

3D Games Programming

CI7500

Prof Vasilis Argyriou

Vasileios.Argyriou@kingston.ac.uk

Design a basic game

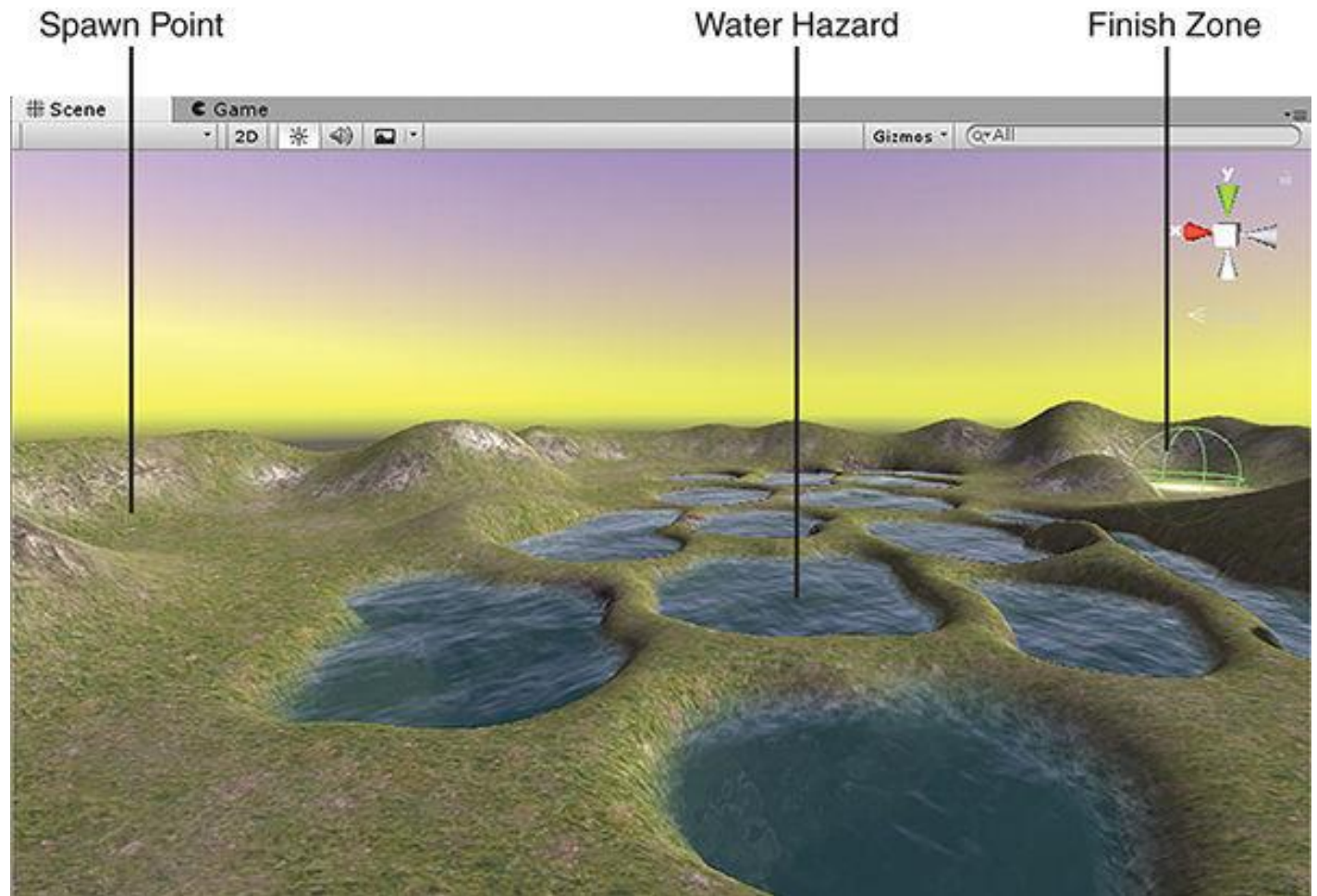
The Rules

- >There is no win or loss condition; only a completed condition. The game is completed when the player enters the finish zone.
- >The player will always spawn in the same spot. The finish zone will always be in the same spot.
- >There will be water hazards present. Whenever the player falls into a water hazard, that player is moved back to the spawn point.
- >The objective of the game is to try to get the fastest time possible. This is an implicit rule and is not specifically built in to the game.

