Public, open-source release of a modernized version of OceanViz

Work package 6, Deliverable D6.2, Due M24

Version 1.3, 21 December 2024

Jeroen Steenbeek, Mike Pan (EII)

Document Information and Version Control

Project Acronym	MarinePlan
Project Title	Improved transdisciplinary science for effective ecosystem-based maritime spatial planning and conservation in European Seas
Grant Agreement Number	EU grant agreement No 101059407; UKRI grant numbers 10038951 & 10050537.
Work Package	WP6
Related Task(s) / Milestone(s)	T6.4 / -
Deliverable Number	D6.2
Deliverable Name	Public, open-source release of a modernized version of OceanViz
Due Date	30 September 2024
Date Delivered	23 December 2024
Dissemination Level	Public — fully open (automatically posted online on the Project Results platforms)

Version Control

Revision-N°	Date	Description	Prepared By	Reviewed By
V0	21/11/2024	1st Draft	Jeroen Steenbeek, Mike Pan	
V1	18/12/2024	1st Complete version	Jeroen Steenbeek, Mike Pan	Maria Vaz
V1.2	19/12/2024	1st revision	Jeroen Steenbeek, Mike Pan	Marta Coll
V1.3	21/12/2024	Submitted	Jeroen Steenbeek, Mike Pan	

TABLE OF CONTENTS

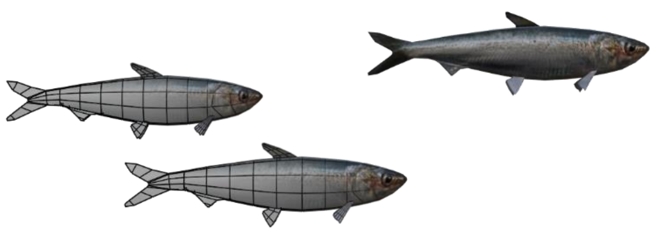
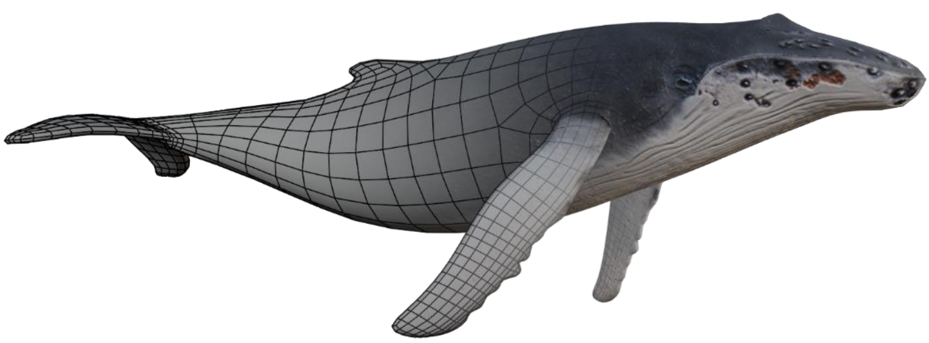


TABLE OF FIGURES

TABLE OF TABLES



EXECUTIVE SUMMARY

The MarinePlan project develops a decision support system (DSS) to promote the implementation of a coherent and adaptive Ecosystem-based marine spatial planning (EB-MSP) framework in Europe, focused on co-development and co-creation of planning actions. The MarinePlan DSS unifies systematic conservation planning tools and marine ecosystem models to explore the trade-offs of planning actions on the marine ecosystems.

Integrating non-experts in the use of complex DSS calls for disseminating scientific data in processed forms. OceanViz is an experimental platform that translates data that quantitatively describes the state of a marine ecosystem into a three-dimensional virtual underwater world and can provide side-by-side views of different states of the ecosystem to convey change caused by, for instance, management interventions (Steenbeek et al., 2021).

The OceanViz approach is a promising tool to engage stakeholders but was built upon now deprecated technologies. Under MarinePlan it was agreed to modernize the OceanViz approach to a common gaming platform, and to share the updated OceanViz approach as an open source and freely available toolkit for visualizing the state of the Ocean.

AIM OF THE DELIVERABLE

This deliverable aims to present the current state of the modernized OceanViz and its capabilities and features, as a programmable engine for visualizing marine ecosystems and how they change over time under influence of climate variability and change, and direct human activities. We also describe the connection points for configuring and interconnecting with OceanViz as an Application Programming Interface (API) and how OceanViz can be further customized and expanded on.

Note that this work occurs in parallel with EU Project NECCTON, which will provide the funding to make the OceanViz system available as an open-source Application Programming Engine (API) and as a visualization toolkit on the European Union Copernicus website.

The new OceanViz will be tested through visualizing the Western Mediterranean Planning Site (PS). The resulting video will be presented separately.

1.1 Contributors

The following people contributed to this deliverable:

Table 1 – Names, affiliations, roles and MarinePlan Planning Site involvement(s) of contributors to this deliverable.

Name	Affiliation	Role	Planning Site
Jeroen Steenbeek	EII	Lead, developer	N/A
Mike Pan	EII	Developer, 3D artist	N/A
Villy Christensen	EII	Scientist, Q&A	N/A
Maria Bas	ICM-CSIC	Scientist, tester	Western Mediterranean
Marta Coll	ICM-CSIC	Scientist, tester	Western Mediterranean
Miquel Ortega	ICM-CSIC	Scientist	Western Mediterranean
Francisco Ramirez	ICM-CSIC	Scientist	Western Mediterranean

2 INTRODUCTION

The aims of the EU Project MarinePlan is to address the need for a cohesive and adaptive Ecosystem-based marine spatial planning framework to aid member states to fulfil the EU Biodiversity Strategy for 2030. Against this background, MarinePlan develops a DSS equipped with tools and guidelines for reconciling conservation and restoration measures with national and transnational MSP applications.

