

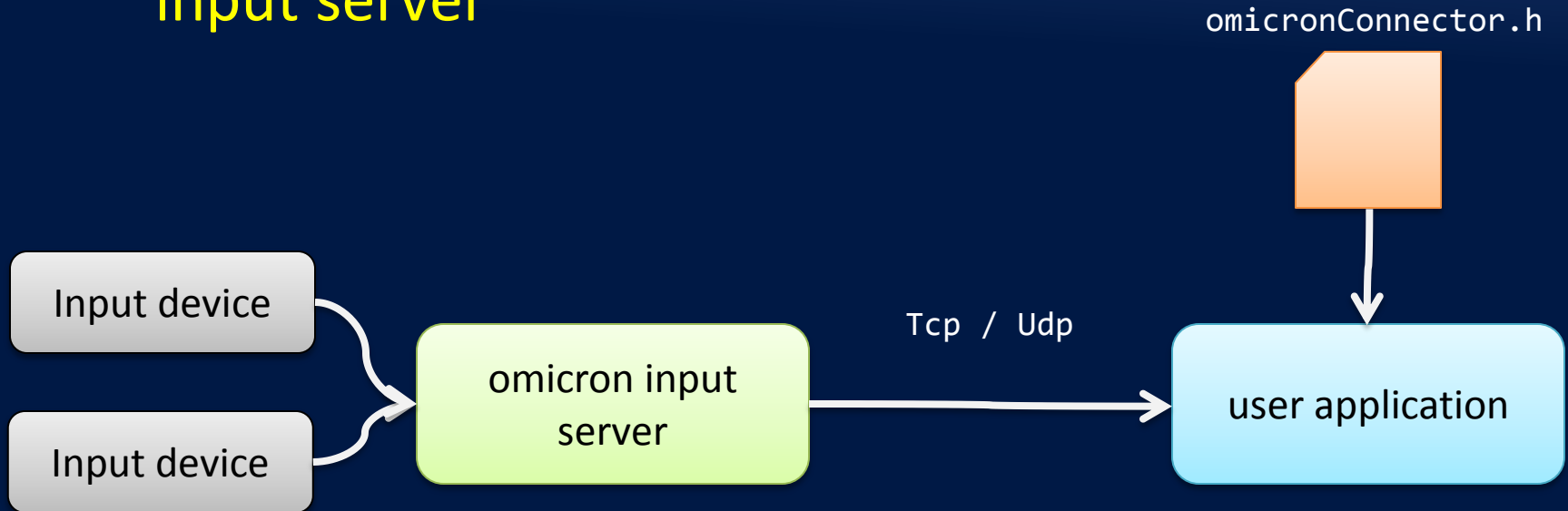
introduction to omegalib and omicron

Alessandro Febretti

Electronic Visualization Lab – University of Illinois at Chicago

omicron vs omegalib

- **omicron**: input management library
 - Can be integrated directly as library or used as **input server**



omicronConnector example

```
#include <connector/omicronConnectorClient.h>
```

```
using namespace omicronConnector;
```

```
int main(int argc, char** argv)
```

```
{
```

```
    OmicronConnectorClient<ConnectorListener> client;
```

```
    client.connect("127.0.0.1");
```

```
    while(true) client.poll();
```

```
}
```

```
class ConnectorListener
```

```
{
```

```
public:
```

```
    static void onEvent(const EventData& e) { printf("Received event!\n"); }
```

```
};
```

omicron events

- omicron events designed as generic input data containers
 - information about event source
 - service class (wand, gamepad, pointer, ui)
 - Services of same class have similar event generation semantics
 - event type, position, orientation, binary flags
 - variable length extra data
 - can be float array, vector3 array, generic char*
- more information: <http://code.google.com/p/omicron-sdk/wiki/EventReference>

