Programming Project III

PROG1700 Logic & Programming

Evaluation : 30% of final grade

Due Date : Dec 13 @ 8:30am

All students will submit their finished PY file to Brightspace (Projects / Programming Project III) on or before the due date / time. The project will then be marked "face to face" with each student during a pre-determined code review during the week. All students must be present for code review. Your project will NOT be marked if you are not present for code review, regardless that you submitted it to Brightspace. No code review = 0 on your project.

Assignment Description

Programming project III will be to design and develop a story application. In other words, you will develop a simple interactive story in which the user can make decisions that will affect the overall plot and outcome.

Requirements include:

1. When the game first starts a main menu and input cursor is presented to the user. The choices are:
    - New game – will start a new game
    - Load game – will load a previously saved game
    - Quit Game – the game (app) will exit
2. When a new game is started the game will show the first "decision point". These decision points consist of some story plot text and two story options the user can select to move the story along.
3. Besides this, every decision point has a third option to save the game in an external text file called "saved.txt". A message should be displayed to the user that the game has been saved and then the user can continue playing the game. For simplicity, only one game can be saved at any given time. It is this saved game that will be loaded if the user selects the "Load game" option from the main menu.
4. The story content must be read from the *provided* CSV file called "story.csv". Each line of this file represents a decision point and consists of the plot text, two story options, and the two destination line numbers: `[Plot Text],[Story Option 1], [Story Option 2],[Destination line number for Story Option 1],[Destination line number for Story Option 2]`
5. The game is over when your game is displaying a decision point that has no choices (indicated in the CSV as empty strings for both). The game will then pause for two seconds and then display the main menu again in which the user can play again, load a game (if one was saved), or quit the game.
6. By studying the story.csv file you will find that this story has four decision points and three possible endings. It is important to note that this might not always be the case. You could completely change the story by adjusting the story.csv file and running the game again. The idea is that the app is like a game engine for any story you want, with any number of decision points or possible endings. In fact, one important test case is I will be swapping out the default story.csv file with a different one of my own design to see if the changes are reflected in your game.

```
****** Text Adventure Game v1.0 ******
*                                    *
*           1 - New Game             *
*           2 - Load Game            *
*           3 - Quit                 *
*                                    *
**************************************
> 1

Should I add a database to my app?
What do you want to do?
1 - Use MS Access
2 - No way - databases suck
3 - Save Game
> 1

You are buried in relational diagrams
What do you want to do?
1 - Get through them and get to the fun stuff
2 - Let's add in a few more relationships
3 - Save Game
> 3

>>> Game Saved

You are buried in relational diagrams
What do you want to do?
1 - Get through them and get to the fun stuff
2 - Let's add in a few more relationships
3 - Save Game
> 2

I miss programming Python - Game Over!

****** Text Adventure Game v1.0 ******
*                                    *
*           1 - New Game             *
*           2 - Load Game            *
*           3 - Quit                 *
*                                    *
**************************************
> 2

You are buried in relational diagrams
What do you want to do?
1 - Get through them and get to the fun stuff
2 - Let's add in a few more relationships
3 - Save Game
> 1

Onto the coding!
What do you want to do?
1 - Which language?
2 - Let's use several!
3 - Save Game
> 2

Brain burnt out - Game Over!

****** Text Adventure Game v1.0 ******
*                                    *
*           1 - New Game             *
*           2 - Load Game            *
*           3 - Quit                 *
*                                    *
**************************************
> 3
```

7. There are two points in the game which require user input. Both must be fully error checked. All error checks must display a descriptive error message to the user and *always* provide the user another chance to enter the correct input.
8. It is possible that the user will try to load a game before previously saving one. Your game cannot assume that a saved.txt exists and therefore must error check for it. If no saved.txt exists when the user tries to load a game, then a new game is started automatically instead and the user is informed.
9. Your application cannot include ANY global variables and must implement at least four user functions (although you may find you want to include more)
10. Be sure to include informative comments in your code
11. You MUST be able to loosely explain your code / logic (blindly copying and pasting code is not programming). Failure to do so will result in a zero on the project. Essentially, if you wrote it you will understand it.

Requirements (Marks breakdown)

Choose Your Own Adventure Game	
Project Requirements Test Cases *Collection of test cases that test all project requirements listed above*	24
Code Requirements *Code design and functionality, minimum user functions, no global variables, commenting, etc.*	6
TOTAL MARK	30

Other Notes

- Divide and conquer is key to tackling this project! Start with your error checking for all input, move on to loading the story data from the CSV, etc. The saving and loading feature can easily be implemented as a final step.
- Remember to test this application thoroughly to meet all project requirements and to avoid any runtime errors.
- This project will be marked during code review at a scheduled day/time on or after the due date. Be prepared to demo your application and explain your code / logic. Since this is the final project and is due the last week of the semester - no late submissions will be accepted.