A cornerstone of the MarinePlan DSS is the active involvement of stakeholders in the co-design and operation of the DSS, and to ensure that the ecological implications of specific planning actions can be understood and interpreted by expert and non-expert users alike. Ecosystem models are increasingly used for policy advice (e.g., Heymans et al., 2020), but their operation is challenging for non-experts, while model results represent abstractions of reality in numbers, which are difficult to comprehend by non-specialists. Although complex scientific data communication can be made more efficient through careful design and the use of standardized and informative indicators (Böttinger et al., 2020; Coll and Steenbeek, 2017; Kelleher and Wagener, 2011), there are inherent communication problems that call for the use of a more processed form of scientific findings (Georgescu et al., 2019; Walling and Vaneeckhaute, 2020). The huge advances in technology and software usability brought about by video games have opened a wide range of opportunities for initiatives outside the gaming realm. Video games have become an integral part of modern life and have helped develop and expand different approaches to involving science in decision-making processes, including the area of particular interest: serious gaming (Georgescu et al., 2019; Ritterfeld et al., 2009). In serious games, the primary focus is not purely on amusement, but on exposing the participant to some form of knowledge via explicit educational experiences, multi-media, and entertainment (Laamarti et al., 2014). They harbour multiple elements that lend itself for scientific outreach, communication, stakeholder involvement, and ecosystem management (Madani et al., 2017).

The OceanViz system (Steenbeek et al., 2021) was conceived to focus on communicating scientific data to non-scientific audiences to foster dialogue, offering experimental, immersive approaches to visualizing complex ecosystems in complement to more traditional scientific data visualization approaches. OceanViz includes an extensive visualization toolkit capable of accurately reflecting marine ecosystem changes through a simulated three-dimensional (3D) underwater environment, intended to immerse audiences into animated virtual realities that appeal to the hedonic value of seeing an ecosystem as it is rather than having to interpret the state of an ecosystem through graphs and charts.

In its first instalment, OceanViz was interconnected with the marine ecosystem modelling (MEM) approach "Ecopath with Ecosim" (Christensen and Walters, 2004; Heymans et al., 2016) to visualize the ecosystem impacts of potential management interventions. The OceanViz concept has a much larger potential for science co-creation, stakeholder engagement and public outreach (Steenbeek, 2024), but technical and conceptual limitations, and the reliance on the now deprecated Blender Game Engine, meant that OceanViz needed urgent updating.

The EU project MarinePlan funded this endeavour, while EU project NECCTON will fund the next phase of making OceanViz publicly available as a visualization engine on the web portal of the Copernicus Earth Observation platform. This document details the main decisions taken when updating and modernizing OceanViz from version 2 to version 3.

3 Redesign phylosophy

To unlock its full potential, the OceanViz approach needed to undergo a few major changes:

- Transfer the source code from the deprecated Blender Game Engine to another gaming platform with a projected long lifespan.
- Decouple OceanViz from the EwE approach to facilitate interconnectivity with other data sources (including other models, and MEMs such as EwE).
- Retain the main feature set of OceanViz but reorganize the OceanViz code for extensibility and modularity to facilitate unknown uses and integrations.
- Make the OceanViz publicly available as a freely available and open-source project in support of FAIR principles and Open Science.

4 Implementation

This section will present, in broad terms, the design decisions and development roadmap adhered to when building OceanViz v3. For further details we refer to the online, living design documentation on GitHub at https://github.com/Official-EwE/oceanviz/blob/main/docs.

4.1 Choice of gaming engine

The main 3D gaming engines were reviewed as potential targets for the new OceanViz. Table 2 shows these engines with some evaluation criteria.

Table 2 – 3D gaming engines reviewed.

Gaming engine	License-free use	License-free development	License-free deployment	Programmable via 3rd party software	Long-term availability
Godot 4	Yes (MIT license)	Yes (fully FOSS)	Yes	Yes, supports GDNative, C#, and external libraries	Strong, maintained by open-source community
Unity 2022	Yes, for non-proprietary projects under revenue limits	Yes, but restricted by a proprietary license	Requires licensing for revenue-generating products	Yes, supports integration via plugins, C#, and external tools	Strong with concerns due to the Unity pricing model
Unreal 4	Yes	Yes (source code available, but not FOSS)	Requires royalties if revenue > $1M/year	Yes, extensive APIs, plugins, C++, and Python scripting supported	Likely strong given Epic Games' scale and ongoing updates

Although the three engines are more than capable to cater to the needs of OceanViz, Unity was selected because of its widest uptake and support. However, recent changes in Unity's pricing model have raised concerns for developers in open-source contexts, which meant that the reliance on Unity needed to be kept to a minimum.

4.2 Technical design

The architecture of the modern OceanViz is summarized in Figure 1. We have adopted the following terminology to describe the structure of OceanViz and its components:

- **Entity** is any living thing (flora and fauna) that populates the OceanViz world. An entity is made up of a 3D model, image textures, and a collection of parameters in a JSON file. An entity can be dynamic (e.g. fish) or static (e.g. oysters).
- **Location** represents an environment, which is effectively our 'level', or 'map' in gaming terms. For example, "Western Mediterranean" is a location. The location is conceptually separated from an Entity. It doesn't define or care about the entities which occupy it.
- Within a location, there can be multiple **habitats**, which are smaller regions that have distinct ecological properties. They are currently mostly used to help to define the distribution of entities. i.e. coral fish will only stick to coral reef habitats, seagrass will only grow in sandy habitats, not rocks.
- **Client** is the side of the application that visualizes the data in 3D. In most cases, it's synonymous with OceanViz.
- **Server** is the application that provides the ecosystem data to the client.

