

1. Brief introduction __/3

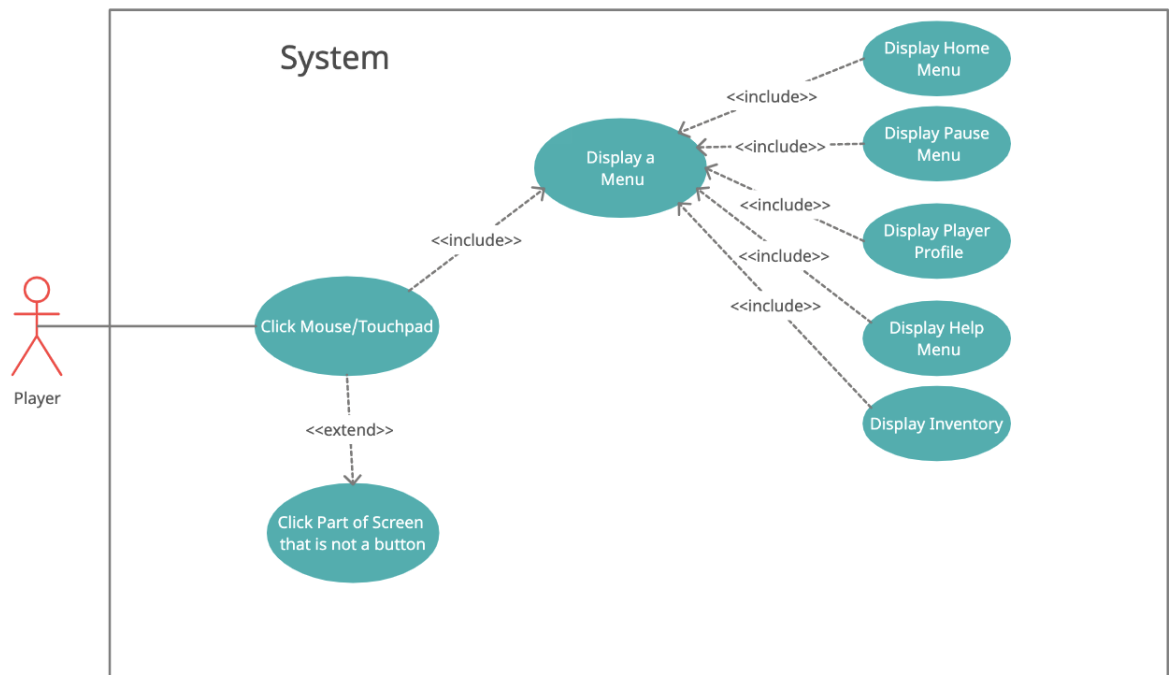
My feature for the game “Exit the Hive” is the user interface and the weapon inventory.

For the user interface, my job is to make sure the user can interact with the game through the keyboard and the mouse or touchpad. For this to work, I need to have a few accepted conditions to what keys the user can press, and certain buttons the user can click to play the game and access menus.

For the inventory, my job is to keep track of weapons the user picks up to use against enemies and bosses. To do this, I need to store the weapons, and make sure the player has access to view the inventory and is able to request the weapons through UI.

1. Use case diagram with scenario __14

User Interface Use Case Diagram:



Scenario:

Name: User Interface

Summary: The player uses the touchpad to interact with the game.

Actors: Player

Preconditions: The game is programmed to accept input from the mouse/touchpad.

Basic sequence:

Step 1: Accept input of device (mouse/touchpad).

Step 2: Process input and display menu (if menu button is clicked)

Exceptions:

Step 1: The mouse/touchpad clicks on something other than a button: ignore input (do nothing)

