Senior Project 2013, Fall

GAME ENGINE

Student: Robert Law, Florida International University
Mentor: *Wei Zeng*, *Assistant Professor*
Instructor: Masoud Sadjadi, Florida International University

FIU School of Computing & Information Sciences

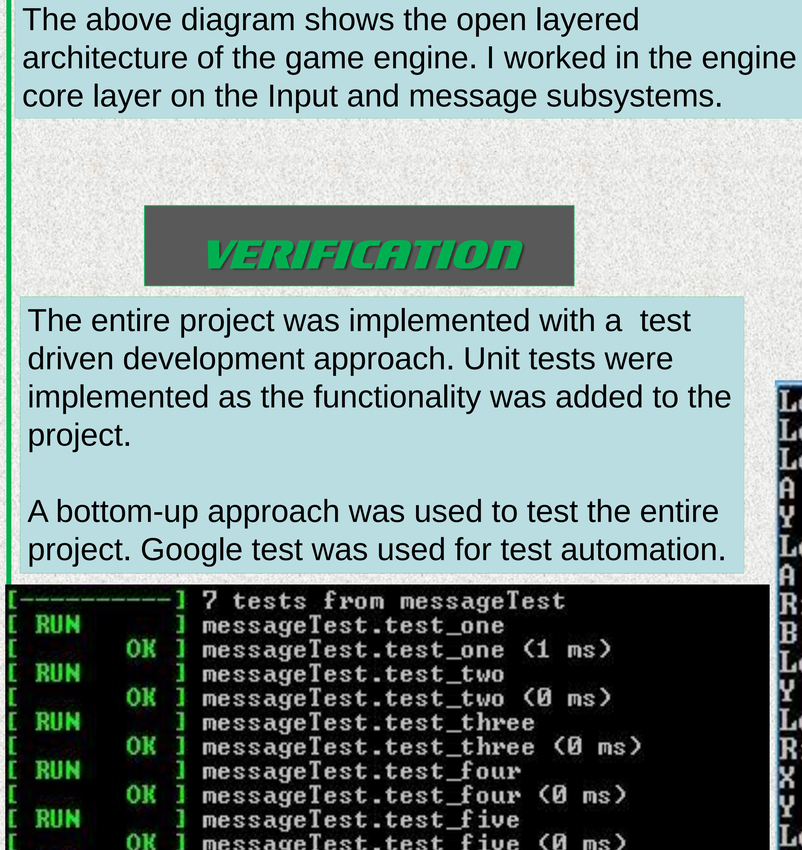
OVERVIEW

- Current multiplatform game engines are complicated for new developers to use and lack the ability to be fully customized and experimented on.

PROBLEM

- A modern multiplatform game engine needs a input handling subsystem that is latency free and easy to work with.
- A game engine also needs a messaging system that allows subsystems to notify each other whenever the state of a subsystem is changed.

SYSTEM DESIGN



The above diagram shows the open layered architecture of the game engine. I worked in the engine core layer on the Input and message subsystems.

VERIFICATION

The entire project was implemented with a test driven development approach. Unit tests were implemented as the functionality was added to the project.

A bottom-up approach was used to test the entire project. Google test was used for test automation.

```
[----------] 7 tests from messageTest
[ RUN      ] messageTest.test_one
[       OK ] messageTest.test_one (1 ms)
[ RUN      ] messageTest.test_two
[       OK ] messageTest.test_two (0 ms)
[ RUN      ] messageTest.test_three
[       OK ] messageTest.test_three (0 ms)
[ RUN      ] messageTest.test_four
[       OK ] messageTest.test_four (0 ms)
[ RUN      ] messageTest.test_five
[       OK ] messageTest.test_five (0 ms)
[ RUN      ] messageTest.test_six
[       OK ] messageTest.test_six (0 ms)
[ RUN      ] messageTest.test_seven
[       OK ] messageTest.test_seven (0 ms)
[----------] 7 tests from messageTest (6 ms total)
```

```
[----------] Global test environment tear-down
[==========] 58 tests from 16 test cases ran. (172122 ms total)
[  PASSED  ] 58 tests.
```

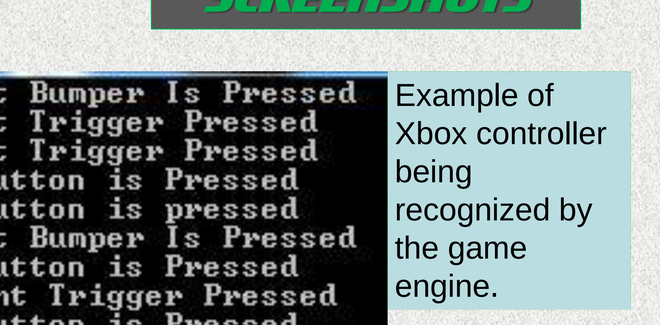
CURRENT SYSTEM

Ogre 3D is an open source rendering engine. While it provides extensive rendering functionality, It does not provide many of the other featured For game design.

