Executive Summary

My software is a text-based game in which a player controls a character they create. Their character will be controlled in an adventure where they are required to explore dungeons, fight monsters, and collect treasure. The game will be displayed in a window primarily through text, but there will be various other buttons and fields for the user to interact with as well.

Core Requirements:

|  |  |  |
| --- | --- | --- |
| Requirement Number | Name | Description |
| 1.0 | GUI based output | The entire application will run in a graphic user interface window. |
| 1.0.1 | View window | All output will exist in a text area in the top half of the screen, and all input fields and buttons will be below. |
| 1.1 | Runnable .jar file | The application should launch from a runnable .jar file when clicked. |
| 1.2 | Gameplay | The game consists of choices to explore a dungeon, or to visit a town to heal and prepare to explore the dungeon. |
| 2.0 | Player/Hero Object Exists | The user should be able to create a new hero to play the game. |
| 2.1 | Player Attributes | The hero has the following items and attributes:  -Name: User created name of the hero.  -Gold: In game currency that can be found and spent in shops. (req. 7)  -Strength: See req. 2.1.1  -Defense: See req. 2.1.1  -Current Health: See req. 2.1.2  -Max Health: See req. 2.1.2  -Pack: See req. 3.0  -Weapon: See req. 4.0 |
| 2.1.1 | Strength and Defense | The strength and defense attributes are used in combat calculations and are tracked and updated automatically. |
| 2.1.2 | Health | Current Health is a measure of damage taken from monsters and effects. It can be recovered but cannot exceed Max Health. |
| 2.1.3 | Display stats | Up to date player stats are displayed throughout the game in the GUI. |
| 3.0 | Pack | Players have a pack that contains items acquired throughout the game. |
| 3.1 | Using Items | Players will be able to access their pack at any time to interact with items. |
| 3.2 | Updating Pack | Pack items are removed and added automatically. |
| 3.3 | Items | Items exist and have:  -description: What the item is.  -effect: what the item does |
| 3.3.1 | List of Required Items | -Potions: Restores Player health. exists in varying strengths  -Talisman Pieces; Collecting three of these gives a boost to player stats |
| 4.0 | Weapon Exists | Players have a Weapon object with its  own attributes |
| 4.1 | Weapon Attributes | Weapon has the following attributes:  -name/type: what the weapon is.  -strength modifier: a measure of how much it improves a player’s strength in combat  -description: a description of the weapon’s appearance and effects  -cost: amount of gold required to purchase from shop if available. |
| 5.0 | Monsters exist | Monster Objects exist and interact with Player objects through both random and predetermined encounters. |
| 5.1 | Monster Attribute | Monsters have attributes similar to player attributes namely:  -name/type: the type of monster that the player encounters  -description: general description of the monster  -strength and defense: As for player, used in internal calculations. |
| 5.2 | Attribute display | Monster Attributes are updated and displayed for the duration of combat scenarios. |
| 5.3 | Generating  Monsters | Monsters are generated randomly and with increasing stats based on game progression. |
| 5.4 | Combat | When a player encounters a monster, a combat sequence starts. The player and the monster take turns dealing damage until combat ends. |
| 5.4.1 | Damage | Damage is calculated through adding the attacker’s base stats to a random number generator results and subtracting the defender’s stats. |
| 5.4.2 | Ending combat | Combat ends when one of the following conditions is met:  -player’s health reaches zero.  -monster’s health reaches zero  -player escapes successfully. |
| 5.4.3 | Escape | Player’s may attempt to escape combat if a randomly generated number roll exceeds a set number determined by monster type. |
| 6.0 | Dungeon Exists | A dungeon object exists for the player to explore |
| 6.1 | Attributes | The Dungeon has the following:  -Floors: The number of floors a player must clear to beat the game.  -Current floor: The floor that a player is currently exploring. |
| 6.1.1 | Room | Each Floor is composed of Rooms that have:  -description: A description of the room  -encounter: what happen when the player enters the room.  -reward: what happens when the player completes the room |
| 6.2 | Generating Rooms | Rooms are generated randomly and procedurally upon entrance of the Dungeon. -Constantly changing layouts as a game mechanic. |
| 7.0 | Shops Exists | Shops exists and are made up of Items. |
| 7.1 | Functionality | Player will only be able to purchase items if they have enough gold. |
| 7.2 | Economy | Gold serves no purpose outside of players purchasing items/weapons. |
| 8.0 | Game/Controls and User Input | No freeform text input will exist outside Hero creation. All user input will be through number menus and buttons. |
| 8.1 | Persistence | Game state is savable and can be loaded and reloaded without losing game progress. |
| 8.1.1 | Specifics | -Player stats and health are persistable.  -Pack contents and Player Items are persistable. |
| 8.1.2 | Exclusions | -Shop, Dungeon, Room, and Monster objects need not be persistable. |
| 8.2 | Menu | Top menu bar will have options for saving and exiting the GUI. |
| 8.2.1 | Exiting | Exit menu button will prompt the user to save the game. |
| 8.3 | Starting | Game window will open to the man menu to show the ‘new game’ or ‘load game’ options |

|  |  |  |
| --- | --- | --- |
| Stretch Requirements | Name | Description |
| 2.0 | Multi-Save | Program supports functionality for multiple save files and will allow players to select save file on program start |
| 3.1 | Text “Buttons” in shop. | Shop will display a list of items that can be clicked to purchase instead of actual buttons. |
| 4.1 | Legendary Weapons | Certain instances of weapons grant special bonus effects to players |
| 8.0 | Keyboard exclusion | Entire game is point and click except for necessary fields such as character name selection. |
| 8.2.1 | Exiting Improved | All methods of exiting will prompt to save the game. |
| 9.0 | Side Quests | Side quests exist and are encountered randomly. |
| 9.1 | Attributes | Side quests have:  -dialogue: specific instruction and story associated with the quest.  -requirements: step(s) a player must complete to finish the quest.  -reward: The outcome of succeeding the quest  -punishment: The outcome of failing the quest. |

