Unity Game Design Workshop 1
Simple Racing Game

For this Game tutorial we create a simple 3D racing game to learn about the Unity Editor, and to learn about terrain, textures, lighting effects, and how to add scripts to game objects.

1. Download the "Raw Terrain File" from the website www.cs.mu.edu/~ktooher/gwc.html
    a. Make sure to download this to somewhere you will remember

2. Go to the Unity asset store (navigate to the store by clicking the tab directly above the scene view window) and type "Standard Asset Package" and download and import the assets to your project

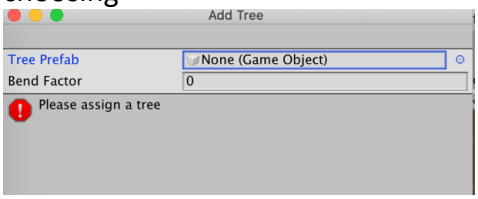
3. Click Create in the hierarchy tab, then 3D objects, then Terrain.
    a. Select this game object and then click the gear icon that appears in the inspector view.
    b. Scroll to the bottom and select import raw
    c. Select the RawTerrainHeightmap file that you downloaded from the website
    d. Change the byte order setting from Windows to Mac
    e. Change the dimensions to be 200 wide by 100 long and 100 tall

4. You may alter the terrain map by navigating to the paint brush setting in the Inspector window (terrain game object must be selected in the hierarchy for this to be viewable) and then selecting "Raise or Lower Terrain" from the drop down
    a. Now you can change the land by changing your brush size, click and drag to raise land, and click and drag while holding shift to lower the land.

5. To add texture (color/details) to the terrain, select the paint brush and change the drop down to say "Paint Texture"
    a. To "paint" on the grass go to **Edit Terrain Layers > Create Layer** and then select MudRockyAlbedo from the search menu appears
        i. Note: the first layer that you import will immediately be put on the entire map, we add more colors on top of this to make it look better.
    b. Then select **Edit Terrain Layers > Create Layer** and search for GrassRockyAlbedo and change your brush tool, opacity (transparent-ness), and size and paint on grass to your terrain
        i. Try out some of the other assets listed in that menu (sand)!

6. You can also add trees to the surface by navigating to the terrain setting (located near the gear and paint brush towards the top of the Inspector View Window) and selecting the "Paint Trees" option.

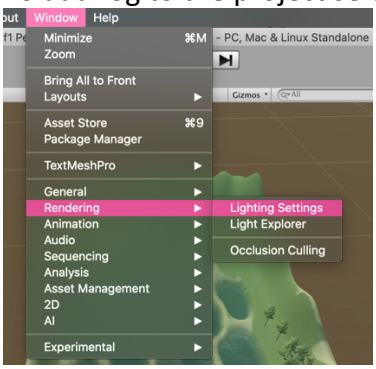
a. Then select **Edit Trees > Add Trees** and then touch the tiny circle to the left side of the text that says Tree Prefab and select “Conifer” or another tree of your choosing



Then change your brush size and tree density and paint on some trees b y clicking and dragging your mouse. To delete trees simply click and drag while holding down shift.

7. Add some water to the scene from the **Standard Assets > Environment > Water > Water4 > Prefabs > Water4Advanced** and click and drag water4advanced prefab directly onto your terrain map
    a. Once you have done this change the position of the water in the inspector view to be 100, 29, 100 and change scale to be 2,1,2

8. To add fog to the project select **Window > Rendering > Light Settings**



Then scroll to “Other Settings” and select the checkbox that says Fog to add fog to the project. You can change the color of the fog to be what you’d like and also change the density to be 0.005

9. Add a sky to your game by clicking the Create drop down in Project View and select **Create > Material**
    a. In the inspector for the new material, click the shader drop-down and select **Skybox > Procedural** and this will give you a dome like world effect
    b. Apply the skybox to the scene by clicking the material and dragging it directly into the “Sky” area of the scene view
    c. You can change the colors on this

10. To be able to test and play your game we need to add a character. To do this go to **Standard Assets > Characters > Prefabs** then click and drag FPSController into the scene where you would like your character to spawn.

