i];
        if ($vertex1['y'] == $vertex2['y'] and $vertex1['y'] == $point['y'] and
$point['x'] > min($vertex1['x'], $vertex2['x']) and $point['x'] < max($vertex1['x'],
$vertex2['x'])) { // Check if point is on an horizontal polygon boundary
            return "boundary";
        }
        if ($point['y'] > min($vertex1['y'], $vertex2['y']) and $point['y'] <=
max($vertex1['y'], $vertex2['y']) and $point['x'] <= max($vertex1['x'],
$vertex2['x']) and $vertex1['y'] != $vertex2['y']) {
            $xinters = ($point['y'] - $vertex1['y']) * ($vertex2['x'] -
$vertex1['x']) / ($vertex2['y'] - $vertex1['y']) + $vertex1['x'];
            if ($xinters == $point['x']) { // Check if point is on the polygon
boundary (other than horizontal)
                return "boundary";
            }
            if ($vertex1['x'] == $vertex2['x'] || $point['x'] <= $xinters) {
                $intersections++;
            }
        }
    }

    // If the number of edges we passed through is odd, then it's in the polygon.

    if ($intersections % 2 != 0) {
        return "inside";
    } else {
        return "outside";
    }
}

function pointOnVertex($point, $vertices) {
    foreach($vertices as $vertex) {
        if ($point == $vertex) {
            return true;
        }
    }
}

```

```

function pointStringToCoordinates($pointString) {
    $coordinates = explode(" ", $pointString);
    return array("x" => $coordinates[0], "y" => $coordinates[1]);
}

}
?>

```

Tworzymy plik config.php do którego wklejamy zawartość:

```

<?php
//Bot Connection
define('BOT_GROUP',      'GROUPNAME');
define('BOT_PASSWORD',   'PASSWORD');
define('BOT_URL',        'http://127.0.0.1:8080');

//constants Second Life
define('TEXTURE_DEFAULT',      '89556747-24cb-43ed-920b-47caed15465f');
define('TEXTURE_BLANK',        '5748decc-f629-461c-9a36-a35a221fe21f');
define('TEXTURE_TRANSPARENT',  '8dcd4a48-2d37-4909-9f78-f7a9eb4ef903');

```

- zamiast GROUPNAME podajemy nazwę grupy do której przynależy bot,
- zamiast PASSWORD podajemy hasło do tej grupy,
- w BOT_URL podajemy URL do serwera HTTP bota (127.0.0.1 - skrypty są wykonywane lokalnie na tym samym komputerze na którym będzie uruchomiony bot, 8080 - port musi być taki sam jak w konfiguracji),
- stałe z prefiksem TEXTURE_ definiują UUID tekstur w Second Life (domyślna, pusta, transparentna).

Teraz powinniśmy się zastanowić, gdzie będziemy przechowywać dalszą konfigurację dla naszego bota. Jeżeli posiadasz materiały dodatkowe dołączone do tego ebooka to w bot_installer\corrade\bot_ai\php to tam konfiguracja jest przechowywana w oddzielnym pliku INI, w tym rozdziale natomiast wszystko będziemy trzymać w pliku config.php.

Gdy będziesz dokonywał zmian w tym pliku, na wszelki wypadek wywołuj phplint.bat i sprawdź czy w pliku phplint.txt zostały wykryte jakieś błędy składniowe.

Zastanówmy się teraz jakie 'akcje' ma przetwarzać nasz bot.

Moje propozycje są podane poniżej.

5.4.2. Proponowane funkcje dla bota

➤ Pobieranie wartości zdefiniowanej w pliku konfiguracyjnej.

W servebot.php w switch dodajemy:

```
case 'getConfig':
    echo getConfig($body['name']);
    break;
//////////
```

Dodaje tą część akcji o nazwie getConfig, która pozwoli na pobieranie wartości zmiennej stałej zdefiniowanej w config.php.

Mamy tutaj wywołanie funkcji, która przyjmuje jako parametr nazwę zmiennej stałej.

Przykład wywołania:

```
php\serverbot.php 'action|getConfig|name|NazwaZmiennejStałej'
```

W pliku functions.php dodajemy:

```
function getConfig($name)
{
    return (defined($name)) ? constant($name) : '';
}
```

Funkcja ta sprawdza, czy stała zmienna w config.php istnieje, jeżeli tak to pobiera jej wartość, jeżeli nie została zdefiniowana zwraca pusty ciąg znakowy.

➤ Obsługa cache

W moim projekcie bota znajduje się obsługa cache, czyli przechowywanie najczęściej używanych wartości przez bota bez potrzeby ponownego wyliczania, przetwarzania, odpytywania Second Life. Są dwa źródła zapisu: do pliku cache.ini lub do bazy danych. Ja zazwyczaj wybieram zapis do pliku cache.

W servebot.php w switch dodajemy:

```
case 'setCache':
    setCacheValue($body['name'], $body['value']);
```

```
break;
//////////
case 'getCache':
    echo getCacheValue($body['name']);
break;
//////////
case 'deleteCache':
    deleteCacheName($body['name']);
break;
//////////
```

Sposób wywołania:

```
php\serverbot.php 'action|setCache|name|nazwaParametruWCache|value|wartośćDlaParametru'
php\serverbot.php 'action|getCache|name|nazwaParametruWCache '
php\serverbot.php 'action|deleteCache|name|nazwaParametruWCache '
```

w config.php dodaj:

```
//Database Connection
define('DB_CONNECTION', 'mysql:host=localhost;dbname=corrade');
define('DB_LOGIN', 'root');
define('DB_PASSWORD', '12345');

define('GENERAL_cacheTypeWriteDatabase', '0');
```

w functions.php dodaj:

```
//Insert or Update
function setCacheValue($name, $value)
{
    if(getConfig('GENERAL_cacheTypeWriteDatabase') == '1'):
        try
        {
            $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
            $pdo->beginTransaction();
            $pdo->exec('LOCK TABLE cache WRITE, cache as cacheRead READ');
            $stmt = $pdo->prepare('INSERT INTO 'cache' ('name', 'value') VALUES (:name,
:value) ON DUPLICATE KEY UPDATE value=:value');
            $stmt->execute(array(
                ':name' => $name,
                ':value' => $value,
```



```
    ));
    $pdo->commit();
    $pdo->exec('UNLOCK TABLES');
}
catch(Exception $e)
{
    writeDebugFile($e->getTraceAsString());
}
else:
    safefilerewriteCache($name, $value);
endif;
}

function writeDebugFile($msg, $type = 'warn')
{
    @mkdir('logs');
    safefilerewrite('logs/log_'.date('d.m.Y').'.txt', date('d.m.Y H:i:s') . ' ' . $type
. ' ' . $msg . "\n", 'a+');
}

function safefilerewriteCache($name, $value)
{
    $iniCache = parse_ini_file("cache.ini", true);
    $iniCache['cache'][$name] = $value;
    write_php_ini($iniCache, "cache.ini");
}

function safefilerewrite($fileName, $dataToSave, $mode = 'w+')
{
    if ($fp = fopen($fileName, $mode))
    {
        $startTime = microtime(TRUE);
        do
        {
            $canWrite = flock($fp, LOCK_EX);
            // If lock not obtained sleep for 0 - 100 milliseconds, to avoid collision
and CPU load
            if(!$canWrite) usleep(round(rand(0, 100)*1000));
        } while ((!$canWrite)and((microtime(TRUE)-$startTime) < 5));

        //file was locked so now we can store information
```

```

        if ($canWrite)
        {
            fwrite($fp, $dataToSave);
            flock($fp, LOCK_UN);
        }
        fclose($fp);
    }
}

function write_php_ini($array, $file)
{
    $res = array();
    foreach($array as $key => $val)
    {
        if(is_array($val))
        {
            $res[] = "[$key]";
            foreach($val as $skey => $sval) $res[] = "$skey = ".(is_numeric($sval) ?
$sval : "'".$sval.'");
        }
        else $res[] = "$key = ".(is_numeric($val) ? $val : "'".$val.'");
    }
    safefilerewrite($file, implode("\r\n", $res));
}

//Get Cache Value
function getCacheValue($name)
{
    if(getConfig('GENERAL_cacheTypeWriteDatabase') == '1'):
        try
        {
            $value = "";
            $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
            $pdo->beginTransaction();
            $pdo->exec('LOCK TABLE cache WRITE, cache as cacheRead READ');
            $stmt = $pdo->prepare('SELECT 'value' from 'cache' as cacheRead WHERE
name=:name');
            $stmt->execute(array(
                ':name' => $name,
            ));
            $data = $stmt->fetch();

```

```
        $value = $data['value'];
        $pdo->commit();
        $pdo->exec('UNLOCK TABLES');
        return $value;
    }
    catch(Exception $e)
    {
        writeDebugFile($e->getTraceAsString());
        return "";
    }
    else:
        $iniCache = parse_ini_file("cache.ini", true);
        return $iniCache['cache'][$name];
    endif;
}

//delete cache
function deleteCacheName($name)
{
    if(getConfig('GENERAL_cacheTypeWriteDatabase') == '1'):
        try
        {
            $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
            $pdo->beginTransaction();
            $pdo->exec('LOCK TABLE cache WRITE, cache as cacheRead READ');
            $stmt = $pdo->prepare('DELETE FROM cache WHERE name=:name');
            $stmt->execute(array(
                ':name' => $name,
            ));
            $pdo->commit();
            $pdo->exec('UNLOCK TABLES');
        }
        catch(Exception $e)
        {
            writeDebugFile($e->getTraceAsString());
        }
        else:
            safefilerewriteDeleteCache($name);
        endif;
    }
}
```

```

function safefilerewriteDeleteCache($name)
{
    if ($fp = fopen('cache.ini', 'w+'))
    {
        $startTime = microtime(TRUE);
        do
        {
            $canWrite = flock($fp, LOCK_EX);
            if(!$canWrite) usleep(round(rand(0, 100)*1000));
        } while (((!$canWrite)and((microtime(TRUE) - $startTime) < 5)));

        if ($canWrite)
        {
            $iniCache = parse_ini_file("cache.ini", true);
            unset($iniCache['cache'][$name]);

            $res = array();
            foreach($iniCache as $key => $val)
            {
                if(is_array($val))
                {
                    $res[] = "[$key]";
                    foreach($val as $skey => $sval)
                        $res[] = "$skey = ".(is_numeric($sval) ? $sval : "'".$sval.'"');
                }
                else{
                    $res[] = "$key = ".(is_numeric($val) ? $val : "'".$val.'"');
                }
            }

            fwrite($fp, implode("\r\n", $res));
            flock($fp, LOCK_UN);
        }
        fclose($fp);
    }
}

//delete all from cache
function deleteAllCache()
{

```

```

if(getConfig('GENERAL_cacheTypeWriteDatabase') == '1'):
    try
    {
        $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
        $pdo->beginTransaction();
        $pdo->exec('LOCK TABLE cache WRITE, cache as cacheRead READ');
        $stmt = $pdo->prepare('DELETE FROM cache');
        $stmt->execute();
        $pdo->commit();
        $pdo->exec('UNLOCK TABLES');
    }
    catch(Exception $e)
    {
        writeDebugFile($e->getTraceAsString());
    }
else:
    @unlink('cache.ini');
    @unlink('php/cache.ini');
endif;
}

```

Kwestia wyjaśnienia.

Dodaliśmy do switch możliwość zapisywania do cache, pobierania wartości z cache o podanej zmiennej, kasacja w cache o podanej zmiennej, kasacja wszystkich zmiennych w cache.

W config.php dodaliśmy wpisy dotyczące bazy danych oraz typ zapisu do cache (GENERAL_cacheTypeWriteDatabase).

Jeżeli ta zmienna ma wartość:

- 0 - zapisuje do pliku cache.ini,
- 1 - zapisuje cache do bazy danych.

Jeżeli wybrano zapis do bazy danych to ustaw poprawne wartości w poprzednich zmiennych oraz dodaj nową tabelę cache do bazy danych:

```

CREATE TABLE IF NOT EXISTS 'cache' (
    'name' varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
    'value' longtext COLLATE utf8mb4_unicode_ci NOT NULL,
    PRIMARY KEY ('name')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

W functions.php dodaliśmy wszystkie funkcje, które obsługują nasz cache.

Dodatkowo znajdziemy tutaj uniwersalne funkcje do bezpiecznego zapisu danych do pliku.

➤ Przetwarzanie wiadomości prywatnych

Jako przykład zrobimy możliwość przetwarzania wiadomości prywatnych wysłanych do bota.

W servebot.php w switch dopisujemy:

```
case 'serveIM':  
    $avatar = AvatarUUIDtoName($body['uuid']);  
    //check ban  
    if(BlackListIMExist($body['uuid'])):  
        SendMessageToAvatar($body['uuid'], $avatar.", I don't talk with you. You are  
very rude.");  
        exit();  
    endif;  
    require_once('part.im.php');  
break;  
//////////
```

Ta część kodu dodaje akcję o nazwie 'serveIM'.

Już na samym początku za pomocą funkcji AvatarUUIDtoName() próbujemy zamienić przesłany UUID awatara, który wysłał wiadomość do bota do postaci jego imienia i nazwiska, które będzie używane przy wysyłaniu zwrotnej odpowiedzi (imitacja spersonalizowanej wiadomości).

Dalej mamy małą część z możliwością sprawdzenia, czy dany awatar nie jest wpisany na czarną listę rozmówców. A jeżeli jest, to jest wysyłany stosowny komunikat.

Tak naprawdę, jeżeli nie zależy nam na zabezpieczeniu przed potokiem wiadomości czy przed korzystaniem z poleceń administracyjnych przez nieuprawnione osoby to możemy skasować tą część.

Uwaga! Aby skorzystać z tej opcji należy skonfigurować połączenie do bazy danych i dodać tabelę:

```
CREATE TABLE IF NOT EXISTS 'blacklist.im' (  
    'agentuuid' varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
    'agentname' text COLLATE utf8mb4_unicode_ci NOT NULL,  
    'time' text COLLATE utf8mb4_unicode_ci NOT NULL,  
    PRIMARY KEY ('agentuuid')  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Dalej mamy załączenie pliku `part.im.php`, w którym będziemy dokonywać przetwarzania wiadomości.

Najlepiej właśnie poszczególne części w switch rozwijać w osobnych plikach PHP, pozwoli to na zapanowanie nad porządkiem w pliku.

W `functions.php` dodajemy:

```
function AvatarUUIDtoName($uuid)
{
    $params = array(
        "command" => "batchavatarkeytoname",
        "group" => BOT_GROUP,
        "password" => BOT_PASSWORD,
        "avatars" => wasArrayToCSV(array($uuid))
    );

    $ret = SendToBot($params);
    $ret = wasCSVToArray(wasKeyValueGet("data", $ret));
    return $ret[1];
}

function BlackListIMExist($uuid)
{
    try {
        $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
        $stmt = $pdo->prepare('SELECT COUNT(*) from 'blacklist.im' WHERE
agentuuid=:uuid');
        $stmt->execute(array(
            ':uuid' => $uuid
        ));
        if($stmt->fetchColumn() == 1):
            return 1;
        endif;
        return 0;
    }catch(Exception $e1){
        return 0;
    }
}

function SendMessageToAvatar($agent, $message)
{
    $params = array(
```

```

        'command' => 'tell',
        'group' => BOT_GROUP,
        'password' => BOT_PASSWORD,
        'message' => $message,
        'entity' => 'avatar',
        'agent' => $agent
    );
    SendToBot($params, 0);
}

function wasArrayToCSV($a)
{
    return implode(',',
        array_map(
            function($o)
            {
                $o = str_replace('"', '""', $o);
                switch( (strpos($o, ' ') !== FALSE) || (strpos($o, '"') !== FALSE) ||
                    (strpos($o, ',') !== FALSE) || (strpos($o, '\r') !== FALSE) || (strpos($o, '\n') !==
FALSE))
                {
                    case TRUE:
                        return '"' . $o . '"';
                    default:
                        return $o;
                }
            }, $a)
    );
}

function SendToBot($param_tab, $ACK = 1, $connectTimeout = 10, $Timeout = 30)
{
    $getTypeProcessingLang = 'was';
    if(file_exists('../Configuration.xml')):
        $xmldata = simplexml_load_file('../Configuration.xml');
        $getTypeProcessingLang = trim(strtolower($xmldata->ScriptLanguage));
    endif;

    if($getTypeProcessingLang == 'json'):
        $postvars = json_encode($param_tab);
    else:

```



```

        array_walk(
            $param_tab, function(&$value, $key)
            {
                $value = rawurlencode($key)."=".rawurlencode($value);
            });
        $postvars = implode('&', $param_tab);
endif;

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, BOT_URL);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, $postvars);
curl_setopt($curl, CURLOPT_ENCODING, true);
curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, $connectTimeout);
curl_setopt($curl, CURLOPT_TIMEOUT, $Timeout);

if($ACK == 1):
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
    $return = curl_exec($curl);
    return $return;
else:
    ob_start();
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 0);
    curl_setopt($curl, CURLOPT_TIMEOUT, 1);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    curl_setopt($curl, CURLOPT_FORBID_REUSE, 1);
    curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, 1);
    curl_setopt($curl, CURLOPT_FRESH_CONNECT, 1);
    curl_exec($curl);
    ob_end_clean();
endif;

curl_close($curl);
}

function wasCSVToArray($csv)
{
    $csv = preg_replace('/\\/','\\', $csv);
    $csv = preg_replace('/,/','"', $csv);
    if(substr($csv, 0, 1) != '"') $csv = '"' . $csv;
    if(substr($csv, -1, 1) != '"') $csv .= '"';
}

```

```

    $csv = preg_replace('/"<([\d]+(?:\.\d+)?)"\s*([\d]+(?:\.\d+)?)"\s*([\d]+(?:\.\d+)?)">"/U', '"<$1,$2,$3>"', $csv);
    preg_match_all('/"(.*)" /U', $csv, $l);
    return $l[1];
}

function wasKeyValueGet($key, $data, $base64decode = false)
{
    $sreturn = '';
    if($LANG == 'json'):
        $sret_json = json_decode($data);
        $sreturn = $sret_json->$key;
        if(is_array($sreturn)):
            $sreturn = implode(',', $sreturn);
        endif;
    else:
        parse_str($data, $response);
        $sreturn = $response[$key];
    endif;

    $sreturn = rawurldecode($sreturn);
    $sreturn = preg_replace('/^\'+/', '', $sreturn);
    $sreturn = preg_replace('/\'+$/ ', '', $sreturn);
    $sreturn = preg_replace('/\'+"/', '', $sreturn);
    $sreturn = trim($sreturn);

    if($base64decode):
        $sreturn = base64_decode($sreturn);
    endif;

    if(strtolower($key) == 'success'):
        $sreturn = (strtolower($sreturn) == 'true') ? '1' : '0';
    endif;
    if(($key == 'data') && ($sreturn == 'data')):
        $sreturn = '';
    endif;
    if(($key == 'success') && ($sreturn == 'data')):
        $sreturn = 0;
    endif;
    return $sreturn;
}

```

```
function BlackListIMAdd($uuid)
{
    try {
        $avatar = AvatarUUIDtoName($uuid);
        if(str_word_count($avatar) == 1):
            $avatar.=' resident';
        endif;

