CGE 401 – CGE Game Dev Assignment #4 – **See due date on Blackboard**

**Step 1:**

* Download the Prototype 3 assets by [direct download](https://connect-prd-cdn.unity.com/20191004/b3882104-93e8-44a1-901c-eb8c0a3f7845/Prototype%203%20-%20Direct%20Download.zip?_ga=2.119332745.1270142453.1598919002-1565378913.1590020947) or from the [Asset Store](https://assetstore.unity.com/packages/templates/tutorials/create-with-code-prototype-3-run-and-jump-146039?_ga=2.82780054.1270142453.1598919002-1565378913.1590020947)
* Import the assets into Unity 2018.4.36 (do not use version 2019, 2020, etc.)
* Follow along with the Prototype 3 Follow Along video to complete Prototype 3
* Note: There are additional required steps in the Prototype 3 video and lecture notes (slides) that are not on the Create with Code website.
* Be sure to include Prototype 3 with your submission of Challenge 3 (separate unity project folders is fine)

**Step 2**:

* Download the Challenge 3 assets by [direct download](https://connect-prd-cdn.unity.com/20191004/5f8592bf-25d8-4f30-8911-fc4d037fb9ed/Challenge%203%20-%20Direct%20Download.zip?_ga=2.52359977.1270142453.1598919002-1565378913.1590020947) or from the [Asset Store](https://assetstore.unity.com/packages/templates/tutorials/create-with-code-challenge-3-balloons-bombs-and-booleans-146045?_ga=2.52359977.1270142453.1598919002-1565378913.1590020947)
* Import the assets into Unity 2018.4.36 (do not use version 2019, 2020, etc.)
* If you want to, watch the video introducing Challenge 3 [here](https://learn.unity.com/tutorial/challenge-3-bouncy-balloons-bombs-and-booleans?courseId=5cf96c41edbc2a2ca6e8810f&projectId=5cf9639bedbc2a2b1fe1e848&tab=overview), but **there will be additional requirements listed below**

**Challenge 3 Requirements Part 1 (Requirements from Create with Code)**

* The player can’t control the balloon - The balloon should float up as the player presses spacebar
* The background only moves when the game is over - The background should move at start, then stop when the game is over
* No objects are being spawned - Make bombs or money objects spawn every few seconds
* Fireworks appear to the side of the balloon - Make the fireworks display at the balloon’s position
* The background is not repeating properly - Make the background repeat seamlessly
* The balloon can float way too high - Prevent the player from floating their balloon too high
* The balloon can drop below the ground - Make the balloon appear to bounce off of the ground, preventing it from leaving the bottom of the screen. There should be a sound effect when this happens, too!

