Finnbarr O'Callahan
Luna Amador

Java Bootcamp: July 15-19

Final Project Handout: HANGMAN

OBJECTIVES

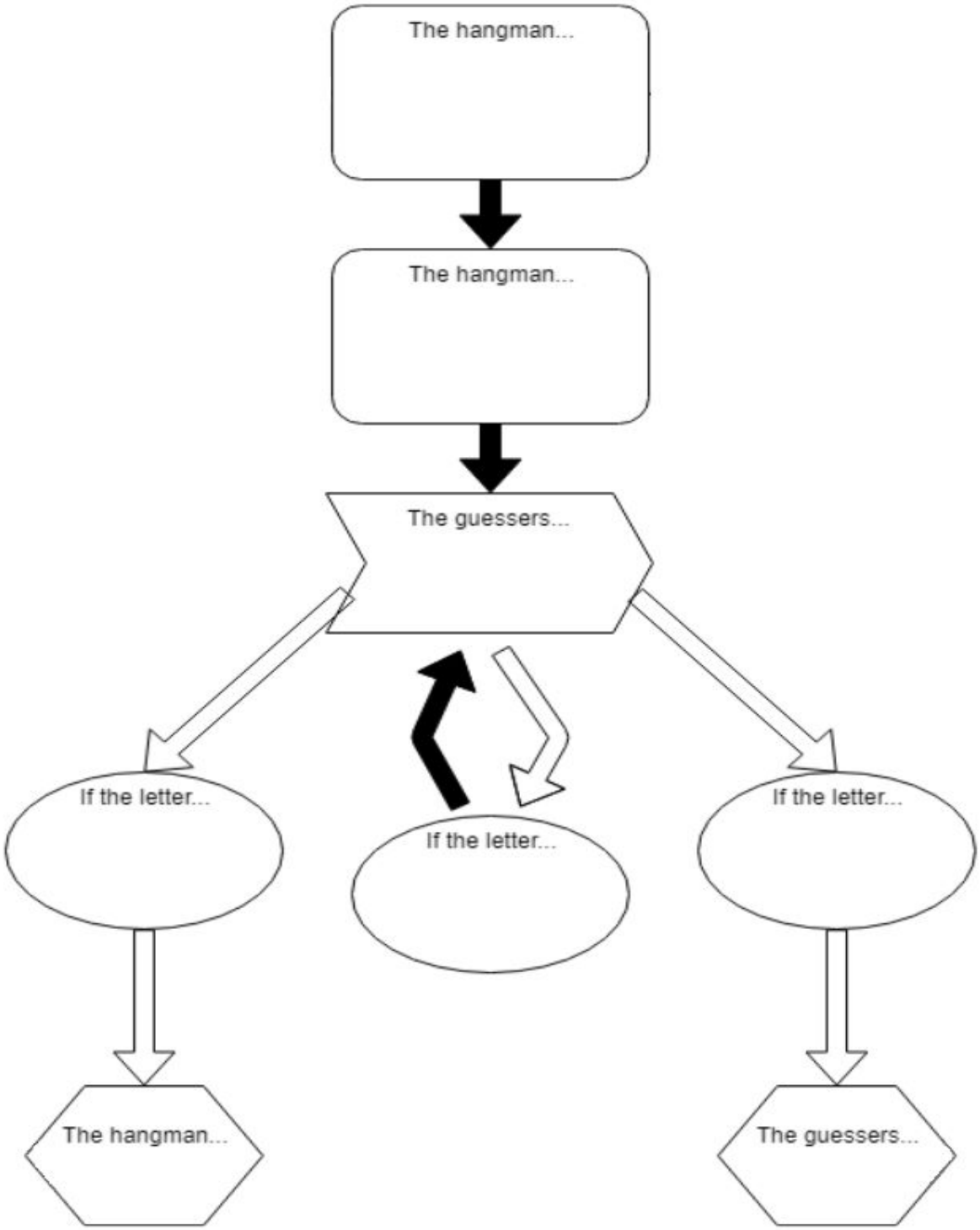
- This project will challenge you in two ways:
    - First, you'll have to use everything we've taught you to make a fully functional game
    - Second, you'll need to translate the rules of hangman into code by first filling in the logic diagram below
- Once you've completed your code, you'll be able to challenge your classmates & instructors to a match (and test your work in the process!)

RULES IN ENGLISH:

1. The hangman comes up with a word, and then draws the correct number of blank spaces "__" for letters.
2. The guesser(s) keep guessing letters until they either fill in the entire word or run out of lives.
    a. If the letter is in the word, then it replaces all the blank spaces where it appears. The hangman must remember that the letter has already been guessed, and no lives are lost.
    b. If the letter has already been guessed, then the guessers must try again. No lives are lost.
    c. If the letter is not in the word, then the hangman must keep track of it. One life is lost.

- BONUS: In Hangman, spaces shouldn't count as letters. We need to:
    - Modify the rules so that players don't need to guess spaces
    - Modify the game output so that spaces aren't printed as underscores "__"
        - This WILL require you to modify the printResults function we've written for you!
- SUPER BONUS: Modify the printResults function to print your own custom-designed platform and character!

RULES IN LOGIC DIAGRAM:



FUNCTION CHEAT SHEET:

- Character.toUpperCase(***exampleCharacter***)
    - returns ***exampleCharacter*** as an uppercase Character if it is currently lowercase
- ***exampleString***.toUpperCase()
    - returns the entire string ***exampleString*** where each Character has been made uppercase
- ***exampleArrayList***.add(***exampleCharacter***)
    - Adds a single Character (or any other type of variable) onto the end of this ArrayList
- ***exampleArrayList***.contains(***exampleCharacter***)
    - Returns “true” if ***exampleCharacter*** is found within ***exampleArrayList*** and “false” otherwise
- new String(***exampleArray***)
    - If ***exampleArray*** is an Array of Characters, then this will create and return a new String composed of all the Characters in ***exampleArray***