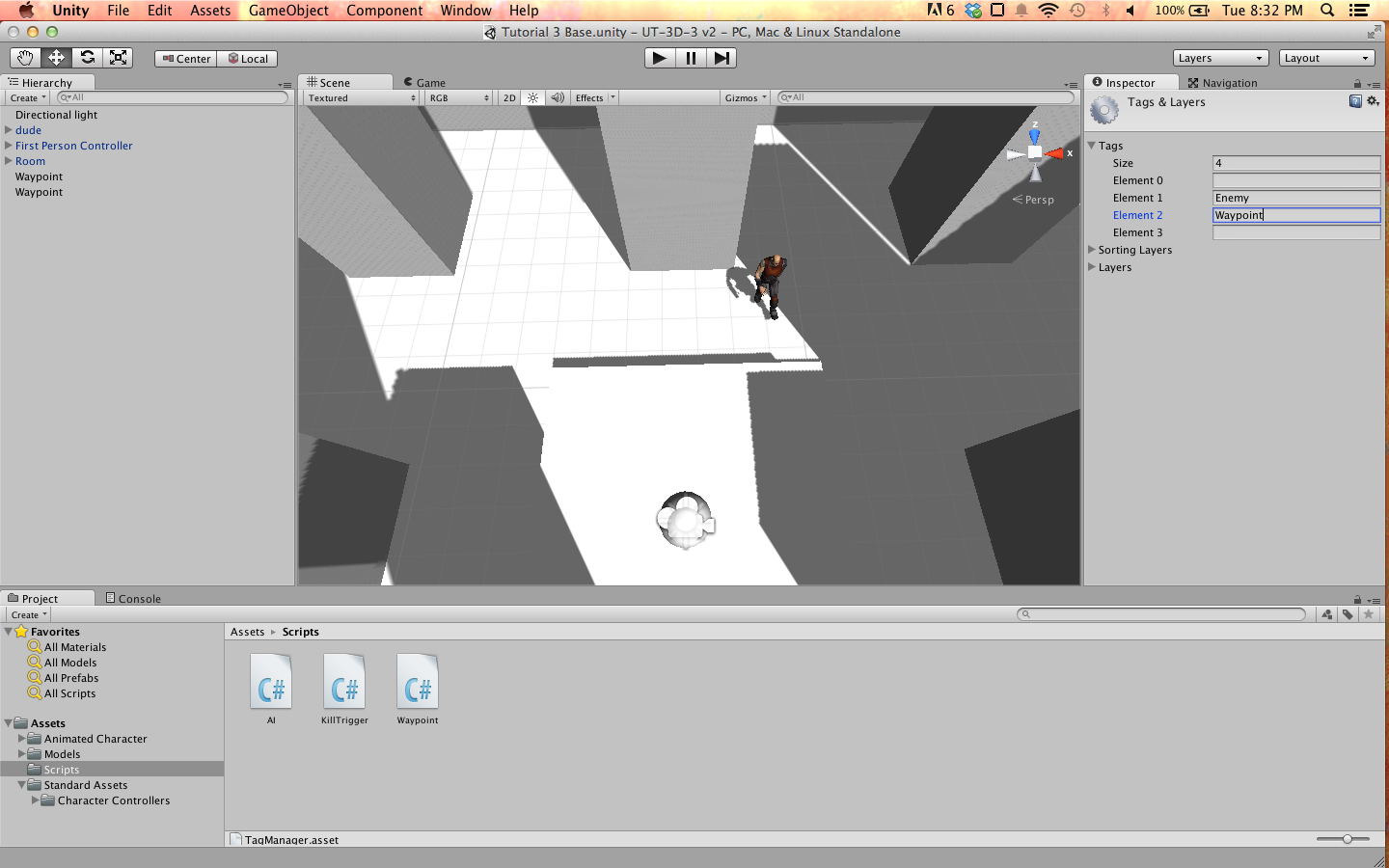
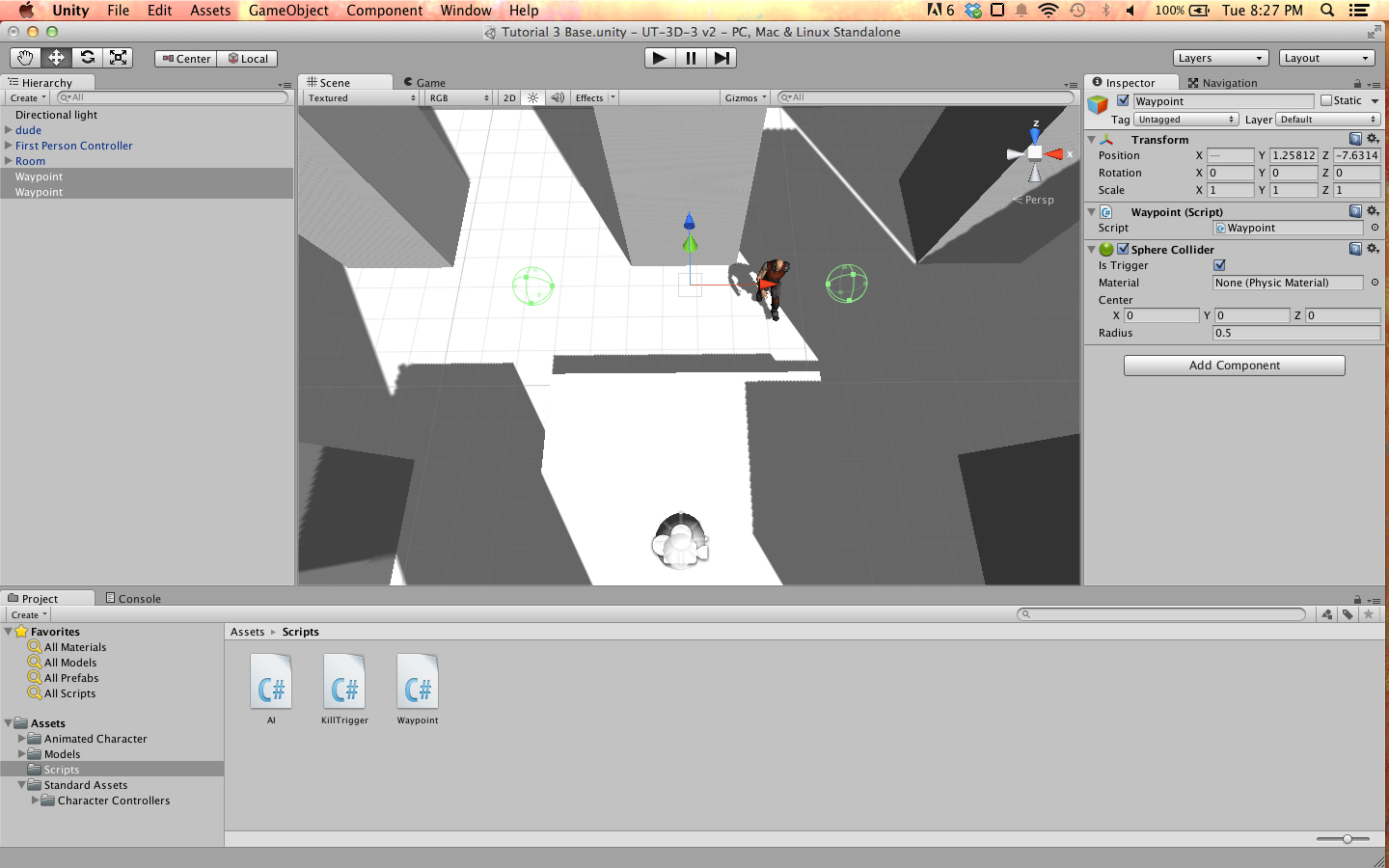
