Nathan’s Hangman Game

**Word choosing, page 1.**

In this report I will be summarising the HCI concepts I followed during the development, testing and designing my program with a user centred philosophy. First off I began by designing a small GUI window, in this window the user is asked to choose how they would like their words selected, in parallel or sequentially? As those are the only two options available and you cannot choose more than one I believed the right component for this would be a radio button with a toggle group.

Once the user selects their method of word choosing, they can click the submit button which will run the first option and once that is done it will run the second option and then it will display three labels below on the scene of this GUI, the first label displays the time it took to print the words in parallel, then the second label displays the time it took to print the words sequentially, and finally the third label will compare both the parallel time and sequential time and it will display which of the options was faster and by how long. Parallel uses threads, I use the read lines method in my file handling class to read words from each of the 4 word files. This class also contains the method I use to serialize the array list of string lists into binary.

After the user learns this information, they can proceed by clicking the next button which takes them to the difficulty page, after a while I thought, what if the user doesn’t care which is faster? If that’s the case then they can just choose a method of word selection and skip the “submit” and click next (this will not display the parallel and sequential time comparison meaning all three labels will not be visible). Whilst testing this GUI, although it was functioning as expected I felt like I could add a little more to it, perhaps some audio and then I had a new idea, I went to a free speech to text website and decided to save some audio for the game, when you first open this page you are greeted with a voice which reads out the first label for you.

I thought this would enhance the user experience as audio provides a richer and more robust environment than a mere graphic feedback.

Finally, as I would like this scene to be user friendly I added a background image, changed the button colours and I changed the colours of the text.

**Difficulty selection, page 2.**

Now that the selection is out of the way, the program will give us an array list of random words from four different question files in my difficulty controller, the word lengths range from 5 letters to 8 letters, some users will want an easy game and others will want a challenging game, I made four levels of difficulty, five letters, six letters, seven letters and eight letters, I named the difficulties “amateur”, “semi – pro”, “professional”, “Legendary” each having their own label, and below these labels I added four more labels to display the number of words in each so that the user has an understanding of what each difficulty means before they choose one.

For each of the four difficulties I created a toggle group and made each choice a radio button so that the user can only select one option, after the user selects the option the option they selected will run a method which will add all words that contain that number of letters into a new array list of strings (for example if the user chooses professional then all 7 letter words will go into this new array list of type string). Once this is all done the user will click submit. Within the FXML file for this particular controller I added some colours for the buttons as well as some for the text, I decided to change the background and add an image instead as I feel this would be more user friendly and again I added some audio speech as this could provide more information. I added two buttons, one for the user to get to the next page which is the game itself and another for the user to quit the game, navigation is important in HCI, everything you should be easy for a user to identify therefore it’s a good idea to add buttons such as “quit” where the user can easily locate it. Now that I’ve kept the user interested up until now let’s get onto the game itself.

**The game, page 3.**

*Navigation bar*

Within the game page there are many GUI components, starting with navigation bar, I added a navigation bar because one of my goals for this project is to make the game easy to use, I want it to be simple for a new user to get into, everyone uses navigation bars because they are easy to use when locating a feature for example if you want to save or you need help etc. I added useful options in the file option such as “new” for new game, “load” to load an old game, “Save” to save the game in the current state and “close/quit” to exit the game. I added an “Edit” option which contains the change difficulty option and next to edit there is an option simply called “help” which opens up a small alert box explaining rules and regulations of the game as well as other useful information.

*Audio*

The speech I added here explains how to start the game once the page opens so that the user will know directly where to click, they are told to click the “start game” button which as described, starts the game. Like I said before, audio helps enhance the user experience as well as keeping them focused so I decided to add some intensive music to the game which will play once the user clicks “start game”, if the user doesn’t want the music to be played they can simply click one of the pause or stop buttons located on the top left of the game screen. There are also other speech sounds and effects that will be played during the game, when you guess a letter right the game will play a ‘ding’ correct sound and when you guess it wrong it plays a ‘buzz’ wrong answer sound.

I want the user to interact with the game well, this is the most important part of the game so a speech once in a while is great, if you guess a letter you’ve guessed previously the voice will say “ you have already used this letter” and a label in red will appear in the bottom of the game screen below hangman saying “this letter was already entered”.

Once the game is complete the voice will say either “Game over you lose” or “ you win” depending on the outcome. When a user is playing a game the last thing they want is a lot of graphical text appearing on their screen so sometimes it’s better for them to just hear it being said to them instead.

*GUI components*

How does this game work? It’s a hangman game, meaning we will be guessing a word user letters… obviously, on the left of the game screen there are 26 buttons for each letter, these buttons will be used for guessing the words, each button runs the “forButton” method which takes two chars, the first char will be the capitalised version of that button and the second will be the lower case version, this method checks if either of the inputs are in the word, if so the game plays the ‘ding sound’, adds the character to the “alreadyUsed” hashset, updates the label that displays the word with letters hidden to show where that letter is in the word for example if the user clicks the B button and b is in the word that label will look like this “B- - b - -” and guessing a letter right adds 100 to your score(I will explain score shortly) else if it does not contain the letter the game will play the wrong answer sound, the hangman image will be updated, the “alreadyUsed” Hashset in the game controller will be updated.

I have a start button as explained earlier, this button simply allows the user to start the game. After a while of playing the game I thought, what if the user wants to give up? Maybe I should add a feature for that, the “give up” button appears on the bottom right of the image, this will reset your score and display the word you gave up on in the middle of the screen.

In summary of the components here, the user has many options whilst playing the game, they can give up, they can start a new game, they can guess letters, turn off the music, play the music and they will also have a play again button located just above the give up button. For a user to enjoy an application they need many features and options, if the options are highly limited it will not be a very good user experience as they could get stuck on something basic such as not wanting to play anymore and they won’t have a simple option for that.

What makes a good game? It has to be fun and most fun games are competitive, I decided to make the game even more competitive by adding high score and your score features, although this isn’t a traditional hangman approach I thought it would make the fun for the user, the user can now compete against themselves as well as others by attempting to beat the previous high score. How are scores registered? Within the forButton method in the game controller, I added a small feature for an integer named score, if the user gets a letter right the score increases by 100, if they get it wrong it decreases by 10 and the voice in the game says in the difficulty page that “the harder the difficulty the more points” which is true as for 5 letters the maximum you can score will be 500 where as for 8 letters the maximum will be 800. Once a game is finished it will check if the score set by the user is greater than the high score(which is originally set to 0), if so the high score updates and when you start a new game the score goes back down to 0 but the high score is now visible. I believe this feature will keep the user interested because they can involve others as well as compete.

The score and high score are shown on the left of the game screen next to the hangman himself, on the right of the screen we see the amount of lives left within that game, usually in a hangman game the number of lives isn’t shown, all we have to indicate how many guesses we have left is the hangman himself. I added the amount of lives so the user will know for sure how many guesses they have left before choosing the next letter. The hangman is located within an image viewer which uses the amount of lives to determine which image to show on the image view, if you have 6 lives left it will show image 6, image 0 is when the game is over (when you have no lives left) and image 9 means you have all 9 of your lives remaining. Under the amount of lives left there is another label which indicates how many words are in this selected difficulty, for example “word:5/250” means you are on the 5th word out of 250. Below the game you have your label which shows your guesses and where they are located within the word and below that I have the hash set “Already Used” so the user knows which words they have used.