Exercise 17: Teddy Bear Explosions

**Getting Started**

Download the zip file below and unzip the file somewhere on your computer.

**Exercise 17 Materials**ZIP File

Open the project in Unity. You can do this by navigating to the Exercise17\Assets\scenes folder and double clicking scene0.

**Problem 1 - Get the explosion working**

Drag the Explosion prefab from the prefabs folder in the Project window onto the Hierarchy window. Run the game and watch the explosion play and play and play ...

Drag the Explosion script from the scripts folder in the Project window onto the Explosion prefab in the prefabs folder in the Project window; this also attaches the script to the instance of the prefab in the scene.

Open the Explosion script in your IDE and add code below the comment in the Update method to destroy the game object if the explosion has finished its animation. You can use the Unity Scripting Reference to explore how to use the Animator GetCurrentAnimatorStateInfo method to figure out how to do this (the anim field is an Animator object).

Run the game. The explosion animation should disappear after it finishes playing the first time. You'll have to watch closely because this happens pretty quickly. If you look at the Hierarchy window as the game runs, you'll see the Explosion game object disappear from the Hierarchy window when that game object is destroyed.

Delete the Explosion game object from the Hierarchy window.

**Problem 2 - Get teddy bears spawning in scene**

Create a TeddyBear prefab using the provided teddy bear sprite and TeddyBear script. Add a Rigidbody 2D component to the prefab, making sure you freeze rotation in z using the Constraints. Add a Box Collider 2D component to the prefab, making sure you set the Material field to the TeddyBearMaterial in the materials folder in the Project window.

If you want to, you can edit the box collider for the TeddyBear prefab. To do that, drag the prefab onto the Hierarchy window. Click the Edit Collider button in the Box Collider 2D component in the Inspector, then drag the green squares on the collider in the Scene view to make the collider fit the sprite more tightly. Click the Edit Collider button again to stop editing the collider. Click the Apply button next to Prefab near the top of the Inspector to apply your changes to the prefab.

Delete the TeddyBear game object from the Hierarchy window (if you have one there).

Remove gravity from the game.

Attach the provided TeddyBearSpawner script to the Main Camera, select the Main Camera, and drag the TeddyBear prefab onto the Prefab Teddy Bear field in the script in the Inspector.

Run the game and watch the teddy bears spawn and bounce off the edges of the screen and each other.

**Problem 3 - Kill teddy bears multiple ways**

Add code to the TeddyBear Update method to destroy the teddy bear if the death timer has finished. No explosion here, the teddy bear just "goes gentle into that good night." From within a script, we can access the game object the script is attached to using gameObject. That means we can destroy the TeddyBear game object the TeddyBear script is attached to using

Destroy(gameObject);

Play around with the TeddyBear TeddyBearLifespanSeconds field to change how quickly the teddy bear dies.

Add a TeddyBear tag to the TeddyBear prefab. To do this, select the TeddyBear prefab in the Project window and click the Untagged dropdown next to Tag at the top of the Inspector. Select Add Tag ..., click the + below and to the right of the List is Empty message, type TeddyBear as the New Tag Name, and click the Save button. Select the TeddyBear prefab in the Project window again, click the Untagged dropdown next to Tag at the top of the Inspector, and select the TeddyBear tag at the bottom of the dropdown.

Set the Prefab Explosion field of the Teddy Bear (Script) component of the TeddyBear prefab to the Explosion prefab.

Add code to the body of the if statement in the TeddyBear OnCollisionEnter2D method to instantiate the prefabExplosion and destroy the teddy bear. You already learned how to destroy the teddy bear above. To instantiate the explosion prefab at the teddy bear's location (which is, of course, where the explosion should be) you can use

Instantiate<GameObject>(prefabExplosion,transform.position,Quaternion.identity);

The \tt{Instantiate}Instantiate method is called a generic method because we tell it what type of object we're instantiating (in this case, a GameObject) between the < and the >. The first argument to the Instantiate method is the prefab we're instantiating. The second argument is the location where we want to instantiate the game object; transform.position is the location of the teddy bear. The third argument tells how much rotation to apply when we instantiate the game object, where Quaternion.identity means don't rotate at all.

Run the game again and watch both teddy bears explode when they collide with each other.