The Requirements

- >A piece of rectangular terrain. The terrain needs to be big enough to present a challenging race.
- >Textures and environment effects for the terrain.
- >A spawn point object, a finish zone object, and a water hazard object
- >A character controller.
- >A graphical user interface (GUI).
- >A game controller.

The Requirements



Sculpting the World

1. Create a new project in a new folder. Add a terrain to the project.
2. Set the resolution of the terrain to 200 wide by 100 long and 100 tall (in the Resolution section of the Terrain Settings).
3. Import the terrain.raw file from Canvas as a heightmap for the terrain (by clicking **Import Raw** in the Heightmap section of the Terrain Settings).
4. Create a **Scenes** folder under assets and save the current scene as **Main**.

Adding the Environment

You need to import the following packages (click **Assets > Import Package**):

- > Standard Assets
- > Improve the terrain adding textures, trees, etc.
- > Add water, fog, skybox, etc.

The Character Controller

1. Import the standard assets **Assets > Import Package**.
2. Drag a **First Person** controller asset from the Characters folder into your scene.
3. Position the First Person controller (with the blue colour) at (160, 32, 64). Rotate the controller 260 on the y axis so that it faces the correct direction.

You may have to adjust these numbers

Adding Game Control Objects

1. Add an empty game object to the scene (click **GameObject > Create Empty**) and position it (at 160, 32, 64).

2. Rename the empty object to **SpawnPoint** in the Hierarchy view.

Next, you want to create the water hazard detector. This will be a simple plane that will sit just below the water. The plane will have a trigger collider, which will detect when a player has fallen in the water.

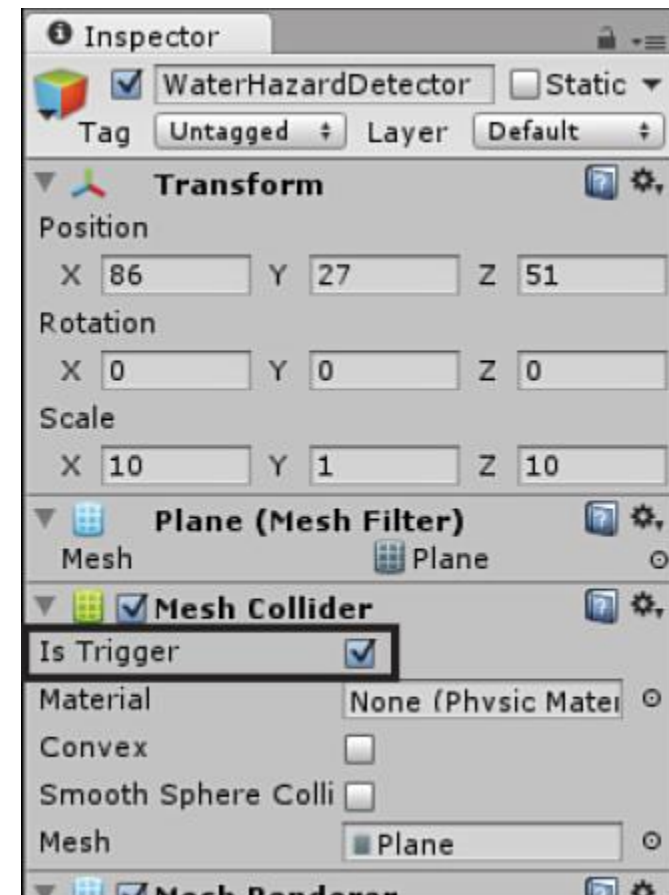
Adding Game Control Objects

1. Add a plane to the scene (click **GameObject > 3D Object > Plane**) and position it (at 86, 27, 51) [Adjust it to fit to the terrain].

Scale the plan (10, 1, 10).