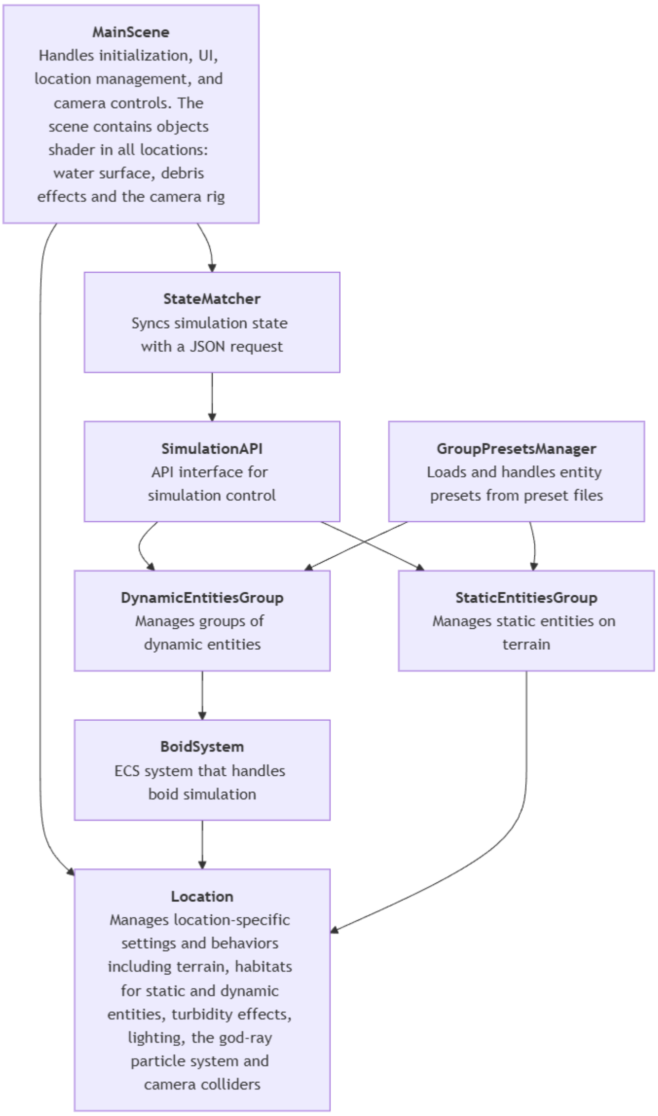


Figure 1 – Conceptual architecture of OceanViz v3.

4.3 Transfer of core features

4.3.1 Technical capabilities

As indicated in Table 3, the transition from OceanViz from Blender to Unity posed an opportunity to mature the code base and to fix technical bottlenecks of the earlier OceanViz versions.

Table 3 – How technical capabilities were implemented in OceanViz versions 2 and 3.

Technical capabilities	OceanViz v2	OceanViz v3
Implementation platform	Blender Game Engine	Unity
Implementation language	Python	C#
Operating system	Windows, can be adapted to Mac and Linux	Native execution on Windows, Mac and Linux
Modularity	Visualization engine coded in Blender environment	Visualization engine is Unity-agnostic, extensible through OOP
Extensibility	Assets and environments coded in Blender environment	Assets and environments can be added without coding
Public availability	Closed-source, accessible via collaboration	Open source (MIT license) via GitHub
Projector compatibility	Monitor, domes, 3D headsets	Monitor(s), domes, 3D headsets
Population dynamics data format	CSV	JSON, CSV
Data interconnectivity	File based, can be delivered via TCP secure sockets (e.g., live MEM)	JSON, can be delivered from anywhere directly, or via a server
Render live data	Yes	Yes
Render scripted movies	Yes	Yes

Choice of platform

The OceanViz logic was ported from Python to C# and was rewritten to work with the modern Unity gaming engine. Unity games can natively run on any popular operating system and a range of projection media such as dome projectors and virtual reality headsets. In addition, the C# programming language facilitates a higher level of sophistication in terms of modularity, extensibility and enterprise-level integration with other systems than Python. However, rewriting a complex 3D visualization software such as OceanViz is a technically complex and costly endeavour, and just to cater to the eventuality that an unforeseen switch to another game engine may be needed in the future, the reliance on Unity was kept to an absolute minimum; all visualization core logic, and the definition of 3D assets and their behaviours, were kept Unity-agnostic.

