**Hangman Functional Specifications**

Alex Xu, Brian Jeong, Pratik Gongloor

Type

WrongGuesses

Hangman

Slot

Man

**Main Class**

Requirements

The main class chooses a random word from the dictionary.txt file and loads the game. It constructs a Type object

Functional Specifications

1. Reads dictionary.txt and selects a random word
2. Creates a Type object with the word and runs it

**Type Class**

Requirements

The Type class generates the JPanel GUI and creates a separate pane for each "function." Additionally, it creates a type field that processes a guess. It uses a HashSet to avoid duplicate guesses, as well as other measures to ensure that a single alphabetic lowercase character is processed. Then, depending on whether or not it is found in the word, the letter is sent to either Slot or WrongGuesses for further processing. If Type receives a boolean flag that the game has ended, it ends the game.

Functional Specifications

1. Called from main class with a random dictionary word.
2. Creates and initializes Slot and WrongGuesses objects
3. Receives an input in the text field upon action event
4. Ensures that the input is a single, alphabetic lowercase character that has not yet been guessed
5. Determine whether or not the letter exists in the word; update text field depending on status
6. Send character to WrongGuesses if incorrect guess, send to Slot if correct guess
7. End game if Slot or WrongGuesses indicate that the game is over

**WrongGuesses Class**

Requirements

The WrongGuesses class constructs a GUI display for the Man object and also creates a text area for the wrongly guessed characters. The Man object is created. Also, the text area that contains the wrongly guessed characters is continually updated whenever a character is guessed, and once a wrong character is updated, the Man object advances a state (game terms: gains an appendage).

Functional Specifications

1. Called from the Type class with a given area and the picture state.
2. Man object is created with the given JLabel picture state.
3. Method public Boolean wrongGuess( char c ) accepts a character guessed in Type class.
   1. TextArea is updated by appending the wrongly guessed character to it.
   2. The state of Man Object is updated
      1. Hang\_state is incremented, thus advancing to the next hangman animation

**Hangman Class**

Requirements

The Hangman class compiles the whole program. It randomly picks a word from the given dictionary.txt file and creates the whole display for the Hangman game.

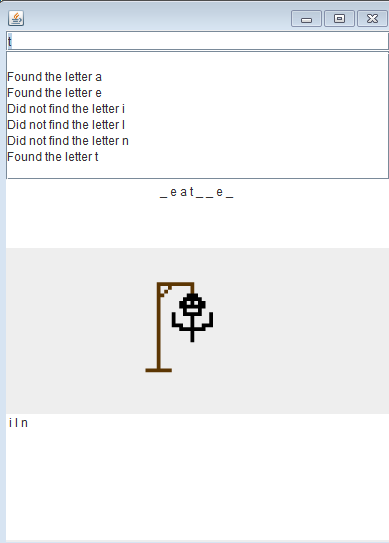
Functional Specifications

1. Random integer is determined from 1 - 19912 and word is chosen from dictionary.txt
2. GUI frame is created
3. Type object is created with the randomly chosen word as the parameter

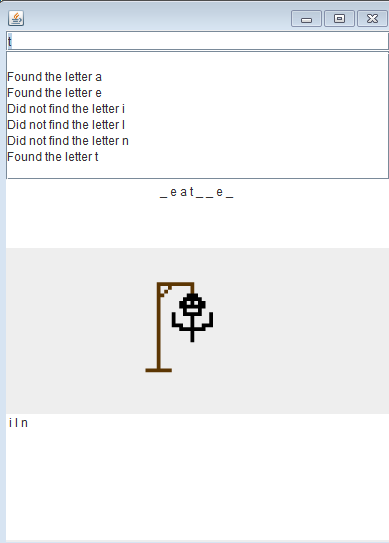
**Slot Class**

Requirements

The slot class keeps track of the letters that are guessed and if they correlate with the word that the program randomly chose. The class creates an array of underscore characters that represent the characters in a given word. Then, the class receives a letter guessed by a user, and if the letter is in the word, the underscore is replaced by the letter. The array is then displayed and updated with every guess.



Functional Specifications

1. Called from the Type class with given String str and JTextPane area
2. Receives randomly chosen word from the dictionary.txt file and creates an character array with “\_”, with one for each letter in the word. Creates text area for slots.
3. Method public Boolean rightGuess( char c ) receives a character guessed by the user. If the character is in string, the underscore that is in the index of the correct character is then replaced by the character. If the character array is filled and represents the randomly chosen word, the method gameOver() is called.
   1. Every time rightGuess( char c ) is called, it is displayed with display().
   2. gameOver() displays the randomly chosen word.

**Man Class**

Requirements

The Man class controls which animations are displayed using the int variable called hang\_state. The display pane is constructed and the animation/man pixelated art is painted onto the display. Finally, when the user loses the game, the game over animation is played.

Functional Specifications

1. Hang\_state is set to 0 and man\_display is set as the first png file.
2. The display for the pictures is constructed
3. The png file is painted onto the display
   1. Total of 5 hangman states
   2. When hang\_state is greater than 5, a Game Over gif animation is called and played on the display
4. The advanceState() method would increment the hang\_state and update the display with a new png file.