Daniel Berekdar

Db19786n@pace.edu

PyQuest

CS32p Project I

**Table of contents Page**

1. **Instructions for running PyQuest 2**
2. **Screen Shots of program running. 3**
3. **Flow chart and Explanation of Design. 6**
4. **Inventory of Core Features. 8**
5. **Testing: Cases, Evidence and Explanation. 9**
6. **Gameplay Testing (Human Players) 13**
7. **List of Known Bugs. 14**
8. **Details of Further Enhancements. 15**
9. **Running PyQuest**

PyQuest is a standalone piece of software that does not have much in the way of getting started. Once the .zip file containing the source code of the program is extracted, one can open the terminal of their computer and execute the program if the dependencies are installed. The dependencies are: NumPy, Pandas, and PyFiglet.

To install the dependencies to the native or virtual environment, one can execute the following commands in terminal:

**pip install numpy**

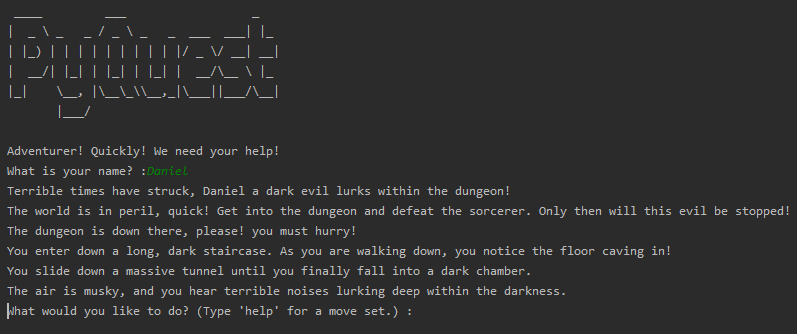
**pip install pandas**

**pip install pyfiglet**

Once the dependencies are installed, one can execute PyQuest by entering the following command in their terminal: **Python3 pyQuestMain.py**

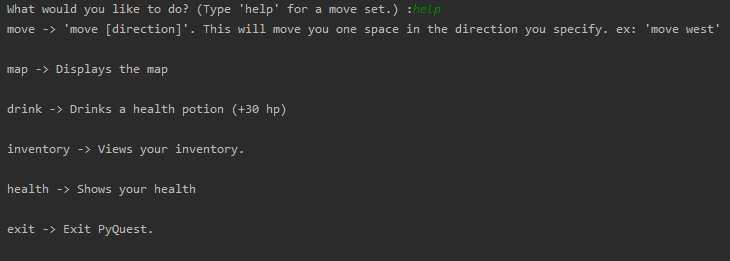
1. **Screenshots of the Game Running**

**2.1 Introduction Screen**



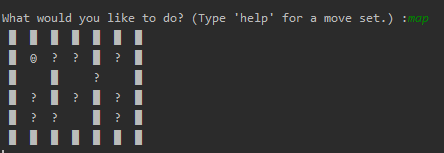
This is the introduction screen of PyQuest.it takes inspiration from classic role playing games and dungeon crawlers by using a traditional hero trope and having a call to action. In PyQuest the player can enter your name to enable characters and events to use your name and enable a deeper level of immersion.

**2.2 Help Menu**



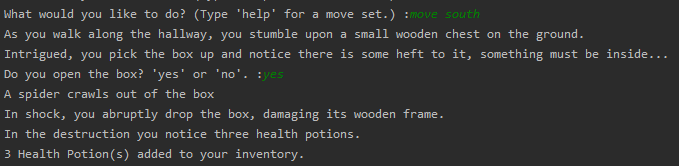
This is the help menu, which allows you to see the options the player has in exploring mode. At any point except combat, the player can carefully read the instructions because they are not at danger unless they begin moving to different coordinates on the grid.

* 1. **Map**



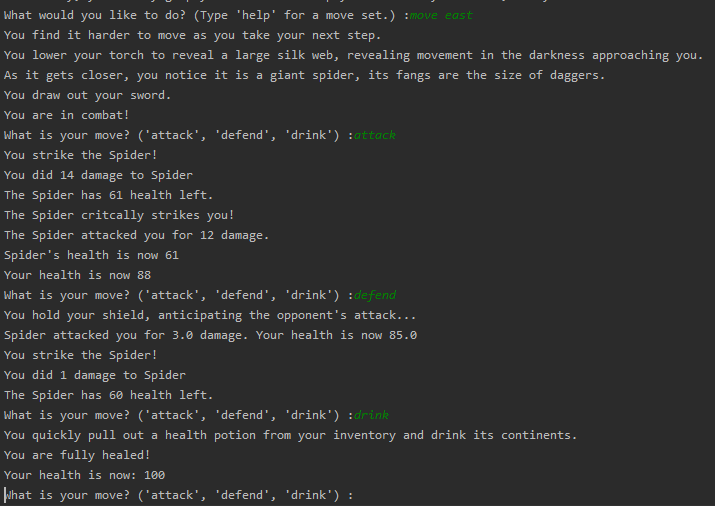
This is the map section of the game that allows the player to see the layout of the dungeon and the different events that take place. Because it is a dark dungeon, the player can’t figure out what an event is until they go to that coordinate.

