Dungeon Explorer

Product Backlog

Team 13

Xinpi Du

Art Malinin

Wei Haow Tan

Jun Xiang Tee

Matt Walters

Lirong Yuan

1. Problem Statement

Our teams project is to create a multi-level simple puzzle game that increases in difficulty with each level. While keeping score that can be posted to a scoreboard, while keeping a personal best.

2. Background Information

Our game needs to be challenging but not too hard from the start, and increase in difficulty as the users moves on through levels. This will allow new users to enjoy the game but still provide a challenge to more experienced users. It will also need to have replayability. We will accomplish this by the score system, to give users a benchmark to aim for and constantly improve on. The scores will be stored in a database so users can compare scores with their friends. There will also be different modes that will have varying difficulty as in amount of tries the users gets, as well as the speed of obstacles.

3. Environment and System Models

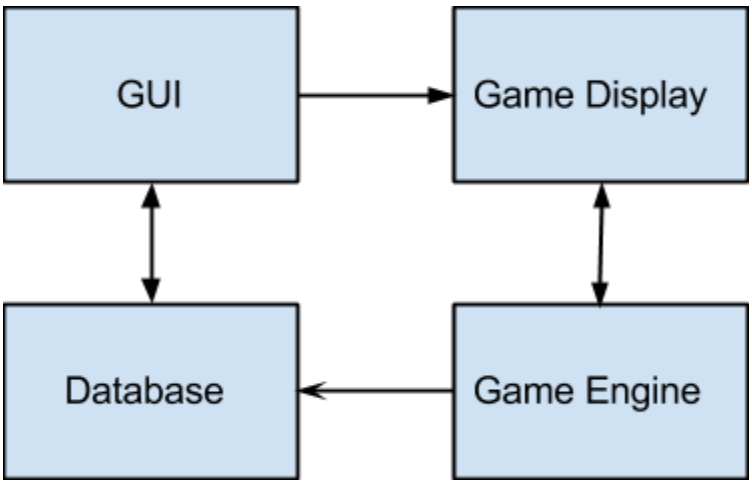
Our game has three main components. The first is the GUI/Game display, second is the underlying game engine, the last part is the database.

The GUI will include user menus, where the user can choose to start a new game, continue from previous, or exit. The GUI will also allow users to check the score list. Game display includes the actual character and object models.

Game engine is the heart of the program that processes information that is being passed from the user. When a user clicks corresponding buttons the game engine handles those requests and acts

according. It also handles collisions between walls, obstacles, enemies, and the character model.

The final part of the game is the database. This is what keeps track of the user scores and it saves the scores locally. The user will need to provide a username and the database will store the user score and username. The user will also be able to check the scores of other players.



4. Functional Requirements

4.1 Interaction between the user and the GUI

- ❑ The user will input their username and password to the GUI .
- ❑ The GUI will create an account for a new user or retrieve the account information from the database after user input.
- ❑ The user can also choose game difficulties and can access old scores

4.2 Interaction between the game display and the game engine

- ❑ The game display will pass data about the location of character model and the engine

will determine how the character model interacts with the environment.

4.3 Interaction between the user and the game display

- ❑ When the user is playing the game, the game display will record input and transfer it to the game engine.

4.4 Interaction between the GUI and the database

- ❑ The GUI will retrieve user's data from the database.

4.5 Interaction between the game engine and the database.

- ❑ The game engine will transfer the high scores to the database.

5. Non-Functional Requirements

Response time

While user play the game, game display should be able to play fluently, and could respond to the user quickly when refresh the graphics. And the score could be uploaded quickly when user wants to upload the score to our database. The game would not get stuck for a long time. The user's information could be stored immediately once the user registered a new account.

Security

To ensure security, nobody is allowed to see the other users' data stored in the database. The user could only get access to his/her own information. As well, the user can not modify his/her score.

Platform

The game language is JAVA, developed on eclipse. So any computer support java could be accessed to our game.

Reliability

When a crash happens, the score should be able to be stored immediately and could be recovered when user restart the game so that user could continue to play game even after crashing. The database can not be accessed by user in case of the data modification. The user can reset his/her personal information and could get his/her password from email if he/she forget password.

Usability

The game should be easy to play and needs a little strategy to accomplish the game. The user can easily modify his/her personal information.

6. Use Cases

Case: The user wants to create an account.

1) The user registers for an account by inputting username and password in the user registration page.

2) Once the username and password are entered, they are verified accordingly.

a) If the inputted username exists, the user will be asked to input another username.

b) If the inputted password is less than the required length, the user will be asked to input another password.

Case: The user wants to sign into the created account.

1) The user signs into his or her account by inputting his or her username and password accordingly.

2) Once the username and password are entered, they are verified accordingly.

a) If the inputted username does not exist, the user will be asked to input another username.

b) If the inputted password does not exist, the user will be asked to input another password.

Case: The user wants to adjust game difficulty prior start of the game.

1) The user adjusts game difficulty by clicking “Difficulty” button.

a) He or she then chooses whether the game is easy, medium, or hard in difficulty.

b) Movement speed of the character model and enemies changes accordingly based on the game difficulty. The harder the difficulty, the slower the movement speed of the character model, and the faster the movement speed of the enemies.

Case: The user wants to have a look of top scores recorded so far.

1) The user is given a list of his or her top five personal scores recorded so far on the main menu.

2) The user is given a list of top five overall scores recorded so far on the main menu.

3) The score lists are updated after end of each game session.

