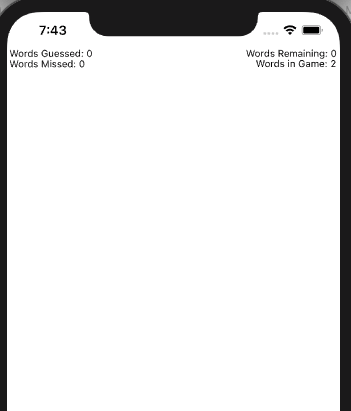
**Assignment 5**

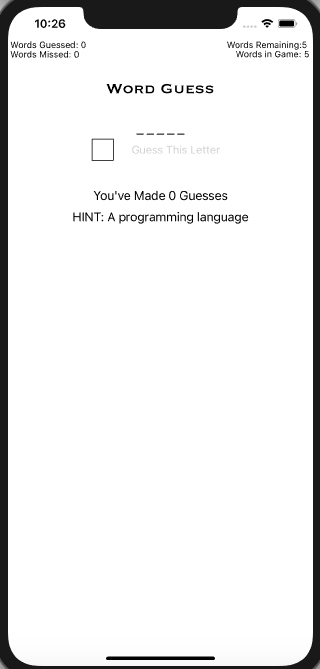
**Points 30**

**Please follow the following instructions to complete this assignment.**

1. Open Xcode from the launchpad of your mac.
2. Click on create a new Xcode project. Select the iOS template and click on the App application.
3. Click on next which will prompt you to choose options for the project.
4. Provide product name as **Lastname\_WordGuess**, “**nwmsu**” for organization identifier, “**storyboard**” as interface and swift as language.
5. Click on next and select an appropriate location to save your app and click on create. A project directory will be loaded.
6. From the project navigator click on “Main.storyboard” file, a blank mobile screen will be loaded where the required fields for an app need to be added.
7. Add 4 labels Words Guessed, Words Missed, Words Remaining, Words in Game. Words Guessed and Words Remaining should be in a horizontal line, add these two labels to a stack. Words missed and Words in Game should be in a horizontal line and also add these two labels to a stack. Now select all the labels and add them to a stack. Give the constraints to the stack as Top,Left,Right – 4,8,8 resepectively, Give a height of 40/50. In the attribute inspector, Alignment and Distribution should be Fill.
8. Create IBOutlet for all the labels and give them the names as **wordsGuessedLabel, wordsMissedLabel, wordsRemainingLabel, totalWordsLabel.**



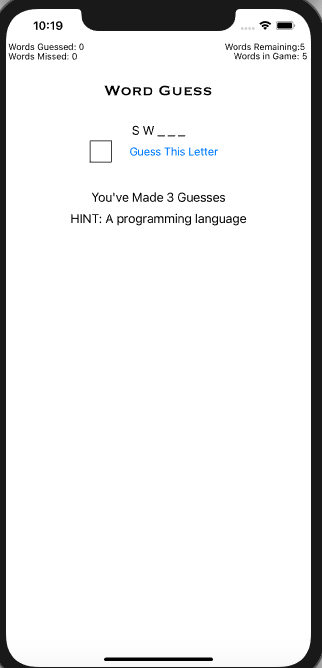
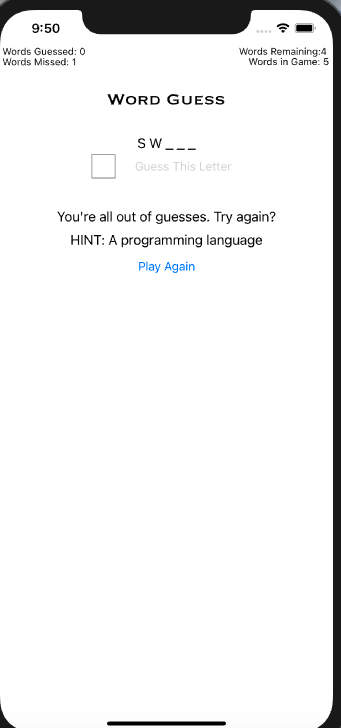
1. Add another label for the title of the app and name it as Word Guess. Use your own styling, Add constraints to this labels Top, left, right – 8,16,16 and also add a height constraint of your own.
2. Add in another label and fill it with “\_ \_ \_” , text in the label should be centered. Constraints are as follows: Top, left, right – 8,16,16 and give it a height : 20. Make a IBOutlet Connection and give it a name as **userGuessLabel.**
3. Now add a Text Field with the constraints Height: 34, Width: 30. In the attribute Inspector select the Spell Checking to NO, Correction: NO, Capitalization to: All Characters. Make a IBOutlet connection for this textField and give it a name **guessLetterField.**
4. Now add a button beside to the above Text Field and give the text as Guess a Letter**.** Constraints are Width: 150, Height: 30. Add the above button and Text Field to a stack and add a Top : 8 constraint for this stack. While making a IBAction connection, give it a name **guessLetterButtonPressed.**
5. Now add a Label below the stack which is a Hint label, and give the constraints as top,left,rigtht,height : 8,16,16,40. Name it as **hintLabel** while making a IBOutlet connection.
6. Now add in another label below the above label for the game status. Give the constraints as Top,left,right,height : 8,16,16,80. Make sure text will show on multiple lines and not be cut off with “…”. Make the default text as “You have Made Zero Guessess” and make sure the text is centered. Name it as **guessCountLabel** while making an IBOutlet connection.
7. Now add a play again button below the label and give the constraints as Top,width,height: 8,90,30. Make the button hidden by default. Name it as **playAgainButtonPressed** while making and IBAction connection