* 1. **Non-Combat Event**



This is a non-combat event in the game that provides the player with some health potions if they choose to open the box. The player is forced to open this chest if they choose to go down this path because the game is incredibly challenging and almost impossible without this health potion drop.

* 1. **Combat Event**



When the player steps on a combat event, various scenarios may take place before finally engaging in combat. When in combat, the player has the option to attack, defend, or drink. This allows the player to use some strategy. Combat is turn based, therefore the player has time to think about which move they would like to use. There is a twenty percent chance for the player and NPC to critically strike, and therefore the player is rewarded for predicting a critical hit and choosing to defend, with the cost of less damage. If the player decides to heal, then they will not take any damage or deal any damage but will have an opportunity to recover a few hit points.

1. **Flow chart and Explanation of Design**

As the flow chart details, the main objective of the game is to reach the end of the dungeon without dying. The player has the option to explore the map without consequence unless they step on a coordinate that triggers an event. The events are either combat or a treasure chest. If the player finds a treasure chest, they can take items from it and continue moving. In combat the player has the option to Drink, Attack, and Defend because these are commands that dictate whether the player will survive and continue the game or die and end the game.

The player can continue moving and exploring the map of PyQuest until they receive the dragon key from a dragon boss fight and defeat the sorcerer in any order. Once the player has the dragon key, they can step on the final space and win the game. The reasoning behind the dragon key is to make the game nonlinear and reward the player for choosing different paths to the end of the dungeon on various playthroughs.

1. **Inventory of Core Features**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Function** | **Description** |
| **4.1 Character Creation** | **4.1.1** Create Character | The character creation feature allows the player to create their character and assign their character a name. |
| **4.2 Movement** | **4.2.1** Move to a location. | The movement feature allows the character to move from one coordinate to the next on the grid. |
| **4.3 Move Set** | **4.3.1** Displays the move set. | Allows the player to see the list of movements they can make in the map explore state. |
| **4.4 Combat** | **4.4.1** Attack  **4.4.2** Defend  **4.4.3** Drink (Combat) | Allows the player to engage in combat with an NPC. The player can attack the enemy, defend from a blow and use a weak attack, or use a potion to heal. |
| **4.5 Inventory** | **4.5.1** Put item in Inventory  **4.5.2** Take out of Inventory  **4.5.3** Check Inventory  **4.5.3** Use Item  **4.5.4** Drink (Non-Combat) | Allows the player to store items in an inventory to use inside and outside of combat, and it enables the player to keep track of how many potions they have. |
| **4.6 Health** | **4.6.1** Display health. | Allows the player to display their health in order to justify drinking a potion. |
| **4.7 Map** | **4.7.1** View map. | Enables the player to see where they are on the map. |
| **4.8 Exit** | **4.7.1** Exit the game. | Allows the player to exit the game if they no longer want to play the game. |

