CS 3160 – Game Programming

Project 3 – Physics Utilities

Learning Objectives

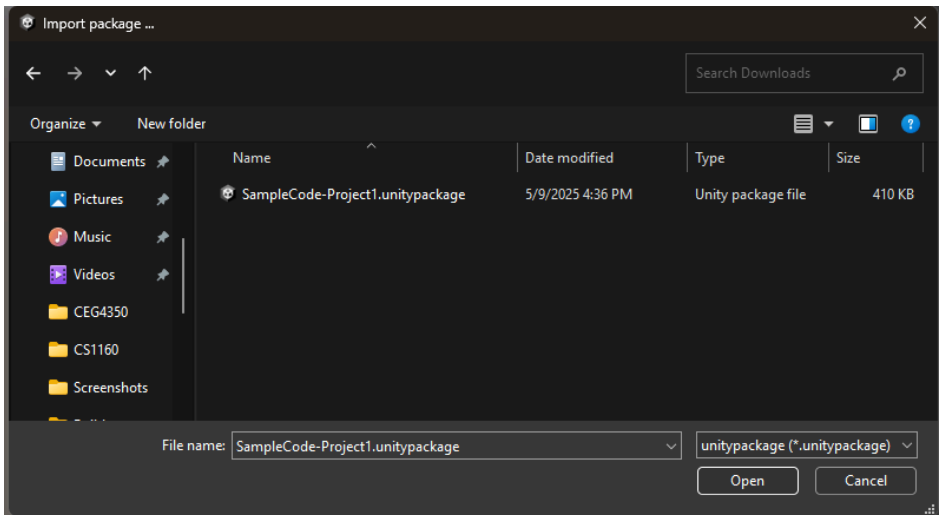
- Learn how to use raycasts, overlaps, and ragdolls.

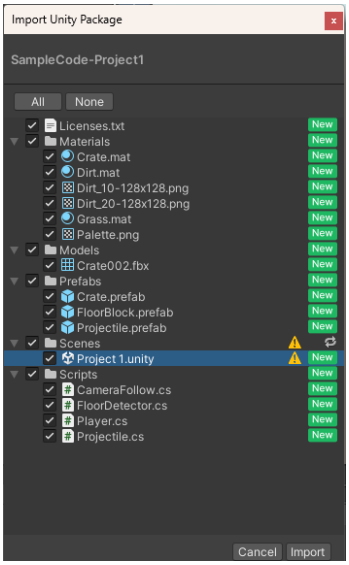
Overview

In this project you will become familiar with performing raycasts, overlapping spheres, and creating ragdolls in Unity.

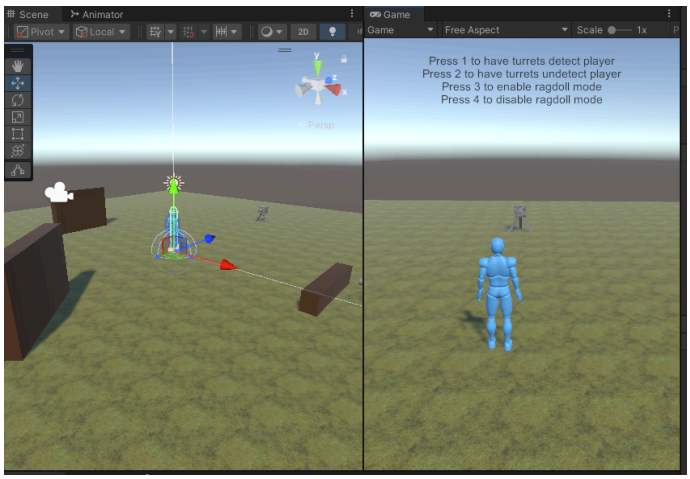
Sample Code

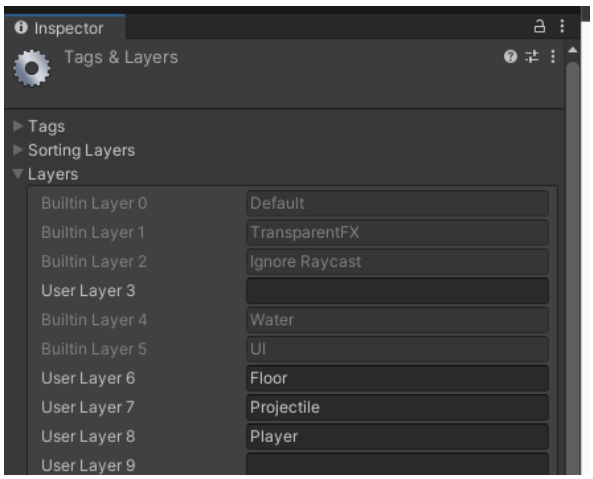
To complete this project some sample code is required. This is provided in the form of a Unity Package. Download the “SampleCode-Project3” Unity Package from Pilot. First, create an empty Unity project using the “3D (Built-In Render Pipeline)” template. Then, go to “Assets” > “Import Package” > “Custom Package…” then locate “SampleCode-Project3.unitypackage” downloaded from Pilot and import all files:





If importing was successful you should be able to load the “Project 3” Scene in the “Scenes” folder and you can test the game and walk forward:





NOTE: Importing a Unity Package does not import Layers. The layers used are listed above.

