COMP 2710: Project 1

Dragons – A Simple Text-Based Game

Points: (100 points) Deadline: 11:59 PM Oct 26th 2018.

Turned in via Canvas

No collaboration between students. Students should NOT share any project code with each other. Collaborations in any form will be treated as a serious violation of the University's academic integrity code.

Goals of Project 1:

- To perform object-oriented analysis, design, and testing
- To develop some proficiency with basic C++ syntax and semantics.
- To learn data flow diagrams.
- To learn use cases and use case diagram.
- To gain experience with unit and system testing.
- To perform separate compilation.
- To use singly linked lists
- To develop a reasonably user-friendly application
- To write a fun application!

Rating

- Design difficulty: 4/5
- Implementation difficulty: 2/5
- Time required: 4/5
- Fun: 3/5

1. Background

A customer wants a cool new video game, similar in style to popular video games like *World of Warcraft* or *Bioshock*. However, the customer recognizes the lack of significant funds and only has a resource poor computer to play this game on. Hence, the game will be a simple text-based adventure concerning a graduate student trying to navigate his way down Shelby Center. In this project, you will help the customer to design and implement a simple text-based game.

2. Requirement Details

2.1. Player

The "player" is represented by *at least* three attributes: *intelligence*, *time*, and *money*. If the player runs out of intelligence, time or money, the player dies. The goal of the player is to survive to the end of the "hall" with the highest combined total of the attributes as possible. "Score" is determined as the three attributes multiplied together.

2.2. The Hall

The player starts the game at the beginning of a hall, which is linear.

The "Hall" is a path that is at least twenty (20) steps long. After a move, the user should be told how far away from the goal they are (in steps). If the player survives to the goal square without any of the attributes falling to 0, they win. Their score should be displayed with a simple ASCII victory message. If the player dies, a "You Lose" message should appear indicating the cause of death (for example, if money falls to zero, you can say that the player starved to death because of poverty). All the attributes should start in some random range (e.g. 8-25).

2.3. Turns

Every turn, the player has (at least) 5 options to choose from:

- **Move**: The player moves one step in the grid, but risks an Encounter or a Puzzle. Moving also takes time.
- **Read technical papers** (boost intelligence): The player loses a fixed amount of time, but increases intelligence by a random amount
- **Search for loose change** (boost money): The player loses a fixed amount of time, but increases money by a random amount.
- **View character**: A simple display should show the character attributes and current position in the hall (ASCII is fine)
- **Quit the game** (shows the "You Lose" screen – optional mockery- and exits the program)

2.4. Encounters

Encounters: Every time the character steps, there is a random change of various events happening. You are free to change the probabilities as you see fit for "game balance," but here are some suggestions:

- 25% chance: nothing happens, you just move forward.
- 30% chance: You encounter a Puzzle (see Puzzle below)
- 10% chance: Encounter a professor. This loses a random extra amount of time, but may slightly increase intelligence.
- 10% chance: Encounter another graduate student. This loses a random amount of time.
- 15% chance: Attacked by grunt work! Lose both time and intelligence.
- 10% chance: Grade papers. Lose time, but gain money.
- 0% chance: Get a huge raise, gain lots of money! (This never happens).

2.5. Puzzles

Puzzles: Puzzles are different from normal encounters since they require interaction from the user. These don't necessarily have to be brilliant, but riddles or even edutainment light puzzles are fine. Examples:

- "What is 2 + 2:" For a correct response, Money + 1. For an incorrect response, Money -20 (you idiot).
- "What can you put in a barrel to make it lighter?" For a correct response, int+2. For an incorrect response, int-2.

2.6. Other options

You are free to add more details and rules to your game, but you must have at least the above specifications. Feel free to be creative – there are many opportunities to do so.

3. Late Submission Penalty

- A ten-point penalty per day for late submission. For example, an assignment submitted after the deadline but up to 1 day (24 hours) late can achieve a maximum of 90% of points allocated for the assignment.
- Assignment submitted more than 3 day (72 hours) after the deadline will not be graded.

4. Rebuttal period

- You will be given a period of 4 days to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

5. Hints

- Start early, you have a good deal of time but you may need it to start implementing your design. Although the following timeline is not mandated, it is

a suggestion of milestone:

- 1/4 time: Design data structures and function prototypes.
- 2/4 time: Design algorithms.
- 3/4 time: Prepare system-level test cases.
- 4/4 time: Write test drivers.