        $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
        $stmt = $pdo->prepare('INSERT INTO 'blacklist.im' ('agentuuid', 'agentname',
'time') VALUES (:agentuuid, :agentname, :time)');
        $stmt->execute(array(
            ':agentuuid' => $uuid,
            ':agentname' => $avatar,
            ':time' => strtotime('now'),
        ));
    }catch(Exception $e1){}
}

function BlackListIMInc($uuid)
{
    try {
        $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
        $stmt = $pdo->prepare('UPDATE 'blacklist.im' SET count=count+1 WHERE
agentuuid=:agentuuid');
        $stmt->execute(array(
            ':agentuuid' => $uuid,
        ));
    }catch(Exception $e1){}
}

function BlackListIMDel($uuid)
{
    try {
        $pdo = new PDO(DB_CONNECTION, DB_LOGIN, DB_PASSWORD);
        $stmt = $pdo->prepare('DELETE FROM 'blacklist.je' WHERE 'avatar' LIKE "%:avatar
%" OR 'uuid' LIKE "%:uuid%");
        return (int)$stmt->execute(array(
            ':uuid' => $uuid
        ));
    }
}
```

```

    }catch(Exception $e1){
        return 0;
    }
}

function checkPerm($avatar, $func)
{
    $retValue = 0;
    $params = array(
        'command' => 'getmemberroles',
        'group' => BOT_GROUP,
        'password' => BOT_PASSWORD,
        'agent' => $avatar,
        'target' => BOT_GROUP
    );
    $ret = SendToBot($params);

    if(wasKeyValueGet("success", $ret)):
        $data = wasKeyValueGet("data", $ret);
        $groups = wasCSVToArray($data);
        for($i=0;$i<count($groups);$i++):
            if($groups[$i] == $func):
                $retValue = 1;
                break;
            endif;
        endfor;
    endif;
    return $retValue;
}

```

W config.php dodaj:

```
define('GENERAL_groupFunctionAdminBot', 'botmanager');
```

Tutaj podajemy nazwę roli w grupie, w której znajduje się bot, gdzie przepisane awatary do tej roli będą traktowane jako administratorzy bota.

Tę rolę oczywiście też tworzymy w Second Life w tej grupie. Nie musi ona mieć przypisanych żadnych uprawnień.

Oprócz wspomnianych dużo wcześniej funkcji, definiujemy ciała funkcji dla ogólnego wysyłania danych do bota (SendToBot), przekształcanie tablicy do CSV oraz na odwrót, pobieranie konkretnych danych uzyskanych od bota.

Utwórz plik part.im.php

```
<?php
$uuid = $body['uuid'];
$message = trim($body['message']);
$type = $body['type'];
//////////
function default_no_command($uuid, $avatar){
    SendMessageToAvatar($uuid, $avatar.', I\'m sorry, but I don\'t understand.');
```

```
}
//////////
if(!empty($type)):
    switch($type):
        //////////
        case 'error_nosameregion':
            SendMessageToAvatar($uuid, $avatar.", Sorry, you don't in the same region
as me.");
            exit();
            break;
            //////////
        default:
            default_no_command($uuid, $avatar);
            exit();
            break;
    endswitch;
endif;

if( (preg_match('/hi/i', $message)) || (preg_match('/hello/i', $message))):
    $params = array(
        "command" => "displayname",
        "group" => BOT_GROUP,
        "password" => BOT_PASSWORD,
        "action" => 'get',
    );
    $ret = SendToBot($params);
    $avatarDisplayName = wasKeyValueGet("data", $ret);
```

```
SendMessageToAvatar($uuid, "Hello ".$avatar.", My name is ".$avatarDisplayName."
and I'm bot who is living in Second Life.");
    exit();
endif;
//////////
if(preg_match('/blim\ -add/i', $message)):
    if(!checkPerm($uuid, getConfig('GENERAL_groupFunctionAdminBot'))):
        if(!BlackListIMExist($uuid)):
            BlackListIMAdd($uuid);
        else:
            BlackListIMInc($uuid);
        endif;

        SendMessageToAvatar($uuid, $avatar.', you have not permissions to manage the
bot.');
```

```
        exit();
    endif;
    $who = trim(substr($message, strpos($message, "blim-add") + strlen("blim-add")));
    if(BlackListIMAdd($who)):
        SendMessageToAvatar($uuid, $avatar.', avatar has been added to blacklist.');
```

```
    else:
        SendMessageToAvatar($uuid, $avatar.', avatar has not added to blacklist.');
```

```
    endif;
    exit();
endif;
//////////
if(preg_match('/blim\ -del/i', $message)):
    if(!checkPerm($uuid, getConfig('GENERAL_groupFunctionAdminBot'))):
        if(!BlackListIMExist($uuid)):
            BlackListIMAdd($uuid);
        else:
            BlackListIMInc($uuid);
        endif;

        SendMessageToAvatar($uuid, $avatar.', you have not permissions to manage the
bot.');
```

```
        exit();
    endif;
    $who = trim(substr($message, strpos($message, "blim-del") + strlen("blim-del")));
    BlackListIMDel($who);
```

```

    SendMessageToAvatar($uuid, $avatar.', avatar has been delete from blacklist.');
```

exit();

```
endif;
//////////
if(preg_match('/\s.getbalance/i', $message)):
    if(!checkPerm($uuid, getConfig('GENERAL_groupFunctionAdminBot'))):
        if(!BlackListIMExist($uuid)):
            BlackListIMAdd($uuid);
        else:
            BlackListIMInc($uuid);
        endif;
        SendMessageToAvatar($uuid, $avatar.', you have not permissions to manage the
bot.');
```

exit();

```
endif;
$params = array(
    "command" => "getbalance",
    "group" => BOT_GROUP,
    "password" => BOT_PASSWORD,
);
$ret = SendToBot($params);
$money = wasKeyValueGet("data", $ret);
SendMessageToAvatar($uuid, $avatar.', I have L$ ' . $money . ' on my bank
account.');
```

exit();

```
endif;
//////////

default_no_command($uuid, $avatar);
```

Wyjaśnijmy:

Zmienna \$uuid przechowuje przesłany UUID awatara.

Zmienna \$message przechowuje przesłaną wiadomość.

Zmienna \$type przechowuje typ wiadomości.

Funkcja default_no_command() pozwala na wysłanie wiadomości do danego awatara, że nie rozumie polecenia. Będzie używana w kilku miejscach.

Dalej wykrywamy, czy nie przesłano pustej zmiennej \$type. Przechowuje ona polecenia nie typowe, np. zaimplementowany parę linijek dalej błąd związany z tym, że bot i rozmówca nie są w

tym samym regionie. Można tutaj także wyjść dalej i dodać wysłanie wiadomości, kiedy bot i rozmówca są za daleko od siebie. Jeżeli nie wykryto odpowiedniego typu wiadomości wysyła ogólną wiadomość, że nie rozumie polecenia.

Jeżeli nie wykryto nic w zmiennej \$type to przechodzimy w zasadzie do głównej części przetwarzania wiadomości.

Na przykład wzięliśmy wykrycie słów: 'hi' lub 'hello' w wiadomości przesłanej do bota. Jeżeli tak się stanie to pobiera nazwę wyświetlaną bota i wysyła zwrótną odpowiedź do rozmówcy.

Dalej mamy 3 przykłady, gdzie są sprawdzane uprawnienia.

- blim-add - dodaje awatara do listy zbanowanych
- blim-del - usuwa awatara z listy zbanowanych
- .getbalance - podaje ile pieniędzy ma nasz bot na koncie

Sprawdzenie odbywa się poprzez wywołanie funkcji checkPerm(), która na podstawie uuid awatara oraz wcześniej zdefiniowanej roli sprawdza, czy awatar został przypisany do danej roli.

Jeżeli awatar nie posiada właściwej roli w grupie, to jest dodawany do małego licznika zbanowanych, gdzie po trzech takich sprawdzeniach jest dodawany do stałej listy zbanowanych. Ta propozycja jest oczywiście zbyt regorystyczna, można ją pominąć i tylko wysłać wiadomość, że dany awatar nie ma uprawnień do realizacji danej komendy. Jeżeli autoryzacja awatara się udała, to jest wykonywana dalsza część kodu. Jeżeli natomiast w żadnej części nie znaleziono odpowiednich poleceń, to jest wysyłana standardowa odpowiedź, że bot nie rozumie rozmówcy.

Oczywiście jest to jakaś propozycja, sam ustalisz co dodasz, poprawisz, itd.

Sposób wywoływania:

```
php\serverbot.php 'action|serveIM|uuid|uuidAwatara|message|wiadomość|type|  
typWiadomości'
```

Do naszego pliku serverbot.php możemy dodawać inne rzeczy, które mają być przetwarzane przez bot.

5.4.3. Pliki wsadowe

Stwórzmy teraz kilka plików wsadowych w głównym folderze bota

start.bat

```
chdir /D %~dp0
```



```
@echo off

break>Cache\AgentCache.xml
break>Cache\CurrentGroupMembers.xml
break>Cache\CurrentGroups.xml
break>Cache\GroupCache.xml
break>Cache\InventoryCache.bin
break>Cache\MapFriend.xml
break>Cache\MuteCache.xml
break>Cache\RegionCache.xml
echo 0 > corrade_cron_inc.txt
echo 0 > corrade_attempt_inc.txt

del /Q State\Notifications.xml

sc.exe config Corrade start= auto
sc.exe start Corrade

exit 0
```

Uruchamia normalnie bota, zapisuje puste pliki cache, kasuje plik Notifications.xml ze starymi powiadomieniami, ustawia usługę Corrade w tryb uruchomieniowy i uruchamia usługę Corrade

startWithCleanCache.bat

```
chdir /D %~dp0
@echo off

php -r "include \"php\config.php\"; include \"php\functions.php\"; deleteAllCache();"
START /MIN /WAIT start.bat
```

Kasuje plik cache lub kasuje dane cache z bazy danych (w zależności od opcji) i uruchamia wcześniejszy plik start.bat

stop.bat

```
chdir /D %~dp0

sc.exe stop Corrade
sc.exe config Corrade start= disabled
```

Zatrzymuje usługę Corrade, sprawia, że usługa jest oznaczona jako wyłączona.

restart.bat

```
chdir /D %~dp0
@echo off

START /MIN /WAIT stop.bat
timeout /T 5 /NOBREAK
START /MIN /WAIT start.bat
```

Plik restartuje Corrade na podstawie wywołania dwóch poprzednich plików.

Jeżeli masz dodatkowe materiały to możesz skorzystać z plików bot_installer\corrade\bot_ai\php kopiując je do botu swojego bota.

5.4.4. Wielozadaniowość w PHP

Podczas pracy z własnym botem okazało się, że musiałem zaimplementować wielozadaniowość. Gdyby mój artbot przekroczył zdefiniowany wcześniej czas to miał wrócić na swoje dawne miejsce. Przed implementacją było to trudne, gdyż bot stałby w miejscu pracy jak słup soli. Dlatego zaimplementowałem wielozadaniowość w PHP, gdzie obecny skrypt uruchamia osobny skrypt, który sprawdza co jakiś czas, czy skrypt go wywołujący nie zakończył swojego cyklu życia. To jest bardzo proste, gdyż skrypt rodzic przekazuje kolejnemu skryptowi swój numer procesu (PID) i który jest wykorzystywany do sprawdzenia, czy ten proces z pierwotnego skryptu nadal istnieje.

Wywołanie osobnego zadania może wyglądać tak:

Windows:

```
pclose( popen('START /B php ' . __DIR__ . '/innyplik.php -p ' . getmypid() . ' inne
parametry' > NUL', 'r' ) );
```

Linux:

```
pclose( popen('nohup php ' . __DIR__ . '/innyplik.php -p ' . getmypid() . ' inne
parametry' < /dev/null &', 'r' ) );
```

Krótko:

W systemie Windows osobny plik PHP jest uruchamiany przy pomocy START w tle, gdzie do tego pliku przekazujemy aktualny PID skryptu rodzica wraz z innymi parametrami.

`popen()` i `pclose()` uruchamiają proces i zamykają uchwyt do niego, gdyż nie interesuje mnie jego dalsza obsługa.

W systemie Linux korzystamy z polecenia `nohup`, które wykonuje polecenie nawet po wylogowaniu się użytkownika z systemu.

W wykonywanym osobno pliku PHP musimy się zastanowić, w jaki sposób odebrać parametry.

Jest tutaj wiele projektów do użycia - ja używam `ulrichsg/getopt-php`, który należy dodać do zależności Composer.

Dodajemy po załączeniu `vendor/autoload.php`

```
use GetOpt\GetOpt;
use GetOpt\Option;
use GetOpt\Command;
use GetOpt\ArgumentException;
use GetOpt\ArgumentException\Missing;
```

Potem dajemy fragment kodu odpowiedzialnego za przetwarzanie argumentów:

```
//PROCESSING FROM COMMANDLINE
$getOpt = new GetOpt();
$getOpt->addOptions([
    Option::create('p', 'pid', \GetOpt\GetOpt::REQUIRED_ARGUMENT)
->setDefaultValue(-1),
]);

try {
    $getOpt->process();
} catch (Exception $exception) {

}

$processId = $getOpt->getOption('pid');
```

Jeszcze musimy przemyśleć w jaki sposób będziemy wywoływać polecenia w samym systemie.

Domyślnie możemy użyć polecenia PHP o nazwie: `shell_exec()` lub jakiegoś podobnego.

Ja jednak użyłem w swoim projekcie symfony/process, który dodajemy do zależności Composer.

Następnie dodajemy:

```
use Symfony\Component\Process\Exception\ProcessFailedException;
use Symfony\Component\Process\Process;
use Symfony\Component\Process\PhpProcess;
use Symfony\Component\Process\PhpExecutableFinder;
```

A następnie kod:

```
while(true):
    if(ISREADY? == "1"):
        //[1] WHAT DO YOU WANT TO DO?
        break;
    else:
        if($processId > -1):
            $os_family = strtolower(PHP_OS_FAMILY);
            switch($os_family):
                case 'windows':
                    $process = Process::fromShellCommandline('powershell -Command "Get-
WmiObject Win32_Process -Filter \"processid = ' . $processId . '\" | measure | %
{ $_.Count }"');
                    break;
                case 'linux':
                    $process = Process::fromShellCommandline('ps --no-headers -aux -q '
. $processId . ' | grep -v "grep" | wc -l');
                    break;
            endswitch;
            try {
                $process->Run();
                if(trim($process->getOutput()) == "0"):
                    //[1] WHAT DO YOU WANT TO DO?
                    break;
                endif;
            } catch (ProcessFailedException $exception) {

        }
        endif;
    endif;
    sleep(10);
endwhile;
```

Przyjrzymy się teraz temu kodowi. W pętli sprawdzamy co 10 sekund, czy skrypt rodzic PHP nadal działa. Na początku sprawdzamy czy skrypt rodzic ustawił ISREADY? po wykonaniu całego skryptu. Przez ISREADY? rozumiem tutaj ustawienie przez ten skrypt zmiennej w cache'u, która potem jest sprawdzana co parę sekund przez skrypt dziecko czy zawiera odpowiednią wartość. Jeżeli tak się stało to wykonuje dalszą część kodu - np. wywołanie określonej funkcji, która zostanie podana zamiast linijki: `//[1] WHAT DO YOU WANT TO DO?`. Tę opcję oczywiście można ewentualnie skasować. Dalej jest sprawdzane, czy przekazano PID rodzica - liczbę od 0 w górę. Jeżeli tak to, sprawdzany jest system, na którym jest uruchomiony skrypt. Jeżeli wykryty system to Windows, to wykonywany jest przez `Symfony\Process` polecenie powershell z argumentami:

```
$process = Process::fromShellCommandline('powershell -Command "Get-WmiObject Win32_Process -Filter \"processid = ' . $processId . '\" | measure | %{ $_.Count }"');
```

Jeżeli wykryty system to Linux, to `Symfony\Process` wykonuje polecenie:

```
$process = Process::fromShellCommandline('ps --no-headers -aux -q ' . $processId . ' | grep -v "grep" | wc -l');
```

Bez względu na system cel jest jeden - sprawdzenie czy proces o podanym PID nadal działa w systemie. Jeżeli nie, to wykonywana jest akcja objęta komentarzem `//[1] WHAT DO YOU WANT TO DO?`, który w zasadzie jest taki sam jak wcześniejszy `//[1] WHAT DO YOU WANT TO DO?`. I to w zasadzie wszystko. Dalej po pętli `while()` możemy dać kod, który będzie zrealizowany kiedy skrypt dostanie informację, że skrypt rodzic zakończył swoje działanie.

5.4.5. Second Life – tworzenie primu i skryptów

Po stronie Second Life tworzymy prim, a w nim notatkę o nazwie 'configuration', gdzie zapisujemy następującą zawartość:

```
BOTID=UUID_BOT  
GROUP=GROUP_NAME  
PASSWORD=PASSWORD_GROUP
```

W miejsce:

- `UUID_BOT` - wstawiamy UUID naszego bota,

- GROUP_NAME - nazwa grupy w której jest bot,
- PASSWORD_GROUP - hasło do grupy.

Następnie tworzymy nowy plik o dowolnej nazwie (najlepiej IM) z następującą zawartością (zależy czy kodujemy z wykorzystaniem WAS czy JSON).

Skrypt LSL z wykorzystaniem funkcji WAS wygląda tak (plik Listings\mybot\was_IM.txt):

