Austin Keelin

Dr. Ku

COMP 2710

2 April 2017

1. **Analysis**

Player Options: Every turn the player has 5 options to choose from. The player can choose to Move, Read technical papers, Search for loose change, View character, Quit the game.

Move: The player moves one step in the grid, but risks encountering an enemy or having to solve a puzzle. Moving takes time.

Read technical papers: The player loses 3 seconds off their time, but will increase the player’s intelligence by a random amount up to 10

Search for loose change: The player loses 5 seconds off their time, but will increase the player’s money by a random number up to 10

View character: Returns a display which shows the character’s attributes (intelligence, money, and time) and shows how far the player is from the goal

Quit the game: Shows the “You Lose” screen and exits the program

Encounters: Every time the character steps, there is a random chance of various events happening.

Nothing: 25% chance nothing happens and the character moves forward

Puzzle: 25% chance you encounter a puzzle

Engineering Fee: 10% chance Auburn implements another new Engineering fee, lose random amount of money between 3 and 5, do not lose time

Midterm: 10% chance you must take a midterm you didn’t know about for a class you don’t really go to, lose random amount of intelligence between 3 and 5, lose 3 seconds off time

Book return: 10% chance you don’t need one of the expensive textbooks you purchased for this semester, and can return it to the book store, gain random amount of money between 3 and 7, lose 3 seconds off time

Study guide: 10% chance your professor posts a study guide on canvas for that upcoming exam, gain random amount of intelligence between 3 and 7, do not lose time

Vocareum: 10% chance your newest COMP class requires a paid subscription for a program that checks the correctness of your coding assignments when that free program you used last semester worked just as well, lose random amount of money between 3 and 7, lose random amount of intelligence between 3 and 7, lose 3 seconds

Puzzles: Requires interaction from the user. Will ask some kind of riddle or simple logic puzzle. Correct answers are rewarded (gain money or intelligence or time) while incorrect answers will be punished (lose money or intelligence or time).

1. **Design**

HighScores

1. Loads, sorts, and collects the high scores
2. Variables: Public: final\_score, name

Private: const MAX\_SIZE, intArray[MAX\_SIZE], stringArray[MAX\_SIZE], hsArray[MAX\_SIZE]

Functions: Score- Calculates the final score for one playthrough

readHighScores- Place data from file into local arrays; also tells size of file

newHighScores- Saves high score and name after completion/loss during one playthrough

printHighScores- Prints the high scores. Used in the main menu

ifexists- Checks that a file exists

askName- Asks for user’s name at beginning of game

getName- Returns the user’s name

1. Encounter, System, Menu

Puzzle

1. Will run puzzles, operates like encounters
2. Variables: Public: pcount, money, intelligence, timer, position

Functions: Puzzles- One function with all puzzle questions, will not repeat questions in one playthrough. Will go through questions in order

1. System

Encounter

1. Will run encounters
2. Variables: Public: money, intelligence, timer

Functions: End- Ends the program when the user loses; handles high score being added

EngineeringFee, MidTerm, BookReturn, StudyGuide, Vocareum [All encounters defined in Analysis]

1. HighScores, System

System

1. Instantiates the other objects and runs encounters
2. Variables: Public: money, intelligence, timer, final\_score, const GOAL, position, play\_option

Functions: Play- instantiates objects, begins the game

End- Exits the program

Move- Uses RNG to determine which option will run (nothing, encounter, puzzle). Encounter and puzzle will run through their respective classes

ReadTechPapers- The player loses 3 seconds off their time, but will increase the player’s intelligence by a random amount up to 10

Search: The player loses 3 seconds off their time, but will increase the player’s money by a random number up to 10

ViewCharacter: Returns a display which shows the character’s attributes (intelligence, money, and time) and shows how far the player is from the goal

1. HighScores, Encounter, Puzzle, Menu

Menu

1. Runs the main menu, handles user errors. Does more of initializing the user interface then actually running the program
2. Variables: Public: option, money, intelligence, timer, position, name

Functions: MainMenu- Start a new game through System, view top 10 high scores through HighScores, quits through system.

1. HighScores, System
2. **Testing**

**[Notes- All tests are ran under the single function test\_All. Test descriptions below are organized as if the test cases are separate functions, but they are not]**

test\_HighScores\_Score:

1.1: Calculate the player’s score with max values for money, intelligence, and timer

1.2: Calculate the player’s score with median values for money, intelligence, and timer

1.3: Calculate the player’s score with min values for money, intelligence, and timer

test\_HighScores\_newHighScore:

2.1: Adds the player’s score and name to the high score list, if applicable

test\_Encounter\_EngineeringFee:

3.1 Runs the EngineeringFee encounter when the player has max money

3.2 Runs the EngineeringFee encounter when the player has a median value for money

test\_Encounter\_MidTerm:

4.1: Runs the MidTerm encounter when the player has max intelligence and max time

4.2: Runs the MidTerm encounter when the player has median values for intelligence and for time

test\_Encounter\_BookReturn:

5.1: Runs the BookReturn encounter when the player has max money and max time

5.2: Runs the BookReturn encounter when the player has median values for money and for time

5.3: Runs the BookReturn encounter when the player has min values for money and time

test\_Encounter\_StudyGuide:

6.1: Runs the StudyGuide encounter when the player has max intelligence

6.2: Runs the StudyGuide encounter when the player has a median value for intelligence

6.3: Runs the StudyGuide encounter when the player has a min value for intelligence

test\_Encounter\_Vocareum:

7.1: Runs the Vocareum encounter when the player has max money, max intelligence, and max time

7.2: Runs the Vocareum encounter when the player has median values for money, intelligence, and time

7.3: Runs the Vocareum encounter when the player has min values for money, intelligence, and time

test\_System\_ReadTechPapers:

8.1: Runs ReadTechPapers when the player has max intelligence and max time

8.2: Runs ReadTechPapers when the player has median values for intelligence and time

8.3: Runs ReadTechPapers when the player has min values for intelligence and time

test\_System\_Search:

9.1: Runs Search when the player has max money and max time

9.2: Runs Search when the player has median values for money and time

9.3: Runs Search when the player has min values for money and time