Project Proposal Draft
for UTM CSCI 352
Battle Of The Professors

Mason Bearden Jordan Taylor

Abstract

We are creating a text adventure game which uses puzzles/trivia/logic problems to test the player. The game is based off UTM CSCI classes and proffesors. The game will include stat management, challenges to solve, and navigation through a map. The player is playing as a student who must traverse a map, solve puzzles, meet other students, and eventually fight a professor. The target audience is students and teachers specific to UTM, or gamers in general who find value in the game.

1. Introduction

The user plays as a character defined as a ”student” who needs to pass through their CSCI class. They will traverse through a map acting as maze with which they must solve events to raise/lower their grade. The students health will act as their grade and as they complete challenges, it will decrease or increase. The student must beat the boss of the map ”Professor” in order to pass the class. All events including the boss fight will unfold as puzzles/trivia/logic games. For example, while traversing the maze, a student may come across a challenge that asks for a series of questions to be solved. Somewhere in the maze will stand a fellow ”Student” who will help the player by giving advice/crucial information on the class. Seperate stats: Intellect and Sanity will determine the values lost/gained when completing challenges. Sanity will be a students defense to health loss, while Intellect will influence how much damage is done to a boss. The expected users that would be interested in this game are the students(Past and Present of CSCI UTM), the professors(UTM), and those who enjoy RPGs (role-playing game). Although the game is specific to a UTM all gamers are welcome.

1.1. Background

We are interested in learning the concepts of a puzzle adventure game.. It will provide a variety of skill sets that we will be able to use in future programs. Having the experience of creating an RPG could potentially open up job opportunities.

1.2. Impacts

This game will hopefully provide enjoyment to those who are stressed or ill, making life a little more exciting. The game should also be a light educational game, which tests a persons knowledge on computer science based information.

1.3. Challenges

I believe that the toughest portion of the game will be all the small things working together such as: traversing through the map, stat loss / gain and implementation of challenges. We would like to implement diversity in said challenges, such as questions, trivia, and management of stats.

2. Scope

Scope: The bare minimum for the project that we want to accomplish is having at least one map with puzzles/trivia/logic problems working. A functional boss and student character and stat implementation. Stretch goals would include: 1. adding all maps/professor 2. Adding additional npc involvement such as an npc giving the player hints at challenges, or even the boss fight. Of course they will not give the full answer as that would be cheating.

2.1. Requirements

The functional and non-functional requirements were difficult for us to determine. We determined requirements by thinking what a player should be able to do in game such as movement, and interaction of events. For the non-functional, we went for things the player should have access to.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Move through map	Player	low	1
2	Solve Challenge	Player	Med	2
3	Challenge Boss	Player	High	3

TABLE 1. SAMPLE USE CASE TABLE

2.1.1. Functional.

- Player should be able to save game state and load game state
- Player should be able to traverse map
- Player should be able initiate challenges and complete them
- Player should be able to create new game and restart progress

2.1.2. Non-Functional.

- Stats - Player should be able to track stats/stat changes
- Save - Player should be able to save progress in game, such as placement and stat changes.

2.2. Use Cases

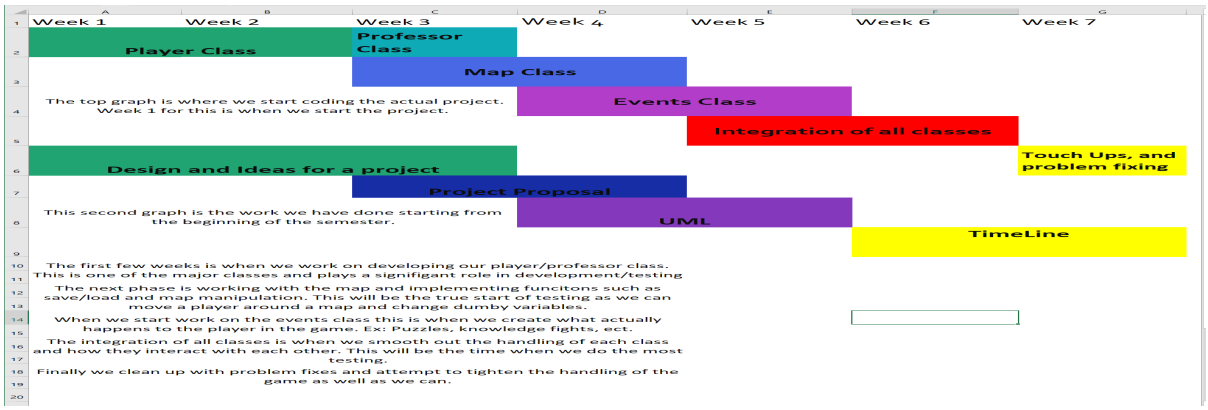
Use Case Number: 1
Use Case Name: Move through map
Description: The player will move through the map to explore/progress
1) Player is presented with map and options for movement.
2) Player picks direction and proceeds.
3) Player position and map are updated.
Termination Outcome: The player is now in a new location.
Alternative: Direction does not exist
1) Player tries to move in a wrong/unaccesible direction
Termination Outcome: Player is given warning that movement can not happen, and map remains unchanged.