```
float distanceBetweenAvatars = 10.0;
key CORRADE;
string PASSWORD;
string GROUP;
string URL = "";
integer Debug = 1;
string TAG;
string val;
list lmessage;
string sjump = "";
key kAvatar;
string body;

integer line;
list tuples = [];
string type;
string agent;
string message;

key queryid;

list lprepare = [];

setDebug(string msg)
{
    if(Debug == 1)
    {
        llOwnerSay "[" + TAG + " " + msg);
    }
}

sendPHP(list value){
    lprepare = [
        "group", wasURLEscape(GROUP),
```

```

        "password", wasURLEscape(PASSWORD),
        "command", "execute",
        "file", "php",
        "parameter", "php/servebot.php " + llDumpList2String(value, "|"),
        "_script", TAG
    ];

    llOwnerSay(wasKeyValueEncode(lprepare));
}

integer StringMatch(list lwhatsearch, string sinwhere){
    sinwhere = llToLower(sinwhere);
    sinwhere = strReplace(sinwhere, "!", "");
    sinwhere = strReplace(sinwhere, "?", "");

    list lsinwhere = llParseString2List(sinwhere,[" "],[]);

    integer imatch = llGetListLength(lwhatsearch);
    integer imatched = 0;

    integer i=0;
    for(i=0; i<imatch; i++){

        list lwords = llParseString2List(llList2String(lwhatsearch,i),["|"],[]);

        integer j=0;
        for(j=0; j<llGetListLength(lwords); j++){
            if(llListFindList(lsinwhere, [llList2String(lwords,j)]) >= 0){
                imatched++;
            }
        }
    }
    if(imatch == imatched){
        return 1;
    }else{
        return 0;
    }
}

list wasCSVToList(string csv)
{

```

```

list l = [];
list s = [];
string m = "";
do
{
    string a = llGetSubString(csv, 0, 0);
    csv = llDeleteSubString(csv, 0, 0);
    if(a == ",")
    {
        if(llList2String(s, -1) != "\\")
        {
            l += m;
            m = "";
            jump continue;
        }
        m += a;
        jump continue;
    }
    if(a == "\\\" && llGetSubString(csv, 0, 0) == a)
    {
        m += a;
        csv = llDeleteSubString(csv, 0, 0);
        jump continue;
    }
    if(a == "\\")
    {
        if(llList2String(s, -1) != a)
        {
            s += a;
            jump continue;
        }
        s = llDeleteSubList(s, -1, -1);
        jump continue;
    }
    m += a;
    @continue;
} while(csv != "");
return l + m;
}

string strReplace(string str, string search, string replace) {

```



```

    return llDumpList2String(llParseStringKeepNulls((str = "") + str, [search], []),
replace);
}

string wasKeyValueGet(string k, string data)
{
    if(llStringLength(data) == 0) return "";
    if(llStringLength(k) == 0) return "";
    list a = llParseStringKeepNulls(data, ["&", "="], []);
    integer i = llListFindList(llList2ListStrided(a, 0, -1, 2), [ k ]);
    if(i != -1){
        string ret = llList2String(wasCSVToList(wasURLUnescape(llList2String(a, 2*i+1))),
0);
        ret = strReplace(ret, "\\r", "");
        ret = strReplace(ret, "\\n", "");
        ret = llStringTrim(ret, STRING_TRIM);

        return ret;
    }
    return "";
}

string wasKeyValueEncode(list data)
{
    list k = llList2ListStrided(data, 0, -1, 2);
    list v = llList2ListStrided(llDeleteSubList(data, 0, 0), 0, -1, 2);
    data = [];
    do
    {
        data += llList2String(k, 0) + "=" + llList2String(v, 0);
        k = llDeleteSubList(k, 0, 0);
        v = llDeleteSubList(v, 0, 0);
    } while(llGetListLength(k) != 0);
    return llDumpList2String(data, "&");
}

string wasURLEscape(string i)
{
    string o = "";
    do
    {

```

```
string c = llGetSubString(i, 0, 0);
i = llDeleteSubString(i, 0, 0);
if(c == "") jump continue;
if(c == " ")
{
    o += "+";
    jump continue;
}
if(c == "\n")
{
    o += "%0D" + llEscapeURL(c);
    jump continue;
}
o += llEscapeURL(c);
@continue;
} while(i != "");
return o;
}

string wasURLUnescape(string i)
{
    return llUnescapeURL( llDumpList2String( llParseStringKeepNulls( llDumpList2String(
llParseStringKeepNulls( i, ["+"], [] ), " " ), ["%0D%0A"], [] ), "\n" ) );
}

default
{
    state_entry()
    {
        TAG = llGetScriptName();
        setDebug("Free memory: " + (string)llGetFreeMemory());
        state ReadConfigurationNotecard;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }
}
```

```
on_rez(integer start_param)
{
    llResetScript();
}

state ReadConfigurationNotecard
{
    state_entry()
    {
        if(llGetInventoryType("configuration") != INVENTORY_NOTECARD)
        {
            setDebug("Sorry, could not find a configuration inventory notecard.");
            return;
        }
        setDebug("Reading configuration file...");
        line = 0;
        llGetNotecardLine("configuration", line);
    }

    dataserver(key id, string data)
    {
        if(data == EOF)
        {
            if(llGetListLength(tuples) % 2 != 0)
            {
                setDebug("Error in configuration notecard.");
                return;
            }

            CORRADE = llList2Key(tuples, llListFindList(tuples, ["BOTID"])+1);

            if(CORRADE == NULL_KEY)
            {
                setDebug("Error in configuration notecard: BOT ID KEY");
                return;
            }

            GROUP = llList2String(tuples, llListFindList(tuples, ["GROUP"])+1);
```

```

        if(GROUP == "")
        {
            setDebug("Error in configuration notecard: GROUP");
            return;
        }

        PASSWORD = llList2String(tuples, llListFindList(tuples, ["PASSWORD"])+1);

        if(PASSWORD == "")
        {
            setDebug("Error in configuration notecard: PASSWORD");
            return;
        }

        state url;
    }
    if(data == "") jump continue;
    integer i = llSubStringIndex(data, "#");
    if(i != -1) data = llDeleteSubString(data, i, -1);
    list o = llParseStringKeepNulls(data, ["="], []);
    string k = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String(o,0), STRING_TRIM), ["\""], []), "\"");
    string v = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String( o, 1 ), STRING_TRIM), ["\""], [] ), "\"");
    if(k == "" || v == "") jump continue;
    tuples += k;
    tuples += v;
    @continue;
    llGetNotecardLine("configuration", ++line);
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{

```

```
        llResetScript();
    }
}

state url
{
    state_entry()
    {
        if ((GROUP == "BOTID") || (GROUP == "GROUP") || (GROUP == "PASSWORD")){
            llResetScript();
            return;
        }

        if ((PASSWORD == "BOTID") || (PASSWORD == "GROUP") || (PASSWORD == "PASSWORD"))
        {
            llResetScript();
            return;
        }

        setDebug("Requesting URL...");
        llSetTimerEvent(0.5);
    }

    timer(){
        llSetTimerEvent(60);
        llRequestURL();
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");
        if(method != URL_REQUEST_GRANTED) return;
        URL = body;
        setDebug("Got URL...");
        state NotifyInstall;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
```

```
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}

state NotifyInstall {
    state_entry() {
        llOwnerSay("Binding to the IM notification...");
        llSetTimerEvent(0.5);
    }

    timer(){
        llSetTimerEvent(60);
        lprepare = [
            "group", wasURLEscape(GROUP),
            "password", wasURLEscape(PASSWORD),
            "callback", wasURLEscape(URL),
            "URL", wasURLEscape(URL),
            "command", "notify",
            "action", "set",
            "type", "message",
            "_script", TAG
        ];

        llOwnerSay(wasKeyValueEncode(lprepare));
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");

        if(wasKeyValueGet("success", body) != "True") {
            llOwnerSay("Failed to bind to the IM notification...");
            state preNotifyInstall;
        }

        llOwnerSay("IM notification installed...");
    }
}
```

```
    state sense;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

state preNotifyInstall
{
    state_entry()
    {
        state NotifyInstall;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state sense {
    state_entry() {
        llOwnerSay("Listen IM...");
    }
}
```

```
}

http_request(key id, string method, string body)
{
    llHTTPRequest(id, 200, "OK");

    type = wasKeyValueGet("type", body);
    agent = wasKeyValueGet("agent", body);
    message = wasKeyValueGet("message", body);

    if(agent == NULL_KEY){
        return;
    }

    if(llStringLength(message) > 255){
        return;
    }

    lmessage = [] + llParseString2List(message,[" "],[]);

    if(llGetListLength(lmessage) == 0){
        return;
    }

    state CheckAvatar;
    return;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}
```



```
state CheckAvatar {
    state_entry() {
        queryid = llRequestAgentData(agent, DATA_ONLINE);
    }

    dataserver(key DataserverQueryId, string data)
    {
        if(DataserverQueryId == queryid)
        {
            if((integer)data == 1){
                state stateIM;
                return;
            }
        }
        state sense;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state stateIM {
    state_entry() {
        list lbot = llGetObjectDetails(CORRADE, [OBJECT_POS]);
        list lagent = llGetObjectDetails(agent, [OBJECT_POS]);

        if(StringMatch(["where", "are", "you"], message)){
            sendPHP(["action", "serveIM", "type", "whereareyou", "uuid", agent]);
            state sense;
            return;
        }
    }
}
```

```

        if(llGetListLength(lagent) == 0){
            sendPHP(["action", "serveIM", "type", "error_nosameregion", "uuid",
agent]);
            state sense;
            return;
        }

//      vector vAgentPos = llList2Vector(lagent, 0);
//      float distance = llVecDist(vBotPos, vAgentPos);

//      if(distance > distanceBetweenAvatars){
//          sendPHP(["action", "serveIM", "type", "error_distance", "uuid", agent]);
//          state sense;
//          return;
//      }

        sendPHP(["action", "serveIM", "uuid", agent, "message", message]);

        state sense;
        return;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state preSense {
    state_entry() {
        state sense;
    }
}

```

```
changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}
```

Plik skryptu wykorzystujące funkcje JSON (plik Listings\mybot\json_IM.txt):

```
float distanceBetweenAvatars = 10.0;
key CORRADE;
string PASSWORD;
string GROUP;
string URL = "";
integer Debug = 1;
string TAG;
string val;
list lmessage;
string sjump = "";
key kAvatar;
string body;

integer line;
list tuples = [];

string type;
string agent;
string message;

key queryid;

list lprepare = [];

setDebug(string msg)
```

```
{
    if(Debug == 1)
    {
        llOwnerSay("[ "+ TAG +" ] " + msg);
    }
}

integer StringMatch(list lwhatsearch, string sinwhere){
    sinwhere = llToLower(sinwhere);
    sinwhere = strReplace(sinwhere, "!", "");
    sinwhere = strReplace(sinwhere, "?", "");

    list lsinwhere = llParseString2List(sinwhere,[" "],[]);

    integer imatch = llGetListLength(lwhatsearch);
    integer imatched = 0;

    integer i=0;
    for(i=0; i<imatch; i++){

        list lwords = llParseString2List(llList2String(lwhatsearch,i),["|"],[]);

        integer j=0;
        for(j=0; j<llGetListLength(lwords); j++){
            if(llListFindList(lsinwhere, [llList2String(lwords,j)]) >= 0){
                imatched++;
            }
        }
    }
    if(imatch == imatched){
        return 1;
    }else{
        return 0;
    }
}

sendPHP(list value){
    lprepare = [
        "group", GROUP,
        "password", PASSWORD,
        "command", "execute",
    ]
}
```

```
        "file", "php",
        "parameter", "php/servebot.php " + llDumpList2String(value, "|"),
        "_script", TAG
    ];

    llOwnerSay(wasKeyValueToJSON(wasKeyValueEncode(lpprepare), JSON_OBJECT));
}

list wasCSVToList(string csv)
{
    list l = [];
    list s = [];
    string m = "";
    do
    {
        string a = llGetSubString(csv, 0, 0);
        csv = llDeleteSubString(csv, 0, 0);
        if(a == ",")
        {
            if(llList2String(s, -1) != "\"")
            {
                l += m;
                m = "";
                jump continue;
            }
            m += a;
            jump continue;
        }
        if(a == "\"" && llGetSubString(csv, 0, 0) == a)
        {
            m += a;
            csv = llDeleteSubString(csv, 0, 0);
            jump continue;
        }
        if(a == "\\")
        {
            if(llList2String(s, -1) != a)
            {
                s += a;
                jump continue;
            }
        }
    }
}
```

```

        s = llDeleteSubList(s, -1, -1);
        jump continue;
    }
    m += a;
    @continue;
} while(csv != "");
return l + m;
}

string strReplace(string str, string search, string replace) {
    return llDumpList2String(llParseStringKeepNulls((str = "") + str, [search], []),
replace);
}

string wasKeyValueGet(string k, string data)
{
    if(llStringLength(data) == 0) return "";
    if(llStringLength(k) == 0) return "";
    list a = llParseStringKeepNulls(data, ["&", "="], []);
    integer i = llListFindList(llList2ListStrided(a, 0, -1, 2), [ k ]);
    if(i != -1){
        string ret = llList2String(wasCSVToList(wasURLUnescape(llList2String(a, 2*i+1))),
0);
        ret = strReplace(ret, "\\r", "");
        ret = strReplace(ret, "\\n", "");
        ret = llStringTrim(ret, STRING_TRIM);

        return ret;
    }
    return "";
}

string wasKeyValueToJSON(string kvp, string T) {
    list data = llParseString2List(kvp, ["&", "="], []);
    list temp = [];
    do {
        string k = llList2String(data, 0);
        string v = llList2String(data, 1);
        string o = "";
        o += "\"" + k + "\"";
        if(T == JSON_ARRAY) {

```

```

        o += ",";
        jump type;
    }
    o += ":";
    @type;
    if((float)v != 0 || v == "0") {
        o += v;
        jump dump;
    }
    o += "\"" + v + "\"";
    @dump;
    temp += o;
    data = llDeleteSubList(data, 0, 1);
} while(llGetListLength(data) != 0);
if(T == JSON_ARRAY)
    return "[" + llDumpList2String(temp, ",") + "]";
if(T == JSON_OBJECT)
    return "{" + llDumpList2String(temp, ",") + "}";
return JSON_INVALID;
}

string wasKeyValueEncode(list data)
{
    list k = llList2ListStrided(data, 0, -1, 2);
    list v = llList2ListStrided(llDeleteSubList(data, 0, 0), 0, -1, 2);
    data = [];
    do
    {
        data += llList2String(k, 0) + "=" + llList2String(v, 0);
        k = llDeleteSubList(k, 0, 0);
        v = llDeleteSubList(v, 0, 0);
    } while(llGetListLength(k) != 0);
    return llDumpList2String(data, "&");
}

string wasURLUnescape(string i)
{
    return llUnescapeURL( llDumpList2String( llParseStringKeepNulls( llDumpList2String(
llParseStringKeepNulls( i, ["+"], [] ), " " ), ["%0D%0A"], [] ), "\n" ) );
}

```

```
string wasJSONToKeyValueData(string JSON) {
    list output = [];
    list symbols = [];
    string data = "";
    integer level = 1;
    // parse JSON
    do {
        string c = llGetSubString(JSON, 0, 0);
        // we are at an object start
        if(c == "{") {
            if(level > 1) {
                data += c;
            }
            // increment the level
            ++level;
            // add the symbol
            symbols += c;
            // and continue
            jump continue;
        }
        // we are at an object end
        if(c == "}") {
            // decrement the level
            --level;
            if(level > 1) {
                data += c;
                output += data;
                data = "";
            }
            // pop symbols
            string s = llList2String(symbols, -1);
            symbols = llDeleteSubList(symbols, -1, -1);
            // got an object start
            if(s == "{") {
                // so continue
                jump continue;
            }
            // error
            return JSON_INVALID;
        }
    }
    // we are at data end or data start
```



```
    if(c == "\\") {
        if(level > 2) {
            data += "\\";
            jump continue;
        }
        // pop symbols
        string s = llList2String(symbols, -1);
        symbols = llDeleteSubList(symbols, -1, -1);
        // if we are at data end
        if(s == "\\") {
            // add the data to the output
            output += data;
            // flush data
            data = "";
            // and continue
            jump continue;
        }
        // we are not at the end of data
        // add the symbol back
        symbols += s;
        // add the current character
        symbols += c;
        // and continue
        jump continue;
    }
    if(level > 2) {
        data += c;
        jump continue;
    }
    // pop symbols
    string s = llList2String(symbols, -1);
    symbols = llDeleteSubList(symbols, -1, -1);
    if(s == "\\") {
        data += c;
        symbols += s;
        jump continue;
    }
    symbols += s;
@continue;
    JSON = llDeleteSubString(JSON, 0, 0);
} while(llStringLength(JSON) != 0);
```

```

// now encode to key-value data
list k = llList2ListStrided(output, 0, -1, 2);
list v = llList2ListStrided(llDeleteSubList(output, 0, 0), 0, -1, 2);
output = [];
do {
    output += llList2String(k, 0) + "=" + llList2String(v, 0);
    k = llDeleteSubList(k, 0, 0);
    v = llDeleteSubList(v, 0, 0);
} while(llGetListLength(k) != 0);
return llDumpList2String(output, "&");
}

default
{
    state_entry()
    {
        TAG = llGetScriptName();
        setDebug("Free memory: " + (string)llGetFreeMemory());
        state ReadConfigurationNotecard;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state ReadConfigurationNotecard
{
    state_entry()
    {
        if(llGetInventoryType("configuration") != INVENTORY_NOTECARD)
        {

```

```
        setDebug("Sorry, could not find a configuration inventory notecard.");
        return;
    }
    setDebug("Reading configuration file...");
    line = 0;
    llGetNotecardLine("configuration", line);
}

dataserver(key id, string data)
{
    if(data == EOF)
    {
        if(llGetListLength(tuples) % 2 != 0)
        {
            setDebug("Error in configuration notecard.");
            return;
        }

        CORRADE = llList2Key(tuples, llListFindList(tuples, ["BOTID"])+1);

        if(CORRADE == NULL_KEY)
        {
            setDebug("Error in configuration notecard: BOT ID KEY");
            return;
        }

        GROUP = llList2String(tuples, llListFindList(tuples, ["GROUP"])+1);

        if(GROUP == "")
        {
            setDebug("Error in configuration notecard: GROUP");
            return;
        }

        PASSWORD = llList2String(tuples, llListFindList(tuples, ["PASSWORD"])+1);

        if(PASSWORD == "")
        {
            setDebug("Error in configuration notecard: PASSWORD");
            return;
        }
    }
}
```

```

        state url;
    }
    if(data == "") jump continue;
    integer i = llSubStringIndex(data, "#");
    if(i != -1) data = llDeleteSubString(data, i, -1);
    list o = llParseStringKeepNulls(data, ["="], []);
    string k = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String(o,0), STRING_TRIM), ["\""], []), "\"");
    string v = llDumpList2String( llParseStringKeepNulls( llStringTrim(
llList2String( o, 1 ), STRING_TRIM), ["\""], [] ), "\"");
    if(k == "" || v == "") jump continue;
    tuples += k;
    tuples += v;
    @continue;
    llGetNotecardLine("configuration", ++line);
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}

state url
{
    state_entry()
    {
        if ((GROUP == "BOTID") || (GROUP == "GROUP") || (GROUP == "PASSWORD")){
            llResetScript();
            return;
        }
    }
}

```

```
        if ((PASSWORD == "BOTID") || (PASSWORD == "GROUP") || (PASSWORD == "PASSWORD"))
    {
        llResetScript();
        return;
    }

    setDebug("Requesting URL...");
    llSetTimerEvent(0.5);
}

timer(){
    llSetTimerEvent(60);
    llRequestURL();
}

http_request(key id, string method, string body)
{
    llHTTPResponse(id, 200, "OK");
    if(method != URL_REQUEST_GRANTED) return;
    URL = body;
    setDebug("Got URL...");
    state NotifyInstall;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

state NotifyInstall {
    state_entry() {
        llOwnerSay("Binding to the IM notification...");
```

```
    llSetTimerEvent(0.5);
}

timer(){
    llSetTimerEvent(60);
    lprepare = [
        "group", GROUP,
        "password", PASSWORD,
        "callback", URL,
        "URL", URL,
        "command", "notify",
        "action", "set",
        "type", "message",
        "_script", TAG
    ];

    llOwnerSay(wasKeyValueToJSON(wasKeyValueEncode(lprepare), JSON_OBJECT));
}

http_request(key id, string method, string body)
{
    llHTTPResponse(id, 200, "OK");

    body = wasJSONToKeyValueData(body);

    if(wasKeyValueGet("success", body) != "True") {
        llOwnerSay("Failed to bind to the IM notification...");
        state preNotifyInstall;
    }
    llOwnerSay("IM notification installed...");
    state sense;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}
```