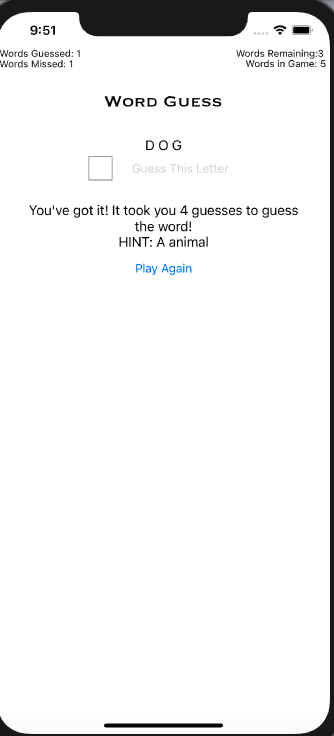


1. Now we are done with the UI part. Make sure all the connections for the controls are made correctly.
2. Now declare an array with atleast 5 words and the related hints. Whenever we build the app, userGuessLabel in the UI should update with number of “\_” same as the letters in the word of the array.
3. Initially totalWordsLabel should contains number of words in the array. Words Guessed : 0, Words Remaining : 5, Words Missed : 0
4. When ever the guessed word is correct/wrong these labels should be updated accordingly.
5. guessLetterField Label should take only the last character even though you enter any number of characters. Type in a letter in this Text Field using the keyboard of the simulator.

If the keyboard is not showing up, then select I/O option for the simulator, then you can find Keyboard and then toggle the keyboard. When ever you press the Guess a letter button, keyboard should resign(use resignfirstResponder)

1. Guess a letter button should enabled only if there is any text in the Text Field else disable it (use editing changed event)
2. Declare a constant maxNumOfWrongGuesses as 10. You are allowed to make only 10 wrong guesses for each word.
3. guessCountLabel displays the count of the guesses you have used. You can refer it from the below picture.
4. If you used all the available guesses for a word, guessCountLabel should display “You have used all the available guesses, Please start again” please refer to the image below for exact message and play again button should be enabled.
5. When ever all the words in the array are completed, guessCountLabel should display “Congratulations, You are done, Please start over again ” and all the app should start over again resetting all the labels.



BONUS:

1. In the Text Field attribute inspector, In the Text Field Traits, select the Return Key as done. Now when we build the app, the keyboard of the simulator shows done key. When ever you press the done the key, same action should be performed as that of Guess a letter button.
2. When ever you have guessed the word correct/wrong, related image for the word should be displayed using aniations with spring damping.

**Please submit your app as compressed file, your compressed files should contain Lastname\_WordGuess folder and Lastname\_WordGuess.xcodeproj file. Please check your submission by downloading the submitted file and rechecking in xcode.**