Assignment 1 – Apple Picker

Due Date: Thursday January 31st 2019

In this lab, you will create a working game, and the end result will be something that you can show to friends and colleagues as shown in Figure 1. After letting them play it for a while, you can ask whether the difficulty feels too easy, too difficult, or just right. Use that information to tweak the variables in the game and custom craft a specific difficulty for each of them.

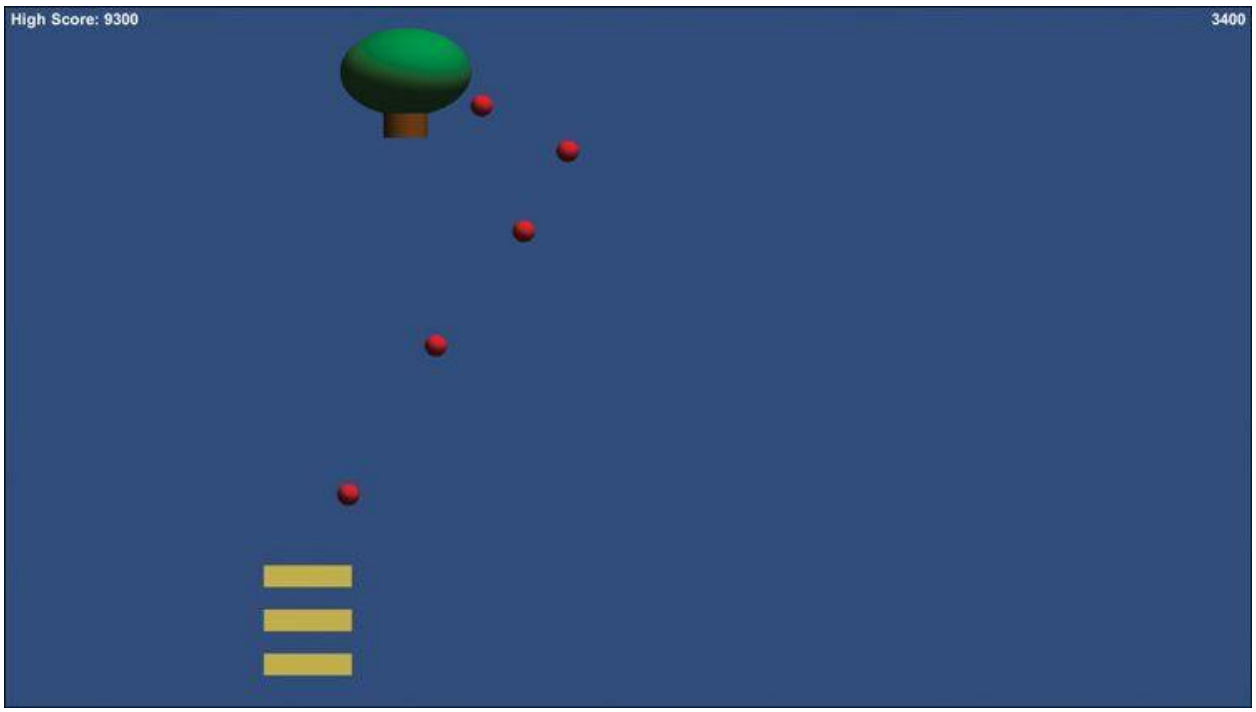

Figure 1

Apple Picker *Basic Gameplay*

The player controls the three baskets at the bottom of the screen and is able to move them left and right using the mouse. The apple tree moves back and forth rapidly while dropping apples, and the player must catch the apples using her baskets before they hit the ground. For each apple that the player catches, she earns points, but if even a single apple hits the ground, it and all other remaining apples will disappear, and the player will lose a basket. When the player loses all three baskets, the game is over.

Apple Picker *GameObjects*

In Unity terminology, any object in the game—usually meaning anything that you see on screen is a GameObject. We can also use this term in discussing the elements seen in Figure 1. For later consistency with our Unity projects, I will capitalize the name of all GameObjects (e.g., Apples, Baskets, and AppleTree) in the following list.

A. Baskets: Controlled by the player, Baskets move left and right following the player's mouse movements. When a Basket hits an Apple, the Apple is caught, and the player gains points.

B. Apples: The Apples are dropped by the AppleTree and fall straight down. If an Apple collides with any of the three Baskets, it is caught and disappears from the screen (granting the player some points). If an Apple hits the bottom of the play window, it disappears, and it causes all other Apples on screen to disappear as well. This destroys one of the Baskets (starting at the bottom), and then the AppleTree starts dropping Apples again.

C. AppleTree: The AppleTree moves left and right randomly while dropping Apples. The Apples are dropped at a regular interval, so the only randomness in the behavior is the left and right movement.

Set Up the Project

Following the standard project setup procedure, create a new project in Unity.

- **Project name:** yourUWindsorID_ApplePickerProject (e.g., ouda_ApplePickerProject)
- **Scene name:** _Scene_0 (The underscore at the beginning of the scene name will keep it sorted at the top of the Project pane.)
- **C# script names:** ApplePicker, Apple, AppleTree, and Basket

Attach the ApplePicker C# script to the Main Camera in _Scene_0. Do not attach the C# scripts Apple, AppleTree, or Basket to anything.

Preparing

You need to do some preparation for this Lab by reading Chapter-16 "Thinking in Digital Systems," in the textbook "Introduction to Game Design, Prototyping, and Development".

Lab Implementation

For more details and specifications please read Chapter 28, in the textbook "Introduction to Game Design, Prototyping, and Development". You need to implements all the steps mentioned in Chapter 28. Chapter 28 is the reference for this Lab.

Submitting

You need to ZIP the Unity project (folder that has the same name as the project name) and submit it electronically through the Black-Board System.

Assignment1 Assessment Rubric

Criteria	Points
Set up the project; e.g., project name, scene name, c# script names, materials for the project such as apple and basket, adding and organize needed folders, camera setup, setting GameObject layers, etc.	25 points
Did you implement the lab mentioned requirements (Coding the Apple Picker Prototype)? e.g., basic movement, making your game time-based, changing direction randomly, dropping apples, stopping apples if they fall too far, instantiating and moving the baskets using the mouse, catching and destroying apples, managing and displaying the score, etc.	40 point
How efficiently do you handle the creation and execution of the methods and statements in the action script?	10 points
Are all of the required elements present and working correctly?	25 points
Total	100 points