Turret Controller Script

A script is provided to you for controlling the turrets in the demo called TurretController.cs. The turrets must be capable of tracking the player when the player gets near and when there are no obstacles in between the player and the turret. You must accomplish this rewriting the private void CheckForPlayer() method:

```
1 reference
private void CheckForPlayer()
{
    // Your code goes here
}
```

CheckForPlayer() must do 3 things:

- Perform an “OverlapSphere” at the position of the turret using the provided “detectionRadius”.
- Iterate through all collisions until a collision is found that has an attached “Player” script (the player)
- Perform a raycast from the barrelEnd position of the turret in the direction of the detected player
    - If the raycast detects an obstacle in between the barrelEnd and the player, call the “PlayerIsNotDetected()” method provided
    - If the raycast detects no obstacle in between the barrelEnd and the player, call the “PlayerIsDetected()” method provided

Player Script

We are going to allow the player to become a ragdoll at the push of a button and then transform back into a controllable animated model. The secret behind this will be instantiating a ragdoll GameObject prefab, and then disabling the animated player model. This will be accomplished using 2 methods, private void EnableRagdoll() and private void DisableRagdoll():

```
1 reference
private void EnableRagdoll()
{
    // Your code goes here
    ragdollIsEnabled = true;
}
```

```
1 reference
private void DisableRagdoll()
{
    // Your code goes here
    ragdollIsEnabled = false;
}
```

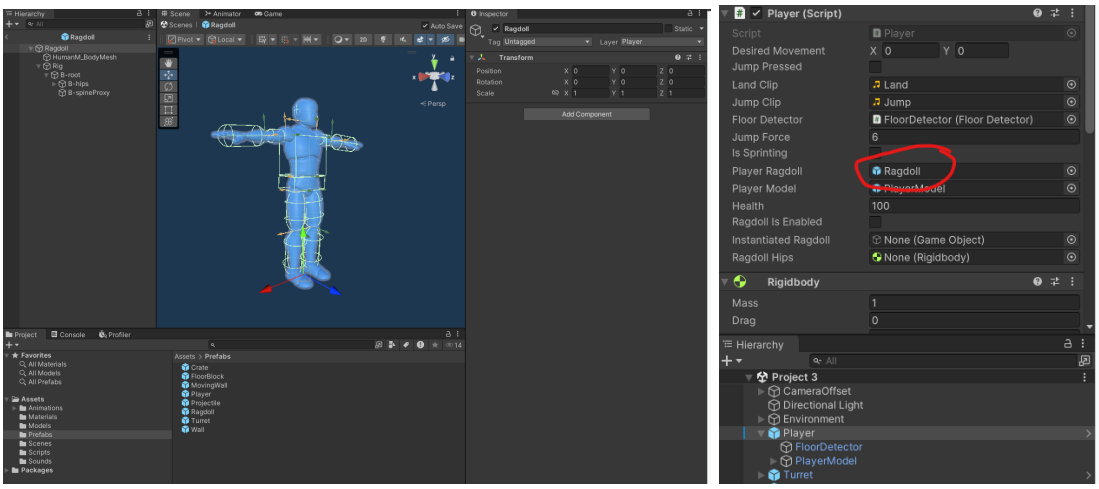
EnableRagdoll() must implement the following:

- Destroy the previously instantiated ragdoll if it exists
- Instantiate a new ragdoll at the position and rotation of the player
- Disable “playerModel” (the animated player model)
    - Hint: use playerModel.SetActive(false);
- Get the rigidbody component on the "Rig/B-root/B-hips" transform of the ragdoll and assign it to “ragdollHips”
    - Hint: use instantiatedRagdoll.transform.Find("Rig/B-root/B-hips"); to get the transform of the hips
- Set “ragdollIsEnabled” to “true”

DisableRagdoll() must implement the following:

- Destroy the previously instantiated ragdoll if it exists
- Enable “playerModel” (the animated player model)
- Set “ragdollIsEnabled” to “true”

