# Course 2 Unit 2 Exercise 7: Finally, A Driving Game!

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 then accept the assignment to create your repository for submitting your work:
    a. Gallant AM: https://classroom.github.com/a/05h7feOV
    b. Gallant PM: https://classroom.github.com/a/5Nm3UPNz
    c. Nunn AM: https://classroom.github.com/a/dAUiPVpV
    d. Nunn PM: https://classroom.github.com/a/G3BwAyR7
    e. Wijaya AM: https://classroom.github.com/a/obpYjqqR
    f. Wijaya PM: https://classroom.github.com/a/8N_sKV3M
2. In GitHub Desktop, clone the repository you just created to your desktop.

## Create your Unity project and prepare for GitHub tracking

1. Use Unity Hub to create a new 2D Unity project named Exercise7. Save the project in your new repository folder.
2. 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**
3. 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.
4. 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 a sprite

1. Rename the SampleScene as Scene0.
2. Add a new Sprites folder and use your Operating System to copy a sprite of your choosing into that folder.
3. Drag the sprite into the Hierarchy window to create a game object in the scene.
4. Run the game and watch nothing happen.
5. In GitHub Desktop, commit your changes with a meaningful message.

## Problem 2 - Drive horizontally

1. Create a new Scripts folder and create a new C# script in that folder called Driver. Open the new script in Visual Studio and add a documentation comment for the class. The Driver class (script) drives the game object based on keyboard input.

2. Add a constant called `MoveUnitsPerSecond` to store how many units your game object moves per second below the line that starts public class.
3. Delete the `Start` method from the script.
4. Add code to the body of the `Update` method to save the value on the Horizontal input axis (already provided in the default Unity project) into a variable called `horizontalInput` and to check if that value is non-zero.
    a. If there is input on that axis, change the x position of the game object (using a local variable as usual).
    b. The appropriate amount to change the x position is `horizontalInput * MoveUnitsPerSecond * Time.deltaTime`.
    c. The Keyboard Processing lecture covers keyboard input processing in detail.
5. Attach the Driver script to your game object in the Hierarchy window and run the game to drive horizontally.
6. In GitHub Desktop, commit your changes with a meaningful message.

### Problem 3 - Drive vertically

1. Add code to the body of the `Update` method to handle input on the Vertical input axis (already provided in the default Unity project) to change the y position of the game object.
2. Run the game to drive vertically (and horizontally if you'd like).

## Submit your work

1. In GitHub Desktop, commit your changes with the message: “Ready for grading”.
2. Push your changes to the remote.
    a. By committing and pushing your updates to GitHub you have submitted your assignment on GitHub Classroom.
    b. If auto-grading is enabled, this will also check your code and provide automatic feedback on your code.
3. Return to CodeHS and respond to the prompt.