

Bryan Roberts

Design Document

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.

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.

Requirements Map

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.
GUI will be a desktop window with multiple scenes that change based on user input. The majority of scenes will contain buttons to receive user choices, text area's to display feedback, and quick links to a main menu		

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.
In the top left corner of the GUI, the player's name and stats will be displayed in a text area. The name will appear at the top of the text area, followed by health. Underneath, strength, defense, gold, and talisman pieces will be shown.		
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
In the bottom section of the GUI, a list of the players items will exist. A player will be able to click on an item they want to use, and then it will disappear from the list of items. Acquiring items will likewise add them to the list.		
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.
<p>The majority of this requirement is for back-end use, but there are some user facing features. A description of the weapon will appear in the shop window (see Req. 7), along with its cost. The other place it will be visible is in the top right display window, along with any modifiers it has.</p>		
5.0	Monsters exist	Monster Objects exist and interact with Player objects through both random and predetermined encounters.
5.1	Monster Attribute	<p>Monsters have attributes similar to player attributes namely:</p> <ul style="list-style-type: none"> -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	<p>Combat ends when one of the following conditions is met:</p> <ul style="list-style-type: none"> -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.
<p>When a player is fighting a monster, the monster's type and stats will be listed in a text area in the top left corner of the window. Text commentary on the fight progress will appear in a text area in the middle of the window. All user controls for the fight will exist as buttons in the bottom control area (see Req. 8).</p>		
6.0	Dungeon Exists	A dungeon object exists for the player to explore
6.1	Attributes	<p>The Dungeon has the following:</p> <ul style="list-style-type: none"> -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	<p>Each Floor is composed of Rooms that have:</p> <ul style="list-style-type: none"> -description: A description of the room

