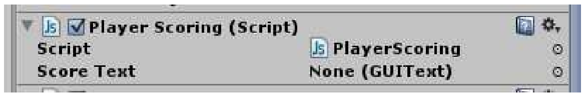
Instructions for using Object Collection and Trigger mechanics in Unity

Jason Fritts

1 Object Collection and Scoring

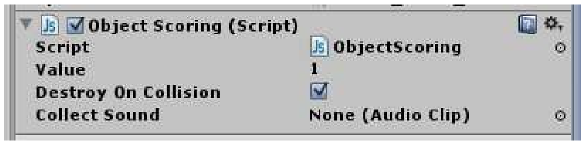
- Add **PlayerScoring** script to **First Person Controller**
  - Select **First Person Controller** object in the Hierarchy window.
  - In Inspector window, click on the Add Component button.
  - In the drop-down menu, select Scripts → **PlayerScoring**. You should now see the following script component attached to **First Person Controller**.



Player Scoring (Script)

Script	PlayerScoring
Score Text	None (GUIText)

- Add **ObjectScoring** script to each of the Rigidbody objects you want to collect for scoring
  - For each object that you want to collect for scoring, first select it in the Hierarchy window.
  - In the Inspector window, make sure that it has the Rigidbody component.
  - In the Inspector window, change the Tag to CollisionObject. If this Tag does not exist yet, you will need to create it by selecting the **Add Tag...** option from the drop-down menu and entering the name of the next Element in the Tag list to be CollisionObject. See page 95 in our Unity text for further clarification on creating new Tags.
  - In the Inspector window, click on the Add Component button.
  - In the drop-down menu, select Scripts → **ObjectScoring**. You should now see the following script component attached to the object.

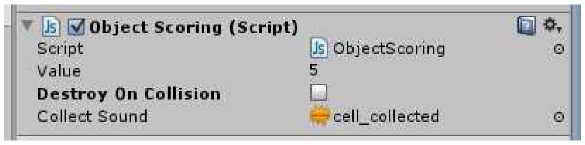


Object Scoring (Script)

Script	ObjectScoring
Value	1
Destroy On Collision	✓
Collect Sound	None (Audio Clip)

  - The parameters in the **ObjectScoring** script are: Value, Destroy On Collision, and Collect Sound. Value is the number of points added to the game score when the player collides with this object. Destroy On Collision dictates whether the object is deleted from the game upon collision – if the box is checked it will be deleted from the game; if not checked, it remains in the game. Finally, Collect Sound is an optional audio file that is played when a collision occurs.

    Below shows an example parameterization that upon collision will add +5 points, will not delete the object from the game, and will play the **cell_collected** sound.



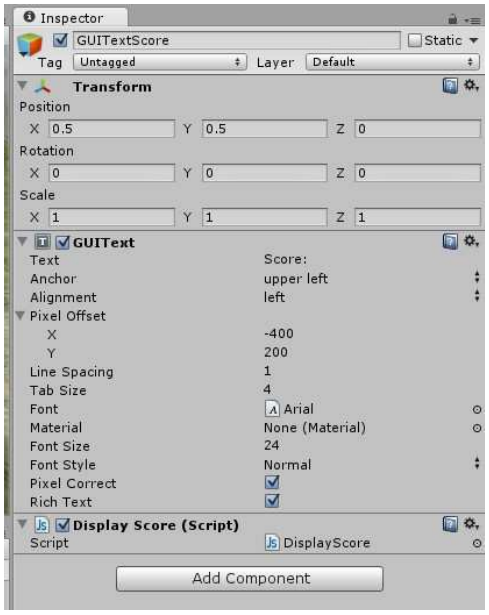
Object Scoring (Script)

Script	ObjectScoring
Value	5
Destroy On Collision	☐
Collect Sound	cell_collected

  - Optionally, if you're creating many objects with the **ObjectScoring** script, you can make a Prefab object that already has the appropriate Tag, Rigidbody component, and **ObjectScoring** script, and then quickly create instances of this Prefab by dragging them into the game.