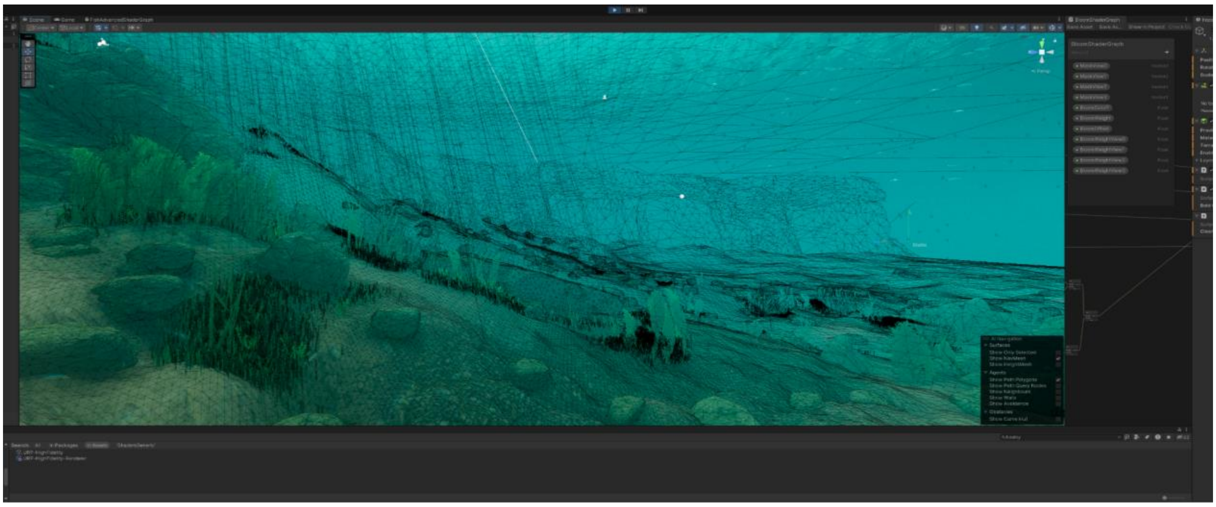


Figure 2 – A OceanViz v3 location while under development, as seen in the Unity IDE.

Extending OceanViz

OceanViz v3 was designed for extensibility via a few simple Object-Oriented coding structures that facilitate overriding or expanding the capabilities of OceanViz. However, as the majority of OceanViz applications will most likely involve showing new environments, flora and fauna, we took care to allow OceanViz to be augmented with new environments and species without the need for writing complex code. OceanViz v3 can dynamically load assets and their interactions, provided as 3D models and an accompanying JSON file that describes the characteristics and preferences of the asset, in a pre-defined folder and file structure (see online documentation).

Interconnecting OceanViz with data

In version 2, changes of the marine ecosystems over time could be brought to OceanViz in two ways: 1) via CSV files of a pre-defined format, and 2) via a TCP socket connection directly connected to the Ecopath with Ecosim (EwE) food web modelling approach software. Although this dual integration was useful to build the first OceanViz prototypes, it posed two severe limitations that would limit a wider uptake of OceanViz:

- The CSV file format was only able to convey ecosystem changes over time but would be unable to carry any other configuration information.
- The inclusion of direct communication with EwE precluded the use of other MEMs.

We decided to make interacting with OceanViz v3 easier and more powerful:

- The CSV file standard was replaced with JSON, which, as a self-describing compound file format, offers the means to convey a much broader amount of information to OceanViz while it initializes and executes.
- The OceanViz client component is made generic and can now be made to connect to a wide range of data servers.

Rendering capabilities

It is imperative that version 3 retains the earlier capabilities of OceanViz to be used for either display of ecosystem dynamics through data streams, but also for the generation of scripted movies.

Whereas a decade ago, the capabilities of the Blender Game Engine and hardware meant that live rendered data at a reasonable frame rate would be visually inferior to the high-quality scripted movies that were rendered by several computers over a several days (Figure 3). These visual differences will be less stark because of the capabilities of Unity, even on computers with mid-range graphical capabilities, in OceanViz v3.

Figure 3 - The difference in visual quality when OceanViz v2 is live connected to EwE (left) or when the v2 engine is used to produce scripted movies (right).

4.3.2 Visualizing the environment

Table 4 summarizes key features how OceanViz renders the environment.

Table 4 – Features related to handling the environment in OceanViz versions 2 and 3.

Virtual environment	OceanViz v2	OceanViz v3
Number of environments	1	Any
Number of entities (flora and fauna)	30	Any
Environments	Hard-coded scenes that can be loaded by restarting OceanViz	Dynamic locations that load on the fly
Ocean floor	Simple	High detail
Habitats	No	Visually distinct, and species-specific importance for flora and fauna
Water turbidity	Volumetric fog, coloured, with particles	Volumetric fog, coloured and blurred, with particles
Water surface	Simple undulations	Realistic waves
General ambience	Post-processing (diffusion, chromatic abberation, color adjustment / attenuation, visual noise)	Post-processing (diffusion, chromatic abberation, color adjustment / attenuation, visual noise)
Split-screen	2-3 states of the same scenario (via shaders)	2-3 ecosystem states, or 2-3 different ecosystems
Camera paths	Fixed	Fixed, user controllable within environmental bounds

Handling environments and entities

OceanViz v2 scenes were dedicated to the display of one single environments, populated with a limited set of entities (flora and fauna). Version 3 addressed these shortcomings by allowing users to visit different areas in any given location, or to switch locations on the fly. A significant improvement has been made to entity management in version 3, where entities (Figure 4) are dynamically loaded only when needed.

