4. System Use Cases

4.1 Display U1

1. Objective - Displaying all of the relevant information ie. inventory, item stats, guide, consolem, player stats and if in the shop, the shop

2. Priority - High

3. Source - User

4. Actors - User

5. Flow of Events

5.1 Basic Flow

- User boots up the program

- Bootup screen shows up

- The method displays data

6. Includes - tba

7. Preconditions - The user runs the program

8. Post conditions - The user now has visual indication on what the program does

9. Notes - If the display has weird formatting, the user should increase the resolution

4.2 Move in inventory U2

Objective - Moves the current cursor to a different item

Priority - High

Source - User

Actors - User

Flow of Events

5.1 Basic Flow

5.1.1 User boots the program and waits for it to inicalise

5.1.2 The user presses the arrow keys

5.1.3 Cursor changes positions

5.2 Alternate Flow

5.2.1 The user boots the program and it inicalises

5.2.2 The pointer is up a border, and the user tries to move it outside

5.2.3 The pointer does not move outside the border

Includes - u2

Preconditions - user must run the programhttps://docs.google.com/document/d/1Oq\_pliZkuIow-lsHPlMatdPbbWgMpyzOvQ5oBXfktME/edit?usp=sharing

Post Conditons - The user can choose witch item he wants to use, sell ect.

Notes/Issues -

4.3 Help

1. Objective - Shows Help window

2. Priority - High

3. Source - User

4. Actors - User

5. Flow of Events

5.1 Basic Flow

5.1.1 User boots the program and waits for it to inicalise

5.1.2 The user presses ‘the ‘H’ key or write help in console mode

5.1.3 help window shows

5.2 Alternate Flow

5.1.1 User does not need help

6. Includes

7. Preconditions - user must run the programhttps://docs.google.com/document/d/1Oq\_pliZkuIow-lsHPlMatdPbbWgMpyzOvQ5oBXfktME/edit?usp=sharing

8. Post Conditons - The user can choose witch item he wants to use, sell ect.

9. Notes/Issues -

4.4 Get Item Stats

1. Objective: To display the statistics of the item at the current position in the inventory grid.

2. Priority: High

3. Source: User

4. Actors: User

5. Flow of Events

5.1. Basic Flow:

The player requests to view item stats.

The system calls the getItemStats function.

The function retrieves and displays the item details if an item is present at the current coordinates.

5.2. Alternative Flow(s): None

5.3. Exception Flow(s):

If there is no item at the current coordinates, a message indicating "Nothing" is displayed.

6. Includes: None

7. Preconditions: The player is positioned over an item in the inventory grid.

8. Post conditions: The item stats are displayed to the player.

9. Notes/Issues: None

4.5 Enter Store

1. Objective: Enters store

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press t key

The system changes shop\_mode to oposite value

The function retrieves and displays the store

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Player see his inventory

8. Post conditions: Player can buy sth in store

9. Notes/Issues: None

4.6 curse item

1. Objective: Curses item

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press c key

The system calls curseItem function

The function retrieves and curse the item

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Selected the default item

8. Post conditions: Selected item are Cursed item now

9. Notes/Issues: None

4.7 drop item

1. Objective: drobs/ deletes items

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press x key

The system calls dropItem function

The function retrieves and drops the item

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: selected item in inventory

8. Post conditions: selected item deleted from inventory

9. Notes/Issues: None

4.8 sell item

1. Objective: sell items from users inventory

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press g key

The system calls sellItem function

The function check if cursor is on the item

item sells and users gold increases

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: selected item in inventory

8. Post conditions: Item sells and user have more gold

9. Notes/Issues: None

4.9 randomise

1. Objective: randomizes items in inventory or shop

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press r key

The system calls randomizeShop or randomizeInventory function depends were the user is

The function randomize all items in Shop or Inventory

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: In inventory are some items

8. Post conditions: In inventory are different items now

9. Notes/Issues: None

4.10 Save

1. Objective: saves players inventory into txt file

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press k key

The system calls saveInventory function

The function saves all inventory into file

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Save file are empty, filled by old save files or does not exist

8. Post conditions: Save file created or updated by current items in inventory

9. Notes/Issues: Does not save users equipment, hp, money etc.

4.11 load

1. Objective: loads players inventory from txt file

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press l key

The system calls loadInventory function

The function load all inventory from file

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Save file exist

8. Post conditions: items from save file are in inventory

9. Notes/Issues: None.

4.12 unEquipWeapon

1. Objective: unequips users weapon and put it into free slot

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press n key

The system calls unEquipWeapon function

The function checks if user has free spaces in inventory

If here some free space:

Put equiped item adress into this slot

Set equiped item as nullptr

if not:

Print in console that user does not have free space

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: User has equiped weapon

8. Post conditions: User does not has equiped weapon

9. Notes/Issues: None.

4.13 unEquipArmor

1. Objective: unequips users armor and put it into free slot

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press m key

The system calls unEquipArmor function

The function checks if user has free spaces in inventory

If here some free space:

Put equiped item adress into this slot

Set equiped item as nullptr

if not:

Print in console that user does not have free space

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: User has equiped armor

8. Post conditions: User does not has equiped armor

9. Notes/Issues: None.

4.14 SortInventoryByPrice

1. Objective: sort invertory by price, from the highest to the lowest

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press f key

The system calls SortInventory function

The function sort inventory uses bubble sort, from the highest price to the lowest

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: some items are placed in different places in inventory

8. Post conditions: items are plased in order by price

9. Notes/Issues: None.

4.15 SortInventoryByPrice

1. Objective: sort invertory by rarity, from the highest(Hakurei) to the lowest(common)

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press v key

The system calls SortInventoryByRarity function

The function sort inventory uses bubble sort, from the highest rarity to the lowest

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: some items are placed in different places in inventory

8. Post conditions: items are plased in order by rarity

9. Notes/Issues: None.

4.16 OpenCase

1. Objective: Opens case

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press o key

The system calls OpenCase function

The function checks if player has free space in inventory

if passes:

The function checks if player has enough gold

if passes:

decrase players gold, and fill one free space by item that has more chance for higher rarity

if not:

print to console that user does not have enough money

if not:

print to console that user does not have free space

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Player has free space in inventory and enough gold to buy and open the case

8. Post conditions: item with higher chance to higher rarity are plased in inventory free space

9. Notes/Issues: None.

4.17 Buy item

1. Objective: Buys item from shop

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press b key

The system checks if user is in shop

The system calls BuyItem function

The function checks if player has free space in inventory

if passes:

The function checks if player has enough gold

if passes:

decrase players gold, and fill one free space by item that been selected by user

if not:

print to console that user does not have enough money

if not:

print to console that user does not have free space

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Player has free space in inventory and enough gold to buy item

8. Post conditions: item that was selected by user are plased in inventory free space

9. Notes/Issues: None.

4.18 quit/exit

1. Objective: exits the program

2. Priority: High

3. Source: User

4. Actors: Player

5. Flow of Events

5.1. Basic Flow:

The player press i key

The system goes into word type mode

The system calls BuyItem function

The function checks if player has free space in inventory

if passes:

The function checks if player has enough gold

if passes:

decrase players gold, and fill one free space by item that been selected by user

if not:

print to console that user does not have enough money

if not:

print to console that user does not have free space

5.2. Alternative Flow(s): None

5.3. Exception Flow(s): None

6. Includes: None

7. Preconditions: Player has free space in inventory and enough gold to buy item

8. Post conditions: item that was selected by user are plased in inventory free space

9. Notes/Issues: None.