1. **Testing: Cases, Evidence and Explanation**

|  |  |  |  |
| --- | --- | --- | --- |
| Error Test Case | Explanation | Prevention Method | Evidence |
| **5.1 Null Character Name** | The player enters a name that is invalid, which would cause errors later for functions which reference the character’s name. | Using a while loop to catch null character name input, and continuously prompting the player for a character name until they enter it. |  |
| **5.2 General Invalid Commands in Map Explore Mode (General)** | The player might enter a command on the movement screen which could crash the game. | Having an else statement in the map explore logic which doesn’t include valid commands, and lets the player know they entered a bad command. |  |
| **5.3 Invalid Movement Command** | The player might try to move to an invalid direction, or misspell their direction, or forget a direction | Having a try catch block and if-else statements to prevent the player from entering input that would crash the game. Since the player’s movement is stored as a list with two indices, if there is no movement[1] value, the program will crash. |  |
| **5.4 Moving Outside of the Map,**  **Moving Through a Wall.** | The player accidentally walks outside of the map and causes an index error. | The map is a coordinate grid on a NumPy 2D Array, therefore Assigning dungeon walls the value of 1 and creating logic that prevents players from walking through “1” spaces prevents the player from walking through those walls. |  |
| **5.5 Event fails to Clear after Completion.** | The character completes an event, but the event isn’t cleared, allowing the player to complete the same event multiple times. | Completing an event and having the game logic detect that the event is over so that the player cannot repeat it by having the number of the event replaced by a zero, which indicates an empty space by the game logic, and optionally moving the character on that spot depending on the event. |  |
| **5.6**  **Invalid Chest Option** | The player tries to open a chest but does not enter the right output. | Allowing the player to be creative in their answer and defaulting to opening the chest unless they specifically type “no” to not open the chest. The chests are not optional because the game is impossible without opening them. |  |
| **5.7 Drinking a potion with 0 potions in your inventory.**  **(Combat and non-combat drinking)** | Due to the player’s inventory being a Python list, if a player drinks a potion with an empty inventory, it will crash the game. | Adding a try catch block to catch a Python ValueError, and letting the player know that they tried to drink a potion on an empty inventory. | (Combat)    (Non-Combat) |
| **5.8 Entering an Invalid Combat Command** | The player enters an invalid command while in combat, causing the game to crash or a turn to be wasted. | The combat screen will not register this command and allow the player to enter a new command via the if-else blocks of the combat system. |  |
| **5.9 Dying and allowing the player to continue the game.** | The player dies and the game did not end. | Setting a global Boolean flag that ends the game in the main logic of the game if the player dies. If dead == True, the player dies and the game is done. |  |
| **5.10 Improper Spelling and Case Handling (Combat and Non-Combat)** | The player misspells a command or plays the game with the caps lock on. | When the player enters a query, it will look for the word in the phrase, the player entered, and then convert it to lower case throughout the entire game. Note: The actual command has to be correctly spelled somewhere in the query. |  |
| **5.11 Trying to end the game without the Dragon Key** | If the player reaches the end of the game but does not have the Dragon Key in their possession, the game will not end. | Adding logic that looks through the characters inventory and confirms that they do indeed have the Dragon Key in their inventory and having a global Boolean flag that is triggered when at the end of the game to exit the program and end the game. |  |

**Gameplay Testing (Human Players)**

Other testing, that was conducted in the production of this game was to have friends and family playing it with only information based on the README file bundled with the game. I provided no help to playing the game aside from running it on the command line of my laptop.

The general feedback was interesting because I received mixed results based on the person who played it. People who are familiar with role playing games, dungeons and dragons, or videogames in general appreciated the text based medium of the game and thought it was a unique experience for them.

On the other hand, my friends that don’t like videogames were very confused about how to play the game and did not enjoy typing commands and input into the terminal in general.

Some of the common feedback I heard from my brutally honest friends was that the storytelling of the game was cheesy but the concept and the creativity behind it was engaging.

1. **List of Known Bugs**

|  |  |  |
| --- | --- | --- |
| **Bug Name** | **Explanation** | **Example** |
| **6.1 Certain Misspellings not caught.** | If a command is misspelled in the middle of the word, it will not register. |  |
| **6.2 Boss Fight has the Potential to be Unwinnable.** | If the final boss has a very lucky streak of critical attacks, you may run out of potions and be unable to kill him. | You don’t have enough potions when encountering the final boss, and then he gets an incredibly rare series of critical strikes on you. There will be no way to deal with his damage. In testing this never happened but it is certainly in the realm of possibility. |
| **6.3 Getting stuck in the beginning of the game without opening a chest.** | If you are very brave and try to get past the beginning of the game without opening the potion chests, you will be stuck. This is to prevent the bug from happening. | There is no path to the end of the game without opening a chest. You will be stuck in the beginning sequence. |
| **6.4 No real “yes” Option on Chests.** | If the player encounters a chest, the options are no, or anything. There is no option for an ambiguous answer, you either leave the chest alone, or take it.  “yes”, “maybe” or “not really” will all equate to a “yes” in the game’s logic. The chests intentionally default to the easier option. |  |

1. **Details of Further Enhancement**

|  |  |
| --- | --- |
| **Enhancement** | **Explanation** |
| **7.1 Bigger Map** | Having a bigger map would make the game more interesting because it can be more maze like and have more story. |
| **7.2 More Intricate Combat System** | The combat system is very rudimentary, and with more time and commitment it would be interesting to add more combat features such as fleeing, charged attack, and area of effect attacks. |
| **7.3 CO-OP (Multiplayer) System** | It would be interesting to have a feature where you can play the game with your friend over the internet via socket programming, or to fit the retro style of the game, have “keyboard coop” where you take turns with your friends on the same keyboard. |
| **7.4 More Dungeon Levels.** | With more time and dedication, it would be nice to extend the game to include more levels, characters, enemies, and bosses. |
| **7.5 Equipment** | Instead of just an inventory of potions, it would be cool to have an inventory, and an equipment list as well to have different items affect different stats. |
| **7.6 Enhanced Character Creation** | Instead of the default melee warrior class, it would be interesting to have the character specify which stats they want to a allocate to their character and allow pure dungeon and dragons style min-maxing. |
| **7.7 Enhanced Graphics** | Although it fits the retro theme to be text based, it would be interesting to have enhanced visuals to upgrade the player’s experience. |