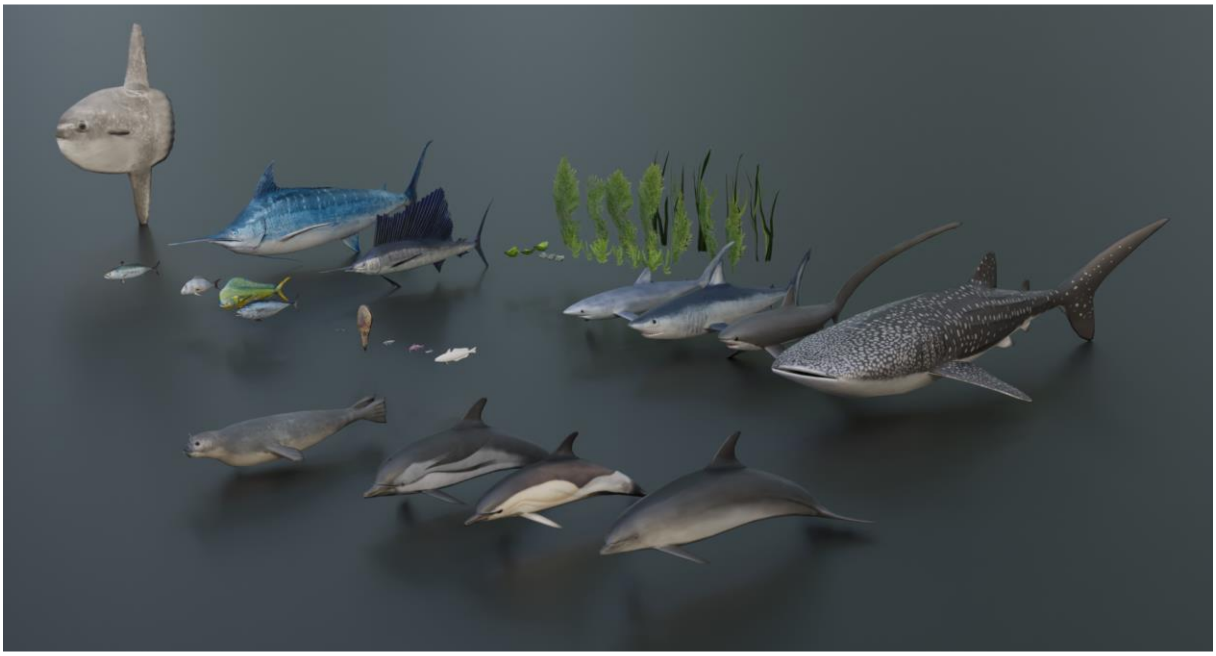


Figure 4 – A small subset of the 3D assets available in OceanViz v3.

Showing the ocean floor

Whereas v2 used a simple ocean floor that was part of the scenery, v3 displays the ocean floor in much higher detail. V3 also introduces the notion of habitats that are visually and functionally distinct (Figure 5).

Figure 5 – An object avoidance experiment in OceanViz showing the detailed bottom. Here, crabs roam the sandy habitat but must avoid the rocks. The avoidance buffer is drawn in red for testing purposes.

Showing the water

The water surface in v3 now shows “Snell’s Window” and renders the moving water surface with much higher fidelity due to Unity’s capabilities. V3 also improves on the display of water clarity and hue, adding blur to the environment to strongly emphasize environmental degradation (Figure 8).

Blur can also be used to represent horizontal thermoclines.

Intercomparing content through split screens

One of the main novelties of OceanViz v2 was the introduction of shader-driven split screens, showing three panels side by side, each showing the state of an ecosystem at a different time, across a continuous landscape. The panel setup provides a compelling visualization technique to demonstrate, for instance, the temporal impact of management decisions. V3 now allows for showing up to four panels and can also show entirely different locations in each panel (Figure 6).

Figure 6 – The three-panel layout in OceanViz v2 (left) and v3 (right), The left panel shows a cinematic rendering of a continuous scene with three time slices in BalticViz. The right panel shows a live rendering of three panels with different locations.

Viewing the entire ecosystem

Viewing an entire ecosystem in one single camera shot is impossible due to the heterogeneity of species distributions and the huge differences in their sizes. OceanViz v2 addressed this by spawning ecosystem entities closer together than in reality, and deployed camera paths that took viewers past the different organisms within a scene.

This functionality was retained in v3.

4.3.3 Visualizing the ecosystem

Table 5 – Features related to reflecting the living ecosystem components in OceanViz.

Ecosystem	OceanViz v2	OceanViz v3
Entity spawning	Spawn where credible (mid-water, near seafloor, at seafloor, rocks, …)	Spawn where credible (depth and habitat-affinity)
Entity management	Keep flora and fauna in FOV, within visible distance	Keep flora and fauna around the user, within visible distance
Entity movement	Probabilistic state machine plus general behaviour	Parameterizable general behaviour
Flocking	Limited	Schooling
Predator-prey avoidance	No	Predator-prey dependence and size-based
Predation events	No	No (and not planned)
Obstacle avoidance	No	Yes
Habitat affinity	No	Species-based
Habitat-based flora	No	Yes
Habitat-based fauna	No	Yes
Fauna hiding	No	Planned (Fauna hiding in cracks and caves, among coral)
Animation of simple entities	Procedural repeating animations	Mathematical undulations with inertia, turn cycles for larger animals
Animation of complex entities	Procedural repeating skeleton animations	Custom animations (no skeletons) with inertia

Spawning and managing entities

OceanViz v2 spawned static flora on specific structures, while fauna was spawned and retained in the vertical space. Lacking a clear definition of species distributions, this system allowed fauna to disperse into environments where species were less likely encountered. In OceanViz v3 we added the notion of habitats, which provide spatial ranges for spawning and retaining habitat-sensitive fauna and flora.

For performance reasons, OceanViz v2 tried to keep the number of entities to a minimum and kept fauna only in front of the camera. This could result odd disappearing and reappearing fish. The better performing Unity engine allows for improvements here by retaining fauna and flora within visible range around the user.