omicron event services

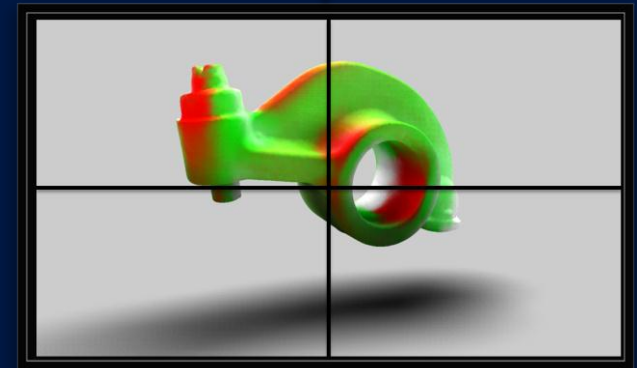
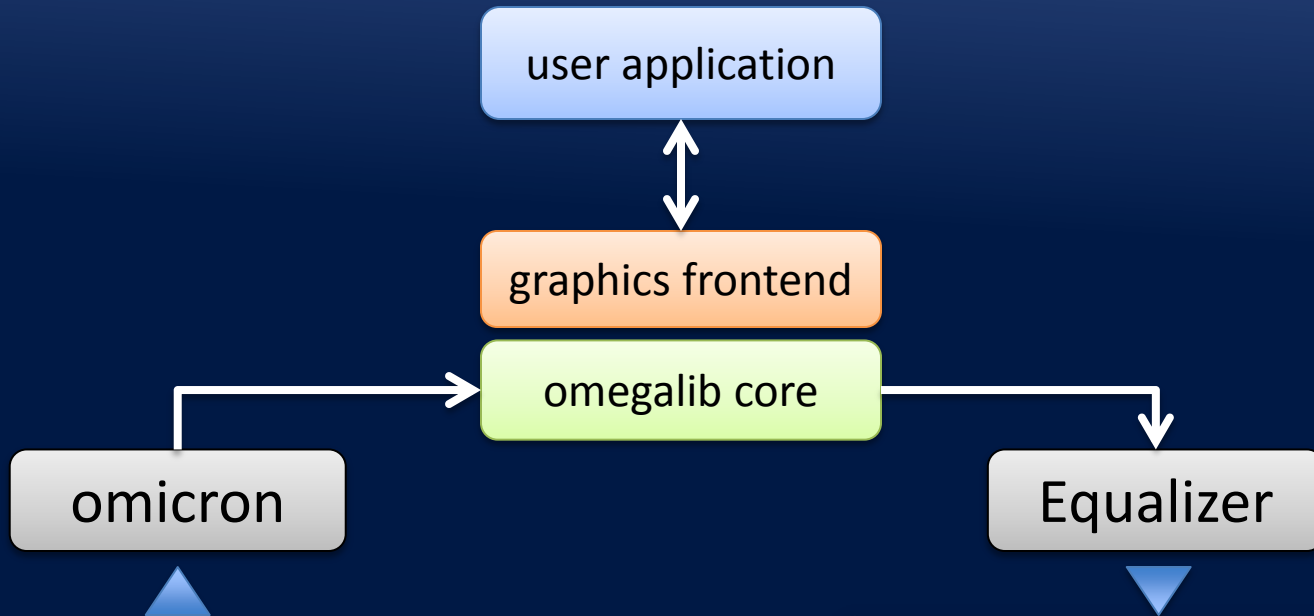
- event service = source of omicron events
- some built-in event services:
 - VRPN-supported trackers
 - Optitrack native
 - XBox360, PS3, Wiimote controllers
 - PQLabs touch overlays
 - Kinect skeletons
 - ThinkGear
 - iPad app (prototype)
 - SAGE Pointer

omicron utilities

- omicron also offers utility APIs for app developers (used by omegalib)
 - config file support ([libconfig](#))
 - xml reading/writing ([tinyxml](#))
 - multithreading ([tinythreads](#))
 - tcp client/server API ([asio](#))
 - math library ([Eigen](#))