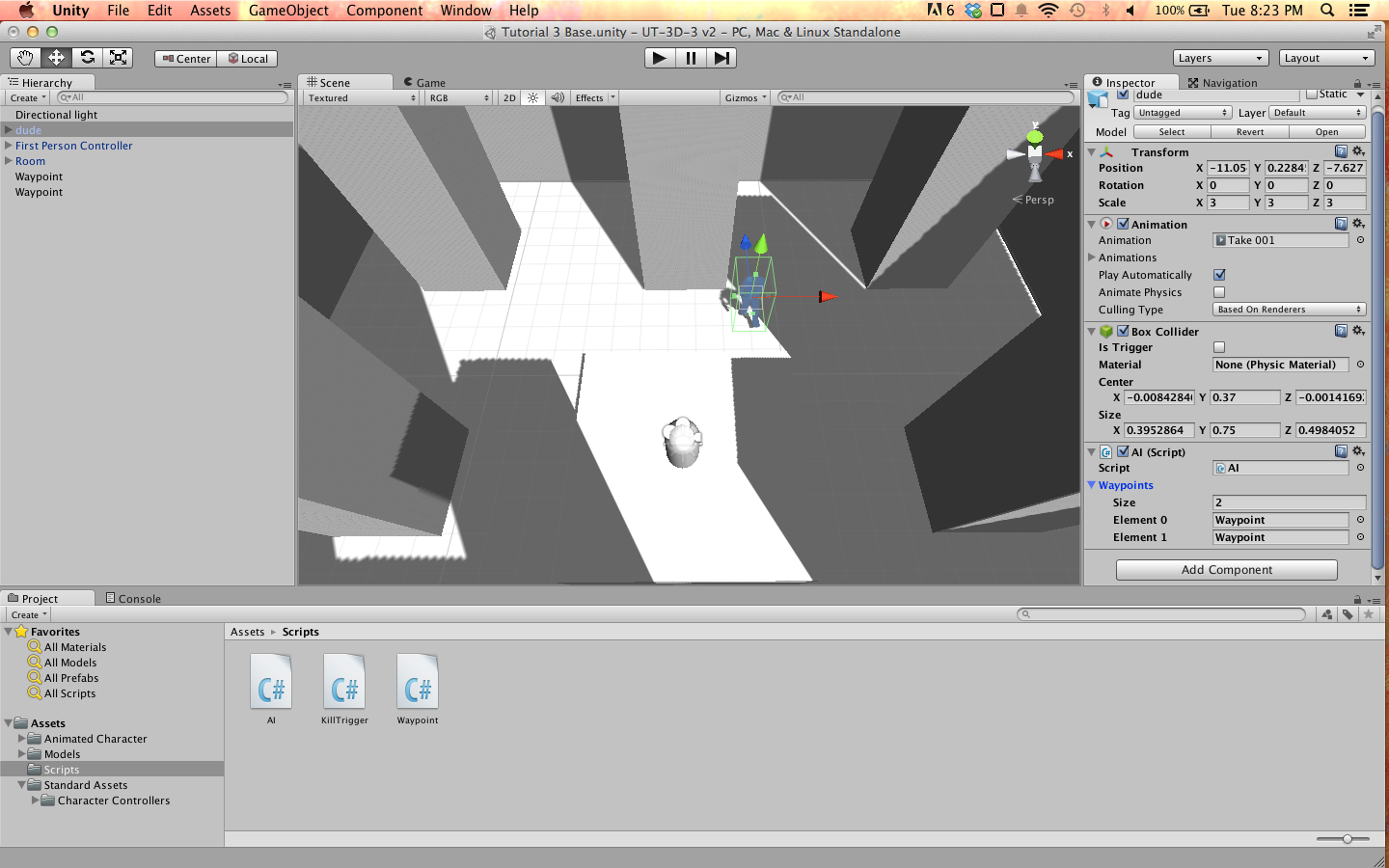
**Making 3D Games with Unity**