Movement

Although species' movement in OceanViz is generally kept simple, movement will need to look credible at a cursory glance. For undulating entities, we switched from skeleton-based animations to a procedural GPU-based animation system (Figure 7) that can display simultaneously thousands of realistically moving fish at high frame rates. Traditional skeleton animations are only used for more complex animals such as cephalopods, birds and large cetaceans.

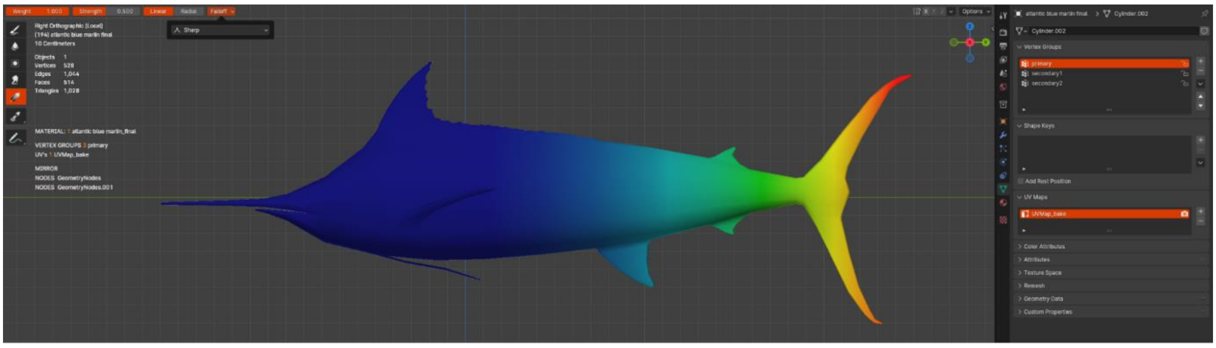


Figure 7 – Assigning priority regions for 3D mesh movement, which indicate the relative strenght of mathematical undulations. This system is much less demanding to compute than traditional animation techniques.

Behaviour

OceanViz v3 contains various animal behaviours that can be combined where necessary: Boid-based schooling, object avoidance, and avoidance of large predators typically apply simultaneous on fish.

We deliberately omitted predation events from OceanViz to avoid distracting carnage.

Under MarinePlan we still plan to expand predator avoidance behaviours by having species hide in the terrain.

4.3.4 Adding light

Light sources and the distribution of light (Table 6) is essential for complementing underwater scenes, and to subliminally convey key properties related to depth, proximity to the coast, and overall ecosystem health.

Table 6 – Light effects in OceanViz versions 2 and 3.

Lighting	OceanViz v2	OceanViz v3
Sun position	Static (light source is fixed)	Dynamic, light rays penetration
Caustics	Ocean floor and entities	Ocean floor and entities
Light attenuation	Depth and distance	Depth and distance
User light source	No	Camera mounted light source
Entity light sources	No	Planned (bioluminescence)

Light sources

OceanViz v3 contains two light sources: the sun and a camera-mounted light source. A third type of light source, bioluminescence, is planned for inclusion under MarinePlan.

Currently, the position of the sun can be set via the JSON control information sent to OceanViz. Sunlight can create light rays that penetrate the surface, and can reflect on particles.

Underwater light effects

In shallow areas, caustics made by the water surface can be shown on the sea bottom and entities (Figure 8). At the time of writing, caustisc are not influenced by objects overhead that may cast shadows, which is something that will need fixing in the future.

In deeper areas, or when viewing objects at a larger distance, shortwave light frequencies are exponentially absorbed by the water, resulting in recoloured and darker imagery. The strength of light attenuation can be configured in OceanViz v3.

Figure 8 – An example now OceanViz v3 renders the water surface as seen from below. This screenshot showcases Snell's Window, seafloor caustics, and water turbidity, greenish colour and blur typical for coastal water.

4.3.5 Features fur future funding opportunities

There are, of course, plenty of features that would add to the realism and utility of OceanViz that we cannot address within the time frame and budget of MarinePlan. Key suggested features are shown in Table 7. This list is by no means definite.

Table 7 – Feature wish list for future funding opportunities.

Follow-up project wishlist	OceanViz v2	OceanViz v3
Procedural location generation	No	High priority
Structures (turbines, etc.)	No	High priority
Fishing pressure	Limited	High priority
Shadows	No	Medium priority
Coastal representation	No	Medium priority
Garbage and litter, wrecks	Limited	Low priority
Aquaculture	No	Low priority
Above water visualizations	No	TBD
User interactivity	Yes	TBD
Living habitats	No	TBD

Procedural environment generation

To make OceanViz operational around the world, it will need the capability to generate locations from data. Factors such as depth, proximity to coast, latitude and longitude, present habitats, proximity to the coast and rivers, and general knowledge about the distributions of flora and fauna can all contribute to the ability to automatically generate plausible locations in OceanViz.

Although we will not be able to add this feature under MarinePlan, its future development is already provisioned. For instance:

- The notion of habitats can be used to create credible benthic environments in a location.
- We also have started a systematic inventory of global species that can be mapped to proxy assets when a specific species 3D asset has not been developed yet.

For procedural generation of locations, widely available high-definition habitat distributions and depth maps (see EMODnet, NOAA, etc.) will be invaluable information sources.

Structures

With the increasing uptake of Systematic Conservation Planning and Marine Spatial Planning, there is a need to visualize various man-made structures, either on the bottom (cables, pylons, oil rigs, etc) or suspended overhead (floating renewable energy structures).

Structures may provide habitat to certain species and may visually affect the quality of the water.