VR toolkit built on top of **omicron** (input) and **Equalizer** (display)

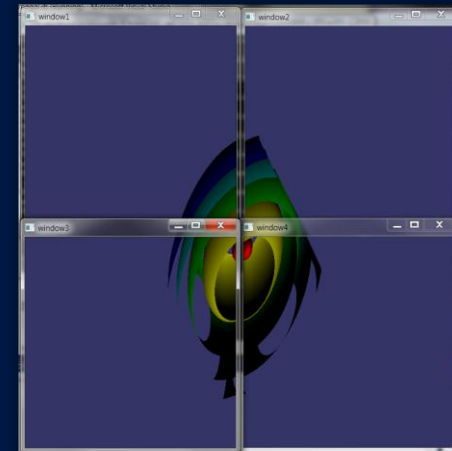
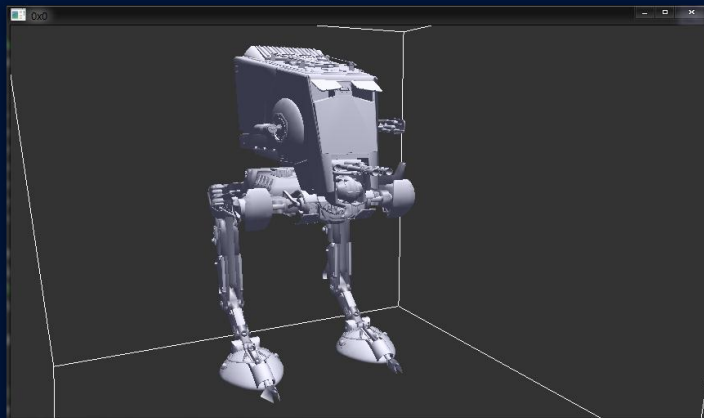
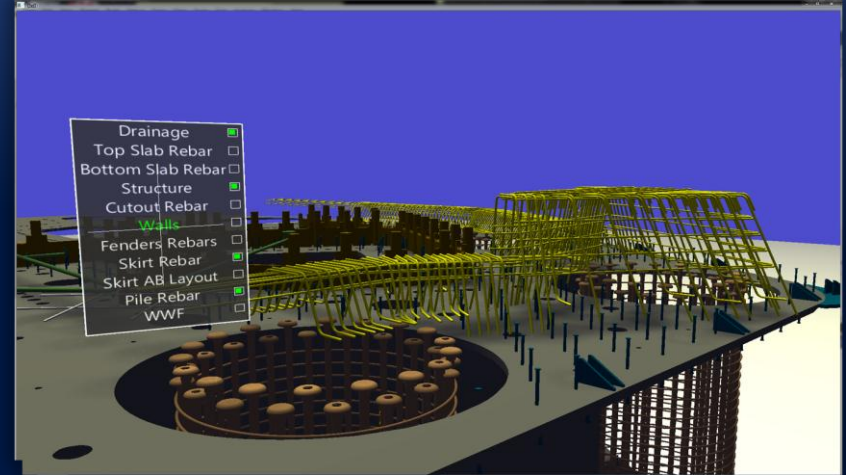




omegalib core = support backend for different **graphics frontends**

Currently supported frontends:

- **OpenSceneGraph**
- **Vtk**
- **Plain OpenGL**





some legacy applications can be ported to omegalib

(<https://code.google.com/p/omegadoom/>)





- Research direction: **multi-application VR**
 - run multiple applications inside separate viewports / windows (a la SAGE)
 - Allow applications to switch to ‘fullscreen mode’ and take over entire display
 - **Challenges:**
 - Load balancing (equalizer helps a bit)
 - Interaction

That's all folks!

- Links:
 - omicron: <http://code.google.com/p/omicron-sdk/>
 - omegalib: <https://code.google.com/p/omegalib/>
 - Equalizer: <http://www.equalizergraphics.com/>