**Tutorial 3**

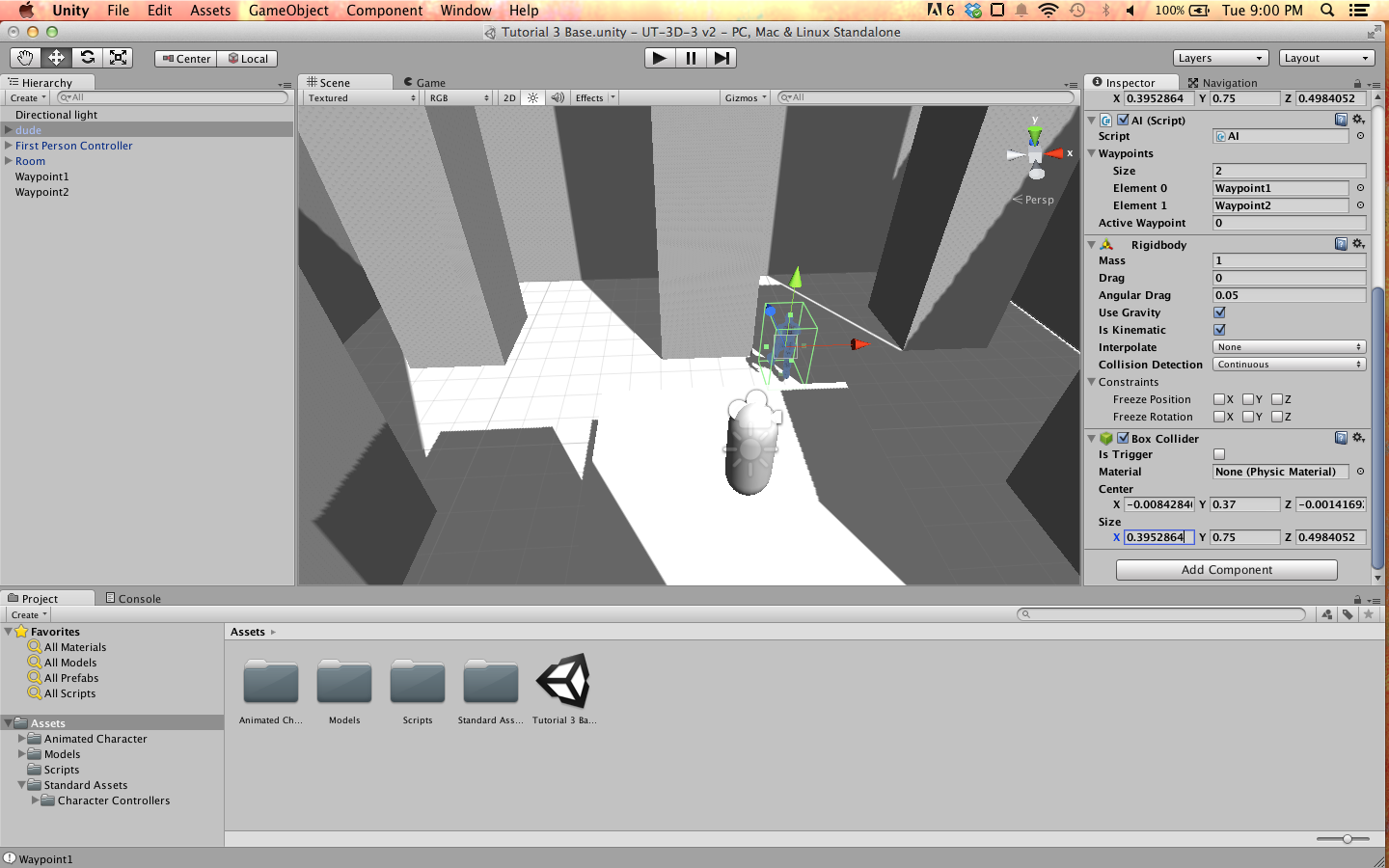
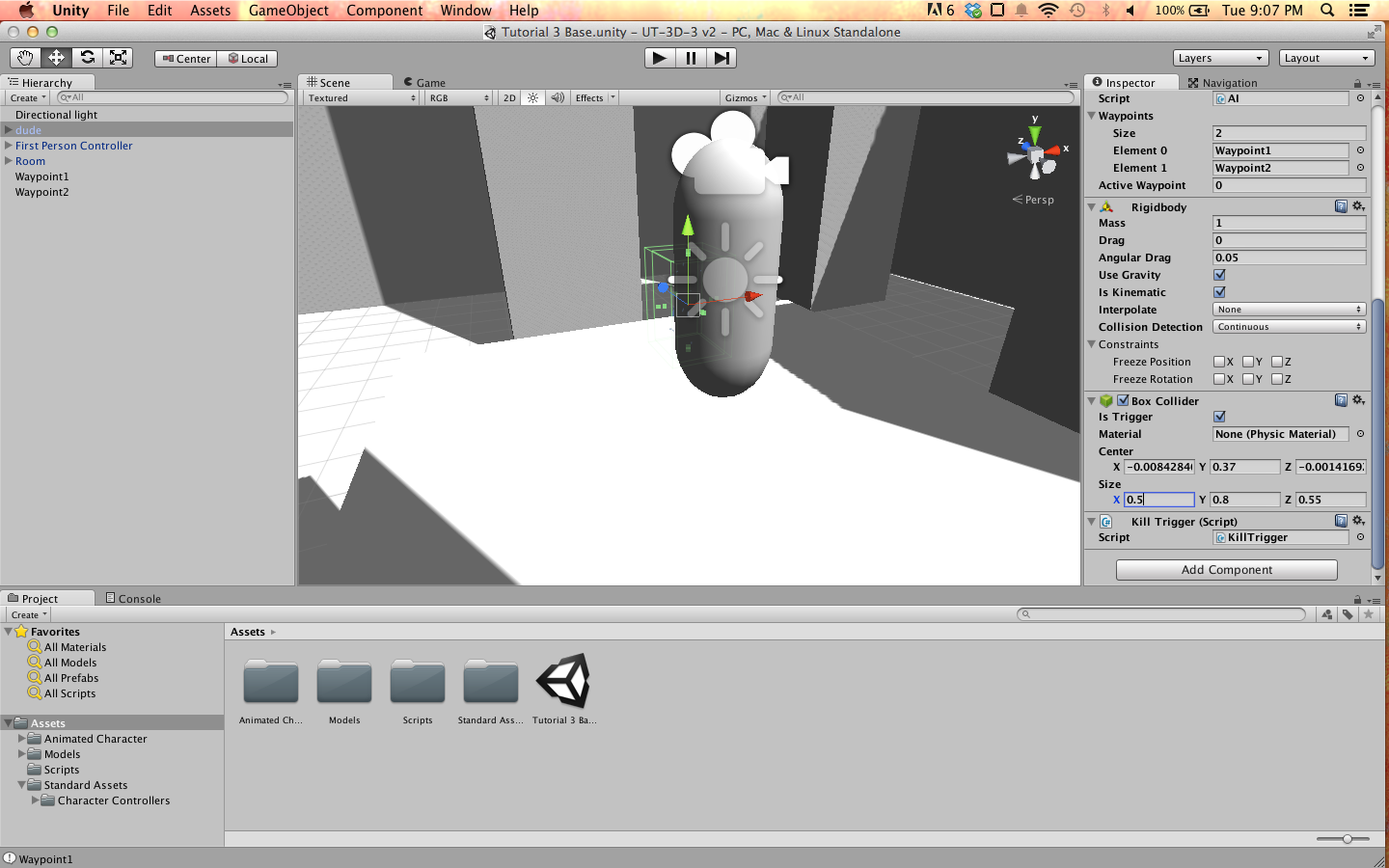
Objectives:

* AI
  + AI moves side to side in a hallway
  + kills player on contact
  + Shooting and killing enemies
* Third Person setup