```
    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state preNotifyInstall
{
    state_entry()
    {
        state NotifyInstall;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

state sense {
    state_entry() {
        llOwnerSay("Listen IM...");
    }

    http_request(key id, string method, string body)
    {
        llHTTPResponse(id, 200, "OK");

        body = wasJSONToKeyValueData(body);

        type = wasKeyValueGet("type", body);
        agent = wasKeyValueGet("agent", body);
        message = wasKeyValueGet("message", body);
    }
}
```

```
    if(agent == NULL_KEY){
        return;
    }

    if(llStringLength(message) > 255){
        return;
    }

    lmessage = [] + llParseString2List(message,[" "],[]);

    if(llGetListLength(lmessage) == 0){
        return;
    }

    state CheckAvatar;
    return;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

state CheckAvatar {
    state_entry() {
        queryid = llRequestAgentData(agent, DATA_ONLINE);
    }

    dataserver(key DataserverQueryId, string data)
    {
        if(DataserverQueryId == queryid)
```



```
{
    if((integer)data == 1){
        state stateIM;
        return;
    }
}
state sense;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

state stateIM {
    state_entry() {

        list lbot = llGetObjectDetails(CORRADE, [OBJECT_POS]);
        list lagent = llGetObjectDetails(agent, [OBJECT_POS]);

        if(StringMatch(["where", "are", "you"], message)){
            sendPHP(["action", "serveIM", "type", "whereareyou", "uuid", agent]);
            state sense;
            return;
        }

        if(llGetListLength(lagent) == 0){
            sendPHP(["action", "serveIM", "type", "error_nosameregion", "uuid",
agent]);
            state sense;
            return;
        }
    }
}
```

```
//      vector vBotPos = llList2Vector(lbot, 0);
//      vector vAgentPos = llList2Vector(lagent, 0);
//      float distance = llVecDist(vBotPos, vAgentPos);
//      if(distance > distanceBetweenAvatars){
//          sendPHP(["action", "serveIM", "type", "error_distance", "uuid", agent]);
//          state sense;
//          return;
//      }

    sendPHP(["action", "serveIM", "uuid", agent, "message", message]);

    state sense;
    return;
}

changed(integer change)
{
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
    {
        llResetScript();
    }
}

on_rez(integer start_param)
{
    llResetScript();
}
}

state preSense {
    state_entry() {
        state sense;
    }

    changed(integer change)
    {
        if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START))
        {
            llResetScript();
        }
    }
}
```

```

    }

    on_rez(integer start_param)
    {
        llResetScript();
    }
}

```

Teraz wyjaśnijmy poszczególne części:

- `sendPHP()` - wysyła zestaw poleceń do pliku `php/serverbot.php` przez bota, argumenty są oddzielone od siebie za pomocą pionowej kreski.
- `StringMatch()` - szuka jednego ciągu tekstowego w drugim ciągu tekstowym.
- `strReplace()` - zamienia jeden ciąg znakowy na drugi ciąg znakowy w zadanym tekście.
- `TAG = llGetScriptName();` - przypisuje nazwę skryptu do zmiennej, w celach analitycznych.
- `llSetTimerEvent()` i `timer()` - pierwszy ustawia czas, po którym wyzwalacz czasowy się uruchomi (stosuję tutaj krótki okres), natomiast drugi to blok czasowy, który wykonuje się co określoną liczbę sekund - domyślnie jest to co 60 sekund, ponieważ Corrade średnio tyle przyjmuje maksymalnego czasu na wykonanie polecenia u siebie. Jeżeli dany kod zostanie wykonany pomyślnie to możemy wyjść z tego bloku do kolejnego stanu.
- `llOwnerSay()` - wysyłamy wszystko bezpośrednio do bota, gdyż `prim` jest własnością bota i edytujemy `prim` i jego skrypty jako bot; jeżeli `prim` jest umieszczony na działce to zaleca się stosowanie `llInstantMessage()`, a przed wysłaniem komendy najlepiej umieścić stan sprawdzający czy bot jest dostępny (przy użyciu `timer`a) w Second Life (sprawdź stan `CheckAvatar`) i dopóki go nie ma, nie wysyłamy żadnych poleceń.

```

state CheckBotAvatar {
    state_entry() {
        llSetTimerEvent(0.5);
    }

    timer(){
        llSetTimerEvent(60);
        queryid = llRequestAgentData(CORRADE, DATA_ONLINE);
    }

    dataserver(key DataserverQueryId, string data) {
        if(DataserverQueryId == queryid) {

```

```

        if((integer)data == 1){
            state NastepnyStan;
        }
    }
}

changed(integer change) {
    if(change & (CHANGED_OWNER | CHANGED_REGION | CHANGED_REGION_START)) {
        llResetScript();
    }
}

