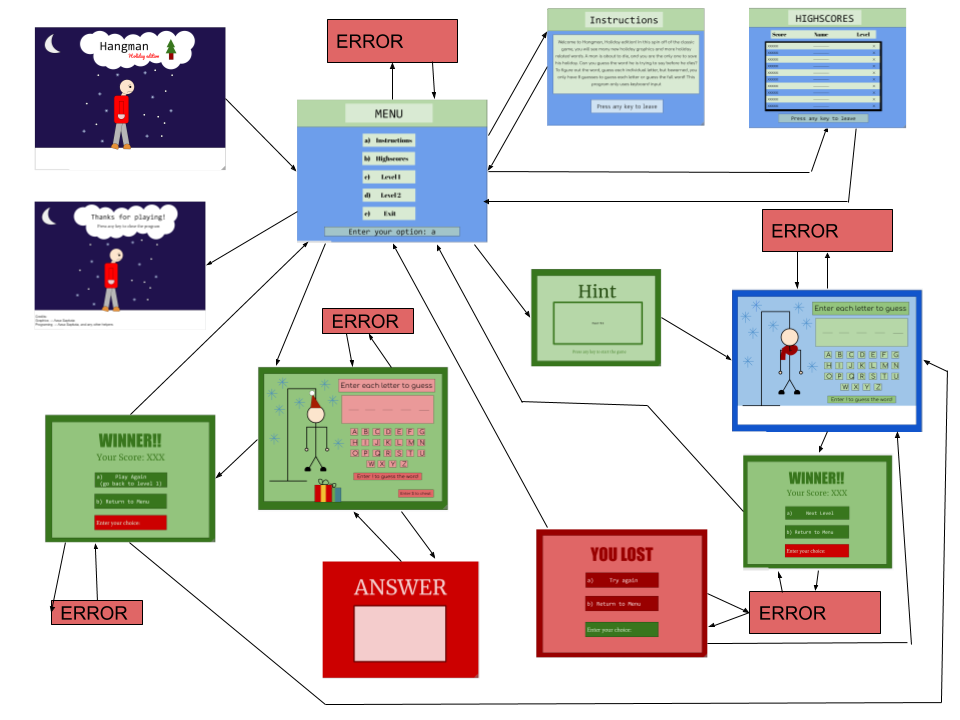
**Theme:** Holiday

**Title:** Holiday Hangman

**About:** This hangman is a new holiday adventure where you the user must guess a word to save a man’s life.

**Program Flow:**

****

**Methods:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name:** | **Type** | **purpose** | **Access level** | **Parameters** |
| splashScreen | void | Run the splashscreen | public | none |
| menu | void | Present splashScreen | public | none |
| instructions | void | Display instructions | public | none |
| highscores | void | Calculate and display top 10 highscores | public | none |
| lvl1 | void | Processes and displays Level one | public | none |
| lvl2 | void | Processes and displays Level one | public | none |
| exit | void | Exits program and credits creators | public | none |
| fileToArray | Return array | Converts the word dictionary into an array | private | String fileName |
| wordGenerator | Return String | Choses a random word from an array of word | private | String[] dictionary |
| hint | void | Displays hint for lvl1 | public | none |
| winOrLose | void | Displays whether user wins or loses | public | String screen, boolean state |
| cheat | void | Displays the cheat function in lvl2 | public | none |

**Variables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **scope** | **Type** | **Purpose** |
| choiceO | global | String | Main method option, to determine overall program |
| highscores | local | String [] | Stores the highscores in the game |
| word | local | String | Stores unknown word |
| win | global | boolean | Determine a game is won |
| letter | local | char | The user input guessing the letter |
| cLetter | local | boolean | Check if letter entered is contained in the hidden word |
| man | local | int[] | Used to display the man in a specific order |
| timer | global | int | Used to time cheat function |
| letterPos | global | int[] | Used to track location of each alphabet letter |
| x | local | int | X Position of man in splash screen |
| guess | global | int | Tracks number of guesses user gets to use |
| deadLetters | local | String | Tracks letters that have already been used or other character that cannot be used |
| liveLetters | local | String | Tracks letters user can use |
| screen | global | String | Tracks what screen user is on |
| state | global | boolean | If user wins or loses a lvl |
| walkingMan | local | int [] | To track the different stages of the man walking in the splash screen |
| score | global | int | Calculates and tracks the score |
| wordBank | local | String[] | Stores all of the words from the wordBank file. |
| writer | local | PrintWriter | PrintWriter’s reference variable |
| reader | local | Buffered  Reader | BufferedReader’s local variable |
| numOfLines | local | int | Counts number of lines in file |
| lines | local | String | Read each line of input |
| wordSelected | local | String | Word outputted by wordGenorator method |

**Main method:**

public static void main (String args[]){

Hangman o = new Hangman();

o.splashscreen();

while(!o.choice.equals(“e”)||!o.choice.equals(“E”)){

o.menu();

if(o.choice.equals(“a”)||o.choice.equals(“A”)){

o.instructions();

}else if(o.choice.equals(“b”)||o.choice.equals(“B”)){

o.highscores();

}else if(o.choice.equals(“c”)||o.choice.equals(“C”)){

o.hint();

o.lvl1();

o.winOrLose();

}else if(o.choice.equals(“d”)||o.choice.equals(“D”)){

o.lvl2();

if(o.letter==‘$’){

o.cheat();

}

o.winOrLose();

}

}

o.exit();

}

**Error Traps:**

**Main Menu:**

Message: Invalid Entry! Please enter one of the options presented.

The only user input used in the menu is where to go afterwards, so if the user enters anything other than a, b, c, d, or e, the message above will be displayed.

**Level 1:**

Message: Invalid Entry! Enter one of the characters presented on the screen.

In Level one, the only expected user input are the letters displayed on the screen, so if a character is entered that is not on the screen, the program will return the message above.

**Winner Lvl1:**

Message: Invalid Entry! Please enter one of the options presented.

Like the Menu, when the program displays Winner Lvl1, it is only accepting 1 of 2 options, a) Next level, and b) Return to Menu. If the user fails to enter one of these two acceptable inputs, the error message above will be displayed.

**Lost:**

Message: Invalid Entry! Please enter one of the options presented.

When the user has lost, they are given the option to re do the level, or return to the main menu to choose a different option. If the user does not enter one of the two options available, the program will return the message above.

**Level 2:**

Message: Invalid Entry! Enter one of the characters presented on the screen.

Exactly like lvl1, if the user enters a character that is not on the screen, the program will return the error above.

**Winner Lvl2:**

Message: Invalid Entry! Please enter one of the options presented.

Like the Menu, when the program displays Winner Lvl2, it is only accepting 1 of 2 options, a) Play again, and b) Return to Menu. If the user fails to enter one of these two acceptable inputs, the error message above will be displayed.