**1. AI walking between Waypoints**

1. Create a new project called **UT-3D-3**
   1. Import the standard character controller package. We will use the standard FPS controller for this tutorial.
   2. In the project window, right click and select Import Package>Custom Package and select the supplied UT-3D-3 Base.unitypackage. Wait for the package to import. This package contains the dude character and a room model along with a scene which has already been setup.
   3. Open the Tutorial 3 Base scene. It is already playable with a character controller and an enemy with animations and collisions. This was created using the same steps as tutorial 2.
2. Add a new script called AI to dude.
   1. Click on dude in the hierarchy window and in the inspector click Add Component, and type AI as the name and hit enter twice to create and attach the script.
   2. See the example AI.cs for an explanation.
3. Change dudes tag to “Enemy”.
   1. Select dude and in the Inspector, change the Tag drop down underneath the object’s name to “Enemy”.
4. Add a rigidbody to dude so that Unity will detect trigger events for dude.
   1. Select dude and in the Inspector, click Add Component and type “rigidbody”, then hit enter.
   2. Under Rigidbody, check the “Is Kinematic” check box.
5. Add an empty GameObject to the scene to be used as a waypoint by the enemy AI. The AI will move between waypoints continually.
   1. Click Menu>GameObject>Create Empty.
   2. Name this object Waypoint by clicking its name in the Inspector window.
   3. Change the Waypoints tag to “Waypoint”.
      1. Click the dropdown next to Tag under the object’s name. Select Add Tag…
      2. In the Inspector window, Type “Waypoint” into an empty element. eg:
      3. Select the Waypoint again and change the Tag drop down to “Waypoint”.
   4. Select the Waypoint again and in the Inspector click Add Component and type “Sphere Collider” then hit Enter to add it to the Waypoint.
   5. Under the Sphere Collider component, check the Is Trigger checkbox. This will disable collisions and allow Unity to use this collider to trigger events.
   6. Add a new script to the waypoint. Click Add Component, type Waypoint and hit enter twice.
   7. See the example Waypoint.cs for an explanation.
6. Duplicate the Waypoint object by selecting it in the Hierarchy window and clicking Menu>Edit>Duplicate.
7. Position both waypoints along a path you’d like dude to walk. Make sure that you place dude somewhere in between these two points, and that there are no obstructions between the two waypoints. For example:
8. Add the waypoints to the AI script.
   1. Select dude and ensure that the AI script component is visible in the Inspector.
   2. Drag the first waypoint from the Hierarchy window to the Waypoints item in the Inspector. Repeat for the second waypoint.
   3. You can expand the Waypoints item in the Inspector to ensure both Waypoints were added properly. Your dudes Inspector should look something like this:
   4. 
9. Play the game, the dude should walk towards a waypoints until he reaches it and switches direction to the other waypoint. He should loop between waypoints in order.

**2. AI kill on player collision**

1. To have the player “die” when in contact with an enemy, we will use triggers in a similar manner as the waypoints. When the player “dies”, it will reload the game and start from the beginning.
2. Create a trigger collider around the enemy.
   1. Select the dude in the Hierarchy window, and in the Inspector right click the name of the Box Collider component and select Copy Component.
   2. Right click the Box Collider again and this time select Paste Component As New.
   3. Scroll to the bottom of the list to see the new Collider. Check the “Is Trigger” checkbox, and change the size x, y and z of the collider to be slightly larger than they currently are. eg: to 
3. Create a new script called “KillTrigger” on the enemy so that entering the trigger area kills the player.
   1. Select dude and in the Inspector, click Add Component. Type “KillTrigger” and hit enter twice to create the script.
   2. See the example KillTrigger.cs for an explanation of what this script should do.
4. Make sure the player is tagged “Player”.
   1. Select the First Person Controller in the Hierarchy window.
   2. In the Inspector, change the Tag drop down to “Player”.
5. Play the game and run into the dude to “die”.