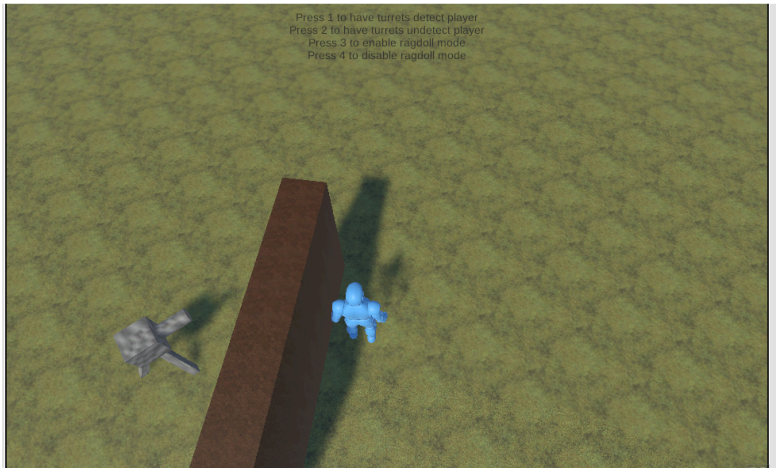
Then, you must create a new ragdoll prefab using Unity's ragdoll dialogue and add this to the player's "Player" Component:



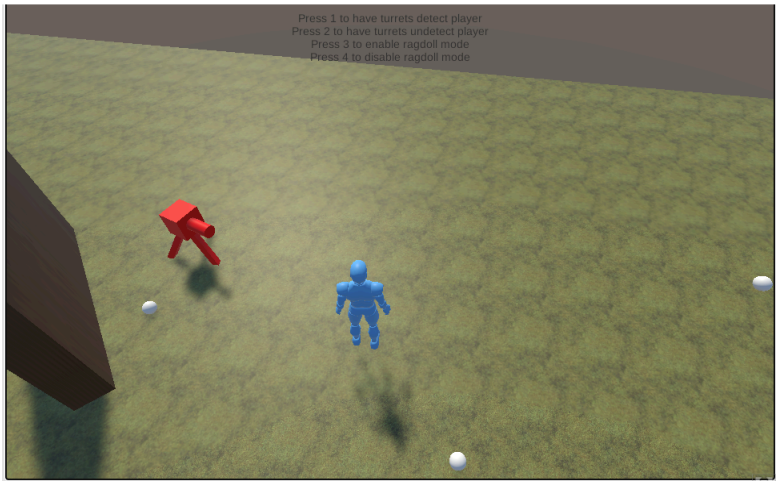
If you want to see what your final project should implement, please try the following demo:

https://maxgilson.itch.io/ragdoll-turret-demo

Example: player is not detected by turret there is a wall in between:

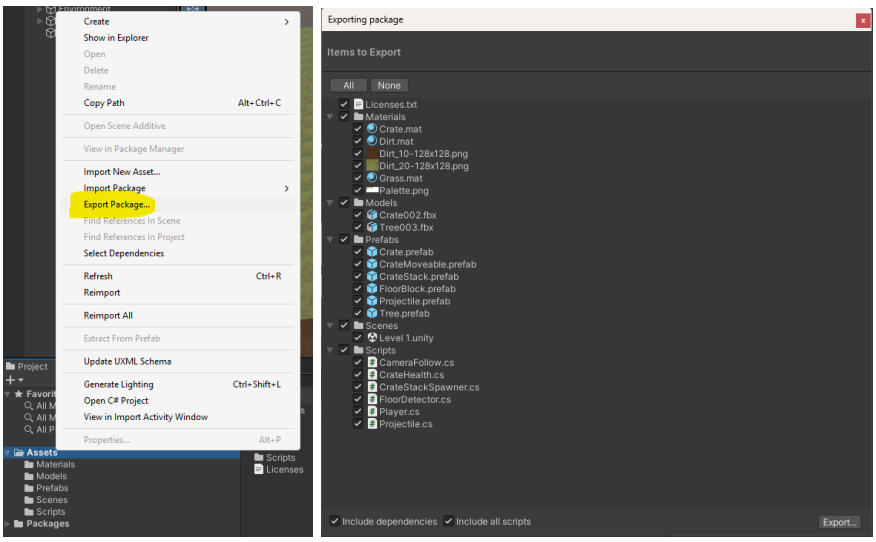


Example: player is detected if player is within radius and no obstacles:



How to Submit

To submit your code, you must pack your project into a Unity Package. To create a Unity Package first right click on the “Assets” folder inside of the Unity Editor, select all items and hit Export. Lastly, name your file “YourLastName-Project2” where “YourLastName” is your actual last name, and submit this file to the Pilot dropbox:



Grading

This lab is worth 5.00 points, distributed as follows:

Task	Points
There are no build or runtime errors	1.00
Turret script correctly performs “OverlapSphere”	1.00
Turret script correctly performs raycast for detecting player	1.00
Player script correctly creates a ragdoll	1.00
Player script correctly destroys ragdoll and enables player	1.00
Total	5.00