Bouncing Shapes Game

CS 4600 Graphics Final Project Report

Video Link: https://youtu.be/wAxenvtCl-U

Michael Wright

Unity 3D Android Game

I created a simple Android game using Unity 3D. The game consists of cubes and spheres falling from above the screen in a 3D scene. There are some objects in the middle of the scene that shapes bounce off of and they also bounce off of each other. The goal of the game is to tap shapes to get the associated point value. Cubes are worth 1 point, while spheres are worth -1 points. If any cube falls off the screen before a player is able to tap it, this also results in -1 points. No penalty or reward happens when a sphere falls off the screen.

Set Up

Learning Unity for Android took much longer than anticipated. I initially thought Unity was just a plugin to Visual Studio and that I'd need Xamarin in combination with Unity to make an Android app. In the end, I wasn't able to use my Android experience and I had to learn the basics of Unity from scratch. That being said, once it was all set up and I had done a tutorial, Unity greatly simplified the process.

Technical Details

I created cube and sphere prefab gameObjects that I instantiate in a Coroutine in the GameManager that waits between 0.4 and 1.0 seconds before generating a new Cube and Sphere. The shapes are instantiated above the screen and are constantly being generated, creating a waterfall effect. The shapes have both randomly generated colors and starting positions. Both cubes and shapes are destroyed when they fall off the screen so no memory leaks occur. I also applied materials and physics to the cube and sphere objects so they would bounce realistically when colliding with each other or with the crossbars in the middle of the screen.

I keep track of the score in the GameManager script. When the shapes are tapped or destroyed by falling off the screen the individual shape scripts call the ChangeScore(int) function in the GameManager Script. Managing objects being destroyed was challenging and forced me to learn how to debug scripts in Unity to prevent null errors.

The game doesn't end, but I did create a “New” button that will restart the game from the beginning. This might be used when people's score becomes negative and they want to start over from the beginning.

Summary

Creating an Android game from scratch by using a program I've never used before was more challenging than I expected. That being said, I'm really pleased with the results. I considered using OpenGl in a native Android application, but there is no way I could have figured out the physics and lighting that Unity provided. It was exciting to see how graphics are generated and used in a modern engine and I plan on continuing to develop and eventually release the game.

References

Unity 2D Basics - https://www.youtube.com/watch?v=h_cUMpXb9QY

How to destroy game objects that fall of the screen - https://answers.unity.com/questions/1230388/how-to-destroy-object-after-it-moves-out-of-screen.html

Creating primitive shapes - https://docs.unity3d.com/ScriptReference/GameObject.CreatePrimitive.html

Timers/Coroutines - https://forum.unity.com/threads/c-timer-event.127920/

Prefabs, how to avoid having a copy of a game object floating off the screen to copy- https://docs.unity3d.com/Manual/InstantiatingPrefabs.html

Coroutines - https://docs.unity3d.com/Manual/Coroutines.html

Random in Unity - https://docs.unity3d.com/ScriptReference/Random.Range.html

Changing text in UI object - https://answers.unity.com/questions/777335/46-ui-changing-the-text-component-via-script.html

Random Color generation - https://docs.unity3d.com/ScriptReference/Random.ColorHSV.html