Scope Exclusions:

-The game will not have any audio components.

-The game’s visuals will be limited to text and inputs. No graphics will be included.

-Alternate game modes and multiplayer support will not be developed.

-GUI layout design will be catered to windowed mode and not full screen.

User Stories:

Req. 1

-A user wants to play the game:

From the desktop-

1: They click on the .jar file shortcut and the game window opens to the main menu. 2: They see options for input in the form of buttons. They click New Game.

3: They enter their name in a text field and click the start button.

Outcome: The game starts

-A user is playing the game:

From the start screen-

1: The user clicks choose to either explore the dungeon or visit the town.

2: The click the corresponding button.

Outcomes:

a: The user explores the dungeon and sees gameplay text

b: The user visits the village and sees gameplay text

Req. 2

-A user wants to check their character

From any point during gameplay-

1: The user looks at the top right corner of the window

Outcome: The user sees their character’s name and stats

Req. 3

-A user wants to use an item

From any point during gameplay-

1: The player clicks the Open Pack Button

2: The play sees and selects from a list of items the item they want to use.

3: They click the item and its’ effect is executed.

Outcome: an item is used

Req. 4

-A player wants to have a weapon

From the main game screen-

1: The user selects the Visit Village

2: The user selects the Visit Weapon Shop button

3: The user is greeted and shown a menu of items available for purchase

4: The use selects the item they would like to purchase, and gold is subtracted

Outcome: The user has a weapon.

Req. 5

-The player wants to fight a monster

From any room in the dungeon-

1: The user presses the explore button until they find a monster.

2: They select the Fight button from the options to Fight or Flee.

3: They are shown a message detailing the outcome.

4: The user continues until they or the monster are defeated.

Outcome: The player has fought a monster

Req. 6

-The user wants to find treasure

From any point in the dungeon-

1: The user selects the Explore button to explore the various rooms

2: The user continues to do so until they find randomly generated treasure

Outcome: The user finds treasure

Req. 7

-A user decides to purchase some other items

From town-

1: The user selects the button corresponding with the shop they wish to visit.

2: The user is greeted and shown a menu of items available for purchase

3: The use selects the item they would like to purchase, and gold is subtracted

Outcomes: The user has bought some items

Req. 8

-A user wants to save and exit game

-From any point during gameplay-

1: The user clicks the menu dropdown from the menu bar at the top of the screen.

2: The user clicks on the Save option from the dropdown menu.

3: The user clicks either the top right x or the exit option from the menu dropdown

Outcome: The game is saved and closes.

-A user wants to load a saved game

From the desktop-

1: The user clicks the game shortcut and the game window opens

2: The user clicks the Load button.

3: The user selects their name from the list.

Outcome: The saved game loads and starts

Req. 9

-A player wants to accept a side quest

From certain random encounters-

1: The user is presented with the story and requirements of the quest.

2: The player selects the Yes button when asked if they will accept the quest.

Outcome: The player has accepted a quest

Technical Stories:

Req. 1

-The runnable .jar file opens the window and starts the game. Button clicks control inputs to the controller and in some instances change the scene and update the display.

Req. 2

-Player base stats are set internally and initialized at the start of the game, and are updated throughout based on encounters

Req. 3

-The pack controls and interacts will all item objects within it.

Req. 4

-Weapon stats and costs are predetermined and add their unique modifiers and abilities to the player

Req. 5

-A player in combat takes damage and their health decreases. The user wants to fight and internal systems calculate the success of an attack and the amount of damage dealt.

Req. 6

-Rooms type and frequency are generated randomly and communicate their details to the controller

Req. 7

-When the user attempts to purchases an item or weapon, internal systems check the item’s cost against the user’s current amount of gold and refuses the sale if the user does not have enough.

Req. 8

-The user selects save game and the software gathers all serializable information and writes it to the correct file.

Req. 9

-A random number generator creates test numbers at set intervals, and when a predetermined condition is met, the user is presented with a side quest

Mock ups:

![A screenshot of a cell phone

Description generated with very high confidence]()A screenshot of a cell phone

Description generated with very high confidence![A screenshot of a cell phone

Description generated with very high confidence]()![A screenshot of a cell phone

Description generated with very high confidence]()

Definitions-

Encounters- the set of actions and effects that occur when a player meets some predefined condition e.g. monster fights.

Stats- The combination of player defining characteristics such as strength, defense, health, etc.

Random- In this context, random refers to a random number result within a certain predetermined range, and not truly random.

User- Individual who uses the software.

Player- Another name for user, one who plays the game.