on_rez(integer start_param) {
    llResetScript();
}
}

```

Dalej oczywiście jest czytanie notatki 'configuration', gdzie znajdują się informacje o: UUID naszego bota, nazwy grupy, hasła do grupy. Po wczytaniu konfiguracji następuje przejście do stanu 'url', gdzie następuje próba pobrania URL z Second Life pod który będą wysyłane powiadomienia. Kolejnym bardzo ważnym krokiem jest instalacja samego powiadomienia typu 'message', które pozwala na odbiór wiadomości prywatnych wysyłanych do bota. Całość jest wysyłana do właściciela primu, czyli bota przez polecenie llOwnerSay. Jeżeli nie uda się ustawić powiadomienia (zwracany jest błąd) to ponownie jest wysyłane żądanie do bota o zainstalowanie powiadomienia. Gdy to wszystko się uda to następuje nasłuch - jeżeli bot otrzyma jakąkolwiek wiadomość prywatną to zostanie ona przekazana do skryptu przez powiadomienie, które zawiera URL zwrotne. Jeżeli już tak się stanie, to w moim przykładowym skrypcie są sprawdzane na początku następujące warunki:

- czy agent nie jest wartością pustą, tzn. nie przyjmuje wartości samych zer,
- czy długość wiadomości nie jest większa niż 255 znaków,
- czy wiadomość przekształcona na listę nie ma zerowej liczby elementów.

Jeżeli na każde pytanie będzie fałsz, to następujące przejście do sprawdzenia czy awatar-nadawca nadal jest dostępny w Second Life, ponieważ jeżeli nie ma nadawcy w wirtualnym świecie, to zbyt często staje się wysłanie odpowiedzi do niego kiedy jest offline.

Jeżeli nasz rozmówca jest dostępny to zaczyna się główna część kodu polegająca na:

- pobraniu współrzędnych bota,

- pobraniu współrzędnych awatara-nadawcy,
- sprawdzeniu, czy w wiadomości są słowa kluczowe: where, are i you, jeżeli tak to wysła polecenie do pliku PHP obsługi bota przez bota jako pośrednika z akcją: serveIM, typu: whereareyou i uuid jako UUID awatara nadawcy,
- sprawdzeniu, czy awatar nadawca jest w tym samym regionie co bot (element listy niezerowy),
- sprawdzenie czy odległość pomiędzy botem a awatarem nie jest większa niż zadana liczba (opcja zakomentowana),
- wysłaniu wiadomości awatara do obsługi PHP wraz z UUID awatara-nadawcy.

Przykładowy kod dzieli się na dwie części przy odbiorze wiadomości.

Pierwsza część sprawdza wiadomość globalnie, np. kiedy nadawca prosi o lokalizację bota. W tej części nie dochodzi do sprawdzenia większości warunków wymienionych wyżej.

Druga część to sprawdzenie określonych warunków i w przypadku gdy wszystko się zgadza, wysłanie do dalszego przetwarzania po stronie PHP.

Jeżeli uruchomimy skrypt, to po paru minutach będziemy mieli możliwość rozmawiania z naszym botem przez prywatne wiadomości.

6. Logiczna siatka do ustawiania obiektów (PosGrid)

Ręczne obliczanie współrzędnych dla obiektów może być pracochłonne, a przy rozbudowanym skrypcie może później pojawić się problem z jego analizą kodu.

W Second Life co prawda można ustawiać rzędy primów z unikalnymi nazwami, jednakże ten pomysł szybko upada, ponieważ jeszcze szybciej skończy się limit primów do wykorzystania na działce lub w regionie.

Dużo lepszym wymyślonym przeze mnie sposobem jest podział logiczny danego obszaru (np. obiektu) na linie pionowe i poziome (nazywany dalej przeze mnie siatką), a następnie określenie pozycji danego punktu jako przecięcie się tych dwóch linii. Im więcej linii pionowych i poziomych, tym określenie pozycji wybranego punktu jest dokładniejsze.

Precyzyjne wyjaśnienie oparte na przykładach znajduje się poniżej.

Dane potrzebne do obliczenia współrzędnej:

- O - współrzędne X,Y punktu, który jest środkiem obiektu w Second Life - zazwyczaj są to współrzędne tego obiektu,
 - S - rozmiar obiektu,
 - R - współczynnik (X,Y) obrotu w zależności od kierunku zgodnego z różą kompasową,
 - Północ (N) – 0,1,
 - Wschód (E) – 1,0,
 - Południe (S) – 0,-1,
 - Zachód (W) – -1,0.
 - L - liczba linii pionowych i poziomych (X,Y), które dzielą obszar na identyczne kawałki,
 - E' - współrzędne (X,Y) punktu końcowego, na którym obiekt powinien być umieszczony.
- Współrzędne są liczone od 0.

Założenia:

- X jest liczbą zmiennoprzecinkową, z 5 miejscami po przecinku, wartości: 0...255,
- Y jest liczbą zmiennoprzecinkową, z 5 miejscami po przecinku, wartości: 0...255,
- Z jest liczbą zmiennoprzecinkową, z 5 miejscami po przecinku, wartości: 0...4096,
- O jest liczbą zmiennoprzecinkową, z 5 miejscami po przecinku,
- S jest liczbą zmiennoprzecinkową, z 5 miejscami po przecinku,
- R jest liczbą całkowitą, która przyjmuje wartości: -1, 0, 1,
- L jest liczbą całkowitą, która przyjmuje wartości od 1 do nieskończoności,
- Jeżeli L jest podzielna przez 2, wtedy zwiększ L o 1,
- E' jest liczbą całkowitą, która przyjmuje wartości od 0 do nieskończoności,
- E' nie może być większa niż liczba linii L,
- Jeżeli E' jest mniejsze od 0, wtedy to $E' = 0$,
- Jeżeli E' jest większa niż liczba linii L, wtedy to $E' = \text{liczba linii} - 1$,
- Jeżeli $L = 1$, wtedy $E' = 0$,
- W przypadku błędnych lub niekompletnych danych, $E' = 0$,

Aby uzyskać uniwersalny wzór do obliczenia współrzędnych końcowego punktu E, rozpatrzmy 4 sytuacje zgodnych z różą kompasową, biorąc tylko główne kierunki: północ, południe, wschód i zachód.

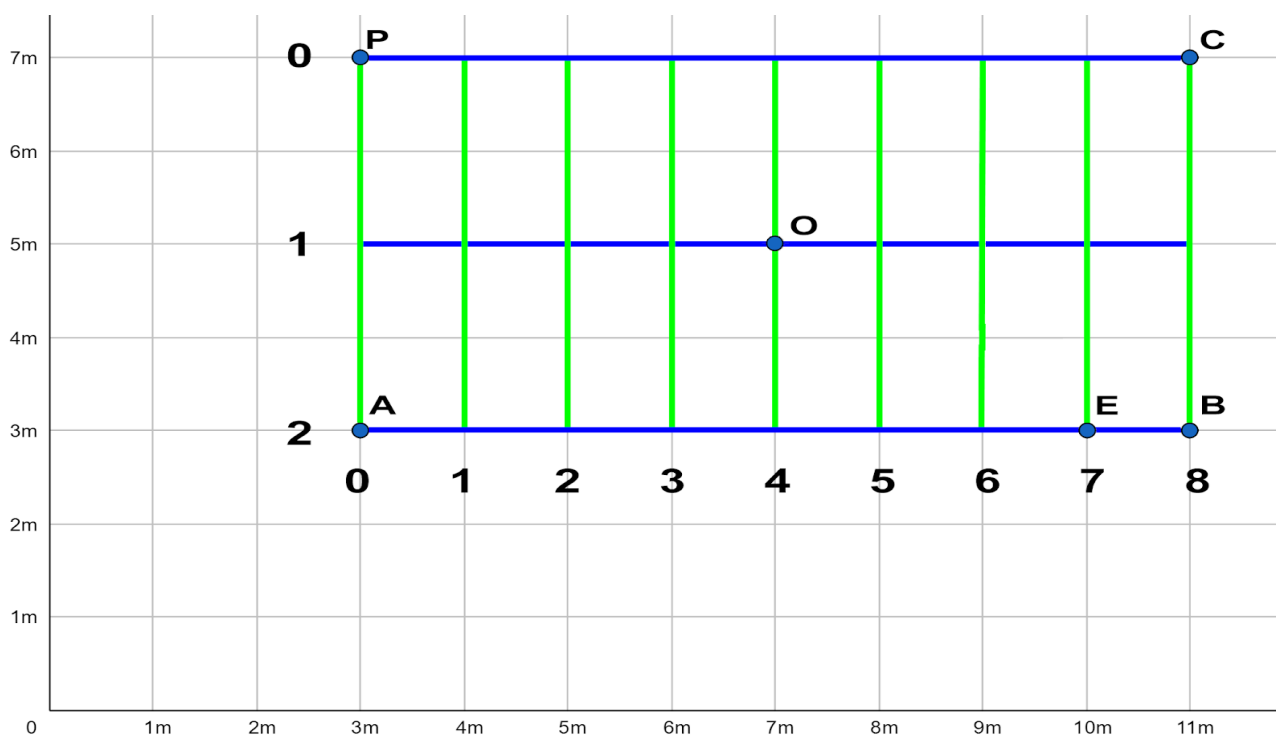
Dane wspólne dla każdego poniższego obrazka:

- $O = (7,5)$,
- $S = (8,4)$,
- R - zależy od obrotu obiektu, który dzielimy wirtualnymi liniami, jest podawany przed obrazkiem,
- $L = (9,3)$,
- $E' = (7,2)$.

Do obliczenia pozostaje nam punkt P, który oznacza lewy górny róg obiektu, który byłby także punktem wejściowym do obliczenia punktu końcowego E oraz szerokość i długość pojedynczego kawałka.

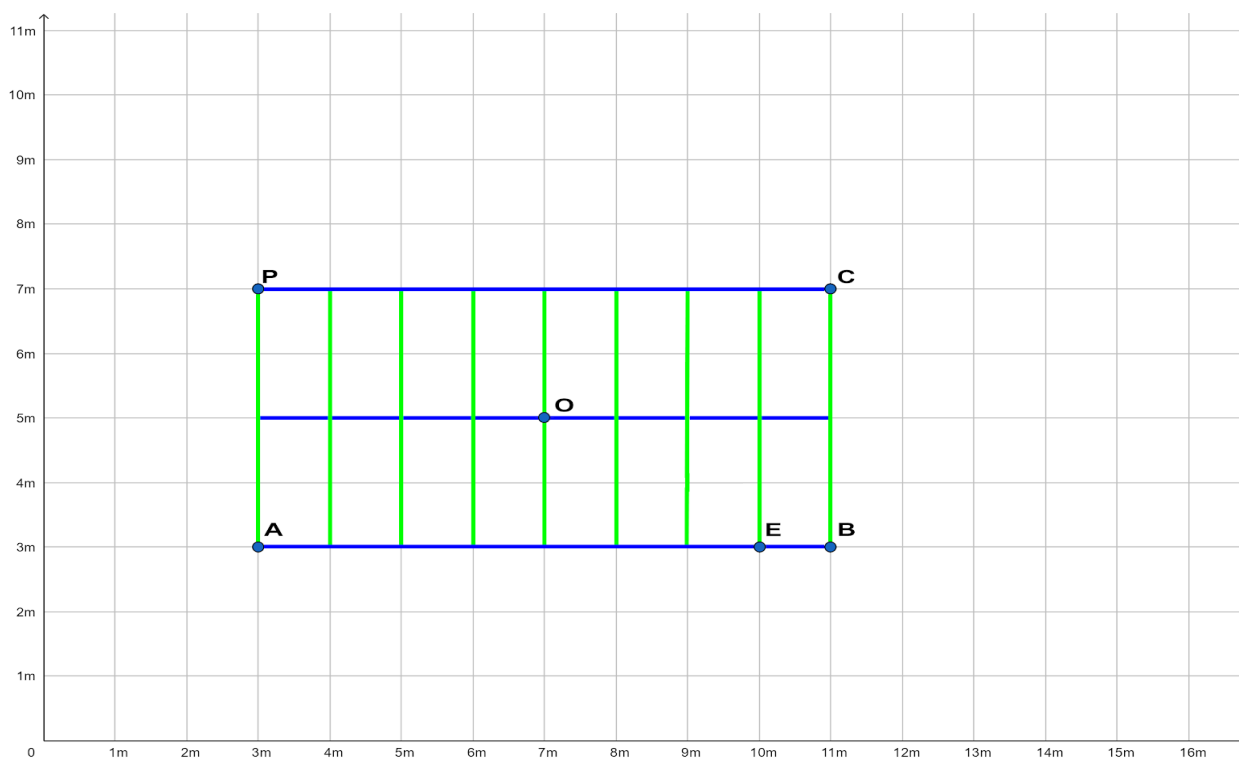
Sposób liczenia współrzędnych E' pokazuje poniższy rysunek.

Indeks jest liczony od 0, np. dla punktu E = (7,3) E' wynosi (7,2), dla punktu O=(4,5) E' wynosi (4,1), itd.



Rysunek 6.1 - Sposób liczenia punktu E'

Północ (N), R = (0,1), obrót obiektu o 0°



Rysunek 6.2 - Sposób liczenia punktu E dla kierunku północnego

Na początku obliczamy w tej części szerokość i długość pojedynczego kawałka. Do obliczenia będzie potrzebny rozmiar obiektu S oraz liczba linii pionowych i poziomych L.

Jeżeli podzielimy obiekt pionowo (zielona linia) od 3 do 11 m, co 1 m, otrzymamy 16 kawałków (8 kawałków \times 2 rzędy).

Jeżeli podzielimy obiekt poziomo (niebieska linia) od 3 do 7 m, co 2 m, otrzymamy 16 kawałków (2 kawałki \times 8 rzędów).

Można dojść do wniosku, że liczba kawałków jest równa gdy liczba linii jest mniejsza o 1.

$$S' = \frac{S}{L-1}$$

Powyższy wzór jest uniwersalny i określa rozmiar pojedynczego kawałka.

Sprawdzenie:

$$S'_x = \frac{S_x}{L_x-1} = \frac{8}{9-1} = \frac{8}{8} = 1$$

$$S'_y = \frac{S_y}{L_y-1} = \frac{4}{3-1} = \frac{4}{2} = 2$$

Obliczmy teraz współrzędne punktu P, czyli punktu od którego będziemy liczyć współrzędne punktu E.

$$P_x = O_x - \frac{S_x}{2}$$

$$P_y = O_y + \frac{S_y}{2}$$

Sprawdzenie:

$$P_x = O_x - \frac{S_x}{2} = 7 - \frac{8}{2} = 7 - 4 = 3$$

$$P_y = O_y + \frac{S_y}{2} = 5 + \frac{4}{2} = 5 + 2 = 7$$

Obliczmy teraz współrzędne interesującego nas punktu E. Do obliczenia E będziemy potrzebować P, obliczone wcześniej długości i szerokości pojedynczego kawałka S' oraz E.

$$E_x = P_x + S'_x \cdot E'_x$$

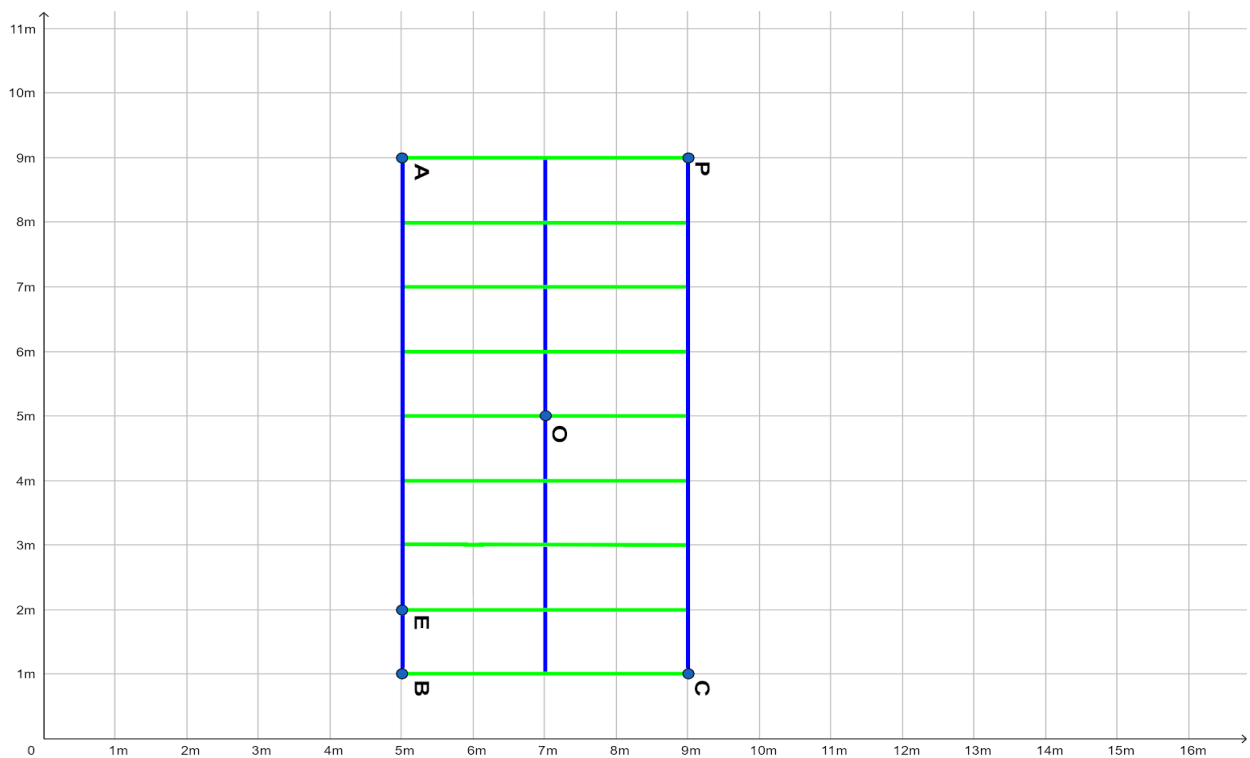
$$E_y = P_y - S'_y \cdot E'_y$$

Sprawdzenie:

$$E_x = P_x + S'_x \cdot E'_x = 3 + 1 \cdot 7 = 3 + 7 = 10$$

$$E_y = P_y - S'_y \cdot E'_y = 7 - 2 \cdot 2 = 7 - 4 = 3$$

Wschód (E), R = (1,0), obrót obiektu o 90°



Rysunek 6.3 - Sposób liczenia punktu E dla kierunku wschodniego

Obliczmy teraz współrzędne punktu P dla obrotu na wschód.

$$P_x = O_x + \frac{S_y}{2}$$

$$P_y = O_y + \frac{S_x}{2}$$

Sprawdzenie:

$$P_x = O_x + \frac{S_y}{2} = 7 + \frac{4}{2} = 7 + 2 = 9$$

$$P_y = O_y + \frac{S_x}{2} = 5 + \frac{8}{2} = 5 + 4 = 9$$

Obliczmy teraz ponownie punkt E dla kierunku wschodniego.

$$E_x = P_x - S_y \cdot E_y$$

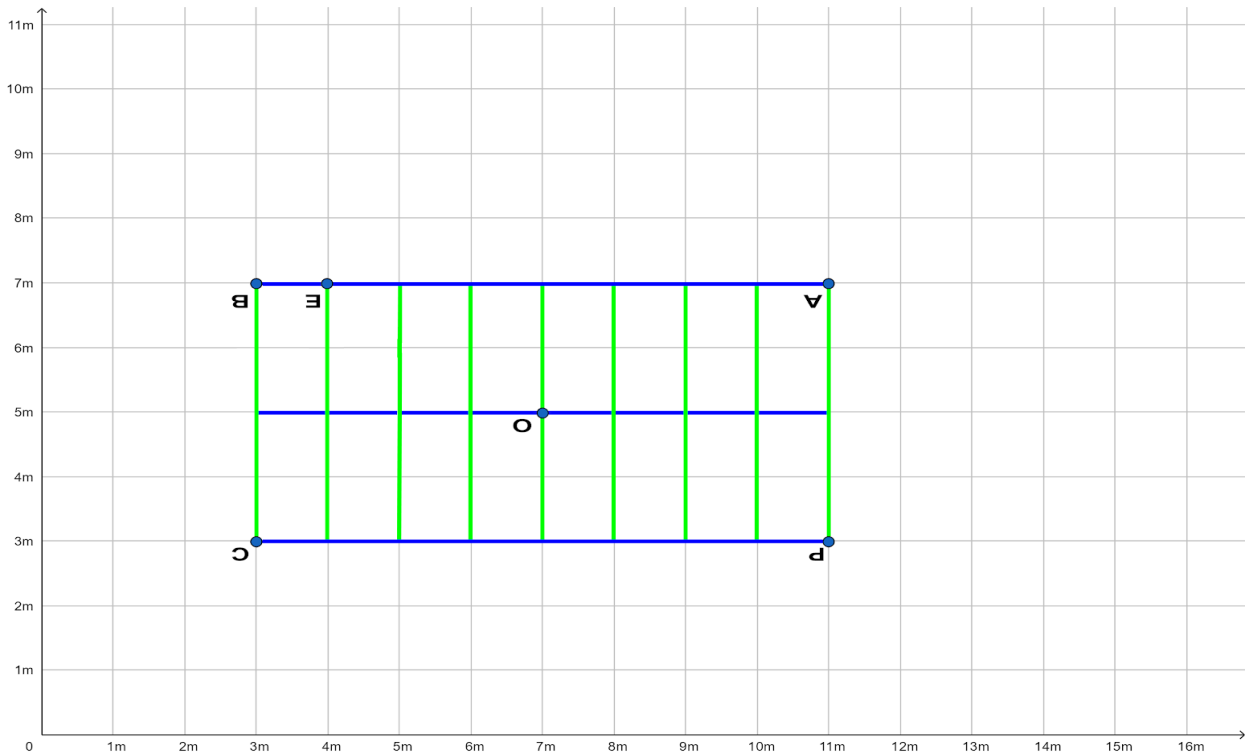
$$E_y = P_y - S_x \cdot E_x$$

Sprawdzenie:

$$E_x = P_x - S_y \cdot E_y = 9 - 2 \cdot 2 = 9 - 4 = 5$$

$$E_y = P_y - S_x \cdot E_x = 9 - 1 \cdot 7 = 9 - 7 = 2$$

Południe (S), $R = (0, -1)$, obrót obiektu o 180°



Rysunek 6.4 - Sposób liczenia punktu E dla kierunku południowego

Obliczmy teraz współrzędne punktu P dla obrotu południowego.

$$P_x = O_x + \frac{S_x}{2}$$

$$P_y = O_y - \frac{S_y}{2}$$

Sprawdzenie:

$$P_x = O_x + \frac{S_x}{2} = 7 + \frac{8}{2} = 7 + 4 = 11$$

$$P_y = O_y - \frac{S_y}{2} = 5 - \frac{4}{2} = 5 - 2 = 3$$

Obliczmy teraz współrzędne E.

$$E_x = P_x - S_x \cdot E_x$$

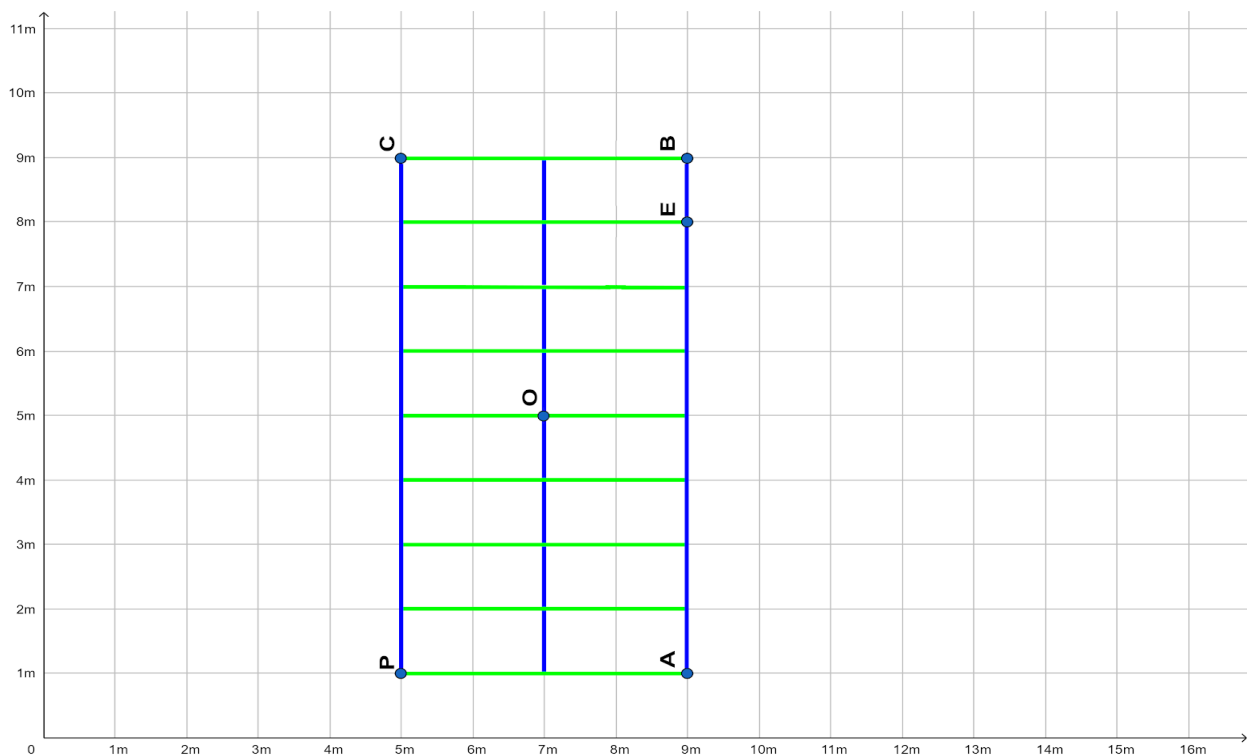
$$E_x = P_x - S_x \cdot E_x$$

Sprawdzenie:

$$E_x = P_x - S_x \cdot E_x = 11 - 1 \cdot 7 = 11 - 7 = 4$$

$$E_x = P_x - S_x \cdot E_x = 3 + 2 \cdot 2 = 3 + 4 = 7$$

Zachód (W), $R = (-1,0)$, obrót obiektu o 270°



Rysunek 6.5 - Sposób liczenia punktu E dla kierunku południowego

Obliczmy teraz współrzędne punktu P dla obrotu na zachód.

$$P_x = O_x - \frac{S_y}{2}$$

$$P_y = O_y - \frac{S_x}{2}$$

Sprawdzenie:

$$P_x = O_x - \frac{S_y}{2} = 7 - \frac{4}{2} = 7 - 2 = 5$$

$$P_y = O_y - \frac{S_x}{2} = 5 - \frac{8}{2} = 5 - 4 = 1$$

Obliczmy teraz ponownie punkt E dla kierunku zachodniego.

$$E_x = P_x + S_y \cdot E_y$$

$$E_y = P_y + S_x \cdot E_x$$

Sprawdzenie:

$$E_x = P_x + S_y \cdot E_y = 5 + 2 \cdot 2 = 5 + 4 = 9$$

$$E_y = P_y + S_x \cdot E_x = 1 + 1 \cdot 7 = 1 + 7 = 8$$

Z powyższych cząstkowych równań, uwzględniając współczynniki obrotu otrzymujemy uniwersalny wzór:

$$E_x = O_x - \left(\frac{S_x}{2} \cdot R_y\right) + \left(\frac{S_x}{L_x - 1} \cdot E_y \cdot R_y\right) + \left(\frac{S_y}{2} \cdot R_x\right) - \left(\frac{S_y}{L_y - 1} \cdot E_x \cdot R_x\right)$$

$$E_y = O_y + \left(\frac{S_y}{2} \cdot R_y\right) - \left(\frac{S_y}{L_y - 1} \cdot E_y \cdot R_y\right) + \left(\frac{S_x}{2} \cdot R_x\right) - \left(\frac{S_x}{L_x - 1} \cdot E_x \cdot R_x\right)$$

Sprawdzenie:

Dane wspólne (przepisane specjalnie z wcześniejszej strony):

- $O = (7, 5)$,
- $S = (8, 4)$,
- R - współczynnik obrotu obiektu,
- $L = (9, 3)$,
- $E' = (7, 2)$.

| | |
|---|--|
| N | $E_x = 7 - \left(\frac{8}{2} \cdot 1\right) + \left(\frac{8}{9-1} \cdot 7 \cdot 1\right) + \left(\frac{4}{2} \cdot 0\right) - \left(\frac{4}{3-1} \cdot 2 \cdot 0\right) = 7 - 4 + 7 + 0 + 0 = 3 + 7 = 10$ $E_y = 5 + \left(\frac{4}{2} \cdot 1\right) - \left(\frac{4}{3-1} \cdot 2 \cdot 1\right) + \left(\frac{8}{2} \cdot 0\right) - \left(\frac{8}{9-1} \cdot 7 \cdot 0\right) = 5 + 2 - 4 + 0 - 0 = 7 - 4 = 3$ |
| S | $E_x = 7 - \left(\frac{8}{2} \cdot -1\right) + \left(\frac{8}{9-1} \cdot 7 \cdot -1\right) + \left(\frac{4}{2} \cdot 0\right) - \left(\frac{4}{3-1} \cdot 2 \cdot 0\right) = 7 + 4 - 7 + 0 - 0 = 11 - 7 = 4$ $E_y = 5 + \left(\frac{4}{2} \cdot -1\right) - \left(\frac{4}{3-1} \cdot 2 \cdot -1\right) + \left(\frac{8}{2} \cdot 0\right) - \left(\frac{8}{9-1} \cdot 7 \cdot 0\right) = 5 - 2 + 4 + 0 - 0 = 3 + 4 = 7$ |
| E | $E_x = 7 - \left(\frac{8}{2} \cdot 0\right) + \left(\frac{8}{9-1} \cdot 7 \cdot 0\right) + \left(\frac{4}{2} \cdot 1\right) - \left(\frac{4}{3-1} \cdot 2 \cdot 1\right) = 7 - 0 + 0 + 2 - 4 = 7 + 2 - 4 = 9 - 4 = 5$ |

| | |
|---|---|
| | $E_y = 5 + \left(\frac{4}{2} \cdot 0\right) - \left(\frac{4}{3-1} \cdot 2 \cdot 0\right) + \left(\frac{8}{2} \cdot 1\right) - \left(\frac{8}{9-1} \cdot 7 \cdot 1\right) = 5 + 0 - 0 + 4 - 7 = 5 + 4 - 7 = 9 - 7 = 2$ |
| W | $E_x = 7 - \left(\frac{8}{2} \cdot 0\right) + \left(\frac{8}{9-1} \cdot 7 \cdot 0\right) + \left(\frac{4}{2} \cdot -1\right) - \left(\frac{4}{3-1} \cdot 2 \cdot -1\right) = 7 - 0 + 0 - 2 + 4 = 7 - 2 + 4 = 5 + 4 = 9$ |
| | $E_y = 5 + \left(\frac{4}{2} \cdot 0\right) - \left(\frac{4}{3-1} \cdot 2 \cdot 0\right) + \left(\frac{8}{2} \cdot -1\right) - \left(\frac{8}{9-1} \cdot 7 \cdot -1\right) = 5 + 0 - 0 - 4 + 7 = 5 - 4 + 7 = 1 + 7 = 8$ |

Tabela 6 - zestawienie wszystkich obliczeń punktów końcowych

Funkcja w języku PHP realizująca powyższy wzór wraz z założeniami została przedstawiona poniżej.

```
function posGrid($beginPoint, $size, $ratio, $lines, $endPoint){
    $returnPoint = new classPoint();

    if($lines->getPointX() % 2 == 0):
        $lines->setPointX($lines->getPointX() + 1);
    endif;

    if($lines->getPointY() % 2 == 0):
        $lines->setPointY($lines->getPointY() + 1);
    endif;

    if($endPoint->getPointX() < 0):
        $endPoint->setPointX(0);
    endif;

    if($endPoint->getPointY() < 0):
        $endPoint->setPointY(0);
    endif;

    if($endPoint->getPointX() > $lines->getPointX() - 1):
        $endPoint->setPointX($lines->getPointX() - 1);
    endif;

    if($endPoint->getPointY() > $lines->getPointY() - 1):
        $endPoint->setPointY($lines->getPointY() - 1);
    endif;

    $x = $beginPoint->getPointX() - (($size->getPointX() / 2) * $ratio->getPointY())
```

```

+ (($size->getPointX() / ($lines->getPointX() - 1)) * $endPoint->getPointX() * $ratio-
>getPointY()) + (($size->getPointY() / 2) * $ratio->getPointX()) - (($size->getPointY()
/ ($lines->getPointY() - 1)) * $endPoint->getPointY() * $ratio->getPointX());
$y = $beginPoint->getPointY() + (($size->getPointY() / 2) * $ratio->getPointY()) -
(($size->getPointY() / ($lines->getPointY() - 1)) * $endPoint->getPointY() * $ratio-
>getPointY()) + (($size->getPointX() / 2) * $ratio->getPointX()) - (($size->getPointX()
/ ($lines->getPointX() - 1)) * $endPoint->getPointX() * $ratio->getPointX());

    if($x > 255):
        $x = 255;
    endif;

    if($y > 255):
        $y = 255;
    endif;

    if(!in_array($ratio->getPointX(), array(-1, 0, 1))):
        $x = $beginPoint->getPointX();
    endif;

    if(!in_array($ratio->getPointY(), array(-1, 0, 1))):
        $y = $beginPoint->getPointY();
    endif;

    if(($lines->getPointX() <= 1) && ($lines->getPointY() <= 1)):
        $x = $beginPoint->getPointX();
        $y = $beginPoint->getPointY();
    endif;

    $returnPoint->setPointX(number_format($x, 5, '.', ''));
    $returnPoint->setPointY(number_format($y, 5, '.', ''));
    $returnPoint->setPointZ(number_format($beginPoint->getPointZ(), 5, '.', ''));
    return $returnPoint;
}

```

Tekst 10 - funkcja zwracająca współrzędne punktu na podstawie siatki logicznej ustawiania obiektu

Aby skorzystać z tej funkcji należy załączyć wcześniej plik class.point.php.

