Game Design Document

Group of the 104

EDIDS 104 project 2024

1 Introduction

Text Adventure Game is a role-playing game based on a horror story, where the player has to explore a series of rooms called '**Nightmares**'. Our goal is to create an immersive experience for the player focused on **solving puzzles** and interacting with various special items and NPCs within the game environment.

2 User stories

Therefore are listed the user stories :

1. **Start a New Game Session** "As a Player, I want to start a new game session so that I can play the game."
2. **Save Game Progress** "As a Player, I want to save the current progress of the game so that I don't lose the progress."
3. **Load a Saved Game** "As a Player, I want to load a saved game session so that I can continue from where I left."
4. **Quit the Game** "As a Player, I want to quit the game so that I can stop playing."
5. **Display List of Commands** "As a Player, I want a list of commands so that I can see what I can do."
6. **Move Around the map** "As a Player, I want to move around the map so that I can explore rooms."
7. **Interact with Items and NPCs** "As a Player, I want to interact with items and NPCs so that I can get information about them."
8. **Look Around** "As a Player, I want to look around so that I know the items in the room."
9. **View Inventory** "As a Player, I want to see the inventory so that I know the items I can use."

10. **Game Over** "As a Player, I want to know if I ended the game and if I cleared the game"

3 Functional requirements

The requirements are invaluable to determine the base functionalities of the game. We have implemented these features so that the player can perceive the best experience out of the game.

1. **Start a New Game Session**
    - The system must allow the player to start a new game.
    - There should be an option in the main menu to start a new game session.
2. **Save Game Progress**
    - The system must allow players to save their game progress.
    - There should be a command to save the game at any point during gameplay.
3. **Load a Saved Game Session**
    - The system must allow players to load a previously saved game session.
    - There should be an option in the main menu to load a saved game.
4. **Quit the Game**
    - The system must allow the player to quit the game at any time.
    - There should be an option in the menu to quit the game.
5. **Display List of Commands**
    - The system must provide a help menu containing a list of available commands to the player.
    - There should be a command to display this list at any time.
6. **Move Around the Map**
    - The system must allow players to move within the game map.
    - The system must allow players to select which room they want to explore.
    - Invalid movements should output an error message to the player.
7. **Interact with Items and NPCs**

- The system must allow the player to interact with items and NPCs in the room.
- It should be possible to perform actions on various items.
- It should be possible to enter combat with NPCs.
- The combat should implement health/hit points (HP) and in case of "Victory" or "Defeat" it displays a feedback message.

8. **Look Around**
   - The system must allow the player to look around to get a description of the surrounding environment.
   - The player should be able to see the room and the items present within it.
9. **View Inventory**
   - The system must provide an inventory that shows the player the items they possess.
   - There should be a command to view the inventory at any time.
10. **Game Over**
    - The system must alert the player in case of game over.
    - There should be an in game feedback message that tells the player in case of game over, the outcome of the session.

4 Non-functional requirements

1. **User Friendliness**
   - The game should be easy to use and the menu should be easy to navigate.
2. **Speed and Performance**
   - The system must give an output whether a selected actions has been performed. If an action has failed, it should display an error code.
3. **Error Resilience and Usability**
   - The program should undergo various testing phases in order to deliver the best experience to the player. To achieve this the system should be able to handle errors during its execution.

5 Acceptance criteria and System test report

1. **Start a New Game Session**
   (a) The player can interact with "**newGame**" command to start a new game session from the menu.
   (b) A new game session has all the variables set to default values.
2. **Save Game Progress**
   (a) The player can interact with "**save**" command at any point during the game.
   (b) Upon saving the current game session, all data is stored in an online cloud storage.
   (c) The game confirms that the progress has been successfully saved.
3. **Load a Saved Game Session**
   (a) The player can load a previous saved game session by interacting with "**loadGame(sessionName)**" command from the menu.
   (b) If the game failed to load a saved session, it displays the error.
   (c) The loaded game restores all data from the previous session, including inventory, variables and progress.
   (d) The player can delete the a previous saved session.
4. **Quit the Game**
   - The player can interact with "**exit**" command at any point during the game to quit the game.
5. **Display List of Commands**
   - The player can interact with "**help**" command at any point during the game to display the list of available commands.
6. **Move Around the Map**
   (a) The player can interact with "**enter(roomName)**" command to enter in the selected room in the map.
   (b) Upon moving the game updates the current location of the player.
   (c) The game prevents movement in directions that lead to non-existent rooms.
   (d) Invalid movements or "roomName" will output an error message.
7. **Interact with Items and NPCs**
   (a) The player can interact with "**use(itemName)**" command to use the item in the inventory, if applicable.

(b) The player can interact with ”**interact(itemName)**” command to interact with the available item in the environment.

(c) Upon issuing a command, the game provides appropriate feedback.

(d) If the player inputs an invalid ”itemName” it will display an error message.

(e) If the the player picks an item successfully, the item is added to the player’s inventor, if applicable.

(f) The player can interact with NPCs with ”**attack**” command to enter combat.

(g) The combat should be based on Hit Points and at the end display the outcome of the combat (**Victory** or **Defeat**), if applicable.

8. **Look Around**

   (a) The player can interact with ”**look around**” command to get a description of the surrounding environment.

   (b) Upon issuing the command, the game provides a description of the current room.

9. **View Inventory**

   (a) The player can interact with ”**inventory**” command to display the inventory.

   (b) The inventory list is accurate and reflects any items the player has picked up or used.

10. **Game Over**

    (a) The system give the player a feedback in case of game over.

    (b) The feedback should display the outcome of the session.

    (c) The game ends properly after game over.

For system test report, please see Table 1

Table 1: System test report

ACCEPTANCE CRITERIA	OK/KO
1 a	OK
1 b	OK
2 a	OK
2 b	OK
2 c	OK
3 a	OK
3 b	OK
3 c	OK
3 d	OK
4	OK
5	OK
6 a	OK
6 b	OK
6 c	OK
6 d	
7 a	OK
7 b	OK
7 c	OK
7 d	OK
7 e	OK
7 f	OK
7 g	OK
8 a	OK
8 b	OK
9 a	OK
9 b	OK
10 a	OK
10 b	OK
10 c	OK