Hangman Assignment Instructions

# Introduction

For this assignment, you are to write a program that plays the game of Hangman.

The computer selects a secret word at random from an array of words read from a file (provided). The program displays dashes for each letter that needs to be guessed. The player guesses a letter. If the letter is in the word, the word is redisplayed with all instances of that letter shown in the correct positions along with any letters correctly guessed in the previous turns and dashes for the letters that still need to be guessed. If the letter does not appear in the word, the player is penalized with a new body part added to the hangman. The player keeps guessing until either all the letters in the word have been guessed or twelve incorrect guesses were made.

# Requirements

You are free to determine the layout of the user interface for the game as you see fit. For example, you could choose to have a button for each letter and, after the player clicks on a button for a guess, make the button not interactable. You could also have an input field to get the player’s selection.

* A main menu that includes a full-screen background, the title of the game, a button to start the game and a button with instructions.
* In the game, the secret word is randomly selected from the list of words provided in the words.txt file. There are 100 words in that file. They should be stored in an array or a list.
* The player is allowed up to 12 incorrect guesses
* Adequate feedback should be provided to the user after each guess. You can choose to display an image or a number of guesses left. But the user interface should let the player know how many guesses they have left, see the updated guessed word, and the letters that were guessed incorrectly.
* You may use the images provided for the hangman or use your own. The images must be school appropriate. You may adjust the maximum number of incorrect guesses as needed. Images are NOT required. But the user interface should display the number of guesses left.
* If the player guesses the secret word, a message should be displayed announcing a victory and the player should be able to play again.
* If the player does not guess the secret word after 12 incorrect guesses, a message should be displayed announcing the loss and providing the secret word. The player should be able to play again.
* All UI elements, variables, and methods must be named meaningfully and follow naming conventions discussed during the lecture.
* All UI elements must be properly anchored to ensure the interface adapts well to different resolutions. Text should be easy to read.
* The project must be organized and named as dictated in the submission guidelines of the course.

# Submission

Your submission must include your compressed Unity project named Hangman-LastName, and your compressed build named HangmanBuild-LastName.

# Sprite Sheet Slicing

If you decide to use the spritesheet provided for the hangman, here are the steps to follow to slice it.

* In your project window, right-click and select the option Import New Asset. Browse to find the image you are importing.
* In the project window, select the image you just imported. In the Inspector, the Texture Type should be set to Sprite (2D and UI). If that is not the case, change it. For the Sprite Mode property, use the drop down menu to select Multiple. Once these two options have been set correctly, click on Apply.
* In the Inspector, while your image is still selected, click on the Sprite Editor button. The Sprite Editor window opens.
* In the Sprite Editor window, click on the Slice menu and select, for the type, Grid by Cell Count.
* Enter 3 for the number of columns (C) and 4 for the number of rows (R). You do not need to change anything else. Click on the Slice button.
* Close the Sprite Editor window. A message will appear “Unapplied import settings for …”. Choose Apply.
* In the project window, take a look at your image. If you click on the arrow, you will see the individual sprites.