

Assignment #5

Due: December 2nd, 11:59PM

Goal: Create a simplified hangman game in App Inventor.

Overview: As in a traditional hangman game, the secret phrase should appear as underscores (only one word is required for this game, but if you would like to try multiple words, you may). The user should guess one letter at a time. If the guess is correct, the letter should be displayed in the proper location(s) in the word, and the letter should be added to the list of guessed letters. If the guess is incorrect, the appropriate body part should be displayed, and the letter should be added to the list of guessed letters. After six incorrect guesses (head, body, 2 arms, and 2 legs), the user loses, and this should be displayed on the screen. If the user correctly guesses the word, then the user should also be notified of their success. You do not need to ask the user to play multiple games (i.e., game ends after one round).

Submission: Download the source of your assignment from the App Inventor site and upload to the appropriate elearning assignment location. To do this, click on My Projects. Select the checkbox next to your hangman game, and click on More Actions, select Download Source. This should produce the zip file that you must submit. If you have any questions, please email Dr. Gray or Amber prior to submitting.

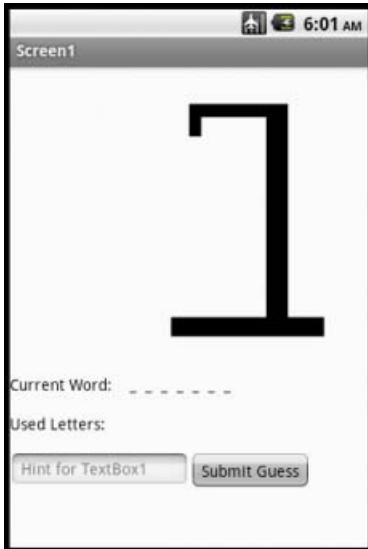
Requirements:

- Procedural abstraction must be used to separate key functionality (e.g., check if current guess is in the secret phrase, check if game over, etc.)
- Must use a list to represent the secret phrase
- Must use iteration (either type of for loop) to process lists, and conditionals (if statements) as part of the solution
- Must display the list of previously guessed letters after each guess
- Must display the secret phrase (underscores initially, with letters revealed after a correct guess)
- Must draw/display the hangman figure after each missed letter

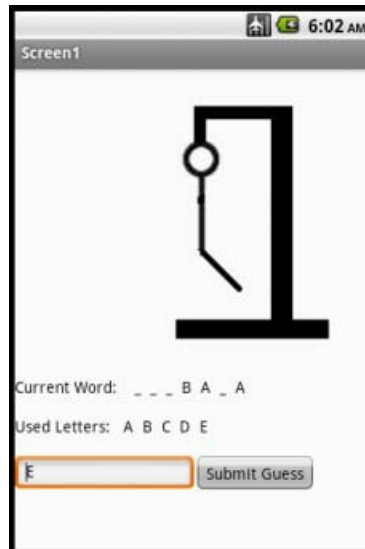
Assumptions and Suggestions: The following are comments aimed at making the assignment more manageable during the end of the semester. We will also spend part of Monday (Nov 28) discussing implementation ideas for this assignment.

- No need to check if new guess is already in used letters; assume user only enters letters that are new and have not yet been selected (i.e., you do not need to report a double guess)
- Use the upcase function to convert the user input to an uppercase letter before processing
- Create a procedure that takes the characters in a list and returns a string result; we will discuss the implementation of this on November 28th and give suggestions for other needed procedures

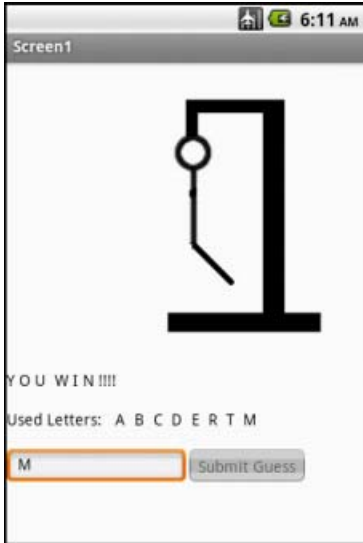
Screenshots: Below are some sample screenshots during various phases of game play, from start to end (two cases of ending a game). Your solution should resemble these screenshots. In the following, the sample phrase is "RTRBAMA"



Start of game



After guessing "A B C D E"



Win: After guessing "A B C D E R T M"



Lose: After guessing "A B C D E F G H"