Case: The user wants to have an in-game menu button that enables him or her to carry out some functionalities while playing the game.

1) The user clicks the in-game menu button located on top right side, and has access to an array of functionalities.

a) The user goes back to the main menu by clicking “Back to Main Menu” button. He or she is directed to the main menu. If the final total score is among the newest top ten scores, it is recorded into the database for future display on the main menu.

b) The user starts from a particular level of the current game by clicking “Retry Level” button. Total

score obtained from previous levels will remain intact.

c) The user exits the program immediately by clicking “Exit” button. The game closes. If the final total score is among the newest top ten scores, it is recorded into the database for future display on the main menu.

Case: The user wants to logout from his or her account.

1) The user clicks “Logout” button at the main menu. He or she will be directed to user login page.

Case: The user wants to exit the program.

1) The user can exit the program by several ways.

a) The user clicks “Exit Game” button at the main menu. The program terminates.

b) The user clicks close button at top right of window. The program terminates.

Case: The character model interacts properly with the environment.

1) Character model should not be able to walk into or pass through walls.

a) The game display will pass coordinates to the engine for verification.

2) Character model needs to be reset to the entrance of the level after making contact with hazards.

a) The character model’s life gets deducted by one for each contact. The game ends if its life reaches zero.

3) After character model reaches the next level checkpoint, the game will then switch to next level.

Case: The user wants to control the character model precisely using keyboard.

1) When the user inputs an arrow keystroke indicating a direction (e.g. down), the character model will move in the indicated direction.

a) When the user holds the direction keystrokes for a period of time, the character model will move continuously in the indicated direction until the user releases the keystrokes.

Case: The user wants to renew his or her total score after each level, and carry the score to next level.

1) The game engine keeps track of the total score throughout the game, and saves the score into the database when the user completes a level.

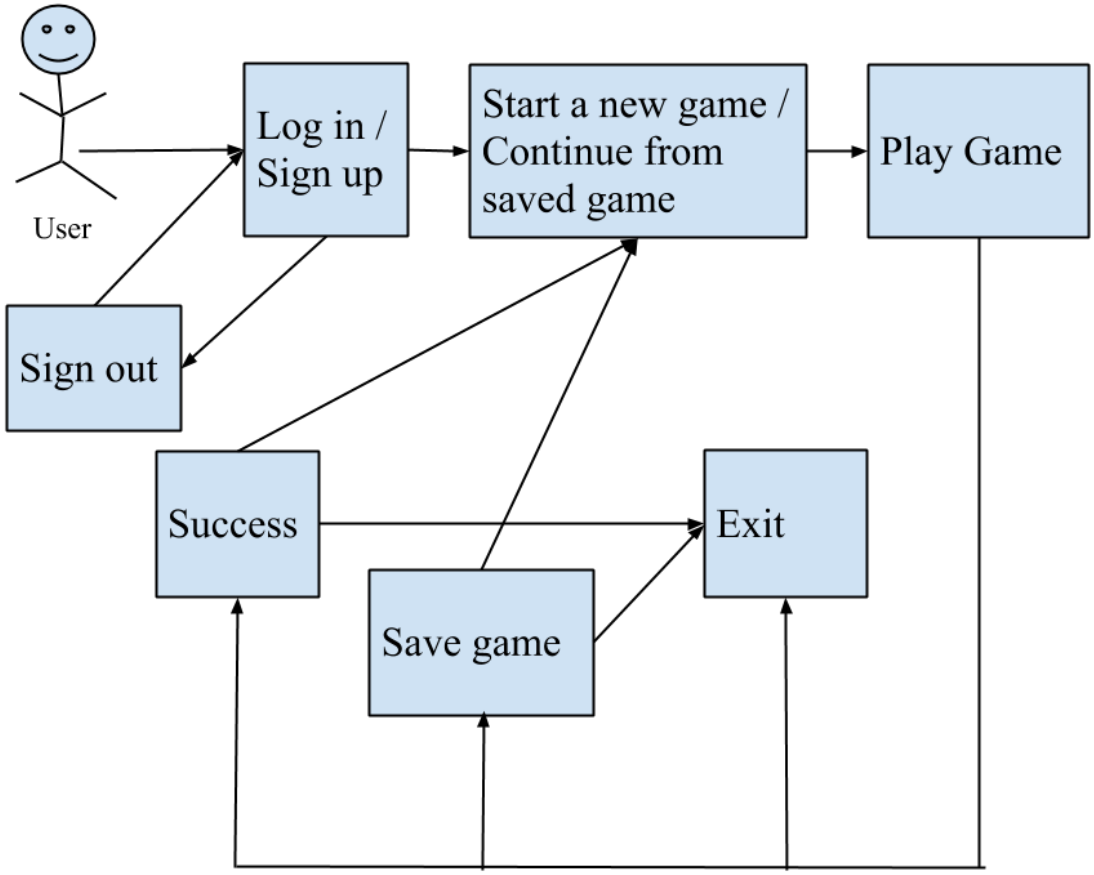
2) If the user fails to complete a particular level, the total score will not be renewed. This ensures that the total score is correctly recorded in the database when the user retries the level.

3) The score for a particular level will be determined by several factors.

a) The score is higher if remaining amount of lives of the character model is higher.

b) The score is higher if the character model opens a less number of doors.

c) The score is higher if the character model found treasure at a particular level.


User
Log in / Sign up
Start a new game / Continue from saved game
Play Game
Sign out
Success
Exit
Save game