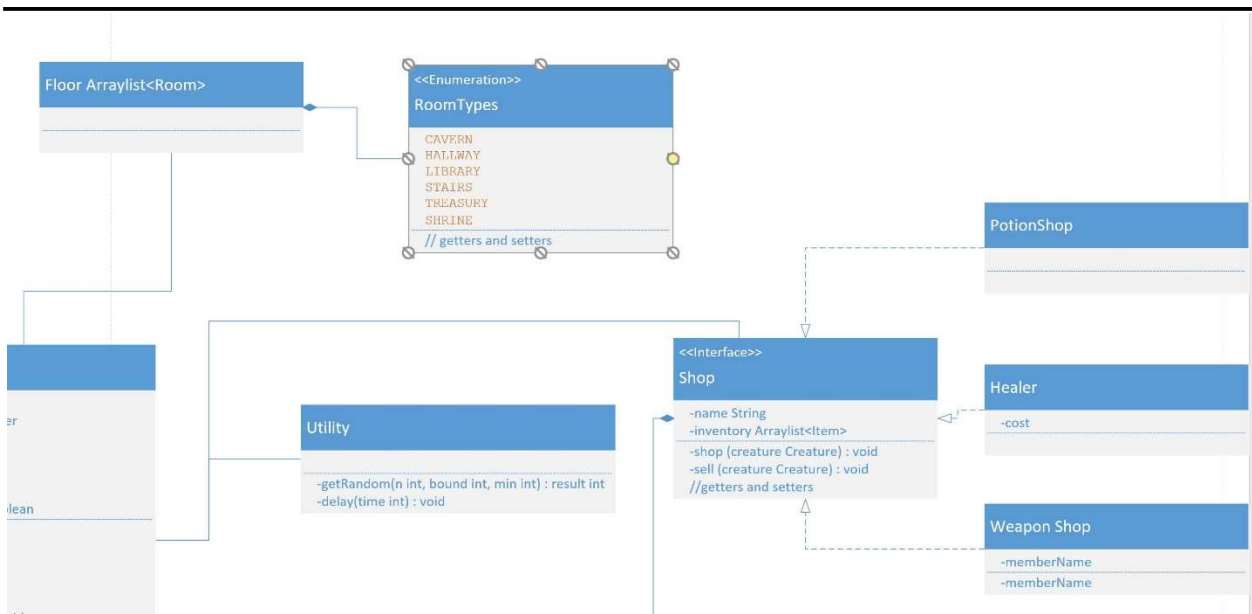
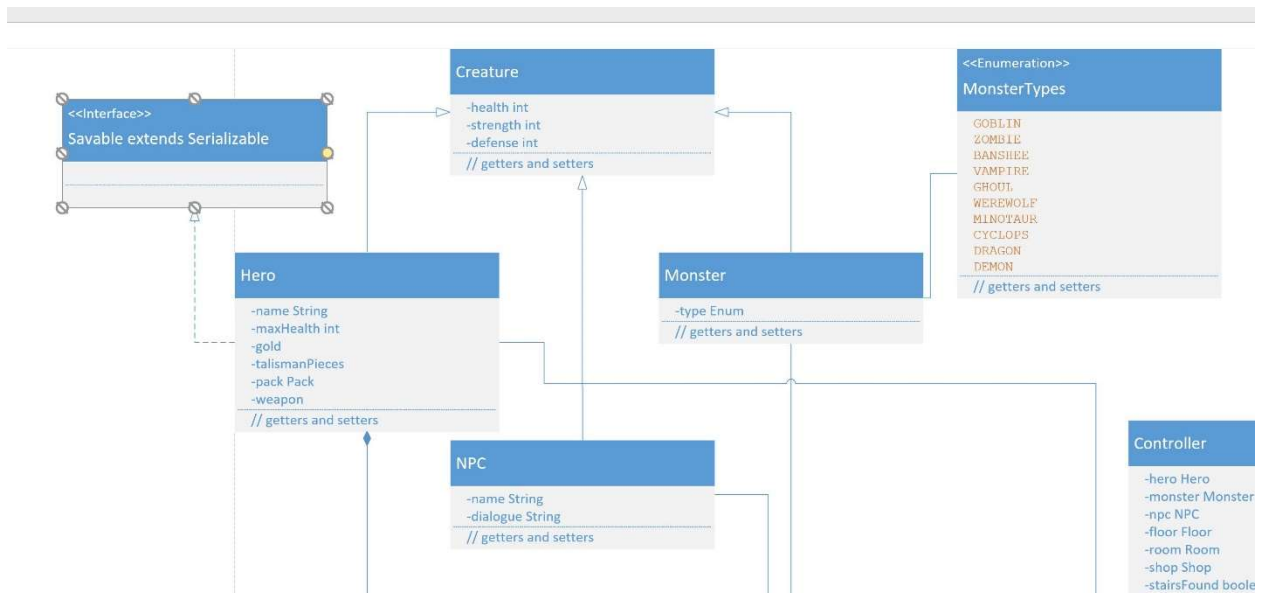
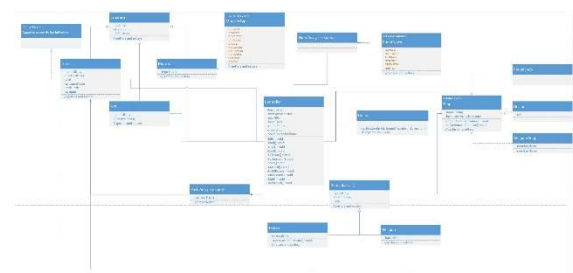
		-encounter: what happen when the player enters the room. -reward: what happens when the player completes the room
6.2	Generating Rooms	Rooms are generated randomly and procedurally upon entrance of the Dungeon. -Constantly changing layouts as a game mechanic.
Mostly back-end. Room descriptions will appear at the bottom of the center display section of the window		
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.
Shop names and interfaces will be displayed in the center display area. A list of items available for purchase will be displayed and clicking them will allow players to purchase them.		
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
Game controls will be mostly in the form of buttons and clickable labels in the bottom section of the GUI. Buttons and labels will change based on the scene, but the layout will remain standard throughout.		