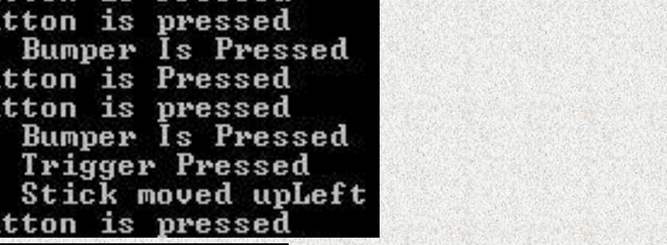
Moai provides and open source game engine that allows the user to create the game with Lua. Unfortunately, this is the only way a user may create a game. By forcing the user to only use Lua paired with the unique architecture makes it impractical for larger scale games.

The Unreal Engine is a very popular game engine used throughout the industry. However, its very high cost typically make it an unreasonable choice for smaller studios.

OBJECT DESIGN



The design of the user input handler is a series of Booleans that return true if keys are pressed. Along with methods to control the mouse.



The design of the user input handler is a series of Booleans that return true if keys are pressed. Along with methods to control the mouse.

SCREENSHOTS

```
Left Bumper Is Pressed
Left Trigger Pressed
Left Trigger Pressed
A Button is Pressed
Y button is pressed
Left Bumper Is Pressed
A Button is Pressed
Right Trigger Pressed
B Button is Pressed
Left Trigger Pressed
Y button is pressed
Left Bumper Is Pressed
Right Trigger Pressed
X button is Pressed
Y button is pressed
Left Bumper Is Pressed
A Button is Pressed
Y button is pressed
Left Bumper Is Pressed
Left Trigger Pressed
Left Stick moved upLeft
Y button is pressed
```

Example of Xbox controller being recognized by the game engine.

REQUIREMENTS

The system shall provide the ability to render 2D/3D graphics, these graphics will be represented by assets like models textures, and shaders.

The system will provide a way to initialize and control the application using a core game loop that will be maintained by the use.

The system shall implement an asset management subsystem which will allow data to be loaded onto game engine memory in an efficient manner.

The system shall provide an engine configuration subsystem which will allow retrieval, storage and modification of engine configuration variables.

The system shall include a debug printing and logging subsystem that will display a data log in a text file that include errors and messages for the user.

The system shall provide an audio system that can play, loop, and fade audio files synchronously with on screen animation.

The system shall contain a gameplay foundation subsystem which will provide the ability to manage and update the state of all core system objects within the game world.

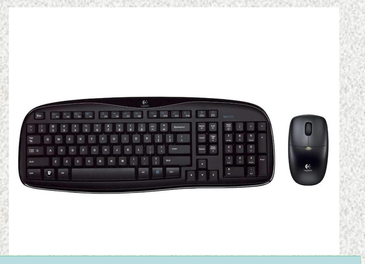
The requirements that relate to my role on the project are:

The system shall provide a user input handling subsystem which will allow handling of current keys and mouse buttons the mapping of keys to functions.

The system shall provide a massaging system which will allow a way for subsystems to send messages to multiple listeners across the engine every time the notify Observers method is called.

IMPLEMENTATION

The message system was created to allow a subject of any class send a message of any type to an observer that is also of any class. This allows any subsystem to implement the subject or observer class. This also allows the messaging system to be adaptable to any future class a user might create. This was done using C++ Anytypes.



The input handling system was created to allow fast raw input directly from the I/O devices. The system was created with platform independence in mind. SFML was used to help achieve platform independence. It provides methods to directly access devices connected to the computer such as a keyboard or joystick.

Methods were written to also allow joystick support. Specifically the Microsoft Xbox controller. The system is user friendly and allows a user to execute functions or any action after a key/joystick state has been changed.

SUMMARY

The game engine supports all functionality required to make a game.

The system allows support for mouse and keyboard as well as PC joysticks in an easy to use and latency free design.

The engine has a fast easy to use message system that can be used with any class/subsystem. And can send and receive an update of any type.

ACKNOWLEDGEMENT

This project was improved thanks to the help I received from my group members Brian Lara, Michael Weschler, Antonio Diaz, and Julian Nodarse