**CS 174A Final Project - Cube Field**

**Introduction**

Our project is a modern recreation of the classic arcade game Cube Field. The goal of the game is to navigate through the field of cubes without a collision and survive as long as possible. The player moves left with the semi-colon key and right with the apostrophe key. The User's current score and high score are displayed in the bottom left of the screen.

**Player Object**

The Player is generated using a for loop and the insert\_transformed\_copy\_into() function. Within each iteration of the loop, a cube that is scaled by one tenth in the z direction is stacked onto the previously added scaled cube. This results in the pyramid shape resembling an arrow that represents the player. The player is given a random color every new round and has the default maximum ambient lighting to make it easier for the user to see the player.

**Obstacle Object**

Each obstacle is a simple unit cube. The cubes are randomly generated with a positive z axis value of 60. As time passes, the cubes' z value decrements to give the user the impression that they are moving forward through the field of cubes. Each obstacle has zero ambient lighting, minimal diffusivity, and maximum specularity. This gives the user the impression that there is very low lighting in the game field and creates an intense reflection off the near side faces of the cubes. The overall effect is similar to a car's headlight where you can primarily see only what's in front of the player object. This will be discussed further in the Camera and Lights section.

The obstacle generation is managed within the manage\_obstacles() and add\_obstacle() methods. In the manage\_obstacles() function, an obstacle is added if enough time has elapsed since the last obstacle was added. This interval was determined through trial and error to make the game most playable. As the user survives longer, the interval is decreased which results in objects being created more frequently to increase difficulty. The initial value on the x axis (left to right with respect to the player) is generated randomly between -100 and 100 of the player's current x value. This results in an even creation of obstacles that always generate with enough spread to prevent users from moving quickly to the side to essentially outrun the generation of obstacles altogether. Once an obstacle's z value indicates it has travelled off the screen behind the player, it is deleted from the array which holds all the obstacle transforms. The add\_obstacle() method simply pushes a new transform onto the array.

The funnel shape at the beginning of each round is simply a collection of obstacles generated with two for loops. Each loops creates one side of the funnel. The funnel behaves just like the other obstacles whereas the player cannot make contact with the wall without dying. The purpose of the funnel is to signal the start of the round and allow the user to get a feel for the player's movement characteristics before entering the obstacle field.

**Camera and Lights**

The camera transform and single light source move in unison with the player along the x axis to create the headlight effect described above. This effect was added to make the game more interesting and difficult as moving very quickly to the left and right results in a greater potential of running into an obstacle that is very difficult to see. The camera always looks straight down the positive z axis with a small y axis position to give the user a slightly raised perspective over the obstacle field.

**Explosion Object**

**Physics - Player Movement**

**Collision Detection**

**Member Contributions**

August - Initial setup of the game and files

Player, Obstacle, Funnel, Camera, and Light objects

Very basic Player, Camera, Light, and Obstacle movement

Random Obstacle creation

Presentation Video

Ambereesh - Physics related to Player movement

Addition of Start screen vs. Actively playing screen

Improved obstacle management including deleting off-screen obstacles

Status Messages on Control Panel

Collision Detection

Explosion Object and Custom Shader

Both members contributed equally to the report.