COSC363 Assignment 1 – Synchronized Coupled Animation

Name: Michael Woodard

Code: mdw85

ID: 24643695

Contents

1.0 Scene Description

The scene consists of four balls (sphere) bouncing on top of raised platforms. As these balls are bouncing a torus attached to a cuboid (a scaled cube) rotates at a 40-degree angle from the y-axis. Each ball's movement is out of time with each other in such a way that at the top of each ball's motion the torus' hole moves perfectly through the ball. The four platforms the balls bounce on are each modelled as a sweep surface. The scene also contains a skybox. The skybox, balls and platforms are all textured. An overall view of the scene can be seen in figure 1 on the next page:

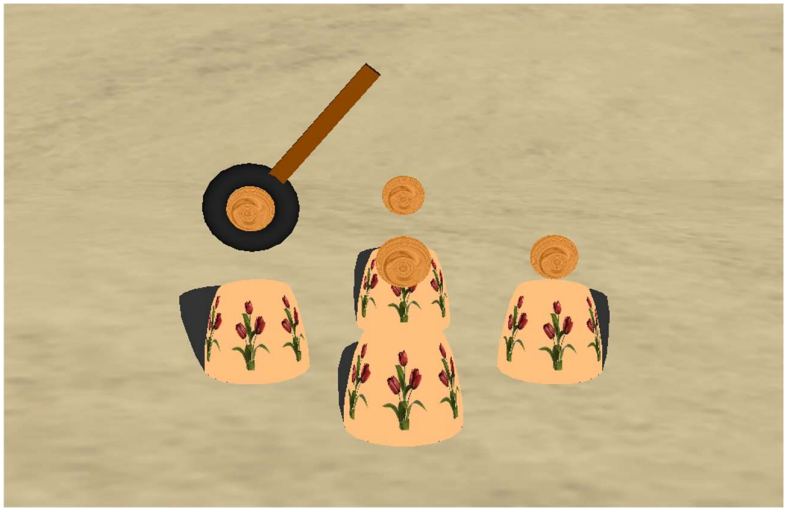

Figure 1: General view of scene.

2.0 Extra Features

2.1 Planar Shadows

All four platforms have planar shadows cast on them. A better view of the shadows can be seen in figure 2 below:

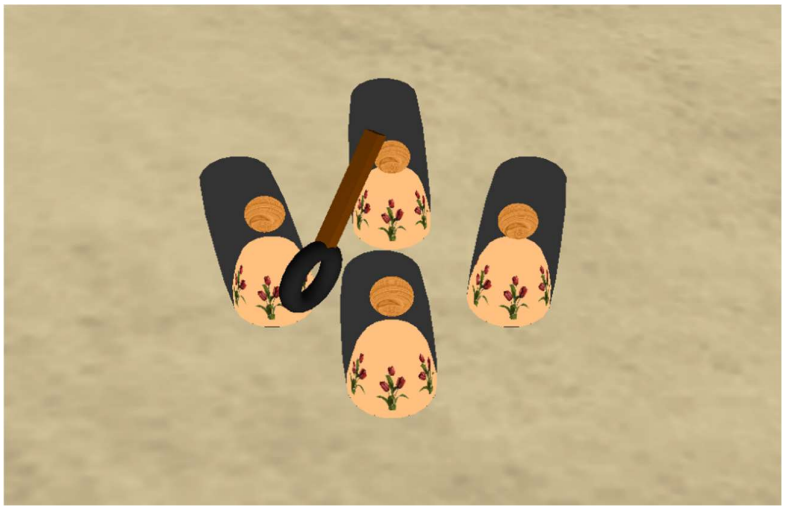

Figure 2: Top-down view of scene to show shadows.

