Exercise 16: Shrink and Grow

**Problem 1 - Grow out of control!**

Create a new 2D Unity project named Exercise16. Add a new scenes folder to the Project window and save the current scene as scene0. Add a new sprites folder to the project window and use your OS to copy a sprite of your choosing into that folder. Drag the sprite into the Hierarchy window to create a game object in the scene. Run the game and watch nothing happen.

Create a new scripts folder in the Project window and create a new C# script in that folder called Resizer. Open the new script in your IDE and add a documentation comment for the class. The Resizer class (script) will shrink and grow the game object over time.

We need fields to support both our standard timer and the resizing process. Write a line comment (just above the comment for the Start method in the script) indicating that the fields you're declaring next are for timer support. Declare a float  TotalResizeSeconds constant and set it to 1 just below your new comment. Declare a float  elapsedResizeSeconds variable and set it to 0 just below your new constant.

Make sure your code compiles without errors (don't worry about warnings -- for example, a warning saying you're not using the elapsedResizeSeconds field yet).

Write a line comment indicating that the fields you're declaring next are for resizing control. Declare a float  ScaleFactorPerSecond constant and set it to 1. Setting this constant to 1 will make the game object double in size in the first second, triple in size over the first 2 seconds, and so on. Declare an int  scaleFactorSignMultiplier variable and set it to 1. We'll use this variable to either grow (when it's 1) and shrink (when it's -1) the game object as appropriate.

Make sure your code compiles without errors.

Delete the Start method, including the comment above it; we don't need that method for this problem.

Add a line comment and code to the Update method to resize the game object. You can remind yourself how to change the localScale of the game object by looking at the YellowTeddyBear script from Exercise 11, but of course you can't just multiply x and y by 4. Instead, you need to use the resizing control fields you declared above and the Time.deltaTime value to change the scale properly. Whatever you come up with for your solution should make the game object twice as large in approximately a second. You can just see if it "looks right" at this point; we'll use the debugger to carefully check it soon.

Run the game. Swear because nothing happens. Attach the Resizer script to your game object in the Hierarchy window. You should see the game object grow (out of control!) as the game runs. Play around with the ScaleFactorPerSecond constant to make the game object grow faster or slower.

**Problem 2 - Shrink and grow**

Add code to the Update method to update the timer and check if it's finished. Look at the solution to Exercise 14 if you need help with this.

In the if body of the code you wrote in the previous step, add code to reset the timer and change the sign of the scaleFactorSignMultiplier field. In other words, if it's 1 make it -1 and if it's -1 make it one. Do this with a multiplication, don't write an if statement to do it!

Run the game and watch the weirdness.

**Problem 3 - Use the debugger**

We said in Problem 1 that we'd use the debugger to check the resizing, so let's do that now. In your IDE, set a breakpoint on the line of code that resets the timer. You set a breakpoint by clicking in the gray bar to the left of the line of code, at which point a red circle appears at that location.

Select Run > Attach to Process... (in MonoDevelop) or Debug > Attach Unity Debugger (in Visual Studio) and double click the Exercise16 Unity project in the popup (it should be the only Unity Editor process that shows up). Go to the Unity editor and run the game. After the game has run for approximately a second, you'll be back in the IDE at the breakpoint (you may have to open the IDE yourself). Hover the mouse over transform.localScale a few lines of code above the breakpoint. If you implemented the scaling properly, the value should be (2.0,2.0,1.0). Select Run > Stop (in MonoDevelop) or Debug > Stop Debugging (in Visual Studio) and remove the breakpoint by clicking on the red circle. Stop the game in the Unity editor.

Play around with the TotalResizeSeconds constant to change the duration of the growing and shrinking.