CSE212: SOFTWARE DEVELOPMENT METHODOLOGIES

YEDITEPE UNIVERSITY

FALL 2021

TERM PROJECT – DUE DATE JANUARY 3RD, 2022

As the term project, you are required to develop a *Hangman* game with multiple lengths of strings. The game should be simple and easy to use with the help of a mouse or a keyboard. Your application should be able to keep scores of users and store them on a file.

You can use the following statements and figures as guidelines:

- For usability purposes, you are required to implement a graphical user interface (GUI) for your application.
- Your application should have a menu bar which would contain *File* and *Help* (see Fig. 1).

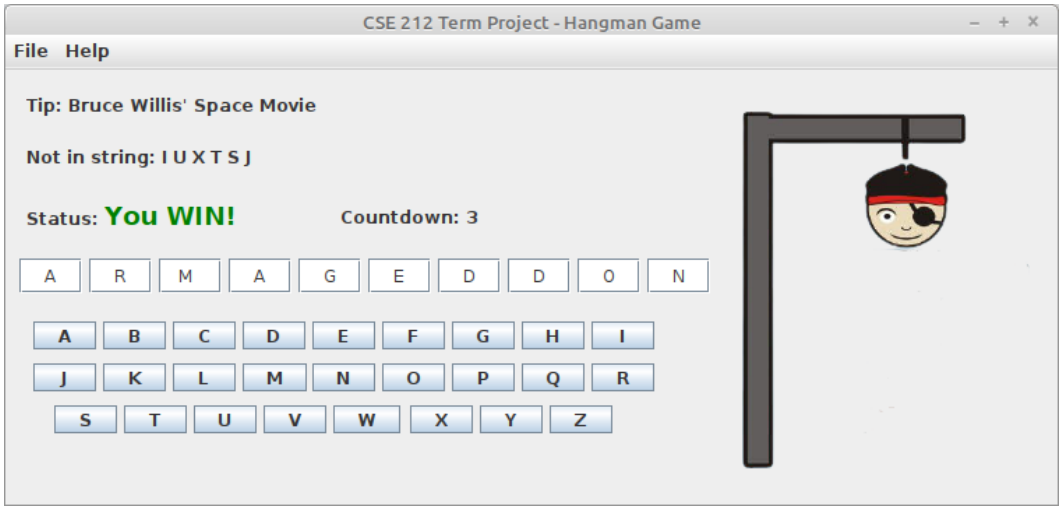

Fig. 1: Example Hangman game win screenshot

- At the end of each game, it should ask for the user's name and record his/her details (*name, time, date, hangman string, string size and time elapsed*) on a file.
- At any time, users should be able to quit the game by clicking on the appropriate key combination (*Ctrl+Q*).

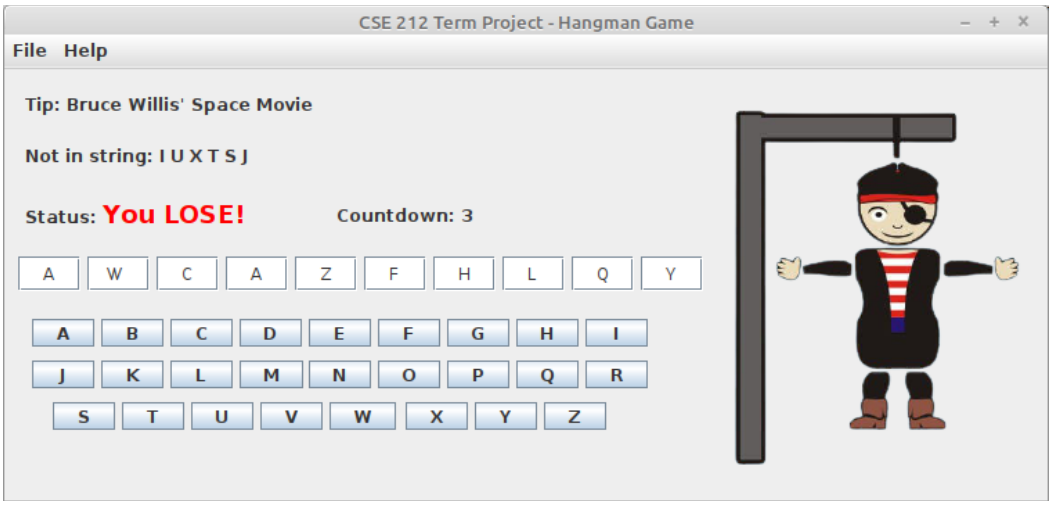

Fig. 2: Example Hangman game lost screenshot

- The game GUI should have a visual keyboard which can be interacted via mouse of arrow buttons.

- The game GUI should include a tip about the string that is expected from the player to guess its letters (Fig. 2) – "*Tip: Bruce Willis' Space Movie*".

- The game GUI should have a list of 6 letters - vowels and non-vowels – that are not in the given string, word or sentence. As shown in Fig. 2: "*Not in string: I U X T S J*".

- The GUI should be also indicated the current status of the game – as in "In progress" in blue, "You WON!" in green or "You LOSE!" in red.

- Depending to the string or sentence length there should be a countdown timer next to the status label. The timeout period should be set based on the difficulty of the string – i.e. easier the keyword shorter the time out period.

- All the questions of the game should be stored comma-separated values (CSV) in a text file. The values separated with a comma should be *the tip*, *not-in-string characters*, *the hidden string* and *countdown timeout value*. This file should be manually created by the programmer. The application should be reading the file and randomly picking a hidden string to ask to the player.