2.2 Physics Based Animation

The ball's movement is under the influence of gravity, however their collision with the platform is elastic. Since the collision is elastic, the ball should rise to the same height each bounce and should also experience no loss of energy (e.g. dure to air resistance or the collision with the platform). Since the only force acting on the ball is gravity, the balls resultant force can be given as:

$$Fres = ma = g$$

Since the objects mass can be ignored in this case, the ball's constant acceleration can be given as:

$$a = -g$$

Where g is negative since the gravitational force is acting downwards. When the ball is released its initial velocity u is 0. As the ball descends to the platform, it's velocity changes by:

$$v = u + at$$

Since the ball moves every time the timer iterates (which is constant each time), we give time t a constant value of 1 unit of time. This means by the equation above that as the ball descends to the ground it accelerates in the downwards direction. When the ball contacts the ground, momentum is conserved given that it is an elastic collision. Therefore, the ball's velocity changes at the point of contact with the ground:

$$v = -u$$

Where u is the velocity before the collision and v is the velocity immediately after the collision. Because the velocity of the ball is now positive, as the ball travels upward the force of gravity acts against the ball. This causes the ball to decelerate until its velocity reaches 0, when the cycle repeats. Since the collision is elastic, the ball will have reached a maximum height which is equal to the initial height at which the ball was dropped at. This means the ball will bounce infinitely.

2.3 Collision Detection

A collision detection system has been set up for the skybox (i.e. the camera cannot leave the skybox). All objects in the scene react as expected when they contact each other (e.g. the ball bounces off the platform). The balls go perfectly through the torus, so no collision detection is needed there.

2.4 Sky Box

A skybox is used in this scene, and a better view of it can be seen in figure 3 below:

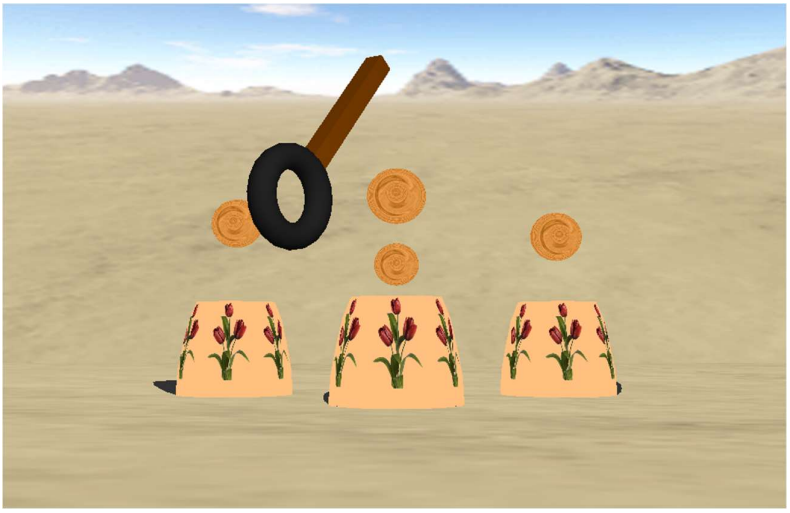

Figure 3: Bottom-up view of scene to show skybox.

2.5 CMakeLists.txt

A proper CMakeLists.txt file is included in the .zip folder that builds the executable from the source code, along with the other submitted files. (See Section 4.2)

3.0 Control Functions

The following controls can be used to interact with the scene:

- Up arrow key: Moves the camera forward.
- Down arrow key: Moves the camera backwards.
- Right arrow key: Shifts the camera at an angle to the right.
- Left arrow key: Shifts the camera at an angle to the left.
- Page up key: Moves the camera up.
- Page down key: Moves the camera down.
- Home key: Shifts the camera at an angle upwards.
- End key: Shifts the camera at an angle downwards.

4.0 Build Commands/Instructions

4.1 Command Line

To compile and run the program from the command line, please follow these steps:

1. Extract the .zip folder provided to your chosen directory.
2. Open the command line inside the extracted folder.
3. Run the following command: 'g++ -o main main.cpp -lGL -lGLU -lglut'. This will produce an output file 'main' inside the extracted folder.
4. To run the program, run the command: './main'

4.2 Qtcreator

To compile and run the program from the command line, please follow these steps:

1. Extract the .zip folder provided to your chosen directory.
2. Open Qtcreator
3. Go to: File →Open File or Project... →Navigate to the extracted folder →select CMakeLists.txt
4. Select the Desktop kit →Configure Project
5. Press the green button in the bottom left to use the CMakeLists.txt to run the program.