Fishing pressure

OceanViz can be augmented with the notion of fishing pressure. OceanViz v2 would draw the bottoms of boats but the rendering of the various types of fishing gear and the interactions of fishing gear with flora, fauna and the benthos were not included.

Aquaculture

In a similar vein, aquaculture sites can be included. As for structures, aquaculture sites may also provide habitat to specific species and may also visually affect the quality of the water.

Shadows

Including underwater shadows may be challenging to implement because of caustics and refraction; shadows are typically blurry at best for objects close to a surface. Although shadows add realism, the computational cost of drawing credible shadows may not be worth the effort. This will need to be explored under future funding.

Coastal representation

Because of higher abundance of nutrients due to wave action, run-off and river transport, coastal areas contain more phytoplankton than the very clear mid-ocean waters. Coastal waters are noticeably greener or bluish-green, and less transparent, than water further from the coast. Whereas coastal proximity can be mimicked per a location (Figure 8), coastal proximity rendering should be automated especially for procedurally generated environments.

Garbage, litter and shipwrecks

People are messy, and people mess should be included in OceanViz especially for visualizations that focus on pollution. Shipwrecks and artificial reefs can also complete a specific scene (Figure 9).

Figure 9 – An example of a shipwreck in OceanViz v2. Ambient light effects and volumetric fog effectively completed the scene.

Above water visualizations

Allowing users to venture out of the water may be desired for future OceanViz applications. The scope of this exercise will need to be decided when dedicated funding becomes available.

User interactivity

To make OceanViz useful as an exploratory tool, the ability to select entities as was present in OceanViz v2 could be brought back. Especially the ability to drag schools of fish between time slice panels can function as a key mechanism to convey the ecological impacts of future climate and fisheries scenarios.

Living habitats

Habitat forming species, such as corals, seagrass and oysters, grow in habitats but in turn provide shelter for other species. This mechanism could be visualized automatically in a future version of OceanViz.

4.4 Implementation roadmap

The feature list above was implemented using the Agile approach, and with Trello boards to track and prioritize the development.

We implemented OceanViz via the following internal development tasks (Figure 10):

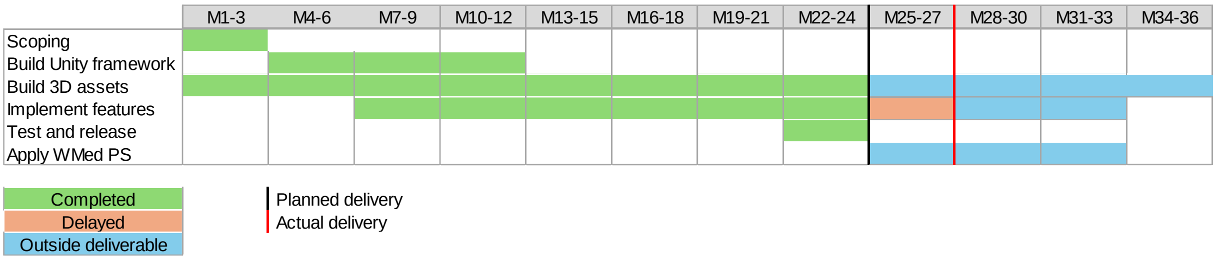


Figure 10 – Internal OceanViz v3 development and application timeline.

Create OceanViz product requirement and define scope of project (Month 1)

This stage was largely completed while writing the MarinePlan proposal and was rapidly confirmed and finalized in September 2022, when MarinePlan launched.

Build core 3D media assets needed for the visualization (Month 1-24, *)

At the start of the OceanViz v3 development we ported over all existing media assets from OceanViz v2 to v3.

Halfway the MarinePlan project we started collaborating with the team of ICM-CSIC (Barcelona), we complemented the list of assets for visualizing the Western Mediterranean case study.

In December 2024 we embarked on a strategic partnership with the NOAA team behind their VES-V viewer, a spin-off of the original OceanViz approach. The VES-V team has provided free access to their expansive library of 3D assets, which are now being incorporated into OceanViz v3 while MarinePlan funding permits.

Build OceanViz using the Unity Game Engine (Month 3-12)

The first year of the MarinePlan project was dedicated to setting up OceanViz in the Unity Game Engine.

Implement OceanViz v2 features in v3 (Month 6-24, delayed)

While the Unity version of OceanViz was taking shape, we started researching how the various features of OceanViz v2 could be best ported over to OceanViz v3. This is where most effort was needed. It is important to realize that very few 3D modelling efforts have focused on the underwater world, which meant that there was limited publicly available experience that could guide the process. Here, we ran into a number of unforeseen bottlenecks where specific game development approaches just did not work for underwater visualizations. Finding creative solutions and testing what would work best took much more time than anticipated, which is the main factor that this deliverable was delayed.

Allow for future extensibility by providing plug-in points and modularity (free)

This requirement was automatically fulfilled by the technical capabilities of Unity.

Create processes and tools for working with OceanViz as a framework (free)

As part of the next step, protocols to develop and work with OceanViz have been provided with the source code documentation on GitHub. As outlined in this documentation, the Blender software is still the best tool for generating 3D assets which can then be ported to use within the Unity framework with minimal effort. The Unity IDE is the best tool for working with the OceanViz code.

Testing, QA, and API documentation (Month 24-36)

The remainder of the MarinePlan project will be used to test OceanViz and apply it to the Western Mediterranean case study.

The newly developed OceanViz has been deployed on GitHub as per December 2024, and this deployment includes the API documentation. The GitHub repository is public and can be found at https://github.com/Official-EwE/oceanviz/.