```

<?php
class classPoint {
    private $X;

```



```

private $Y;
private $Z;

public function setPoint($x, $y, $z) {
    $this->X = $x;
    $this->Y = $y;
    $this->Z = $z;
}

public function setPointX($x){
    $this->X = $x;
}

public function getPointX(){
    return $this->X;
}
}

```

Tekst 11 - kod źródłowy pliku class.point.php [bot_installer\corrade\bot_ai\php\class.point.php]

Teraz definiujemy nasz kod do wyliczenia, używamy przykładowych danych wymienionych w tym rozdziale:

- \$beginPoint - współrzędne środka obiektu (O) - przykład: (7,5),
- \$size - rozmiar obiektu (S) - przykład: (8,4),
- \$ratio - współczynnik obrotu (R) - przykład: (1,0),
- \$lines - liczba linii (L) - przykład: (9,3),
- \$endPoint - punkt końcowy (E') - przykład: (7,1).

```

$beginPoint = new classPoint();
$beginPoint->setPoint(7, 5, 100);

$size = new classPoint();
$size->setPoint(8, 4, 0);

$ratio = new classPoint();
$ratio->setPoint(1, 0, 0);

$lines = new classPoint();
$lines->setPoint(9, 3, 0);

```

```
$endPoint = new classPoint();  
$endPoint->setPoint(7, 2, 0);  
  
$grid = posGrid($beginPoint, $size, $ratio, $lines, $endPoint);  
  
echo 'Result: X:' . $grid->getPointX() . ', Y:' . $grid->getPointY() . ',Z:' . $grid->  
>getPointZ() . PHP_EOL;
```

Tekst 12 - próba wyliczenia punktu końcowego z użyciem PosGrid [bot_installer\corrade\bot_ai\php\functions.php]

Co da nam jako wynik:

```
Result: X:5.00000, Y:2.00000,Z:100.00000
```

Tekst 13 - wynik funkcji PosGrid na podstawie wcześniej zdefiniowanych danych

7. Najczęściej zadawane pytania

Większość poniższych pytań i odpowiedzi można znaleźć na stronie producenta oprogramowania.

Skontaktuj się ze mną, jeżeli masz pytanie odnośnie Corrade.

1. Czy Corrade wymaga hostingu?

Tak, Corrade jako program wymaga hostingu. Możesz wynająć jakiś VPS lub przeznaczyć jakiś swój wolny komputer pod niego, aby nie płacić co miesiąc za wynajem.

2. Co mogę zrobić z Corrade?

Corrade ma zaprogramowane wszystkie możliwości przeglądarki do Second Life. Przy pomocy Corrade możesz, np. przyjmować automatycznie płatności za wynajem ziem, czy sprzedawać przedmioty.

Możliwości Corrade są szerokie, wystarczy wczytać się w dokumentację.

3. Czy Corrade jest darmowy?

Tak, Corrade jest darmowy. Jest licencjonowany pod własną licencję, która nie jest restrykcyjna. W zasadzie producent prosi o oznaczanie go podczas używania w innych projektach oraz niedokonywanie inżynierii wstecznej na binarkach programu.

Więcej informacji o licencji znajdziesz na stronie grimore.org²¹ oraz w rozdziale:  **Licencja**.

4. Czy jeżeli Corrade jest darmowy, to jestem "produktem"?

Nie potrafię tego powiedzieć. Producent zapewnia, że nie zbiera żadnych danych o użytkowniku w celach monetyzacji. Jeżeli chcesz dołożyć swój mały wkład w rozwój projektu, możesz, np.

21 Licencja WAS PC & OD 1.0 - <https://grimore.org/licenses/was-pc-od>

włączyć opcję wysyłania danych diagnostycznych w Nucleus, zareklamować projekt, napisać skrypty i udostępnić je szerszej publiczności.

5. Czy Corrade jest OpenSource?

Nie, kody źródłowe projektu są zamknięte dla publiczności. Jednakże szablony, skrypty PHP czy LSL są otwarte i możesz je spokojnie modyfikować.

6. Na których platformach można uruchomić Corrade?

Obecnie Corrade został zbudowany w oparciu o .NET Core i można uruchomić go na systemach: Linux, Windows, MacOSX, platformach (x64, ARM) które pozwalają na zainstalowanie .NET Core.

7. Jak rozpocząć pracę i programowanie z Corrade?

Musisz ściągnąć Corrade i go skonfigurować. Jeżeli chodzi o programowanie musisz znać podstawy języka LSL od Second Life. Nie jest to trudny język do opanowania. Na stronach Second Life znajdziesz bogate źródło wiedzy o tym języku²².

Na stronach Corrade oraz w sklepie producenta Corrade (Second Life Marketplace)²³ znajdziesz darmowe przykłady używania Corrade.

8. Jak Corrade zarządza kontrolą dostępu?

Corrade zarządza kontrolą dostępu przypisaną do grupy. W innych słowach określasz grupy z hasłami, które będą miały możliwość do wykonania komend na Corrade. Corrade nie dba o to, jaki awatar wywołuje komendy. Ma to swój plus, gdyż możesz stworzyć skrypt w Second Life z odpowiednimi uprawnieniami (np. brak dostępu do podglądu pliku), sprzedać go, wówczas kupujący tylko będzie mógł zmodyfikować notatkę z konfiguracją bota.

²² LSL Portal - http://wiki.secondlife.com/wiki/LSL_Portal

²³ Corrade Second Life Marketplace - <https://marketplace.secondlife.com/stores/165275>

9. Dlaczego Corrade używa grupy zamiast awatarów do dostępu czy uprawnień?

Grupy w Second Life są na najniższym poziomie odnośnie uprawnień.

Operatorzy siatki mają największe uprawnienia w Second Life, zarządcy majątku mają uprawnienia tylko do zarządzania ziemią, uprawnienia zwykłych awatarów ogranicza się do swoich zasobów. Większość akcji w Second Life, tj. dzielenie się ziemią z kimś innym, streaming muzyki na urządzeniu, zezwolenie na rezz obiektu i inne proste operacje, tj. ustawienie landmarku są ograniczone przez uprawnienia grupy, która jest przypisana do ziemi.

Aby zapewnić pomyślne wykonanie poleceń, wszystkie polecenia muszą zapytać siatkę o uprawnienia, ponieważ w przypadku braku graficznego interfejsu użytkownika, Corrade nie będzie w stanie raportować błędy i próby, czy polecenie zostało pomyślnie wykonane. Większość kontroli przeprowadzanych przez Corrade kończą się zapytaniem o uprawnieniem grupy.

10. Dlaczego Corrade nie potrafi sam siebie zrestartować?

Ponieważ są narzędzia, które lepiej radzą sobie z restartem danego programu, uwzględniając takie warunki jak: stan i obciążenie sieci, obciążenie procesora, ruchu, itd., i na tej podstawie decyduje czy zrestartować program.

11. Ściągnąłem Corrade, skonfigurowałem, ale w konsoli widzę komunikat, że jest problem z logowaniem.

Zazwyczaj jest problem z nieprawidłowym hasłem podanym podczas konfiguracji bota.

```
login failed : key
```

Otwórz plik Configuration.xml i znajdź 'Password'.

Otwórz też sobie jakieś narzędzie do kodowania w MD5 (w google wpisz: md5 online).

```
<Password>$1$842e4818f8ede223c9b920d4f7425c9b</Password>
```

Usuń zawartość pomiędzy <Password> a </Password>.

Dodaj \$1\$ a następnie zakodowane hasło w MD5.

Jeżeli twoje hasło ma więcej niż 16 znaków, to zmień je na hasło o długości 6-16 znaków, ponieważ tylko takie hasła Second Life przyjmuje.

12. Co oznacza "presence"?

W konsoli i logach programu możesz zobaczyć błąd:

```
login failed : presence
```

Oznacza to, że awatar jest już zalogowany w Second Life, być może sam ręcznie się zalogowałeś przy użyciu przeglądarki albo Corrade poprzednio niepoprawnie się wylogował ze świata.

Próba zalogowania się na konto, gdzie awatar jest już zalogowany spowoduje, że wcześniejsze zalogowanie jest rozłączane, a użytkownik widzi komunikat, że wylogowano z Second Life.

Przy tym błędzie Corrade spróbuje się ponownie połączyć.

Jeżeli natomiast widzisz błąd:

```
login failed : ban
```

Oznacza, to, że zastosowano tymczasową blokadę na zalogowanie się, w takiej sytuacji poczekaj parę minut i spróbuj zalogować się ponownie.

13. Czy Corrade pracuje na OpenSim?

Corrade jest w zasadzie kompatybilny z protokołami Second Life, ponieważ zostały one dobrze opisane. Corrade może pracować na siatce OpenSim, ale producent nie zapewnia, że wszystkie polecenia będą poprawnie pracować, gdyż OpenSim nie jest kompatybilny z protokołami Second Life i jest mocno obciążony błędami.

14. Wysłałam URL przez Corrade, które zawiera spacje, ale Corrade psuje je

Dzieje się to najczęściej dla SLURL, np. secondlife://Some%20Simulator/32/13/34, gdzie Corrade przekształca spację do %20 w taki sposób, że link przestaje być rozpoznawalny przez przeglądarkę jako link.

Wyślij link jako: secondlife://Some%2520Simulator/32/13/34, który będzie już działał (%2520 będzie przekształcane do %20).

15. Próbuję dodać grupę do konfiguracji Corrade, która zawiera znaki specjalne.

We wpisach XML, niektóre znaki są specjalne i muszą być zastąpione swoimi odpowiednikami.

| Znak | Odpowiednik |
|------|-------------|
| " | " |
| ' | ' |
| < | < |
| > | > |
| & | & |

Tabela 7: Specjalne znaki w HTML

Na przykład grupa o nazwie Wizardry & Steamworks, musi być w pliku konfiguracyjnym przekształcona do postaci: Wizardry & Steamworks

16. Kiedy wysyłam komendy z LSL dla ukrytej grupy, dostaję zawsze informację o braku dostępu lub to, że grupa nie została znaleziona.

Zamiast przypisywać nazwę grupy do parametru grupy, przypisz do niego UUID grupy. Powodem jest to, że Corrade nie może znaleźć grupy wyświetlanej w usługach katalogowych.

17. Corrade i Proxy

Corrade ma kiepskie doświadczenie z sewerami proxy czy Torem.

Najlepiej nie ustawiać proxy na komputerze, na którym jest uruchomiony Corrade.

Najczęstsze symptomy korzystania z proxy to:

- szafa nie do końca się wczytuje, a próba wskazania do elementu w szafie zwraca błąd, że pozycja nie istnieje,
- Corrade nie wie, w jakich grupach jest,
- niektóre powiadomienia często zawodzą.

Więc jak widzisz, najlepiej nie korzystać z proxy i go w systemie wyłączyć.

18. Mój bot ma problem z rezzowaniem (albo pojawia się jako chmura)

Są dwa znane możliwości rezzowania wyglądu bota jako chmury:

- Corrade nosi więcej niż jeden załącznik na slot; Corrade obsługuje jeden załącznik na jeden slot,
- Corrade nie nosi: skóry, oczów, włosów lub kształtu; w Second Life jedna część ciała musi mieć jakieś akcesorium; Przeglądarki mają tendencję do zakazu pokazywania awatarów, które nie noszą akcesoriów na części ciała.

19. Autopilot z wysokością

Second Life zawiera od dawna występujący błąd, który sprawia, że agent zachowuje się arbitralnie na wysokości określonego dla polecenia autopilota. Jeśli chcesz, aby bot poleciał do lokalizacji zamiast spacerował, rozważ użycie polecenia 'flyto', które jest znacznie bardziej wszechstronne lub połącz oba:

- Użyj autopilota, aby poruszać się po powierzchniach/ziemi,
- Użyj 'flyto', by Corrade poleciał do celu.

20. Dlaczego niektóre akcesoria do noszenia nie są zgłaszane przez Corrade?

Jest to problem występujący w bibliotece libopenmetaverse z którego Corrade korzysta, gdzie nosząc niektóre akcesoria (np. spódnicę) nie są pokazywane na bocie. Ten błąd był zgłaszany kilka razy, naprawiany i znów zgłaszany.

To powoduje, że komendy dotyczące noszenia przedmiotów mają niespójne zachowanie. Kiedy te błędy zostaną naprawione w bibliotece, komendy powinny pracować już poprawnie.

21. Bot spada z pojazdów podczas przekraczania regionów

Ten błąd pojawia się we wszystkich znanych oprogramowaniach do spraw botów. Nikt nie wie, co powoduje ten błąd. Jest on spowodowany najpewniej w bibliotece libopenmetaverse, był on kilka razy zgłaszany, bez efektów naprawczych.

22. Czy muszę aktualizować?

Corrade otrzymuje często aktualizacje, które naprawiają zgłaszane błędy.

Przed pobraniem można zapoznać się z plikiem tekstowym, który zawiera krótki opis, co zmieniono w danej wersji.

Z mojego osobistego doświadczenia należy się cieszyć z rozwoju programu, itd., jednakże zalecam ostrożność do nowych wersji programu, ponieważ może okazać się, że nowa wersja coś zepsuje funkcjonalności bota. Najlepiej w takim przypadku zrobić zrzut ekranu, zgłosić błąd u producenta na stronie.

Ja osobiście mam dwa boty - jeden operacyjny - działa na stabilnej wersji programu, drugi testowy - na którym jest testowane nowe wydanie programu i jeżeli tutaj wszystko będzie działało, wówczas bot operacyjny jest aktualizowany do nowszej wersji.

23. Które porty są używane przez Corrade?

Corrade używa tych samych portów co Second Life

TCP - 53, 80, 443, 12043, 12046, 21002

UDP - 53, 3478, 3479, 5060, 5062, 12000-29999

24. Czy muszę restartować Corrade po zmianach w konfiguracji?

Nie, nie musisz, Corrade sam powinien wykryć te zmiany i je zastosować, jeżeli były dokonywane przez Nucleus.

25. Czy Wizardry and Steamworks lub biblioteki pobierają jakieś prywatne dane?

Nie, nie pobierają żadnych prywatnych danych.

26. Dlaczego nie należy używać proxy?

Corrade i Second Life raczej nie lubią kiedy komputer korzystania z Internetu przez serwer proxy. Powoduje to po stronie Corrade problem z uzyskaniem czy wysłaniem wyników, pobieranie danych z Second Life zwłaszcza odnośnie szafy awatara, obsługą powiadomień, informacji w jakich grupach się znajduje. Dlatego zaleca się, aby komputer na którym działa Corrade łączył się z Internetem bezpośrednio bez pośrednictwa serwera proxy.

27. Mam problem z powiadomieniami lub danymi zwrotnymi.

Czy korzystasz z llRequestURL()? Jeżeli tak to zamień na llRequestURL(), które owszem nie jest bezpiecznym URL, jednakże wynika to z tego powodu, że Second Life posługuje się samopodpisanymi certyfikatami dla serwerów, a Corrade może mieć z nimi problem.

8. Słownik

A

API (Application Programming Interface) – jest to zbiór reguł, które określają w jaki sposób program lub programy komunikują się ze sobą.

ATC (Air Traffic Control) – kontroler ruchu powietrznego.

B

Bot – program, który wykonuje czynności szybciej w sposób zautomatyzowany naśladując człowieka.

D

Działka – wirtualne miejsce w Second Life, które jest wynikiem podziału regionu na mniejsze części ziemi.

I

IM (Instant Message) – prywatna wiadomość wysłana pomiędzy dwoma użytkownikami.

L

LSL (Language Second Life) – pseudo język stworzony przez Liden Lab., będący kombinacją C#, C++, Javy, który pozwala na podstawową animację obiektów oraz interakcję z awatarem.

M

Mesh – rodzaj grafiki w Second Life, która została stworzona w zewnętrznym programie graficznym, np. Blender i zaimportowana do Second Life.

O

Obiekt – połączone ze sobą primy (lub primy z meshami) w jedną całość.

P

PHP – język do tworzenia dynamicznych stron internetowych.

Prim – podstawowy element budowy w Second Life.

Proxy – serwer pośredniczący pomiędzy komputerem użytkownika a serwerem docelowym z którego ma zostać pobrana zawartość.

R

Region – wirtualna ziemia w Second Life.

Rezz – może oznaczać, np. umieszczać obiekt z szafy, ładowanie ubrania, ładowanie tekstury na obiekcie.

Róża kompasowa – wskazuje kierunki: północ, południe, wschód i zachód.

S

Second Life – wirtualny świat dostępny przez Internet z własną społecznością, systemem montearnym, itd.

Second Life Marketplace – sklep Second Life oferujący różne wirtualne dobra tworzone przez użytkowników.

U

URL – adres internetowy, który wskazuje na konkretny zasób w Internecie, np. stronę WWW lub plik.




V

VPS (Virtual Private Server) - wirtualny serwer dedykowany, który jest wydzielany jako pewien obszar na serwerze.

W

Współrzędne – określają pozycję obiektu, awatara.

9. Użyte obrazy

- **Okladka** – obraz pobrany z mepixels.com²⁴ na licencji Public Domain Dedication (CC0),
- **Obraz przypisanego DOI** – pobrany ze strony zenodo.org²⁵,
- **Rysunek 4.1 - Logo Corrade** - obraz pobrany ze strony grimore.org²⁶ na licencji 'free for compatibility',
- **Rysunek 4.2 - Sposób przepływu informacji w Corrade** - obraz pobrany ze strony grimore.org²⁷,
-  - obraz obrany z wikimedia.org²⁸ na licencji public domain,
-  - obraz obrany z wikimedia.org²⁹; autor: derex99; na licencji CC BY-SA 3.0,
-  - obraz obrany z wikimedia.org³⁰; autor: Fabrice Florin; na licencji CC BY-SA 3.0.

24 Strona - <https://www.mepixels.com/photo/bot-is-working-in-pc-51>

25 Strona zenodo - <https://zenodo.org/>

26 Strona grimore.org z logo - https://grimore.org/_detail/secondlife/scripted_agents/secondlife_scripted_agents_corrade_banner.png?id=secondlife%3Ascripted_agents%3Acorrade%3Alogos

27 Strona grimore.org – Corrade - https://grimore.org/secondlife/scripted_agents/corrade

28 Strona wikimedia.org - https://commons.wikimedia.org/wiki/File:Info_icon-72a7cf.svg

29 Strona wikimedia.org - https://commons.wikimedia.org/wiki/File:CD_icon_test.svg

30 Strona wikimedia.org - https://commons.wikimedia.org/wiki/File:Media_Viewer_Icon_-_Link_Hover.svg

10. Materiały dodatkowe



Materiały dodatkowe są dostępne wraz z tym ebookiem lub na stronie <https://ko-fi.com/nitropl/shop> za wsparcie autora.

10.1. Instalator bota na VPS (skrypty Powershell, Windows)

Mój autorski instalator Corrade, który zainstaluje Corrade wraz z dodatkowymi programami, np. na VPS.

Działanie skryptu:

- dodaje odpowiednie porty do zapory sieciowej,
- instaluje zestaw bibliotek umożliwiający uruchamianie w systemie Windows wielu aplikacji stworzonych przy pomocy narzędzi programistycznych Microsoftu (Visual C++ + Redistributable Package for Visual Studio 2019),
- instaluje .NET Framework,
- instaluje szereg aktualizacji podnoszących Powershell do wersji 5.1,
- instaluje .NET Core,
- instaluje 7-ZIP,
- instaluje XAMPP (opcjonalnie)
 - dodanie reguły portów używanych przez XAMPP do zapory sieciowej,
 - wypakowanie XAMPP na dysk C:\ do ścieżki C:\xampp,
 - kopiuje konfigurację Apache,
 - kopiuje przydatne moduły Apache,
 - kasuje folder htdocs i kopiuje wcześniej przygotowany folder,
 - kopiuje pliki konfiguracyjne - my.ini i mysql.ini,
 - instaluje usługi dla Apache i MySQL,
 - ustawia hasło root dla MySQL,
 - ładuje zapytania SQL z pliku,
 - dodaje PHP do ścieżki systemowej,
 - dodaje użytkownika 'corrade' z wcześniej zdefiniowanym hasłem w tabeli uprawnień MySQL i przyznaje uprawnienia do bazy 'corrade', którą też tworzy.

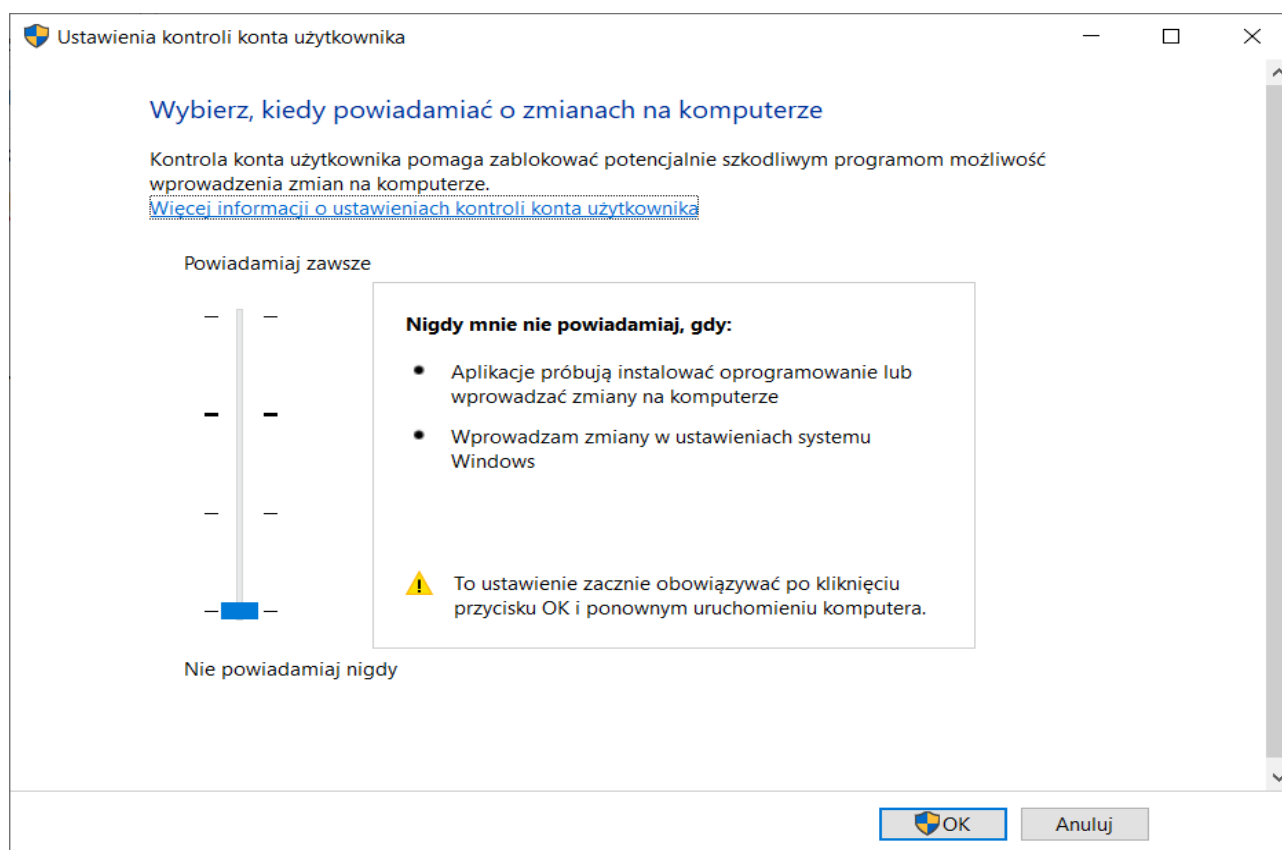
- Jeżeli XAMPP nie został zainstalowany:
 - rozpakowuje PHP do C:\php,
 - dodaje PHP do ścieżki systemowej,
 - kopiuje plik php.ini.
- instaluje Corrade
- kopiuje AI bota do folderu,
- kopiuje konfigurację bota i Nucleus,
- tworzy skróty do bota na pulpicie, instaluje usługę w systemie, dodaje zadanie do harmonogramu zadań, które sprawdza dostępność bota (o ile w folderze bota jest plik mcorrade.ps1).
- instaluje Firefox (opcjonalnie),
- instaluje Notepad++ i kopiuje domyślne ustawienia (opcjonalnie),
- konfiguruje komputer (opcjonalnie),
- wyłącza zbędne usługi (bufor druku, defragmentacja dysku, fax, tematy, asystent zgodności programów, Windows Update),
- wyłącza animację podczas uruchamiania komputera,
- kasuje obecne kopie zapasowe woluminów,
- wyłącza opcję przywracania systemu,
- wyłącza hibernację systemu,
- włącza aktualizację znacznika czasu ostatniego dostępu na każdym katalogu, który jest wymieniony na woluminie NTFS,
- włącza szyfrowanie plików i katalogów na woluminie NTFS,
- wyłącza szyfrowanie pliku stronicowania pamięci w Windows,
- wyłącza tworzenie nazw plików o długości 8.3 znaków na woluminach NTFS i FAT,
- wyłącza DEP,
- ignoruje błędy podczas uruchamiania systemu,
- pozwala na połączenie zdalne z komputerem,
- wyłącza poprawki dla Meltdown i Spectre,
- instaluje Rainmeter (opcjonalnie)
- instaluje HWINFO w systemie i w Harmonogramie zadań,
- instaluje 'lepsze' ustawienia dla wiersz polecenia,
- dodaje Harmonogram zadań jako skrót na Pulpicie,

- dodaje i ustawia inny plan zasilania (opcjonalnie).

Instalator znajduje się w folderze: bot_installer. Wszystkie czynności skryptu będą zapisywane do pliku install.log.txt. Aby użyć instalatora potrzebujesz m.in. 5 GB wolnego miejsca na dysku (nie, raczej instalator nie zużyje, to jest tylko jako rezerwacja).

Przedstawiam krok po kroku co musisz zrobić:

1. Wyłącz UAC (kontrolę dostępu użytkownika) - daj na niższy poziom suwaka, kliknij na OK, zrestartuj komputer docelowy.



Rysunek 10.1 - Wylączania kontroli dostępu użytkownika w systemie Windows

2. Teraz przygotowujemy się do instalacji Corrade na innym komputerze.
3. Przejdź do materiałów dodatkowych, do folderu bot_installer.
4. Masz tam wiele plików.
5. Otwórz plik configs\installer.config.ps1.


