**Battle Zone**

**PA11**

**By: Hannah Munoz – CS 480**

**Connor Scully-Allison - CS 680**

**Kurt Andersen – CS 480**

**Sections:**

Overview

Dependency

User Manual

Build Instructions

Keyboard Inputs

Game Controls

Camera Controls

Shader Controls

Figures

Technical Manual

Issues

Things We Would Have Done Differently

Changes

**Overview:**

**Dependency Instructions:**

* For both of the operating systems to run this project installation of these three programs are required [GLEW](http://glew.sourceforge.net/), [GLM](http://glm.g-truc.net/0.9.7/index.html), and [SDL2](<https://wiki.libsdl.org/Tutorials>).
* This project uses OpenGL 3.3. Some computers, such as virtual machines in the ECC, cannot run this version. In in order to run OpenGL 2.7 follow the instructions at [Using OpenGL 2.7]( <https://github.com/HPC-Vis/computer-graphics/wiki/Using-OpenGL-2.7>)
* This project uses Assimp 3.2. Instructions for downloading and running Assimp can be found at [Main Downloads](http://www.assimp.org/main\_downloads.html)
* This project uses ImagicMagick 6.8.9-9. Instructions for downloading and running ImageMagick can be found at [Install Source] (<http://www.imagemagick.org/script/install-> source.php)
* This project uses Bullet 2.86. Instructions for downloading and running Bullet can be found at [Releases](<https://github.com/bulletphysics/bullet3/releases>)

**Battle Zone**

**Technical Manual**

**By: Hannah Munoz**

**Connor Scully-Allison**

**Kurt Andersen**

**Issues:**

When continuously rotating the camera, the tank increases in speed due to the increased application of torque to the object. Also, the camera begins to veer away from the center of the tank while rotating, but snaps back to its correct position once the tank stops rotating.

**What would we have done differently?**

Our biggest issue when designing the project was coming up with a simple way to rotate the tanks around an axis when the ‘a’ or ‘d’ key were pressed. This required conversion between Euler angles and quaternions. After brushing up on our trigonometry we were able to solve this problem, but then we need the tank to know which direction was the new forward. While the solution did not end up being difficult to implement, the process that went into calculating the angles took us a bit of time. We should have considered how to rotate objects before we begun the project.

Our second issue was getting the enemy tanks to face a tank with in firing range. Once again, without knowing which way was forwards for the tank, it was difficult to get tanks to attack on another successfully.

**Battle Zone**

**User Manual**

**By: Hannah Munoz**

**Connor Scully-Allison**

**Kurt Andersen**

**Build Instuctions:**

To run Battle Zone, ensure that you are in the proper build directory [PA11/build] and compile the code as follows.

cd build

cmake ..

make

To begin playing Battle Zone, enter the following command:

./ BattleZone

A new window should appear with the game ready to play. The score will be displayed in the terminal once the game has ended and the user has run out of lives.

If you make a top 10 score you will be prompted to input your name at the end of the game in the terminal.

**Keyboard Inputs**:

*Game Controls:*

W : Forwards

A : Left

S: Backwards

D: Right

Esc – Quits the game.

*Shader Controls:*

*Numpad:*

'0' – increases specularity on land

'.' - decreases specularity on land

'1' – increases skybox specularity.

'2' – decreases skybox specularity

'4' – increases ball specularity. [Fig.6]

'5' – decrease ball specularity

'7' – increases flipper specularity. [Fig. 7]

'8' – decrease flipper specularity

'9' – increase spotlight height [Fig. 8]

'6' – decrease spotlight height.

'+' - increase ambient lighting. [Fig. 10]

'-' - decrease ambient lighting.

'\*' - increase spotlight ambient lighting. [Fig.11]

‘'/' - decrease spotlight ambient lighting.

**Figures:**