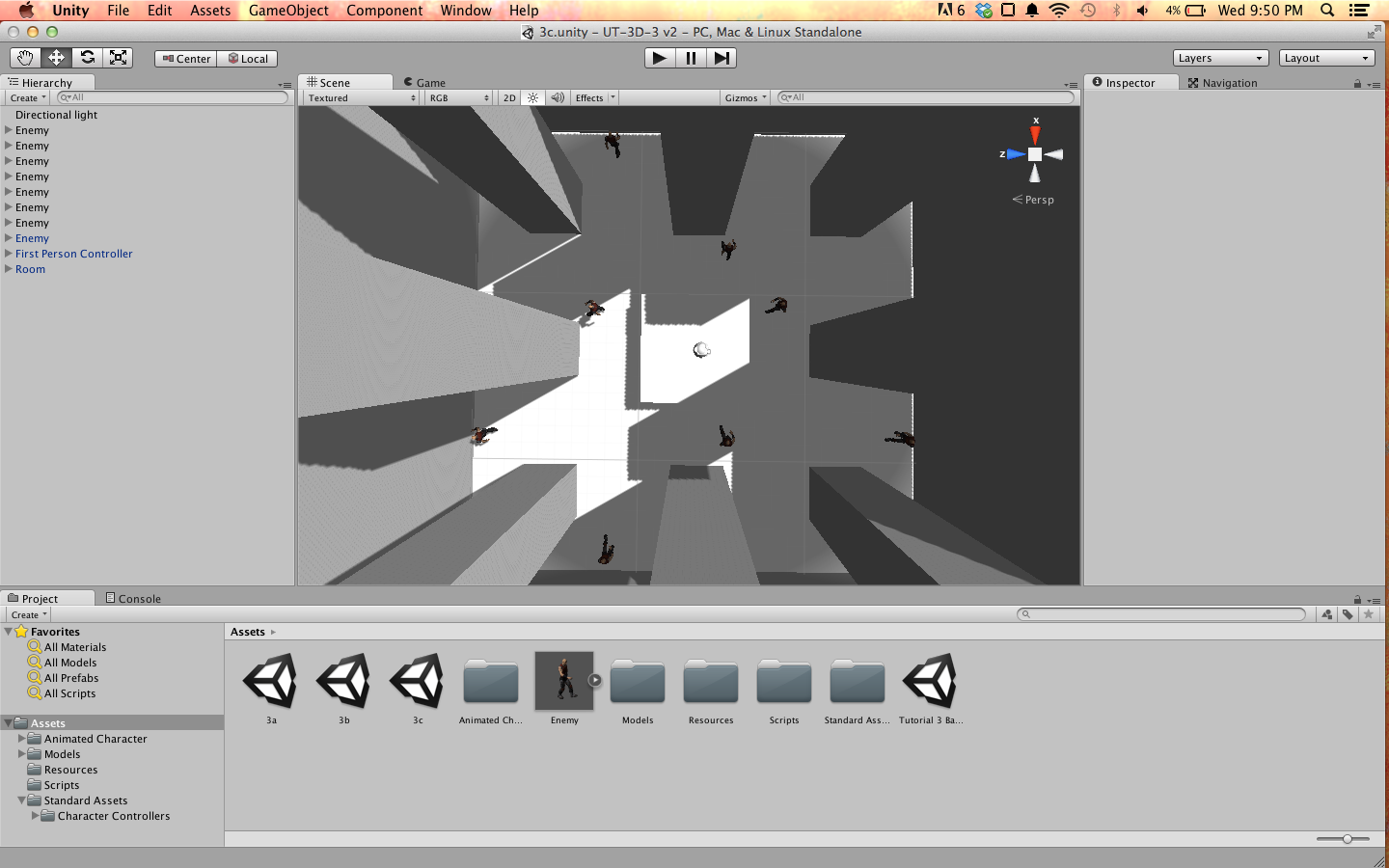
**3. Multiple Enemies and Shooting Enemies**

1. Create a Projectile object to use as a template for projectiles shot from the player.
   1. In the Project window, create a new folder called “Resources”.
   2. Create a new sphere object by navigating to Menu>GameObject>Create Other>Sphere.
   3. Name this new sphere “Projectile”. Create a new tag called Projectile and tag this object with it.
   4. Drag the Projectile object from the Hierarchy to the Resources folder in the Project window. Delete the Projectile from the Hierarchy window.
   5. Add a script to the Projectile which will detect if an enemy has been hit, and kill it.
      1. Click on the Projectile prefab in the Resources folder. In the Inspector, click Add Component and type “Projectile”, then hit enter twice.
      2. See the example Projectile.cs for an explanation of what this script should do.
2. Add a Die function to the AI script.
   1. Open AI.cs. Add this function to the class:

public void Die(){

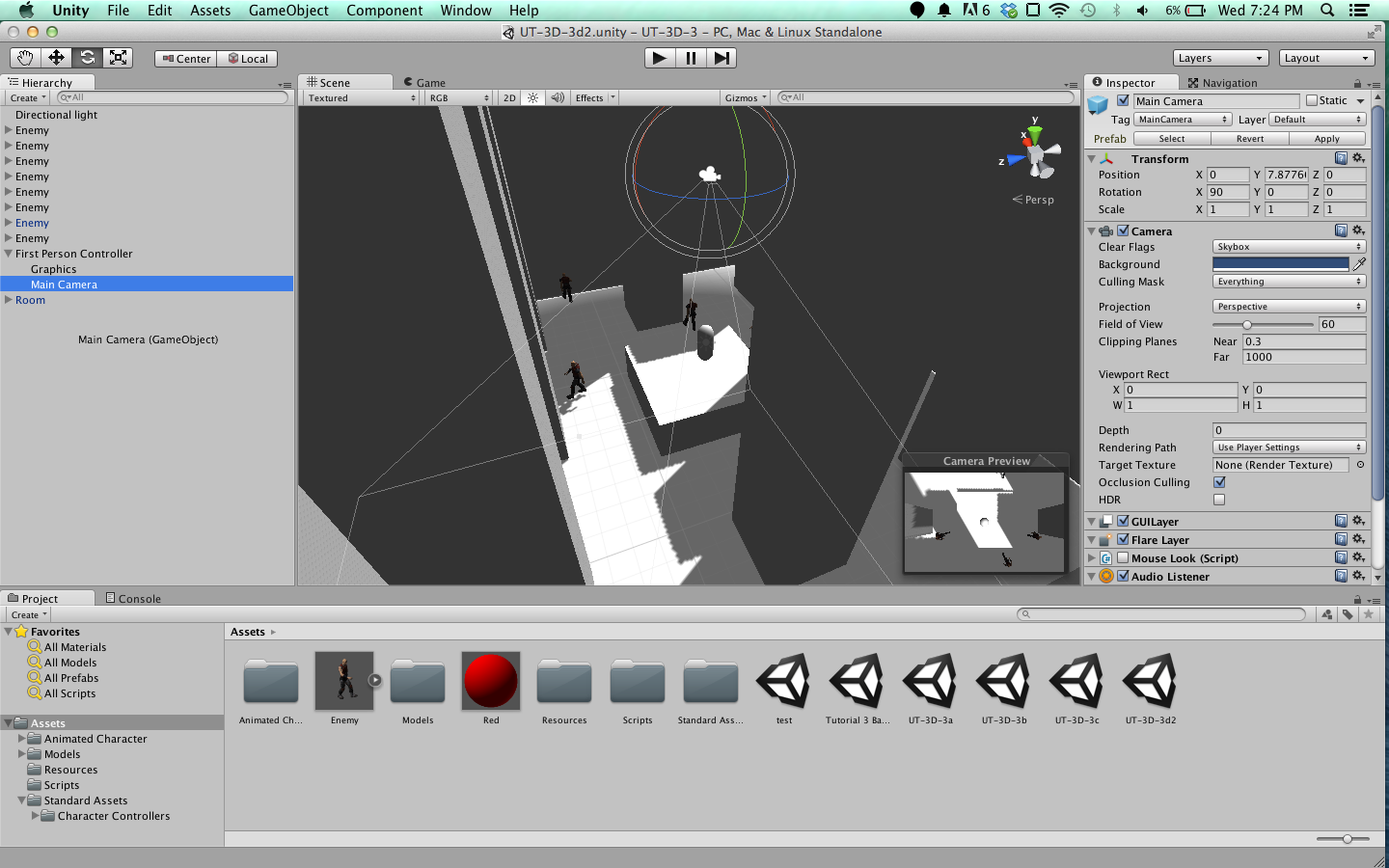
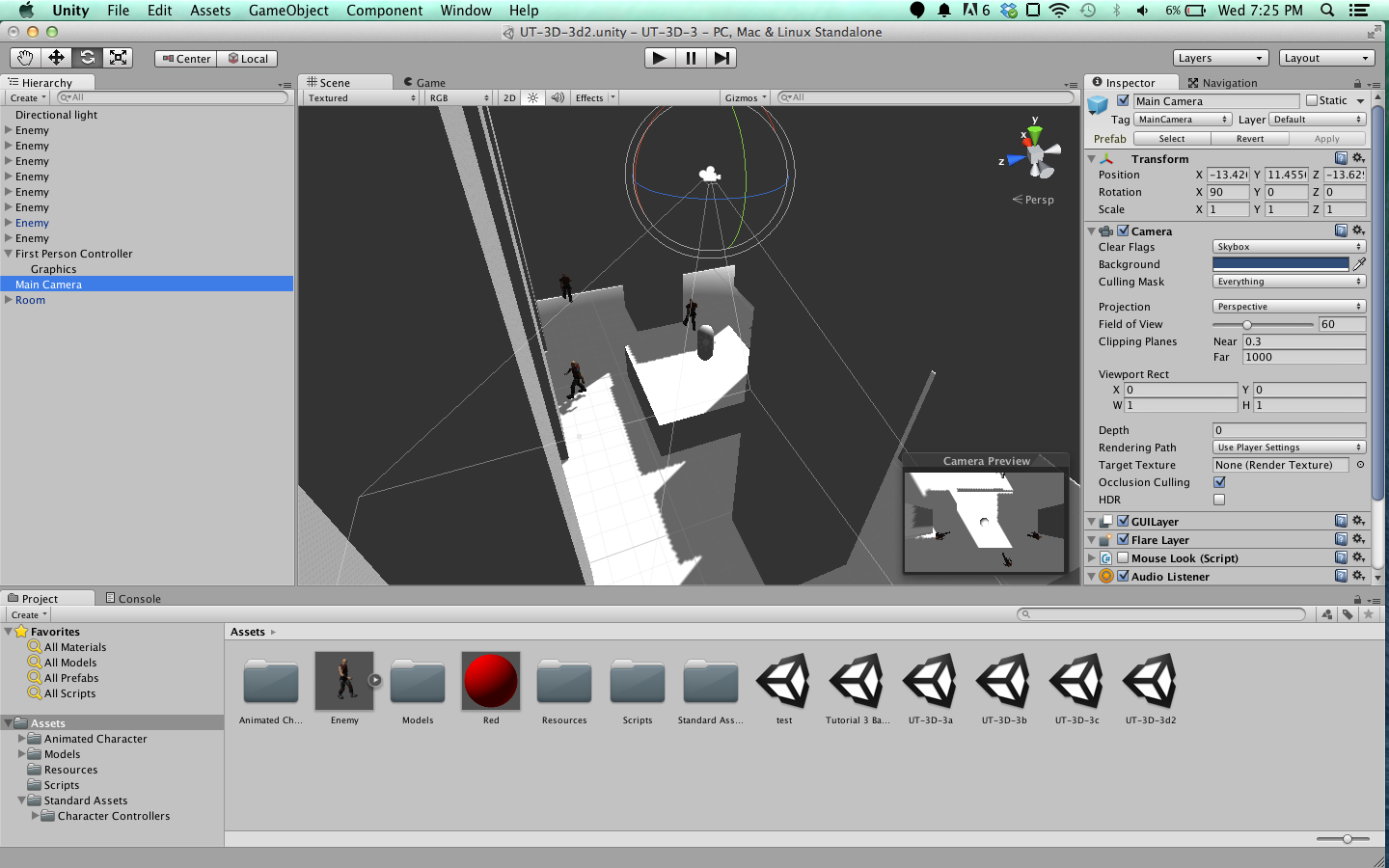
Destroy(this.gameObject); *//this will remove the GameObject attached to this script and handle garbage collection.*