Use Case Number: 2
Use Case Name: Solve challenge
Description: The player is presented with a challenge that will determine health loss/gain, this is activated by moving to specific locations on the map. Events may be made random if we can functionally acomplish this.
1) Player is presented with a challenge in the form of riddle/puzzle/logic all are possible challenges.
2) Player chooses the solution to challenge, either through choice of option or typing in a specific answer.
3) Player health is increased or decreased based on decision.
Termination Outcome: Player is now able to move on. Player is not able to escape challenge, but possible rechallenge may be implemented such as a limited number of times a rechallenge can be done.

Use Case Number: 3
Use Case Name: Challenge Boss
Description: The player is presented with a boss that will determine if they complete the map and move to the next class
1) Player is presented with a challenge in the form of a boss
2) Player chooses battles boss in challenge
3) Player is able to move to the next class
Termination Outcome: Player is now able to move on.

2.3. Interface Mockups

3. Project Timeline

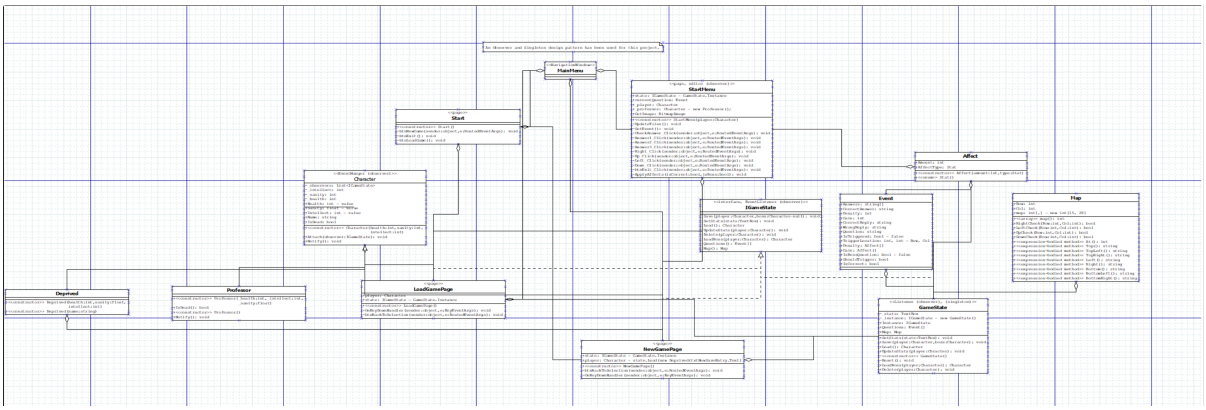


With the timeline we went with a block schedule with the top showing previous work being done such as design, latex building, creating timeline, creaing UML, etc. The bottom is the work we will do on the project itself as we continue through the weeks. This will include things like making the classes such as the character class, the map class, etc. We have included desriptions of what steps will entail on the timeline itself, which is located in the bottom left area.

4. Project Structure

The project structure so far has gone with several design changes as we have moved forward. Graphically speaking we have changed the visual representation of the map and player character, although not neccesarilly permenant as we may change them in the future. We have at this time opted for a branching maze design with random or psuedo random events as the plaer progresses. Events are still being worked on and may be changed to fit our current skills as programmers and with what time we have remaining. Update: Events are not random and are instead in specific locations.

4.1. UML Outline



Our UML has two major classes, the character class and the map class. The character class creates the playable character as well as the professor and holds stats for each. The map class creates a map and defines functions for moving through said map. We have an gamestate interface and a gamestate class which tracks the variables needed to save the game. This includes player position and stats and also handles the load function which allows the player to continue a game they previously saved. The MainMenu is the non class code which handles button presses, global variables and textbox updates.

4.2. Design Patterns Used

1. Observer Adds children of GameState into an observer where they can be notified during events. Through this, they are saved and the stats are updated using an interface, IGameState. 2. Singleton: Uses the IGameState and GameState classes to create an Instance where all classes can speak to. This is used in the main driver of the project named state. This field calls methods from IGameState to add functionality to Character children.

5. Results

1st Results Summary: We have managed to get a functional map with with full functionality of use case 1. We have partial functionality of use case 2 but is soon to be completed. Use case 3 is still in development. 2nd Result Summary: We have managed to get a functional map and updated some of the core feautres and designs of said map. We have also implemented full funciionality of use case 2. Use case 3 is still in development. 3rd Resulty Summary: All use cases are implemented and now we are working on clean up of systems and adding features.

5.1. Future Work

1st Future Summary: We are on schedule as it stands and hope to finish the event implementation and then work on the boss implementation. So far goals are being met, although I do not know if we will be able to achieve multiple maps. 2nd Future Summary: We completed the game however were not able to implement more maps due to time constraints. We had also hoped to update some of the graphical designs but was not able to.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.