```
#SET XAMPP INSTALL
$xmlpp_install=0
#SET MYSQL ROOT PASSWORD
$mysqlRootPassword="12345"
#SET MYSQL CORRADE PASSWORD
$mysqlCorradePassword="qazwsxedc"
#INSTALL FIREFOX
$firefox_install=0
#INSTALL NOTEPAD++
$notepadpp_install=0
#CONFIGURE COMPUTER
$configurePC=0
#SET RAINMETER INSTALL
$rainmeter_install=0
#ADD AND SET POWER PLAN
$powerplan=0
```

Decydujemy tutaj, jakie składniki chcemy zainstalować:

1. `$xampp_install=0`

1 - instaluje i konfiguruje XAMPP³¹

0 - omija process związany z XAMPP

Poniżej bardziej szczegółowy opis dla tej opcji.

- `xampp_config\apache` - w tym folderze możesz zdefiniować wstępną konfigurację Apache (folder 'conf') i dodatkowe moduły dla niego ('modules'). Te foldery odzwierciedlają drzewo folderów i plików w XAMPP. Skopiuj tutaj swoją konfigurację, pamiętając, że na komputerze docelowym serwer jest instalowany domyślnie do folderu 'C:\xampp'. W folderze 'modules' natomiast dodaj moduły, które mają być skopiowane do folderu modułów Apache na komputerze docelowym. Pamiętaj, aby te moduły były zgodnie z wersją Apache pakietu XAMPP, którego chcesz użyć oraz aby w konfiguracji docelowej były one włączone.
- Wstępnie znajdziesz moduły, które ja używam - ograniczenie prędkości transferu (moduł `mod_bw`), ochrona przez atakiem DDoS - Slowloris (moduł `mod_antiloris`) oraz zbyt dużą ilość połączeń z jednego adresu IP (moduł `mod_evasive`, moduł `mod_limitipconn`) i mam nadzieję, że będą przydatne, inaczej możesz je usunąć z folderu.

31 Strona projektu: <https://www.apachefriends.org/>

- `xampp_config\htdocs` - zawiera wstępnie zdefiniowany katalog publiczny dla XAMPP. Umieść tutaj swoje skrypty, strony, itd. Zostanie on przekopiowany po instalacji XAMPP. Ode mnie masz przeglądanie bazy MySQL (PHPMyAdmin³²), wyświetlanie informacji o systemie (phpsysinfo³³) oraz skrypt na temat aktualnego ustawienia PHP (skrypt: `phpsysinfo.php`).
- `xampp_config\php.ini` - wcześniej przygotowany plik `php.ini`, który będzie przekopiowany po instalacji XAMPP.
- `xampp_config\my.ini` - wcześniej przygotowany plik `my.ini` (dla MySQL), który będzie przekopiowany po instalacji XAMPP.
- `xampp_config\sql.sql` - plik SQL z zapytaniami SQL, które będą wykonywane po instalacji XAMPP - tutaj tylko dodałem tworzenie bazy `corrade`.

2. `$mysqlRootPassword="12345"`

Ustala wstępne hasło dla root MySQL z pakietu XAMPP, raczej zastosuj jakieś proste hasło na początek (użyj liter, cyfr, ale bez znaków specjalnych - później możesz je zmienić), gdyż skrypt ma problem z ustawieniem takich skomplikowanych haseł i będziesz musiał ręcznie odzyskać dostęp do bazy danych.

3. `$mysqlCorradePassword="qazwsxedc"`

Ustala wstępne hasło dla użytkownika 'corrade' podczas tworzenia bazy danych z pakietu XAMPP. Sposób ustawienia hasła opisane są jak wyżej, czyli na początku litery i cyfry bez znaków specjalnych.

4. `$firefox_install=0`

Kopiuje wcześniej zdefiniowany folder z programem Firefox.

W `programs\FirefoxPortable` znajdziesz program w wersji przenośnej, który będzie kopiowany na komputer docelowy. Możesz normalnie go uruchomić i ustawić profil, ustawienia, dodać zakładki, itd. - wszystko będzie zapisywane w jego folderze.

Ode mnie masz zakładki do modułów Apache, szybszy dostęp do Nucleus, PHPMyAdmin, PHPSysInfo, dokumentacji Corrade w wersji online.

5. `$notepadpp_install=0`

Instaluje Notepad++, po instalacji kopiuje profil z `configs\Notepad++`, który może być wcześniej modyfikowany.

6. `$configurePC=0`

32 Strona projektu: <https://www.phpmyadmin.net/>

33 Strona projektu: <https://phpsysinfo.github.io/phpsysinfo/>

Konfiguruje komputer (zalecana opcja na VPS z Windows), np. wyłącza poprawki odnośnie Meltdown i Spectre, wyłącza niektóre funkcje, wyłącza tematy, itd.

7. `$rainmeter_install=0`

Instaluje Rainmeter (gadżety dla pulpitu) kopiując folder z programs\Rainmeter, który może być dostosowany do indywidualnych potrzeb.

Wraz z Rainmeterem jest instalowany HWINFO (programs\hwinfo) do zbierania temperatury z systemu oraz zadanie do Harmonogramu zadań (tasks\hwinfo.xml).

8. `$powerplan=0`

Instaluje i ustawia nowy plan zasilania dla systemu (plik others\server.pow), który nie pozwala na wyłączenie się dysków twardych; wyłącza po minucie ekran, itd.

Po skonfigurowaniu pliku konfiguracyjnego instalatora skryptu i ewentualnie dodatkowych folderów dla każdej opisanej wyżej opcji, idziemy dalej.

- corrade\bot_ai - ten folder zawiera pliki i foldery, które zostaną przekopiiowane do folderu bota. Można tutaj umieścić mózg i serce bota.
- W tym folderze możesz umieścić także pliki już skonfigurowane: Configuration.xml oraz Nucleus.xml od Corrade.

W tym folderze znajdują się ode mnie pliki (czyli w zasadzie nic do konfiguracji):

- corrade-kill.bat - zabija proces bota,
- Create ShortCuts.bat - tworzy skróty na pulpicie w folderze 'ArtBot Menu',
- mcorrade.ps1 - plik w Powershell, który pozwala na zarządzanie botem,
- nssm.exe - pozwala na instalację programu jako usługi w systemie,
- restart.bat - restartuje bota,
- start.bat - startuje bota,
- stop.bat - zatrzymuje bota,
- update.bat - aktualizuje bota.

Inne pliki w folderze instalatora:

- installer.ps1 - główna część instalatora,
- installer.bat - program wsadowy, który uruchamia plik installer.ps1 z odpowiednimi parametrami,

- functions.ps1 - funkcje stworzone dla instalatora,
 - configs\php.ini - plik php.ini, który będzie kopiowany w przypadku nie instalowania XAMPP,
 - programs\7zip.exe - program 7-ZIP,
 - programs\php.zip - archiwum zip z PHP,
 - programs\Root_Certificate_Updater - program do aktualizacji certyfikatów w systemie,
 - programs\VC_redist.x64.exe - Visual C ++ Redistributable Package for Visual Studio 2019 x64,
 - programs\VC_redist.x86.exe - Visual C ++ Redistributable Package for Visual Studio 2019 x86,
 - registry\console-restore-settings.reg - ustawienia dla wiersza poleceń,
 - Aktualizacje systemowe, aktualizując Powershell do wersji 5.1:
 - patches\kb4457144-x64.msu,
 - patches\kb4457144-x86.msu,
 - patches\W2K12-KB3191565-x64.msu,
 - patches\Win7-KB3191566-x86.msu,
 - patches\Win7AndW2K8R2-KB3191566-x64.msu,
 - patches\Win8.1-KB3191564-x86.msu,
 - patches\Win8.1AndW2K12R2-KB3191564-x64.msu.
6. Jeżeli chcesz, możesz zmodyfikować plik installer.ps1, dodać swoje programy do folderu i skryptu instalacyjnego.
 7. Jeżeli wszystko jest gotowe, spakuj folder bot_installer, najlepiej z użyciem 7-ZIP.
 8. Gotowe archiwum przenieś na komputer docelowy.
 9. Rozpakuj folder.
 10. Otwórz folder bot_installer.
 11. Kliknij 2x na installer.bat.
 12. Zaczekaj aż skrypt zainstaluje wszystko i uruchomi komputer.
 13. Instalator cały proces instalacyjny rejestruje do pliku install.log.txt w katalogu bot_installer, więc potem możesz przejrzeć ten plik w poszukiwaniu błędów.
 14. Po uruchomieniu zobaczysz następujący obrazek:



Rysunek 10.2 - Pulpit po instalacji przez skrypt

15. Oznacza to, że wszystko jest gotowe do pracy.

FAQ

1. Po instalacji Corrade nie działa

Nie działa, ponieważ domyślnie usługa Corrade jest wyłączona.

Aby uruchomić Corrade, na pulpicie masz folder 'ArtBot Menu' a tam skrót o nazwie 'Artbot Start', kliknij 2x na niego, a Corrade powinien się uruchomić w tle.

2. Serwer XAMPP się nie uruchomił (brak dostępu do adresu <http://127.0.0.1>)

- Uruchom panel XAMPP C:\xampp\xampp-control,
- Zobacz na zdarzenia programu,
- Być może usługi zostały źle zainstalowane
 - W takim razie zatrzymaj Apache i/lub MySQL przez kliknięcie na przycisk STOP,

- Następnie obok programu kliknij na ✓ aby odinstalować usługę,
- Następnie kliknij ponownie w to samo miejsce na ✕, aby zainstalować program jako usługę,
- Daj na START obok każdego programu.
- Jeżeli Apache się nie uruchomił, to jest wiele przyczyn
 - Aby zobaczyć co się stało, w katalogu C:\xampp uruchom apache_start.bat,
 - Zobaczysz komunikat z Apache co się stało,
 - Być może coś źle jest w plikach konfiguracyjnych.
- Zachęcam tutaj do skorzystania dalej z Google.

3. Nie mogę się zalogować do serwera jako 'root' na serwerze MySQL (reset hasła)

- Uruchom panel XAMPP,
- zatrzymaj serwer MySQL,
- Otwórz w notatniku plik: C:\xampp\mysql\bin\my.ini,
 - Na początku sekcji [mysqld] dopisz:

`skip-grant-tables`

- Zapisz plik.
- W panelu XAMPP przy MySQL daj na START,
- W folderze C:\xampp\mysql\bin uruchom wiersz poleceń,
- Wpisz i zatwierdź: C:\xampp\mysql\bin\mysql.exe --user=root,
- Po zalogowaniu się jako root bez hasła, wykonujemy dalsze polecenia SQL:

`FLUSH PRIVILEGES;`

`ALTER USER 'root'@'localhost' IDENTIFIED BY '12345';`

`ALTER USER 'root'@'127.0.0.1' IDENTIFIED BY '12345';`

`exit`

- Teraz kasujemy wcześniej dopisany skip-grant-tables w pliku my.ini i zapisujemy plik.
- W panelu XAMPP zatrzymujemy i uruchamiamy MySQL.
- Teraz możemy zalogować się do MYSQL jako root z użyciem hasła 12345.

10.2. Aktualizacja programów w bot_installer

Przed spakowaniem instalatora Corrade wraz z dodatkowymi programami, przemyśl aktualizację programów, gdyż te dostępne w dodatkowych materiałach mogą być już dawno przestarzałe.

Listę programów wymagających uwagi znajdziesz w pliku bot_installer\README.txt.

Wystarczy ściągnąć program z podanego adresu URL do wskazanej lokalizacji podanej przez ścieżkę PATH, gdzie zgadzamy się na zamianę plików. Należy pamiętać, że jeżeli program był jako archiwum ZIP to ściągamy ZIP, jeżeli EXE to EXE, MSI to MSI, itd.

10.3. mcorrade.ps1 – zarządzanie botem (Powershell, Windows)

Jest to mój skrypt w Powershell, który pozwala na zarządzaniem Corrade.

Znajdziesz go w materiałach dodatkowych: bot_installer\corrade\bot_ai.

Aby go użyć, przekopiuuj wszystkie pliki i foldery z tego folderu do folderu bota.

Uruchom wiersz poleceń (nie od PowerShella), aby wywołać program z parametrami wpisz:

```
powershell.exe -ExecutionPolicy Bypass -File .\mcorrade.ps1 -PARAMETR
```

gdzie PARAMETR może być:

- -Latest - pobiera i aktualizuje Corrade do najnowszej wersji,
- -Status - pokazuje aktualny status usługi Corrade,
- -Start - uruchamia Corrade,
- -Stop - zatrzymuje Corrade,
- -Restart - restartuje Corrade,
- -InstallService - instaluje Corrade jako usługę,
- -DeleteService - usuwa Corrade jako usługę,
- -EnableService - włącza usługę Corrade,
- -DisableService - wyłącza Corrade jako usługę,
- -GetVersion - pobiera aktualną wersję Corrade,
- -CreateShortCuts - tworzy skróty do bota.

10.4. Opis plików w bot_installer\corrade\bot_ai

- bot_ai\php\botconfig.ini

```
[general]
MonitorServices = 1
MonitorServicesAttempts = 3
MonitorServicesCheckEvery = 1
MonitorServiceCreatePrim = 1
MonitorServiceCreatePrimAxisZ = 10
botTitle = "Bot Manager"
botTitleTarget = "Bot Group"
cacheTypeWriteDatabase = 0
[homeback]
enabled = 0
slurl = "http://maps.secondlife.com/secondlife/XXXXXX/112/112/23"
polygon = "<127,97,21>|<97,97,21>|<97,127,21>|<127,127,28>"
```

Ten plik jest korojoną wersją pliku konfiguracyjnego dla mojego bota.

| | |
|---------------------------------------|---|
| general.MonitorServices | włącza monitorowanie Corrade |
| general.MonitorServicesAttempts | określa maksymalną liczbę prób, po których bot zostanie zrestartowany |
| general.MonitorServicesCheckEvery | co ile oznaczać próbę sprawdzenia, czy bot działa |
| general.MonitorServiceCreatePrim | czy podczas sprawdzenia bota, próbować utworzyć prim tymczasowy w Second Life |
| general.MonitorServiceCreatePrimAxisZ | w jakiej odległości umieścić prim tymczasowy |
| general.botTitle | jaki tytuł bota ustawić (z grupy) |
| general.botTitleTarget | grupa docelowa ustawiania tytułu |
| general.cacheTypeWriteDatabase | typ zapisu cache: 0 - plik, 1 - baza danych |
| homeback.enabled | czy procedura sprawdzenia czy bot jest na właściwej działce jest włączona |
| homeback.slurl | SLURL regionu i działki, na którym |

| | |
|------------------|--|
| | powinien być bot |
| homeback.polygon | granice działki w których bot powinien się poruszać; granica jest wyznaczana przez narzędzie Nitro Position Recorder ³⁴ . |

- **composer.phar** (z plikami: **composer.json**, **composer.lock**, folder: **vendor**)

Zarządzanie zależnościami dla projektu PHP (dałem tutaj PHPLint do sprawdzania składni) za pomocą Composer³⁵.

- **class.point.php**

Mój skrypt do definiowania punktu

| | |
|---|---|
| \$point = new classPoint(); | definiuje nowy punkt |
| \$point->setPoint(1,3,4); | ustawia współrzędne punktu: X - 1, Y - 3, Z - 4 |
| \$point->setPointX(1); | ustawia współrzędną X |
| \$point->setPointY(3); | ustawia współrzędną Y |
| \$point->setPointZ(4); | ustawia współrzędną Z |
| \$point->getPointX(); | pobiera wartość współrzędnej X |
| \$point->getPointY(); | pobiera wartość współrzędnej Y |
| \$point->getPointZ(); | pobiera wartość współrzędnej Z |
| \$point->getPoint(); | pobiera punkt do tablicy |
| \$point->setPointFromArray(array(1,3,4)); | ustawia współrzędne punktu z tablicy |

- **corrademonitorservice.php**

Skrypt PHP, który sprawdza dostępność bota.

Wymagane jest zainstalowanie odpowiedniego zadania w harmonogramie zadań w systemie Windows.

- **cron.bat/cron.php**

³⁴ Second Life Marketplace - <https://marketplace.secondlife.com/p/Nitro-Position-Recorder/20479529> lub folder sources w materiałach dodatkowych

³⁵ Strona - <https://getcomposer.org/>

Cron napisany w PHP, który okresowo może wywoływać funkcje dla bota (np. ma coś wykonać w Second Life).

Wymagane jest zainstalowanie odpowiedniego zadania w harmonogramie zadań w systemie Windows.

Tutaj może sprawdzać, czy bot jest u siebie na działce.

- **inPolygon.class.php**

Skrypt który sprawdza, czy punkt znajduje się w wielokącie.

- **logs**

Folder, który zawiera pliki tekstowe z zarejestrowanymi zdarzeniami przez funkcję writeDebugFile().

- **phplint.bat**

Program wsadowy, który sprawdza składnię plików PHP w bieżącym katalogu wraz z podkatalogami.

- **config.php**

Plik konfiguracyjny w PHP używany przez inne pliki PHP, które są częścią mózgu mojego bota.

```
<?php
error_reporting(0);
//Bot Connection
define('BOT_GROUP', 'Bot Group');
define('BOT_PASSWORD', 'PASSWORD');
define('BOT_URL', 'http://127.0.0.1:9199');

//Database Connection
define('DB_CONNECTION', 'mysql:host=localhost;dbname=corrade');
define('DB_LOGIN', 'corrade');
define('DB_PASSWORD', '');

//constants Second Life
define('TEXTURE_DEFAULT', '89556747-24cb-43ed-920b-47caed15465f');
define('TEXTURE_BLANK', '5748decc-f629-461c-9a36-a35a221fe21f');
define('TEXTURE_TRANSPARENT', '8dcd4a48-2d37-4909-9f78-f7a9eb4ef903');
```

| | |
|--------------|---|
| BOT_GROUP | nazwa grupy do której przynależy bot |
| BOT_PASSWORD | hasło do uwierzytelnienia komendy w grupie |
| BOT_URL | URL do bota (taki sam jak w ustawieniach serwera) |

| | |
|---------------------|---|
| | HTTP w pliku konfiguracyjnym bota) |
| DB_CONNECTION | połączenie z bazą danych (o ile używasz) |
| DB_LOGIN | login do bazy danych |
| DB_PASSWORD | hasło do bazy danych |
| TEXTURE_DEFAULT | UUID tekstury domyślnej (przy Second Life zostaw tak jak jest) |
| TEXTURE_BLANK | UUID tekstury pustej (przy Second Life zostaw tak jak jest) |
| TEXTURE_TRANSPARENT | UUID tekstury przezroczystej (przy Second Life zostaw tak jak jest) |

- **functions.php**

Funkcje, które używam przy swoim bocie.

| | |
|--|--|
| writeDebugFile(\$msg, \$type = 'warn') | loguje zdarzenia \$msg do pliku; domyślny typ - 'warn' |
| write_php_ini(\$array, \$file) | zapisuje dane do pliku INI |
| safefilerewrite(\$fileName, \$dataToSave, \$mode = 'w+') | zapisuje dane do pliku w sposób bezpieczny - argumenty: nazwa pliku, dane do zapisu, tryb zapisu (domyślnie nadpisuje) |
| safefilerewriteCache(\$name, \$value) | zapisuje dane do pliku cache w sposób bezpieczny |
| safefilerewriteDeleteCache(\$name) | kasuje wszystko z pliku cache w sposób bezpieczny |
| getConfig(\$nameval) | pobiera wartość z pliku konfiguracyjnego INI o nazwie \$nameval |
| clearConfig(\$section, \$name) | kasuje wartość \$name z sekcji \$section w pliku konfiguracyjnym INI |
| setConfig(\$section, \$name, \$value) | zapisuje wartość \$value dla argumentu \$name w sekcji \$section |
| setCacheValue(\$name, \$value) | zapisuje wartość \$value dla argumentu \$name w pliku cache |

| | |
|--|---|
| getCacheValue(\$name) | pobiera wartość dla argumentu \$name z pliku cache |
| deleteCacheName(\$name) | kasuje wartość dla argumentu \$name z pliku cache |
| deleteAllCache() | kasuje plik cache |
| wasURLEscape(\$param_tab) | funkcja od WAS, która koduje URL |
| SendToBot(\$param_tab, \$ACK = 1, \$connectTimeout = 10, \$Timeout = 30) | wysyła tablicę parametrów \$param_tab do bota; \$ACK - żądanie odesłania odpowiedzi (wartość: 1) lub wyślij i zapomnij (0); \$connectTimeout - czas na połączenie z botem w sekundach; \$Timeout - czas oczekiwania na odpowiedź z bota w sekundach |
| SendMessageToAvatar(\$agent, \$message) | wysyła wiadomość \$message do awatara o podanym UUID \$agent w Second Life |
| wasKeyValueGet(\$key, \$data, \$base64decode = false) | funkcja WAS - pobiera wartość z żądania \$data z klucza \$key; jeżeli dane są zakodowane w base64, może od razu je odkodować \$base64decode |
| CSV2Array(\$csv) | przekształca CSV do tablicy |
| wasCSVToArray(\$csv) | funkcja WAS - przekształca CSV do tablicy |
| wasArrayToCSV(\$a) | funkcja WAS - przekształca tablicę do CSV |
| vectorStringToArray(\$vector) | przekształca wektor w postaci znakowej, np. <1,5,4> do postaci tablicy array(1,5,4) |
| vectorArrayToString(\$vector) | przekształca tablicę wektora do postaci znakowej |
| vectorDistance(\$v1, \$v2) | oblicza odległość pomiędzy wektorami, które są tablicami |
| AvatarNameToUUID(\$name) | próbuję pobrać UUID dla awatara o konkretnej nazwie \$name |
| randomPassword(\$length, \$keyspace = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ')) | losuje hasło o danej długości \$length z podanego zestawu znaków \$keyspace |
| checkPerm(\$avatar, \$func) | sprawdza czy dany awatar z UUID \$avatar |

| | |
|---|---|
| | jest przypisany do konkretnej roli \$func |
| AvatarUUIDtoName(\$uuid) | próbuję pobrać nazwę awatara na podstawie jego UUID \$uuid |
| getBOTUUID() | pobiera UUID bota |
| BlackListIMAdd(\$uuid) | dodaje awatar \$uuid do czarnej listy; awatar nie będzie mógł rozmawiać z botem prywatnie (o ile powiadomienie jest zainstalowane) |
| BlackListIMExist(\$uuid) | sprawdza, czy dany awatar \$uuid został dodany do czarnej listy |
| BlackListIMDel(\$uuid) | usuwa awatara z czarnej listy |
| isValidUuid(\$uuid) | sprawdza czy UUID jest prawidłowy |
| getRegionANDPos(\$slurl) | pobiera region i pozycję z danego SLURL i podaje je w postaci tablicy |
| getCurrentRegion() | pobiera aktualny region, w którym jest awatar |
| RandomNumber(\$min, \$max, \$step = 1, \$float = false, \$places = 5) | losuje liczbę z zakresu od minimalnej \$min do maksymalnej \$max; z możliwością podania kroku \$step; czy losować jako liczbę całkowitą lub zmiennoprzecinkową \$float; z podaną ilością miejsc po przecinku \$places |
| calcObjectBorderPoints(\$point, \$size, \$ratio) | podaje w tablicy punkty brzegowe dla danego obiektu; \$point - środek obiektu; \$size - rozmiar obiektu; \$ratio - obrót obiektu (zobacz temat związany z PosGrid) |
| BotIsActive() | sprawdza czy bot jest aktywny |
| getBotPosition() | podaje aktualną pozycję bota |
| setTitle() | ustawia tytuł botowi |
| posGrid(\$beginPoint, \$size, \$ratio, \$lines, \$endPoint) | zobacz temat: PosGrid |
| isInParcel(\$point = "", \$polygon = "") | sprawdza czy dany punkt \$point znajduje się w wielokącie |

- **corradeMonitorService.bat**

Plik wsadowy, który uruchamia monitor dostępności bota.

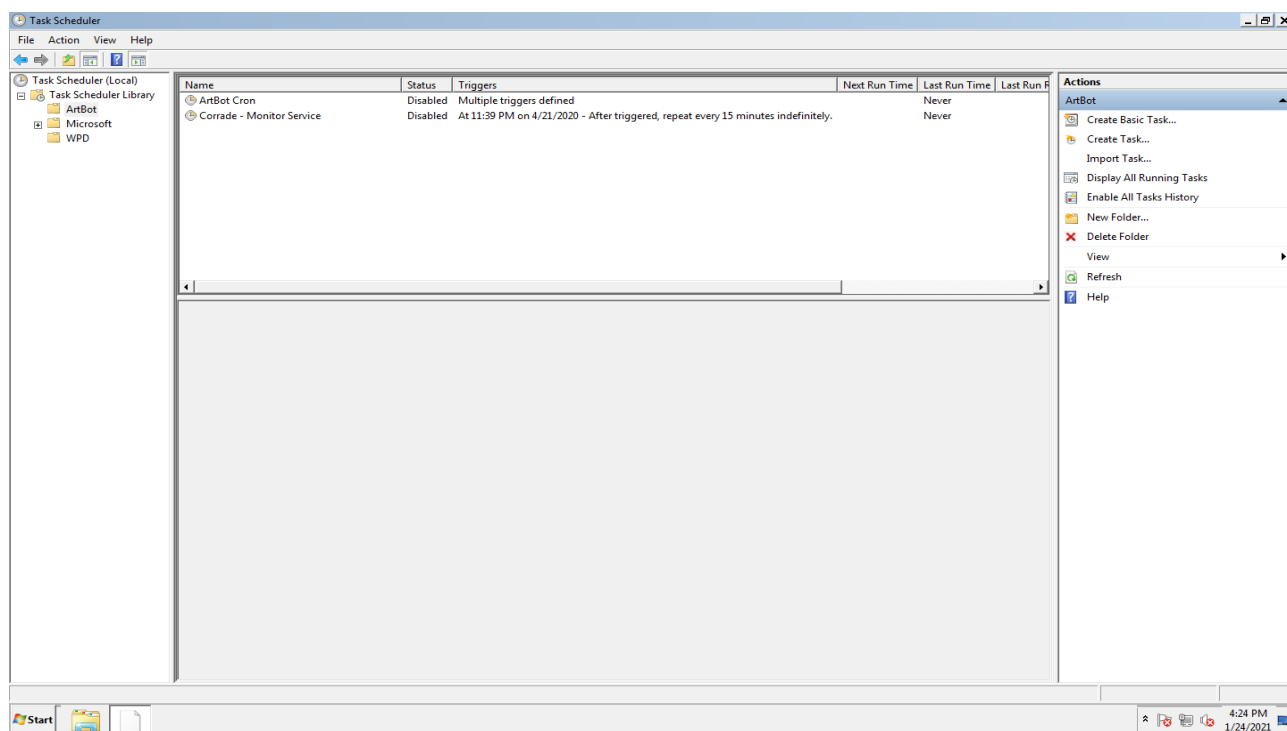
10.5. Harmonogram Zadań

Jeżeli skorzystałeś, np. z mojego autorskiego skryptu do instalacji bota na komputerze, to zostaną dodane 2 zadania do systemowego harmonogramu zadań:

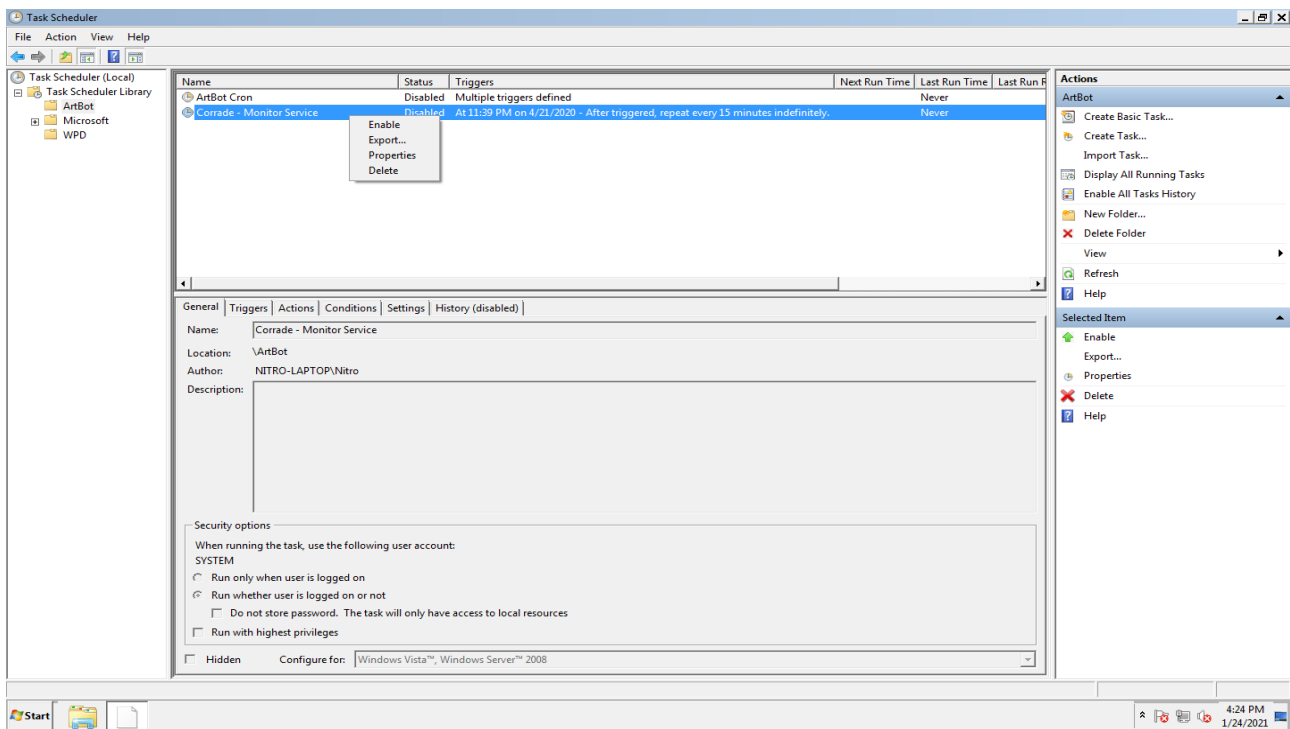
- ArtBot Cron – zadania, które są uruchamiane dla bota co, np. 5-10 minut
- Corrade – Monitor Service – zadanie, które monitoruje proces programu Corrade

Jeżeli chcesz, możesz je zmodyfikować pod siebie.

Pamiętaj, aby zadanie było przypisane do uprawnień użytkownika SYSTEM.



Rysunek 10.3 - Zadania bota w Harmonogramie Zadań systemu Windows



Rysunek 10.4 - Opcje dla zadania