- The *File* menu should have four submenu items – namely *New Game*, *Reset Game*, *Score Table* and *Quit*.

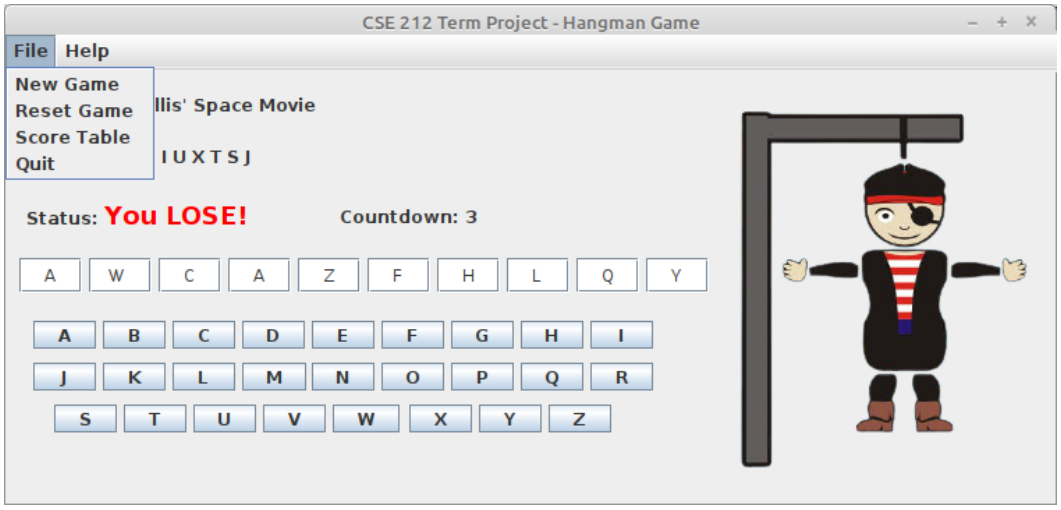

Fig. 3: Example Hangman game File Menu

- Players should be able to start a new game by clicking on the (*File->New Game*) menu item (Fig. 3).
- Players should be able to restart their current attempt without changing the hidden keyword/string by clicking relevant menu (*File->Reset Game*) item.
- Players should also be able to see the scores list, which contains the list of ten highest scoring players on a separate window by clicking the appropriate menu item (*File-> Score Table*).

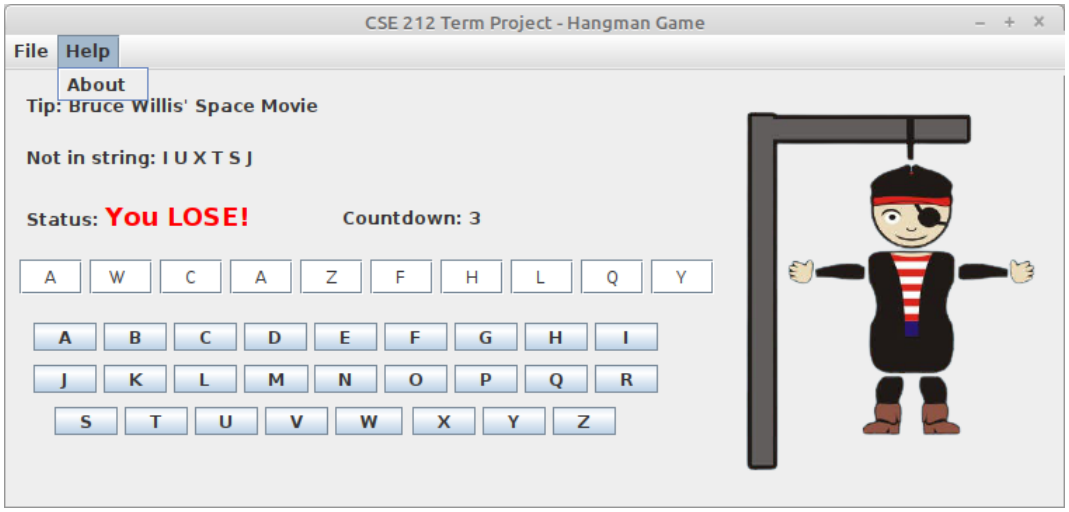

Fig. 4: Example Hangman game Help Menu

- The About (*Help->About*) option should pop-up a small window that contains information (*Name, Surname, School Number and Email*) about the application developer (Fig. 4).

- Every time a player fails to identify a character in the hidden string, one of the six body parts – *head*, *torso*, *left arm*, *right arm*, *left arm* and *right arm* – should be added to the picture. After the sixth failed attempt game should and the player should be notified that it has lost the game.

- Last but not least, if a player would like to exit the game should click on the relevant menu item (*File->Quit*).

- **[BONUS]** The application should play relevant sounds after each character selection of the player – achievement indicating sound if the chosen character is in the hidden string or failure sound if not.

- **[BONUS]** The number of text boxes in the GUI should be dynamic based on the size of the hidden string – each box one letter.

Submit your assignments in a zip file, which has your name_surname_studentNumber as name, using COADSYS (**https://yulearn.yeditepe.edu.tr/**) by the end of Monday, January 2nd, 2022. All submitted source files will be checked for plagiarism among classmates and with any existing open source code available on the Internet. Furthermore, all students will be required to demonstrate their work for 15 minutes. DO NOT submit somebody else's work.