**Hints**

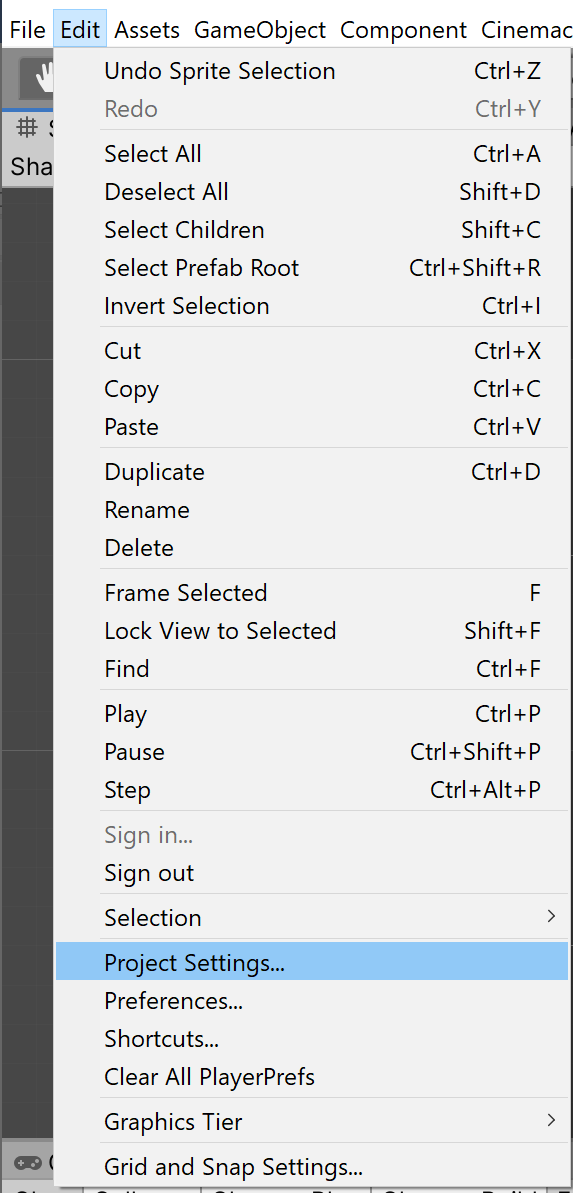
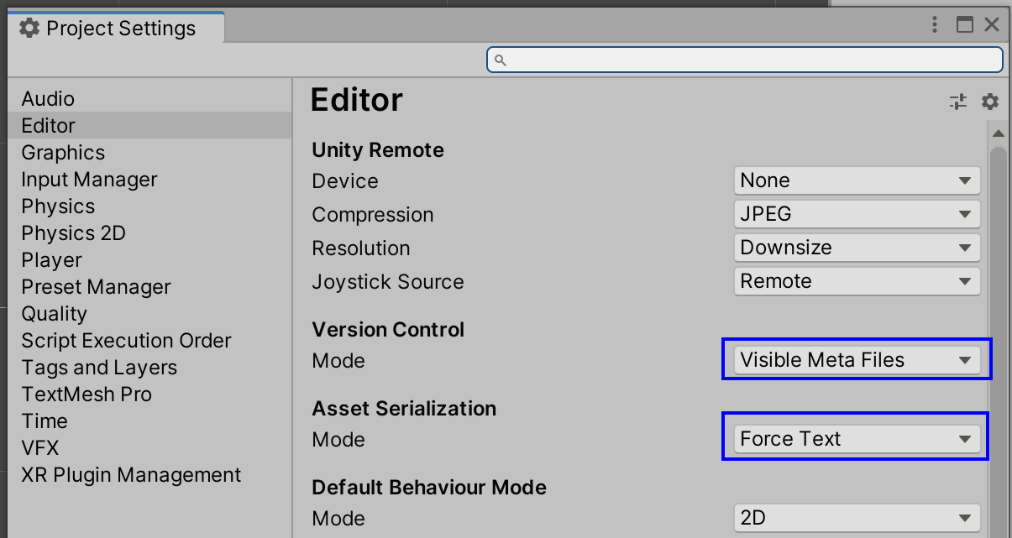
* **The balloon should float up as the player presses spacebar** *Hint* - There is a “NullReferenceExcepton” error on the player’s rigidBody variable - it has to be assigned in Start() using the GetComponent<> method
* **The background should move at start, then stop when the game is over** *Hint* - In MoveLeftX.cs, the objects should only Translate to the left if the game is NOT over
* **Make bombs or money objects spawn every few seconds** *Hint* - There is an error message saying, “Trying to Invoke method: SpawnManagerX.PrawnsObject couldn't be called” - spelling matters
* **Make the fireworks display at the balloon’s position** *Hint* - The fireworks particle is a child object of the Player - but its location still has to be set at the same location
* **Make the background repeat seamlessly** *Hint* - The repeatWidth variable should be half of the background’s width, not half of its height
* **Prevent the player from floating their balloon too high** *Hint* - Add a boolean to check if the balloon isLowEnough, then only allow the player to add upwards force if that boolean is true
* **Make the balloon appear to bounce off of the ground, preventing it from leaving the bottom of the screen. There should be a sound effect when this happens, too!** *Hint* - Figure out a way to test if the balloon collides with the ground object, then add an impulse force upward if it does

**Challenge 3 Requirements Part 2 (Additional Requirements)**

* In addition, the spawned objects only move left when the game is over rather than moving left until the game is over. Make the spawned objects stop moving left when the game is over (Important requirement left out of Create with Code)
* Display a score and increment the score when colliding with the money
* After collecting a set amount of money (e.g. 30), display the text “You win! Press R to try again!”
* On gameOver, display the text “You lose! Press R to try again!”
* When the game is over, make pressing the R key restart the game

**Uploading to GitHub**

If you are having difficulty uploading your unity projects to GitHub, see the video under Blackboard Modules -> Week 2 -> Creating a GitHub Repo for Unity Projects. If you are still having difficulty, see [this video](https://www.youtube.com/watch?v=qpXxcvS-g3g) and this [blog post](https://www.studica.com/blog/how-to-setup-github-with-unity-step-by-step-instructions).

**Simmer.io links: Build Prototype and Challenge projects to WebGL**

* Build your challenge and your prototype to WebGL and post them on a website where the public can play your game if they have the link, such as Simmer.io. You need to include two working simmer.io (or similar) links with your submission.
* See <https://www.youtube.com/watch?v=JZqTHjjtQHM> to see how to build to WebGL and posting to Simmer.io

**Required Deliverables**:

* Upload the following to Blackboard under this assignment:

1) a URL web address linking to one GitHub repository with separate project folders and files for your completed prototype and challenge and

2) two working URL web addresses linking to WebGL Builds – one for the prototype and one for the project.

* Please use one github repository link with both unity project folders in your one repository. You can use the same repository for all individual assignments, but you will need to submit your repository URL on Blackboard for each assignment.
* You must put the .gitignore file on Blackboard Modules in your git repository and commit and push the .gitignore file to your repo **before adding your project files or creating a unity project in your git repository**. You will lose points if you do not properly use the .gitignore file.
* You must set your github repo to public – be sure to **set your repo to public** so I can access your files.
* Include the **name of the scene** in Assets/Scenes/ you want me to open, and the path to them if the scene is not there.
* If your script files for the assignment are not in Assets/Scripts/, then note the path to your script files.
* To receive credit for this assignment, all script files that you write or change **must** include the header comment below with your name on it:

/\*

\* (Student Name)

\* (File Name)

\* (Assignment)

\* (Describe, in general, the code contained.)

\*/