Apply OceanViz v3 (Month 25-33)

Last, as an extended test cycle, the new OceanViz v3 will be applied to the Western Mediterranean Planning Site to convey key findings of the EwE application under MarinePlan. This work will be presented separately from this deliverable.

5 BIBLIOGRAPHY

Böttinger, M., Kostis, H.-N., Velez-Rojas, M., Rheingans, P., Ynnerman, A., 2020. Reflections on Visualization for Broad Audiences, in: Foundations of Data Visualization. Springer, pp. 297–305.

Christensen, V., Walters, C.J., 2004. Ecopath with Ecosim: methods, capabilities and limitations. Ecol. Model. 172, 109–139. https://doi.org/10.1016/j.ecolmodel.2003.09.003

Coll, M., Steenbeek, J., 2017. Standardized ecological indicators to assess aquatic food webs: The ECOIND software plug-in for Ecopath with Ecosim models. Environ. Model. Softw. 89, 120–130. https://doi.org/10.1016/j.envsoft.2016.12.004

Georgescu, A., Gheorghe, A.V., Piso, M.-I., Katina, P.F., 2019. Serious Gaming and Policy Gaming, in: Georgescu, A., Gheorghe, A.V., Piso, M.-I., Katina, P.F. (Eds.), Critical Space Infrastructures: Risk, Resilience and Complexity, Topics in Safety, Risk, Reliability and Quality. Springer International Publishing, Cham, pp. 215–226. https://doi.org/10.1007/978-3-030-12604-9_8

Heymans, J.J., Bundy, A., Christensen, V., Coll, M., de Mutsert, K., Fulton, E.A., Piroddi, C., Shin, Y.-J., Steenbeek, J., Travers-Trolet, M., 2020. The Ocean Decade: A True Ecosystem Modeling Challenge. Front. Mar. Sci. 7. https://doi.org/10.3389/fmars.2020.554573

Heymans, J.J., Coll, M., Link, J.S., Mackinson, S., Steenbeek, J., Christensen, V., 2016. Best practice in Ecopath with Ecosim food-web models for ecosystem-based management. Ecol. Model. 331, 173–184. https://doi.org/10.1016/j.ecolmodel.2015.12.007

Kelleher, C., Wagener, T., 2011. Ten guidelines for effective data visualization in scientific publications. Environ. Model. Softw. 26, 822–827. https://doi.org/10.1016/j.envsoft.2010.12.006

Laamarti, F., Eid, M., El Saddik, A., 2014. An Overview of Serious Games. Int. J. Comput. Games Technol. 2014, 358152. https://doi.org/10.1155/2014/358152

Madani, K., Pierce, T.W., Mirchi, A., 2017. Serious games on environmental management. Sustain. Cities Soc. 29, 1–11. https://doi.org/10.1016/j.scs.2016.11.007

Ritterfeld, U., Cody, M., Vorderer, P. (Eds.), 2009. Serious games: Mechanisms and effects. Routledge, New York.

Steenbeek, J., 2024. Ecosystem modelling for the ocean decade - facing the challenges (Doctoral thesis). TDX Tesis Dr. En Xarxa. Universitat Politècnica de Catalunya. https://doi.org/10.5821/dissertation-2117-417801

Steenbeek, J., Felinto, D., Pan, M., Buszowski, J., Christensen, V., 2021. Using Gaming Technology to Explore and Visualize Management Impacts on Marine Ecosystems. Front. Mar. Sci. 8. https://doi.org/10.3389/fmars.2021.619541

Walling, E., Vaneeckhaute, C., 2020. Developing successful environmental decision support systems: Challenges and best practices. J. Environ. Manage. 264, 110513. https://doi.org/10.1016/j.jenvman.2020.110513

6 LIST OF ABBRIVATIONS

Bioluminescence	Emitting of light by living organisms
Blender	3D computer graphics software
Bender game engine	Deprecated software for making 3D interactive content
Boids	A computer algorithm that mimics the flocking behaviour of birds
C#	A popular programming language
Caustics	Rippling concentration of light patterns on the sea bottom, caused by light passing through the water above
Cross-platform	Software that runs on different operating systems
EB-MSP	Ecosystem-based marine spatial planning
Ecosystem-based marine spatial planning	Marine Spatial Planning that considers both the health of the ecosystem and human uses
FOSS	Free and Open-Source Software
FOV	Field-of-view, the area visible by the user in a 3D environment
FPS	Frames-per-second, see Frame rate
Frame rate	The rate of images per second that can be produced, with higher frame rates perceived as smoother animations
IDE	Integrated Development Environment, a software for application development
JSON	JavaScript Object Notation, an open standard for self-describing data files that are human and machine legible
Marine Ecosystem Model	A mechanistic model that represents the dynamics between biotic and abiotic factors in a marine ecosystem over time / time and space
Marine Spatial Planning	A public process of analysing and allocating the spatial and temporal distribution of human activities in marine areas
MEM	Marine Ecosystem Model
MSP	Marine Spatial Planning
OOP	Object Oriented Programming
Operating system	Software that manages computer hardware
Planning Site	A location studied by MarinePlan (e.g., a case study area)
PS	Planning Site
Snell's Window	The limit circle of surface detail visible from under water, caused by refraction of light entering the water
TBD	To be decided
Thermocline	A boundary between horizontal water layers of different temperatures. Thermoclines are blurry in appearance due to the different density of water at different temperatures
Unity	A cross-platform game engine