2. Rename the plane to WaterHazardDetector in the Hierarchy view.

3. Check the **Is Trigger** check box on the Mesh Collider component in the Inspector view and the Convex option.



Adding Game Control Objects

Next you want to add the finish zone to your game. This zone will be a simple object with a point light on it so that the player knows where to go.

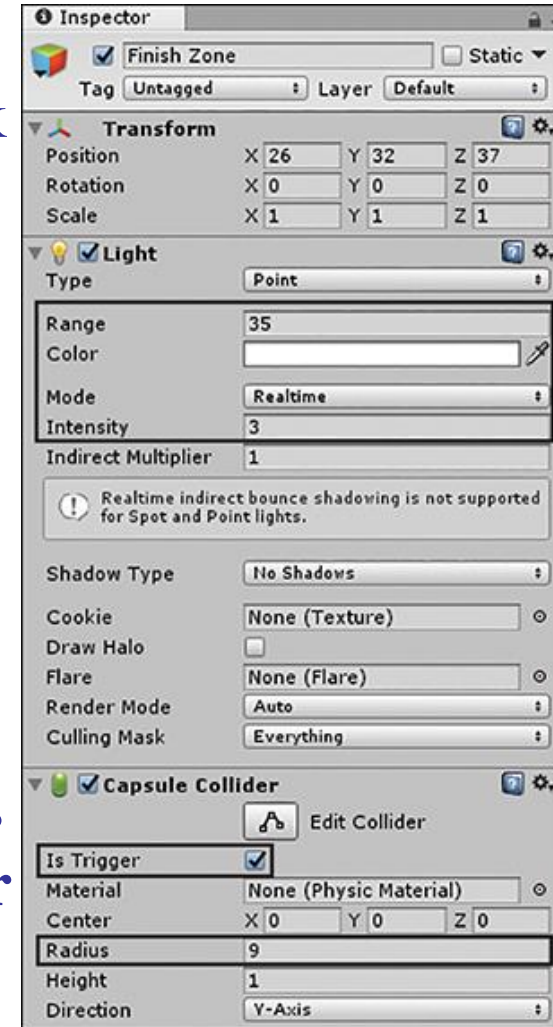
The object will have a capsule collider attached to it so that it will know when a player can enter the zone.

1. Add an empty game object to the scene and position it at (26, 32, 24).
2. Rename the object to **FinishZone** in the Hierarchy view.

Game Control Objects

3. Add a light component to the finish object. (With the object selected, click **Component > Rendering > Light**.) Change the type to **Point** if it isn't already and set the range to **35** and intensity to **3**.

4. Add a capsule collider to the finish object by selecting the object and clicking **Component > Physics > Capsule Collider**. Change the Radius property to **9** and check the **Is Trigger** check box in the Inspector view



Adding Game Control Objects

The final object you need to create is the game control object. This object doesn't technically need to exist.

You could instead just apply its properties to some other persistent object in the game world such as the Main Camera.

You generally create its own object to prevent any accidental deletion, though.

1. Add an empty game object to the scene.
2. Rename the game object to **GameManager** in the Hierarchy view.

Adding Scripts

Scripts specify behaviors for your game objects

1. Create a Scripts folder under Assets in the Project view.
2. Locate the Scripts on Canvas
3. Click and drag the scripts into the Scripts folder in Unity. There should be three scripts: FinishZone, GameManager, and PlayerRespawn.

To apply a script, simply drag it from the Project view onto whatever object you want to apply it to

Adding Scripts

Apply the following scripts:

- >Apply the FinishZone script to the FinishZone game object.
- >Apply the GameManager script to the GameManager object.
- >Apply the PlayerRespawn to the WaterHazardDetector object.

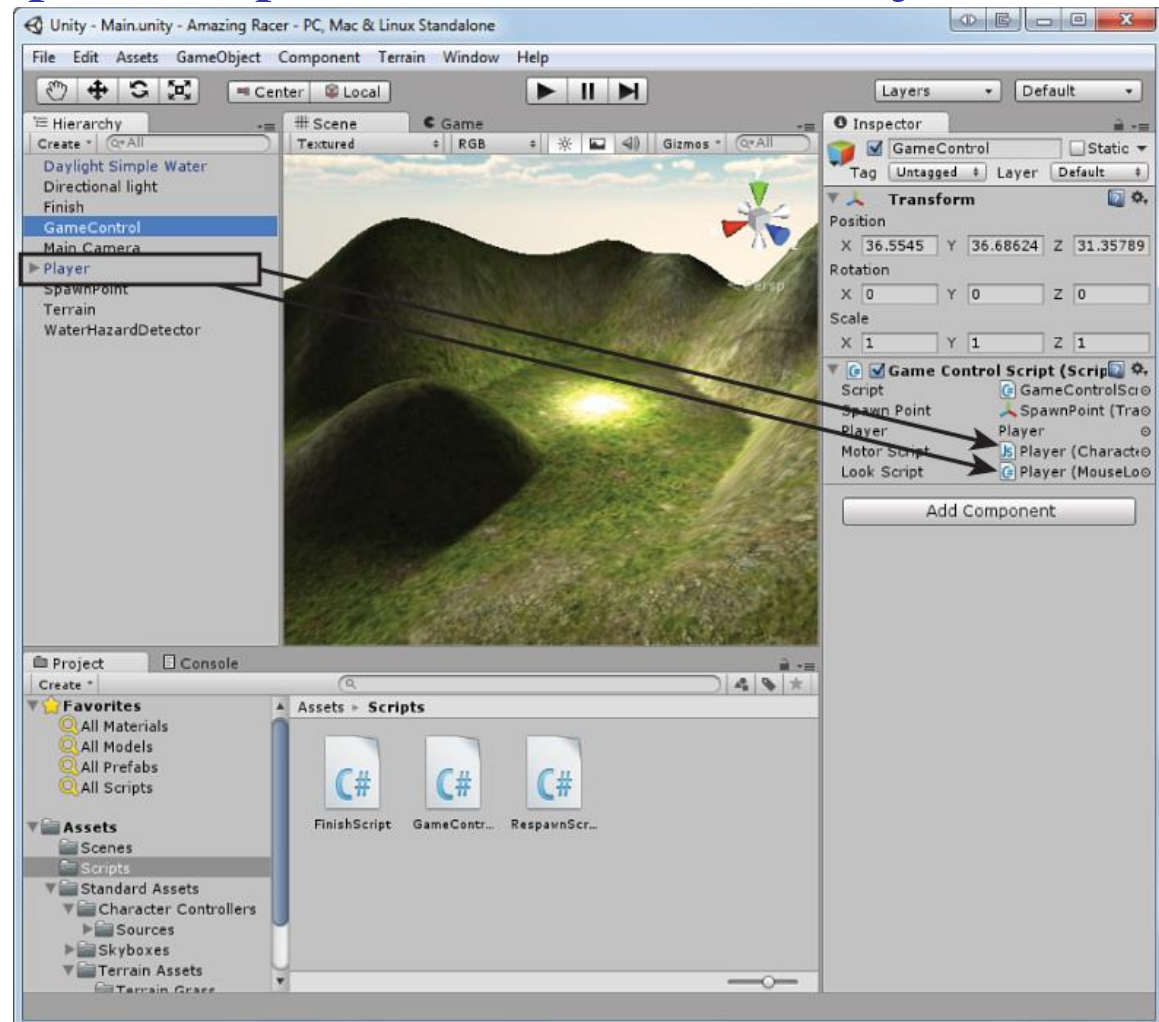
Adding Scripts



Either way works

Connecting the Scripts Together

The scripts have placeholders for other objects.



Connecting the Scripts Together

With the **WaterHazardDetector** object selected, click and drag the SpawnPoint object from the Hierarchy view onto the Respawn Point property of the Respawn Script component.

With the **Finish** game object selected, click and drag the GameController object from the Hierarchy view onto the Game Control Script property of the Finish Script component in the Inspector view.

Connecting the Scripts Together

1. Click and drag the **SpawnPoint** object onto the Spawn Point property of the Game Control Script component of the GameController.
2. Click and drag the Player object (this is the character controller) onto the Player property, the Motor Script property, and the Look Script property of the Game Control Script of the GameController.
3. Repeat the same for the last Controller property.

Summary

- ❑ Design a basic game