- Optionally add an Audio Source to the **First Person Controller**
  - If you want an audio sound played when the player collects a CollisionObject, you need to add an Audio Source component to the **First Person Controller**.
  - In the Inspector window, click on the Add Component button.
  - In the drop-down menu, select Audio $\rightarrow$ Audio Source. When the player now collides with an object, it will play the sound file specified by that object's Collect Sound setting in its ObjectScoring script.
  - *Note:* If you haven't already, be sure to remove the Audio Listener component on the **Main Camera**. A scene can only have one Audio Listener for audio to work correctly. Since both the **Main Camera** and the camera in the **First Person Controller** have an Audio Listener, you need to remove/delete one. While playing the game, the player should hear the audio from the perspective of the player controller, so the Audio Listener on the **Main Camera** is the one that should be deleted.

- To display the game score, add a **GUI Text** object to the Scene
  - In the Hierarchy window, click on the Create button and select **GUI Text**.
  - For reference, you may want to change the name of this object to something more descriptive like **GUI Text Score**, though this is not required.
  - *Note:* After adding this element, you will not immediately see it displayed in the Scene window. It only appears in the Game window, so you won't see it until you begin playing the Scene.
  - To modify the paramters of the **GUI Text** element, select it in the Hierarchy window, and you will see the following in the Inspector window.



Inspector
GUITextScore ☑ Static
Tag Untagged Layer Default

Transform
Position X 0.5 Y 0.5 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

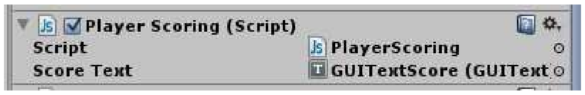
GUIText ☑
Text	Score:
Anchor	upper left
Alignment	left
Pixel Offset	
X	-400
Y	200
Line Spacing	1
Tab Size	4
Font	Arial
Material	None (Material)
Font Size	24
Font Style	Normal
Pixel Correct	✓
Rich Text	✓

Display Score (Script) ☑
Script DisplayScore

Add Component

  - To change the position of the **GUI text** element, in the Inspector window, change the Pixel Offset values for X and Y as desired. For example, to position it in the upper left-hand corner, set your X and Y values to something like -500 and 400, respectively (these values will vary based on your screen resolution).
  - To make the text more visible, in the Inspector window, change Font Size to a larger value like 20 or 24.

- Finally, link the **GUI Text** element for the score to the **PlayerScoring** script
  - In the Hierarchy window, select the **First Person Controller**. Scroll down in the Inspector window until you see the parameters for the **PlayerScoring** script.
  - Click and hold on the **GUI Text** element that displays the score. While holding down the mouse button, drag the **GUI Text** element into the Score Text parameter in the **PlayerScoring** script.
  - After completing this, the **PlayerScoring** script should show the name of your **GUI Text** element after the Score Text parameter. This links the two objects together, so now the score should display correctly when you play the game.

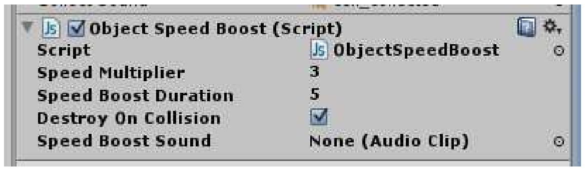


Player Scoring (Script)

Script	PlayerScoring
Score Text	GUITextScore (GUIText

2 Speed Boost from Object Collection

- Add **PlayerSpeedBoost** script to **First Person Controller**
  - Process is identical to the first bullet above, where you added the **PlayerScoring** script to the **First Person Controller**. Here you simply use the **PlayerSpeedBoost** script instead.
  - The only difference is that the **PlayerSpeedBoost** script is simpler; it does not have any parameters that need to be defined.
- Add **ObjectSpeedBoost** script to each of the Rigidbody objects you want to give a speed boost upon collection
  - Process is identical to the second bullet above, where you added the **ObjectScoring** script to the Rigidbody objects. Here you're adding the **ObjectSpeedBoost** script instead, which has a few different parameters, as shown below:



Object Speed Boost (Script)

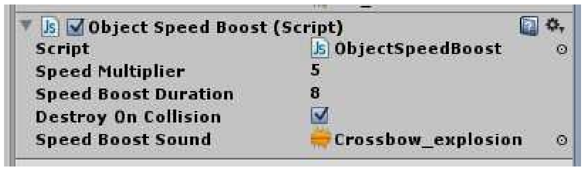
Script	ObjectSpeedBoost
Speed Multiplier	3
Speed Boost Duration	5
Destroy On Collision	✓
Speed Boost Sound	None (Audio Clip)

  - The parameters in the **ObjectSpeedBoost** script are: Speed Multiplier, Speed Boost Duration, Destroy On Collision, and Collect Sound. Speed Multiplier indicates how many times faster the player becomes (e.g. 3 would indicate three times faster). Speed Boost Duration indicates how many seconds the speed boost will last. Destroy On Collision and Speed Boost Sound are identical to **ObjectScoring**, determining whether the object will be deleted and which sound (if any) will be played upon collision.

    Note: If you don't want an object deleted after a collision, and both the **ObjectScoring** and **ObjectSpeedBoost** scripts are attached to an object, then Destroy On Collision must be unchecked in both scripts.

    Below shows an example parameterization that will increase speed by 5× for 8 seconds, will delete the object from the game, and will play the **Crossbow_explosion** sound.

  - Optionally, if you're creating many objects with the **ObjectSpeedBoost** script (and/or **ObjectScoring** script), you can make a Prefab object that already has the appropriate Tag, Rigidbody component, and one or both scripts, and then quickly create instances of this Prefab by dragging them into the game.

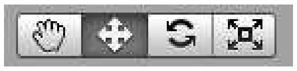

Object Speed Boost (Script)

Script	ObjectSpeedBoost
Speed Multiplier	5
Speed Boost Duration	8
Destroy On Collision	✓
Speed Boost Sound	Crossbow_explosion

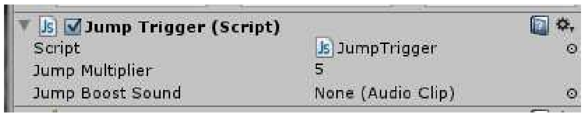
- Optionally add an Audio Source to the **First Person Controller**
  - Just like bullet three for **PlayerScoring**, if you want to play an audio sound when the player collects a CollisionObject, you need to an Audio Source component to the **First Person Controller**.

3 Super Jump Trigger

- Change the Tag of **First Person Controller** to Player
  - Select **First Person Controller** object in the Hierarchy window.
  - In the Inspector window, change the Tag to Player.
- Add one or more **SuperJumpTrigger** prefab objects into your Scene at the desired places
  - To facilitate ease of use, we created a Prefab for the **SuperJumpTrigger**. However, this Prefab was easy to make – it is nothing more than an empty object with a Box Collider, and Audio Source, and an attached **JumpTrigger** script.
  - In the Project window, select the Assets/CSCI 130 Assets/Prefabs/ folder. Inside you will find the **SuperJumpTrigger** prefab object.
  - For each location you desire to have a jump trigger in your scene, move your camera to that position and then drag the **SuperJumpTrigger** prefab object from the Project window into the Scene window. Each time you drag one into the Scene, it will create a unique instance of the jump trigger.
  - After creating an instance of the **SuperJumpTrigger**, you may position and re-size it as desired, either through the Inspector window, or via the Unity selection, translation, rotation, and sizing tools available in the upper left-hand corner of the Unity editor (see below):



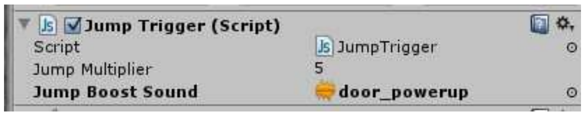
  - In addition, the **SuperJumpTrigger** includes a **JumpTrigger** script, which has the following paramters:


Jump Trigger (Script)

Script	JumpTrigger
Jump Multiplier	5
Jump Boost Sound	None (Audio Clip)

  - The parameters in the **JumpTrigger** script are: Jump Multiplier and Jump Boost Sound. Like the speed boost, Jump Multiplier indicates how many times higher the the player can jump when inside the jump trigger area, while Jump Boost Sound is an optional sound that plays if the Player jumps when inside the jump trigger area.