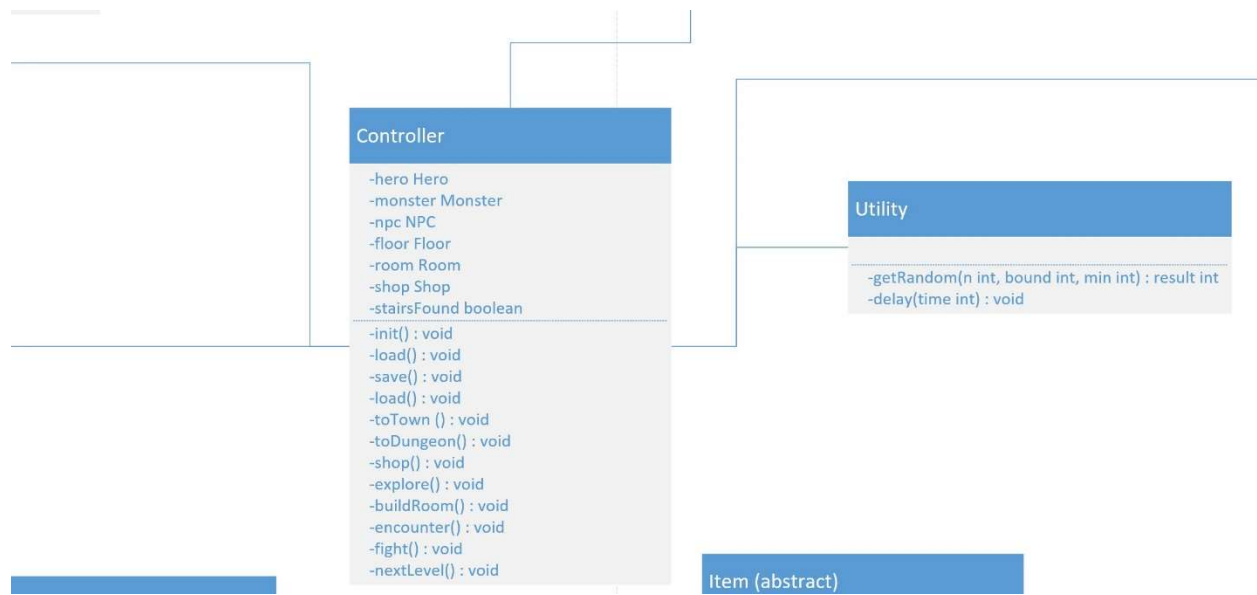
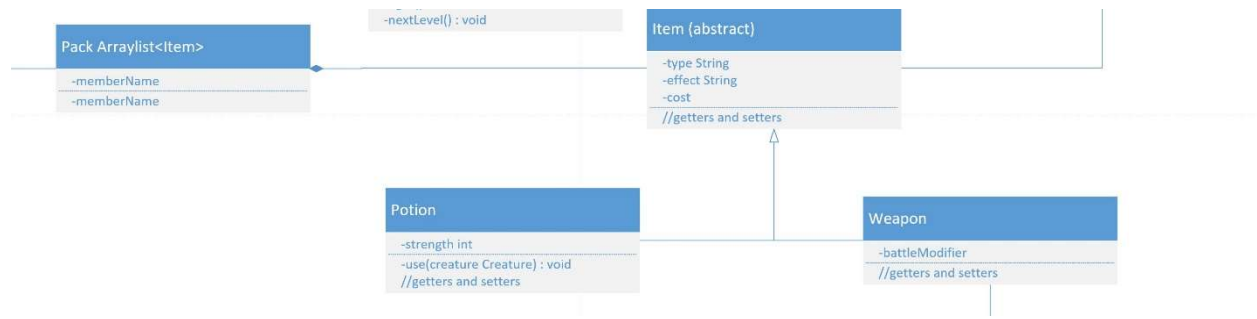
Stretch Requirements	Name	Description
10.0	Multi-Save	Program supports functionality for multiple save files and will allow players to select save file on program start
Back-end only.		

11.0	Text “Buttons” in shop.	Shop will display a list of items that can be clicked to purchase instead of actual buttons.
As described in Req. 7.		
12.0	Legendary Weapons	Certain instances of weapons grant special bonus effects to players
Any bonuses granted will be displayed as text in the player stat text area.		
13.0	Keyboard exclusion	Entire game is point and click except for necessary fields such as character name selection.
13.1	Exiting Improved	All methods of exiting will prompt to save the game.
14.0	Side Quests	Side quests exist and are encountered randomly.
13.0-14.0 are back-end only		
14.1	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.
Any current side quest information will be displayed at the bottom of the player stats text area.		