11. Now to add scripts to make it an actual game. First we need to create a special game object, a Game Manager, that will dictate the game where and when to do what. To do this create an empty game object **GameObject > Create Empty**
    a. Position the game object at (165, 32, 125) and rotate it (0,62,0)
        i. You can change this around to fit your terrain if you changed it
    b. Rename the empty object **Spawn Point** (right click and select rename)

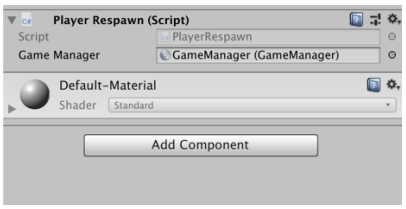
12. To create a water hazard detector (to see if you fell into water), add a plane to the scene with **GameObject > 3D Object > Plane** and position it at (100, 27, 100) and scale it to be (20,1,20)
    a. Again, you can change these dimensions and positions to fit your game.
    b. Rename the new plane game object to be "Water Hazard Detector"
    c. Check the **Convex** and **Is Trigger** check boxes in the mesh collider component in the Inspector View
    d. Then finally make the plane invisible by disabling Mesh Renderer component and do this by unchecking the box next to the Mesh Render components name in the Inspector

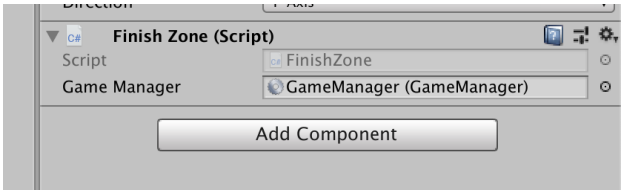
13. Add the finish zone to the game by first creating an empty game object to the scene and positioning it at (26, 32, 37)
    a. Rename it Finish Zone
    b. Add a light component to the Finish Zone Object so that it stands out to the player
        i. **Component > Rendering > Light** and change the type of the light in the drop down to say "point light" and set the range to 35 and the intensity to 3
            1. You may change these settings as you desire
    c. Add a capsule collider to the Finish Zone by selecting the object and clicking **Component > Physics > Capsule Collider**
        i. Check the IsTrigger check box and change the radius to what you'd like I used 9

14. Download the scripts from the website to your desktop or somewhere you are able to navigate to (desktop for this is easier)
    a. Then create a new folder called "Scripts" in the assets folder of your project view
    b. Then drag and drop the code from your desktop or other location into this scripts folder

c. Drag the FinishZone.cs file onto the Finish Zone Game object in the hierarchy view
d. Drag the GameManager.cs code onto the game manager object in the hierarchy view
e. Drag the PlayerRespawn.cs script onto the water hazard detector game object located in the hierarchy view

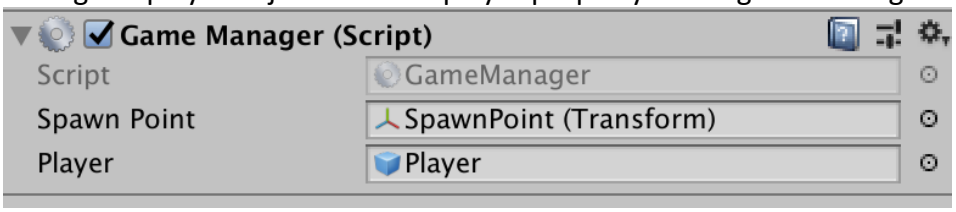
15. Select the water hazard detector object in hierarchy view and notice that the player respawn component has a game manager property – click and drag the Game Manager object from the hierarchy view onto the game manager property of the player respawn script component.
    a. Do the same for the Finish Zone Game object





16. Select the Game Manager Object and click and drag the Spawn Point object onto the spawn point property of the game manager

17. Click and drag the player object onto the player property of the game manager object



18. Now test it out!