COSC 345

“Small is Beautiful”

Assignment 4: Final Release Maintenance Documentation

Group: CodeBound

Project Title: Devolution

Marcus Anderson
(ID: 7049664)

Surayo Akramkhanova
(ID: 7497059)

Grace Auckram
(ID: 7659739)

Josef Bode
(ID: 5636437)

How to build and run the project:

1. Go to the GitHub and download the file tagged with “Final”.
2. Open the Devolution command line project file.
3. Create a new scheme in Xcode for the assignment.
4. Build and run the program in Xcode.

Storyline and text files:

The story is complete and the text files are reviewed for grammar and story flow errors. However, there is no limit for the story expansion.

The placeholder variables throughout the text indicate where to change the text, which is handled by the software code:

- [NAME] - Name of the main character
- [Xe] - He and she
- [Xer] - Her and him
- [Xis] - his and her
- [Xers] - hers and his
- [Xself] - himself and herself

The links to the choices and following paths are held by the placeholder variables too.

The program code:

The code is 666 LOCs. The code includes title screen and integrated text manipulation. The source files and associated modules written in C contain the comments that describe what functions within the code do. Valgrind was used to fix identified memory allocation issues throughout the testing.

Playing the game:

Follow the instructions displayed on the screen:

- Once the title screen gets displayed press “Enter” to start a new game.
- Enter the name of the character at the prompt saying, “please enter your name”.
- Enter either *f* or *m* for the selected gender after the next prompt.
- The game requires the confirmation step; if the parameters were entered wrongly the character creation process can be reset.
- Hitting the space bar or return button allow to progress through the story.
- After reading the displayed scene and making the choice press the corresponding number of the next scene (from 1 to 3 possible paths available).
- Escape button allows exiting the game.
- The game is over when the main character dies, and the game is finished when the storyline is complete.

A description of each source file and their associated functions is provided below:

main.c

The application file for where the program runs from. It handles displaying the title screen from a text file at a particular speed. The introduction with game instructions is also read from the file. All the char operations displaying prompts and accepting inputs by the player as well as the keyboard operations are handled in the main source file using getch method.

Text handling algorithms:

mylib.c

mylib.c holds two memory allocating methods emalloc and erealloc. These two methods are used for calling malloc and realloc as well as checking that the memory was allocated correctly.

fileManager.c

fileManager.c has six main functions, it can get the location of current working directory, open and close the selected file, all of which have error handling implemented if the file or directory cannot be found. It also has functions to save the current path and open the saved path, which are not implemented and are for the future development of the software. Concat is a helper method to path saving methods that concatenates two strings.

textManipulation.c

The setFile method loads all of text from the file into memory and returns a pointer to the loaded text. textManipulation.c also contains functions to get the current file, set the choices and text blocks as well as to add all pointers to the array and a function to set character name and gender and implement them instead of placeholder variables. The setChoices method allocates memory for the choices and text of the choices. The setStoryText method sets the text blocks and deals with placeholder variables to continue with choices, finish the game or set the game to be over.

Milestones achieved:

- Written and edited complete science fiction adventure story
- The story has been integrated into a working text-based game
- Multiple choice version of the game implemented
- The character creation function implemented
- Escape function implemented to stop the game
- Simple graphics used in contrast to proposed version of GUI as the team was running out of lines of code
- Using the feedback of the testers the version of a game that allows gradual read through was chosen over displaying the whole amount of a text for a particular path

Ideas for the future development of the project:

- Implement the proposed GUI version which has been partially developed
- Implement the save path and continue the game functions that were not integrated due to time constraints of the team
- Have a map of the paths available to players
- Future graphics and music add-ons can be developed for the game
- Write more paths to the story, expand the story characters and the events happening throughout
- Debug if any bugs are identified