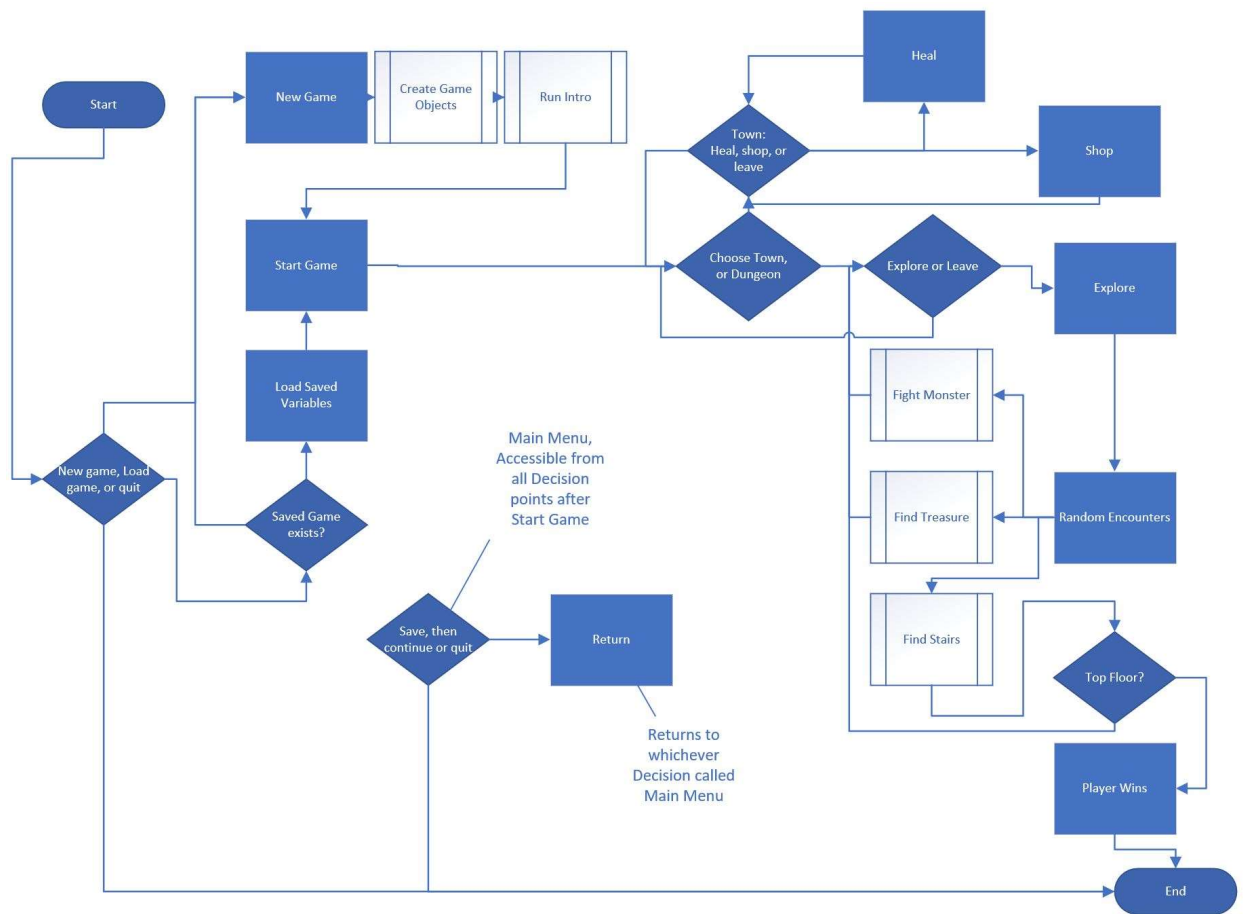
UML

overview -

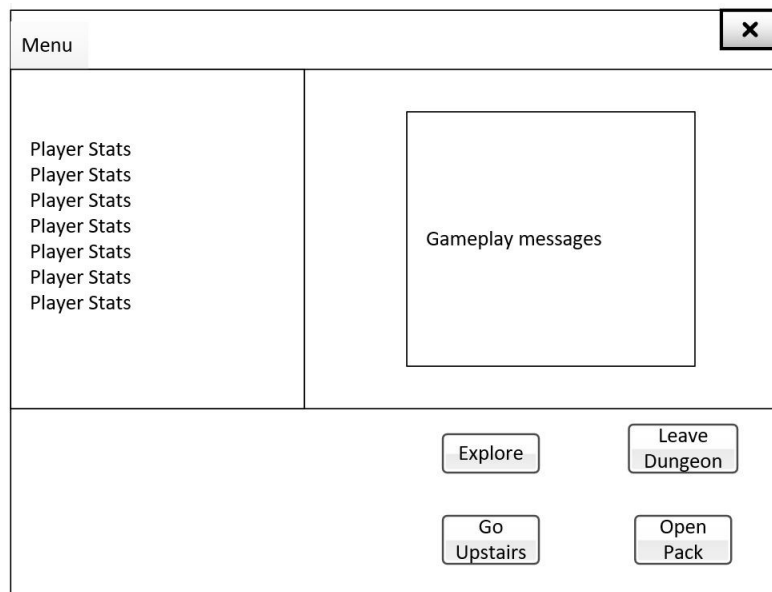


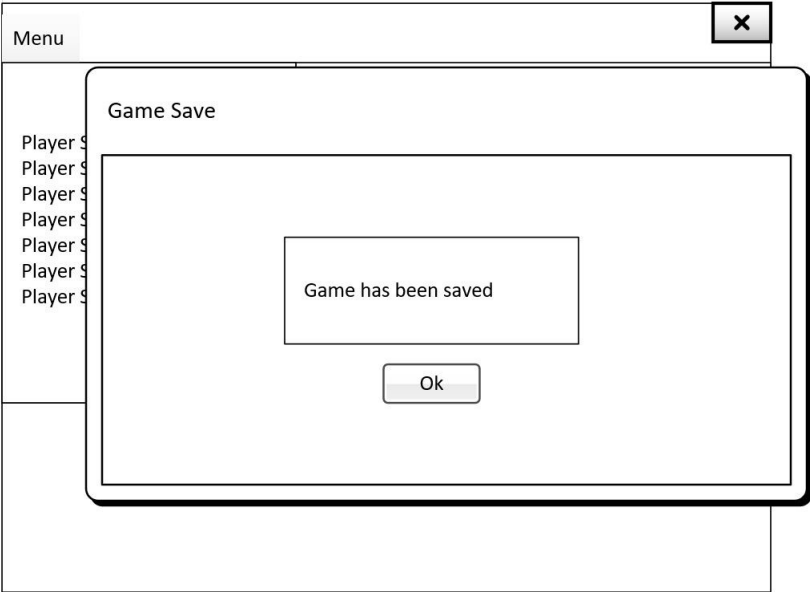
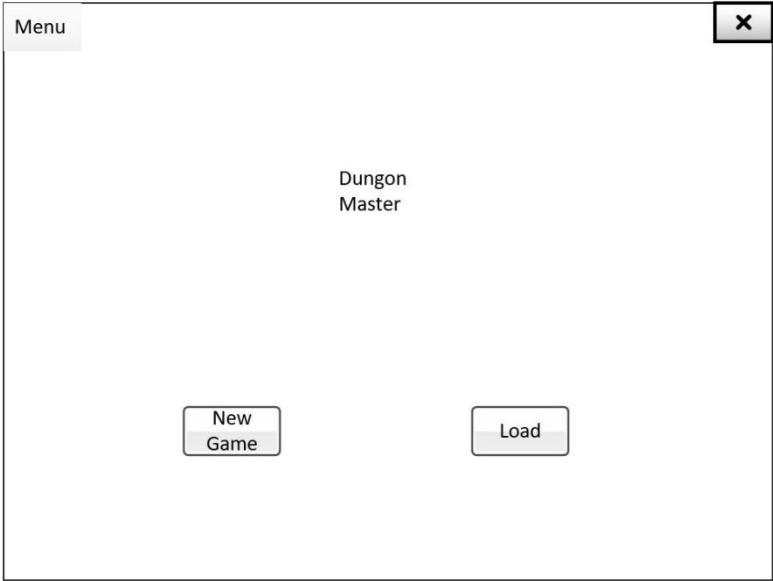


Process Flows



Mock ups:





Test Cases:

Req. 1

Pre-condition: 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

Pre-condition: 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

Pre-condition: 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

Pre-condition: 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

Pre-condition: 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

Pre-condition: 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

Pre-condition: 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

Pre-condition: 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 user selects the item they would like to purchase, and gold is subtracted

Outcomes: The user has bought some items

Req. 8

Pre-condition: 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.

Pre-condition: 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

Pre-condition: 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