Computer Graphics Assignment 4

Joshua Chapman a1607942
Shasheen Ediriweera a1212249

This program has been tested and verified on multiple systems running Ubuntu version 12.04 and 14.04.

The program is ran by first typing ‘make’, then ‘./kartBattle’. There are two players driving cars.

To exit this program the Esc key can be used at anytime.

The ‘W’,‘A’,‘S’,‘D’ keys move Player 1 around. The ‘E’ key shoots P1’s weapon.
The ‘I’,‘J’,‘K’,‘L’ keys move Player 2 around. The ‘U’ key shoots P2’s weapon.

The cars have to be moving forwards or backwards in-order for the cars to turn left or right.

The objective of the game is to fire a ball to hit the other player (to gain points)

Josh worked on the multi-key input system, the lighting and map objects, such as the walls and floor.

Shasheen worked on the foundational framework of the program, and then game mechanics such as collision detection.

Early on Shasheen decided to split the monolithic UsingNateRobbin’s object reader into separate files and abstracted away the object drawing to functions. This really made our code simple and easy to work with early on in the project.

A multi-key system was required to allow both cars to move at the same time. The previous key function that we were familiar with only allowed one key to be recognised at a time. By creating 2 new functions, on key press and on key release I was able to create an array that stored the values in real time to see if they were currently being pressed. Then all that was required was a function to loop through all possible commands and execute the response.

Two lights have been placed in the center of this map to allow lighting to the player’s cars, as the cars approach the edge of the map the darker they will become. Extra lights can be added by increasing the noumberOfLights variable and creating the appropriate light to place within.

For collision, we decided to use a bounding box for each Entity. Every cycle, we would check for any collisions between the bounding box and the location of the other object. When the weapon collides with a player, it simply bounces off at a 45 degree angle.

Due to lack of time, we haven’t done a full collision detection implementation by checking BOTH bounding boxes and bouncing the ball at a correct angle (angle of incidence equal to angle of reflection). The 45 degree bounce actually gives a very nice looking outcome given its simplicity

- particularly both players weapons bouncing against each other.

A simple scoring mechanism has been implemented, but there is as yet no timer between points, so points get added for every collision on the car, in which sometime multiple collisions are detected before the ball leaves the car’s area.

We had some trouble finding objects to use within our game in that most object did not come with a .mtl file or appropriate .jpg images. This limited our options and we eventually chose a soccer ball. This ball has detail but it is hard to see since the object is coloured solid black.