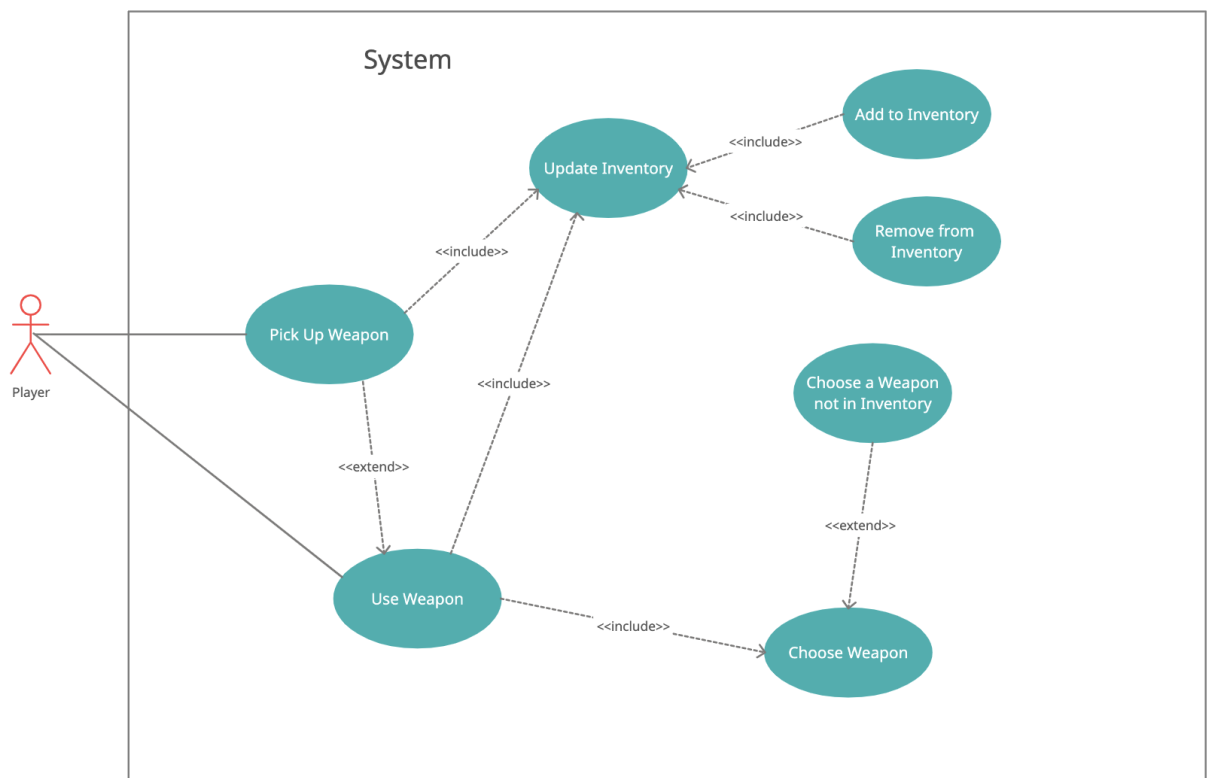
Post conditions: Correct menu is displayed.

Priority: 1*

ID: NN1

*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Inventory Use Case Diagram:



Scenario:

Name: Inventory

Summary: The player picks up and requests weapons from inventory.

Actors: Player

Preconditions: The inventory is initialized and empty at the beginning of the game

Basic sequence:

Step 1: Collect weapons

Step 2: Add weapons to inventory

Step 3: Request to use a weapon from inventory

Step 4: Use weapon

Step 5: Update Inventory to keep track of weapons that have been used

Exceptions:

Step 3: Request to use a weapon that is not in the inventory.

Post conditions: Chosen weapon is used and inventory is up to date.

Priority: 2*

ID: NN2

*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

1. Data Flow diagram(s) from Level 0 to process description for your feature ____14

CONTEXT DIAGRAM (level 0)

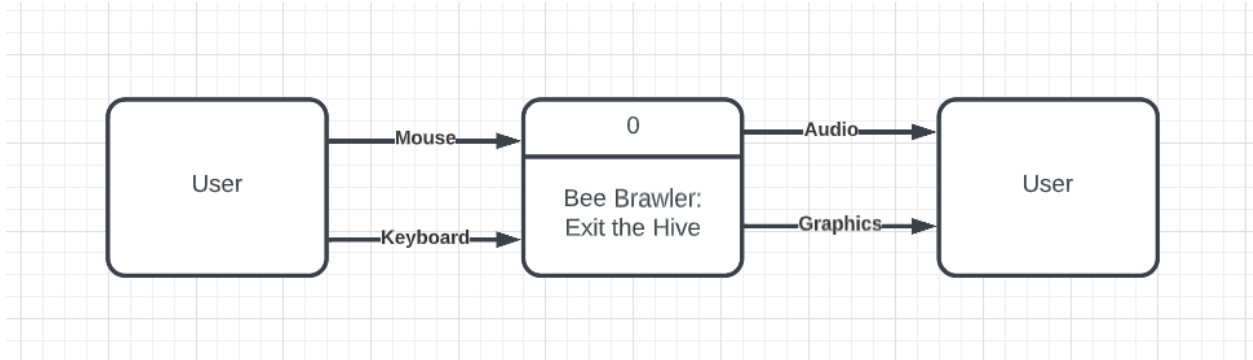
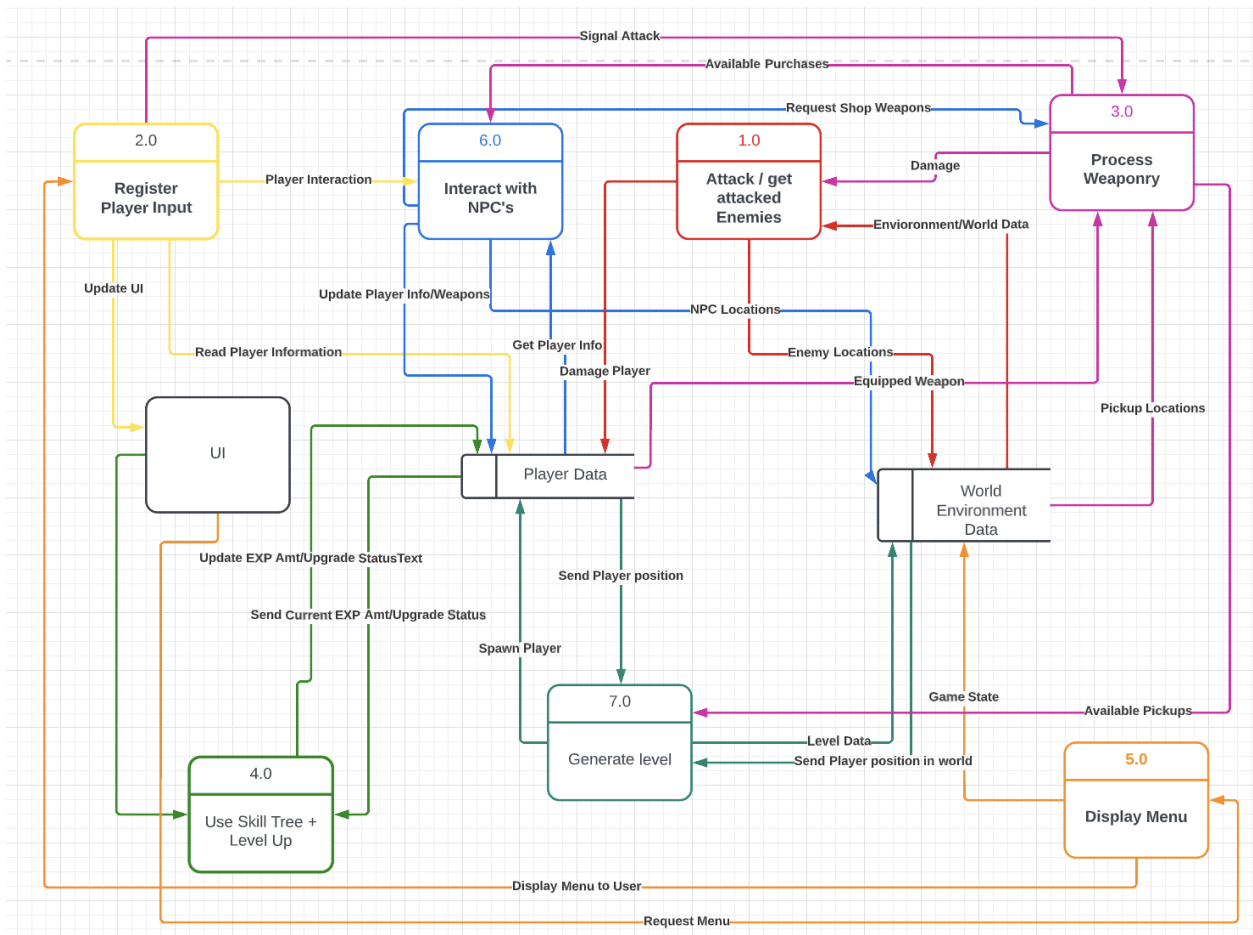
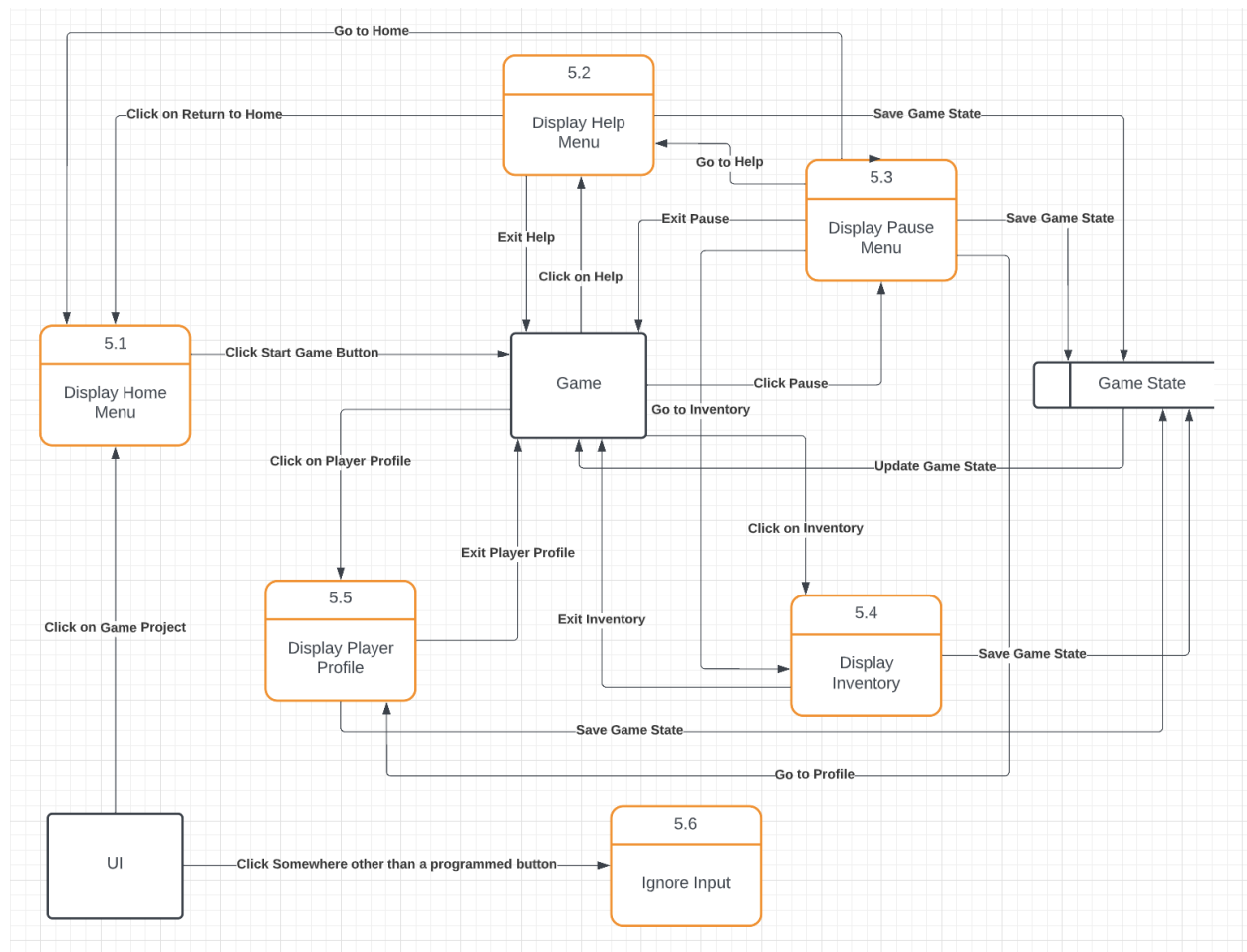


DIAGRAM 0 (level 1)



USER INTERFACE DATA FLOW DIAGRAM: process 5.0 Display Menu



Process Descriptions ???

Display Pause Menu*:

IF help menu is clicked

Display help menu

ELSE IF home menu is clicked

Display home menu

ELSE IF player profile is clicked

Display player profile

ELSE IF inventory menu is clicked

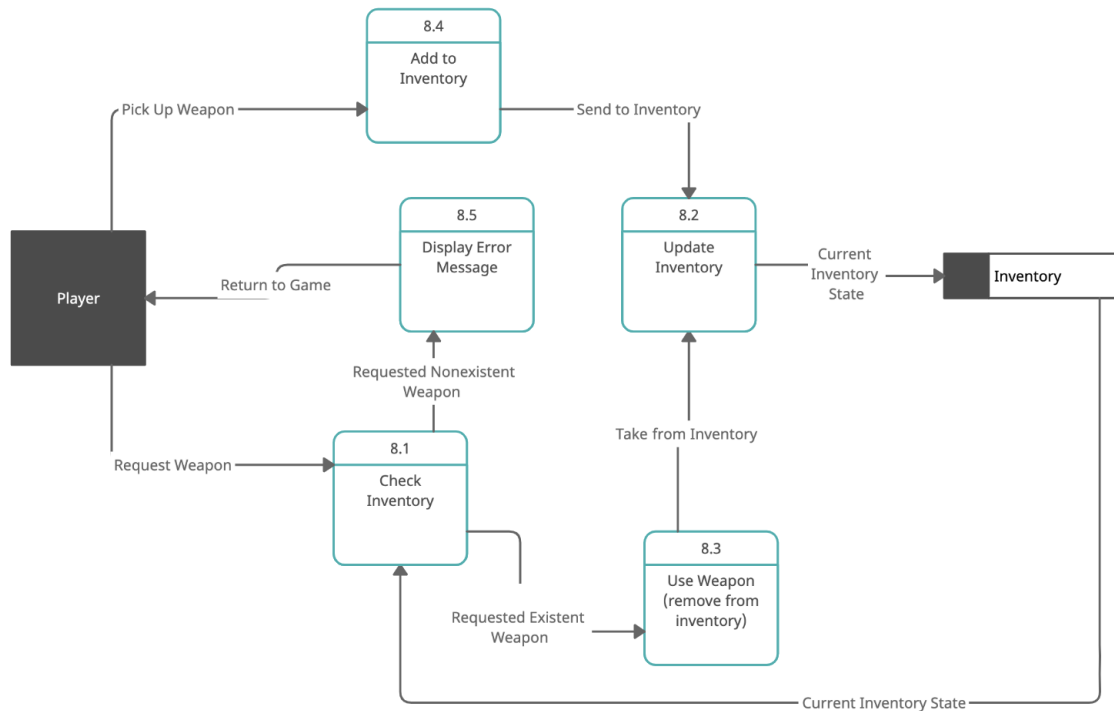
Display inventory menu

```
ELSE IF 'x' button is clicked
```

Exit pause menu and return to game

INVENTORY DATA FLOW DIAGRAM

Note: my team decided not to put this on the team data flow diagram because it falls under the data store Player data. Since it is still a feature I am in charge of, I made a data flow diagram for it to see the processes, but it does not flow from diagram 0.



Process Descriptions

Update Inventory*:
IF pick up weapon
 Add weapon to inventory
ELSE IF request to use weapon
 Remove from inventory
Endif

1. Acceptance Tests _____9

INVENTORY TESTING

For inventory, I will use a unit test to test how many weapons inventory can hold before displaying an error message, and if it correctly displays an error message when trying to use a weapon when inventory is empty. I will also use a stress test to test if inventory has the correct amount in it, and thus is responding correctly to user input.

Inventory Acceptance Test

Run test 20 times (20 = max number of weapons to hold). Each test adds one weapon to inventory as input, and repeats this 20 times until max inventory storage is met. The output of the first 20 tests should display a message telling the user that a weapon has been added to inventory. The test will conclude on the 21st test where it will display an error message telling the user that inventory is full.

Example output

Test #	input	output	success	fail
1	Add to inventory (pick up a weapon)	Success message “ weapon added to inventory”	y	n
2	Add to inventory	Success message “ weapon added to inventory”	y	n
3	Add to inventory	Success message “ weapon added to inventory”	y	n
...
19	Add to inventory	Success message “ weapon added to inventory”	y	n
20	Add to inventory	Success message “ weapon added to inventory”	y	n
21	Add to inventory	Error message “inventory is full, remove a weapon before adding another”	n	y

Similarly, run another test 3 times. The first test will add a weapon to inventory as input, and display a successful message as output. The second test will remove the weapon from inventory as input, and display a weapon graphic as output. The third test will remove a weapon from inventory as input, and should display an error message telling the user that inventory is empty.

Example output

Test #	input	output	success	fail
1	Add to inventory (pick up a weapon)	Success message “ weapon added to inventory”	y	n
2	Remove from inventory	Weapon object appears on screen	y	n
3	Remove from inventory	Error message “inventory is empty”	n	y

Inventory Stress Test

Stress test: how many inputs can inventory respond to simultaneously until it doesn't respond

For this stress test, start with inventory half full, so with ten weapons. Simultaneously add and remove (using two different threads) 1 weapon on the first test, 2 weapons on the second test, and so on. If inventory is not half full after the test, then something was not added or removed and the test number will reveal the level of responsiveness.

Example output

Test #	input	output	success	fail
1	Add 1 weapon to inventory, remove 1 weapon from inventory	10 weapons in inventory	y	n
2	Add 2 weapons to inventory, remove 2 weapons from inventory	10 weapons in inventory	y	n
...
9	Add 9 weapons to inventory, remove 9 weapons from inventory	10 weapons in inventory	y	n
10	Add 10 weapons to inventory, remove 10 weapons from inventory	Error/no response	n	y
11	Add 11 weapons to inventory, remove 11 weapons from inventory	Error message; trying to add more than what is accepted and trying to remove more than what is available	n	y

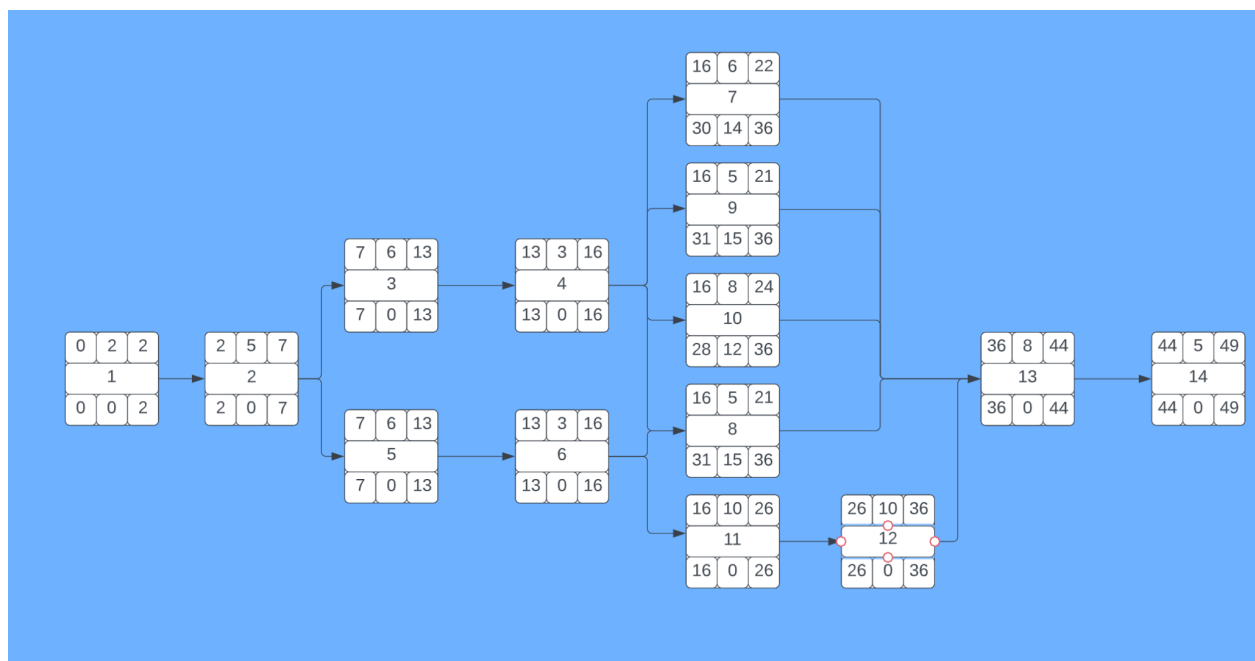
1. Timeline ____/10

Work items

Task	Duration (Hours)	Predecessor Task(s)
1. Setting up Unity	2	-
2. Requirements Collection	5	1
3. UI Framework	6	2

4. UI Testing	3	3
5. Inventory Framework	6	2
6. Inventory Testing	3	5
7. Home Menu design	6	4
8. Pause Menu design	5	4,6
9. Help Menu design	5	4
10. Player Profile design	8	4
11. Inventory Design	10	6
12. Inventory Menu design	10	11
13. Add in Graphics	8	7,8,9,10,12
14. Final Testing	5	13

Pert diagram



Gantt timeline

