Exercise 2: Writing Your First Unity Script

Clone your repository

1. Click on the appropriate link then accept the assignment to create your repository for submitting your work:

    a. **Gallant AM:** https://classroom.github.com/a/Oa4iFolq
    b. **Gallant PM:** https://classroom.github.com/a/XBS9uVSa
    c. **Nunn AM:** https://classroom.github.com/a/biVokKAN
    d. **Nunn PM:** https://classroom.github.com/a/VJg2XklT
    e. **Wijaya AM:** https://classroom.github.com/a/oLRphwwz
    f. **Wijaya PM:** https://classroom.github.com/a/vlL7dMyE

2. In GitHub Desktop, clone the repository you just created to your desktop.

Create your Unity project and prepare for GitHub tracking

3. Use Unity Hub to create a new 2D Unity project named Exercise2. Save the project in your new repository folder.
4. Once the project opens in Unity, go to File Explorer and move the **_UnityProjectRoot.gitignore** file into the Unity project folder and rename it to **.gitignore**
5. Go to GitHub desktop and commit your changes with the message: "Create initial Unity project" Make sure there are only about 30 files being committed.
    a. If you have thousands of changed files, return to step 2 to make sure you've named the gitignore file properly and that it is placed at the root of the *Unity* project not in its original location. Ask for help if you are unsure.
6. Push your changes to the remote.

At this point you are ready to proceed with this assignment. We encourage you to make interim commits as you go. Use your commit message to indicate which step (e.g.: "Completed through step 5").

Problem – Output your favorite three games

7. In Unity Editor, change the name of SampleScene to Scene0.
8. Add a Scripts folder and add a new C# script named PrintFavoriteGames.
9. Open the script, fill in the comment near the top of the script, and delete the **Update** method.
10. Next, add code to the **Start** method to output the names of your three favorite games.
11. Attach the script to the Main Camera in the scene and run the game to see the output in the Console window.
12. Go to GitHub Desktop and commit your changes with message: "Completed through step 8"
13. Test your game and fix any issues.

Caution: Don't try to use **Console.WriteLine()** in your Unity scripts to print to the Console window; you should use **Debug.Log**

Submit your work

1. Make a final test of your code and copy the output from the console window.
2. If you need to make any additional changes to your code, make sure you commit them.
3. By committing and pushing your updates to GitHub you have submitted your assignment on GitHub Classroom.
4. Return to CodeHS. Paste your output into the code window to complete the assignment.