

Exercise 13: More Unity Practice

Although this exercise isn't worth any points, it gives you valuable programming experience. You're almost definitely going to have to complete the exercises to succeed in the course.

Getting Started – Clone your repository

1. Click on the appropriate link below and accept the assignment to create your repository with starter code and for submitting your work:
 - a. Gallant AM: <https://classroom.github.com/a/0S4JzXkY>
 - b. Gallant PM: <https://classroom.github.com/a/y5DmvShq>
 - c. Nunn AM: <https://classroom.github.com/a/MSJudBUu>
 - d. Nunn PM: <https://classroom.github.com/a/lG8VynWU>
 - e. Wijaya AM: <https://classroom.github.com/a/Og7sNlVV>
 - f. Wijaya PM: <https://classroom.github.com/a/Ow3uoWbc>
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 **Exercise13**. 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.
 - b. 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 1 - Create a project and add sprites

7. Select the Main Camera and change Size to 3.
8. Rename **SampleScene** to **Scene0**.
9. Add a Sprites folder in the Project window and use your OS to add the two teddy bear images I provided to that folder.
10. In GitHub Desktop, **commit** your code with the comment: "Completed Problem 1"

Problem 2 - Add sprites to scene

11. In Unity Editor, drag the green teddy bear from the Sprites folder in the Project window onto the Hierarchy window.
12. Create a new Prefabs folder in the Project window and drag the sprite from the Hierarchy window onto the Prefabs folder in the Project window.
13. Rename the prefab to TeddyBear.
14. Rename the game object in the Hierarchy window GreenTeddyBear.
15. Adjust the Position X and Y values in the Transform component of the GreenTeddyBear to move it toward the upper left of the scene.
16. Drag the TeddyBear prefab from the Project window onto the Hierarchy window.
17. Rename the new game object in the Hierarchy window PurpleTeddyBear.
18. Adjust the Position X and Y values in the Transform component of the PurpleTeddyBear to move it toward the lower left of the scene.
19. Drag the purple teddy bear from the Sprites folder in the Project window onto the Sprite value of the Sprite Renderer component in the Inspector pane. As you can see, this changes the sprite used by the sprite renderer, turning the teddy bear purple.
20. In GitHub Desktop, commit your changes with the message: "Completed Problem 2".

Problem 3 - Make teddy bears move

21. In Unity Editor, select the TeddyBear prefab and add a Rigidbody 2D component.
 - a. To do this, click the **Add Component** button near the bottom of the Inspector and select **Physics 2D > Rigidbody 2D**.
22. Run the game and watch both teddy bears fall.
23. Remove gravity from the game by selecting **Edit > Project Settings > Physics 2D** from the main menu bar and setting the Y component of Gravity to 0.
24. Create a new Scripts folder in the Project window and add the **TeddyBear.cs** file I provided to that folder.
25. Open up the script in Visual Studio and add code below the comment in the **Start** method to get a teddy bear moving to the right at a reasonable speed.
26. Compile your code in Visual Studio to make sure you don't have any compilation errors. Once your build succeeds, close Visual Studio.
27. Attach the TeddyBear script to your GreenTeddyBear game object and run the game. You'll see that the green teddy bear moves to the right but the purple teddy bear doesn't move.
28. What we want to do next is apply the changes we made to the GreenTeddyBear game object back on the prefab so that all instances of the prefab share those changes. To do that,

- a. Go to the Prefab area near the top of the Inspector, click the Overrides dropdown, and click the Apply All button on the right.
 - b. This applies the changes you made to the instance of the prefab (the game object in the scene) to the prefab itself.
29. Run the game again. Both teddy bears should now move to the right.
30. In GitHub Desktop commit your changes with the comment: "Completed Problem 3".

Submit your work

31. Make a final test of your code to make sure it is working as expected.
32. If you made any additional changes to your code, make sure you commit them before **pushing** your changes to the Remote.
 - a. By committing and pushing your updates to GitHub you have submitted your assignment on GitHub Classroom.
33. Return to CodeHS and respond to the prompt to complete this assignment.