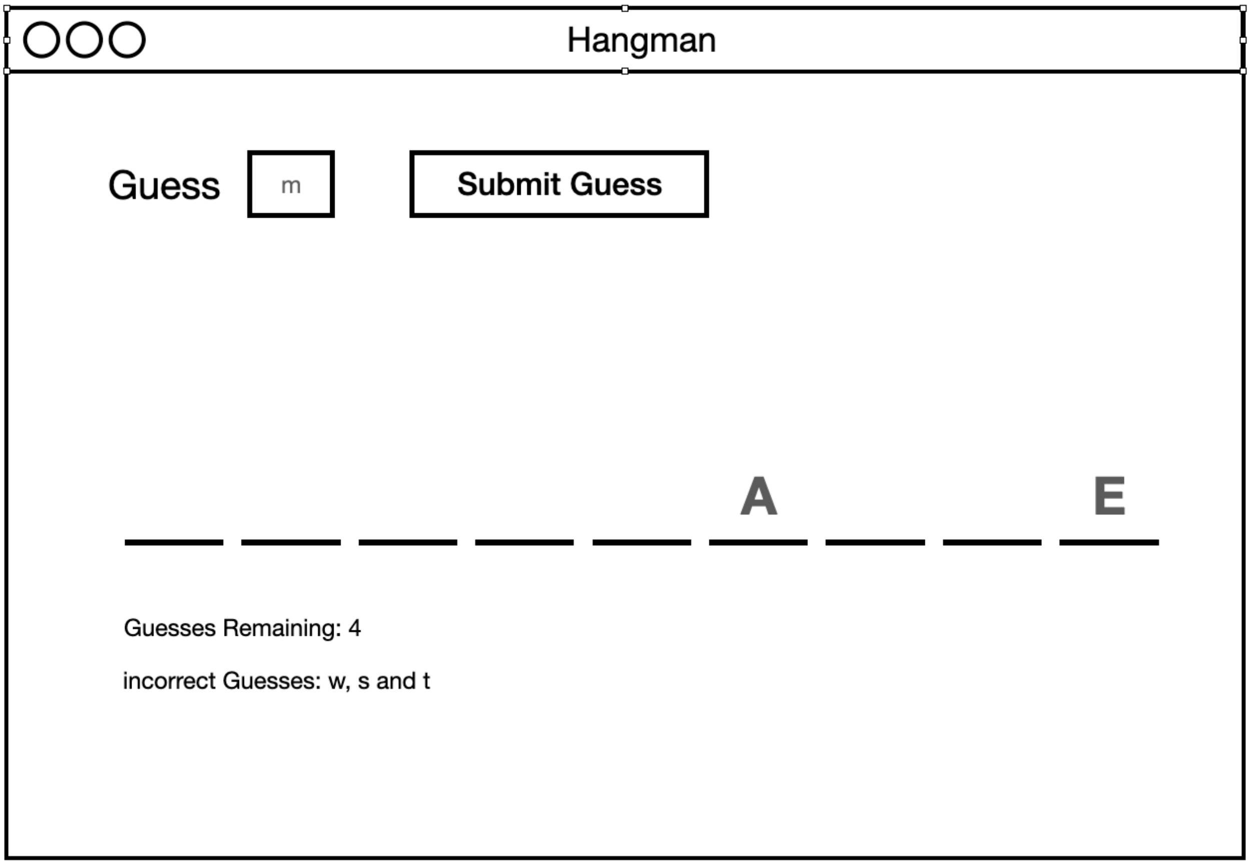
Extended Java Resit Final Assignment

In this assignment you are going to write a Java implementation of the classic game. Hangman. In the game players have to guess all the letters in the selected word before they run out of lives. (Or a hangman has been completely drawn.) when complete a simple implementation might look like the following. In the example below the drawing of the gallows and victim has been replaced by a simple guesses remaining label, you can represent this however you want.



Guesses Remaining: 4

incorrect Guesses: w, s and t

The main rules of the game are:

- Words are selected from a word list provided in the IntelliJ template. They are between 7 and 10 letters long.
- A new word is selected each time the game is run. This will be the target word players have to guess
- Players guess a single letter at a time. Your implementation should check for erroneous input.
- At the start of the game the player has 7 guesses (gallows, head, body, right arm, left arm, right leg, left leg)
- The player wins if they correctly guess all the letters in the target word.
- The player loses when they run out of guesses.
- You should create a Java Swing GUI. You must not use interface builder software or other GUI libraries.

- The game should display the target word in the console (this happens automatically if you use the static method in Main). This will help us mark your game.

The template includes a Main class that loads in the target words list. There is also a static method that returns a selected word and displays it to the console. Use this class as the launch point for your other classes that make up the games GUI.

Marking guide

Your implementation will be marked out of 50 as follows.

- **15 marks** for basic functionality. Have you implemented a playable version of hangman.
- **10 marks** for error checking and stability. Does the game correctly detect incorrect input and not crash.
- **10 marks** for an effective GUI and use of Swing components and layout
- **10 marks** for extending beyond the basic game implementation and adding additional features
- **5 marks** for code quality and modularity.

It’s up to you how you implement the game but for the extension marks we will be looking for you to have used the language features and techniques shown in class to make the game more interesting. But please remember the requirements above and keep your game markable.