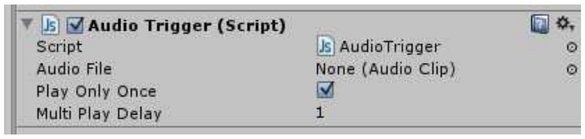
    Below shows an example parameterization that will increase the jump height by 5× and will play the **door_powerup** sound when the Player jumps (while inside the jump trigger area).
  - Unlike the scoring and speed boost gameplay mechanics, you do not need to add an Audio Source component to the **First Person Controller** in order to play the jump boost audio sound. The **SuperJumpTrigger** prefab already contains the requisite Audio Source component.



```
Jump Trigger (Script)
Script                 JumpTrigger
Jump Multiplier        5
Jump Boost Sound       door_powerup
```

4 Audio Trigger

- Change the Tag of **First Person Controller** to Player
  - Process is identical to the first bullet in **SuperJumpTrigger**, above.
- Add one or more **AudioTrigger** prefab objects into your Scene at the desired places
  - Process is again identical to the second bullet above, where you added instances of the **SuperJumpTrigger** prefab element into the game. The only difference is that the **AudioTrigger** has its own script, with different parameters, as shown below:

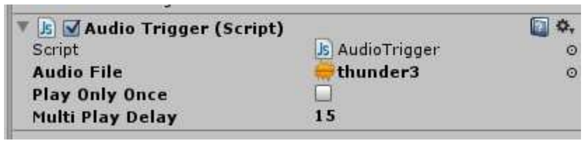


```
Audio Trigger (Script)
Script                 AudioTrigger
Audio File             None (Audio Clip)
Play Only Once         ✓
Multi Play Delay       1
```

  - The parameters in the **AudioTrigger** script are: Audio File, Play Only Once, and Multi Play Delay. The Audio File parameter is obviously the sound file that will be played when the Player enters the trigger area. The Play Only Once option is a checkbox that indicates whether the audio clip can be played more than once per game – if checked, the audio clip can be played only once during the entire game; if unchecked, it will play again whenever the Player re-enters the trigger area.

    The last parameter, Multi Play Delay is only relevant if Play Only Once is unchecked. It's purpose is to prevent the audio trigger from re-playing the audio clip too frequently by enforcing a minimum wait period until the audio clip can be played again.

    Below is an example parameterization that will play the **thunder3** sound when the Player enters the trigger area. It will play this sound each time the Player enters the trigger area, provided at least 15 seconds has passed since the sound file was last played.



```
Audio Trigger (Script)
Script                 AudioTrigger
Audio File             thunder3
Play Only Once         ☐
Multi Play Delay       15
```

  - Similar to the **SuperJumpTrigger**, it is not necessary to add an Audio Source component to the **First Person Controller** in order to play the audio clip. The **AudioTrigger** prefab already contains the requisite Audio Source component.
  - *Note:* Again, if you haven't already, be sure to remove the Audio Listener component on the **Main Camera**. A scene can only have one Audio Listener for audio to work correctly. Since both the **Main Camera** and the camera in the **First Person Controller** have an Audio Listener, you need to remove/delete one. While playing the game, the player should hear the audio from the perspective of the player controller, so the Audio Listener on the **Main Camera** is the one that should be deleted.
- Optionally modify the **AudioTrigger**'s play parameters, in the attached Audio Source component
  - The **AudioTrigger** includes an Audio Source component that specifies further parameters regarding how the audio clip is played. These include whether it is 2D or 3D audio, whether it's looping, the attenuation characteristics for 3D audio, etc. Modify these as needed for each individual audio trigger.