}

1. Add the ability for the player to shoot a projectile.
   1. Select the First Person Controller and in the Inspector click Add Component. Type “Shoot” and hit enter twice.
   2. See the example Shoot.cs for an explanation.
2. Play the game. You can move around and click the mouse to shoot a sphere. If a projectile hits the dude, he will “die” by disappearing.
3. It is a bit boring with only one enemy in the level, so we will create duplicates which follow different paths.
   1. Create a new empty object to act as a parent for the enemy. Menu>GameObject>Create Empty. Name this object “Enemy”.
   2. In the Hierarchy window, select dude and both waypoints by holding Ctrl and clicking the three objects. Click and drag them onto the Enemy object in the Hierarchy window.
   3. Now drag Enemy from the Hierarchy window to the Project window into the Assets folder. This will create a prefab of the Enemy.
   4. Drag the Enemy prefab from the Project window anywhere into the scene to create a copy of it. Move it somewhere you like.
   5. You can also move the waypoints of this object by clicking the arrow next to Enemy in the Hierarchy window and selecting the waypoint you want to move.
   6. You can also select an Enemy object in the scene and Duplicate it by navigating to Menu>Edit>Duplicate or pressing Ctrl + D.
   7. Repeat this a few times. Example of a scene with multiple enemies:

**4. Third Person Camera and Controls**

1. First we need to disable camera movement. Disable the Mouse Look script on the First Person Controller and the Main Camera.
   1. Expand the First Person Controller and Ctrl+Click it and the Main Camera so that they are both selected.
   2. In the Inspector, the Mouse Look script should be visible. Check the checkbox next to its name to disable it for both objects at once.
2. We will move the camera to look down on the player for an aerial view.
   1. In the Hierarchy window, expand the First Person Controller and select the Main Camera.
   2. In the Inspector, change the Main Camera’s X Rotation to 90.
   3. Move the camera with the move tool or in the inspector so it is Y Position is around 6.
   4. Play the game. You should be able to move around the world, but when you click to shoot, the bullet will appear in the middle of the screen and then vanish.
3. Add a shooting script which shoots from the player to the mouse position, instead of the camera.
   1. Select the First Person Controller and disable the Shoot script in the Inspector.
   2. Click Add Component and type “Third Person Shoot” and hit enter twice to create the script. It will be very similar to our existing script “Shoot” but change the origin and direction of the bullet to be relative to the player.
4. Rotate the player to look in the direction of the mouse.
   1. Select the First Person Controller and in the inspector, uncheck the MouseLook and Shoot scripts.
   2. Expand the First Person Controller and drag the Main Camera out of it’s parent and into the root of the Hierarchy, like this:

 -> 

* 1. Move the camera up in the Y axis roughly 6 units and then rotate the camera so the X axis is 90 degrees, facing straight down at the player.
  2. Add a new script to the camera and call it “FollowPlayer”. See the example FollowPlayer.cs for an explanation of what this script should do.
  3. Select the First Person Controller and add a new script called “ThirdPersonMouseLook”. See the example ThirdPersonMouseLook.cs for an explanation.
  4. Add another new script to the First Person Controller called “ThirdPersonShoot”. See the example ThirdPersonShoot.cs for an explanation.

1. Play the game, and you should be able to move the player with the WASD keys while moving the mouse around the player to look, and click to shoot in the direction of the mouse. The player will move “forward” with the W key towards the mouse due to the nature of the standard First Person Controller. The game should look like this: