Name: Sasha Bearman-Drummond Mark _____/50

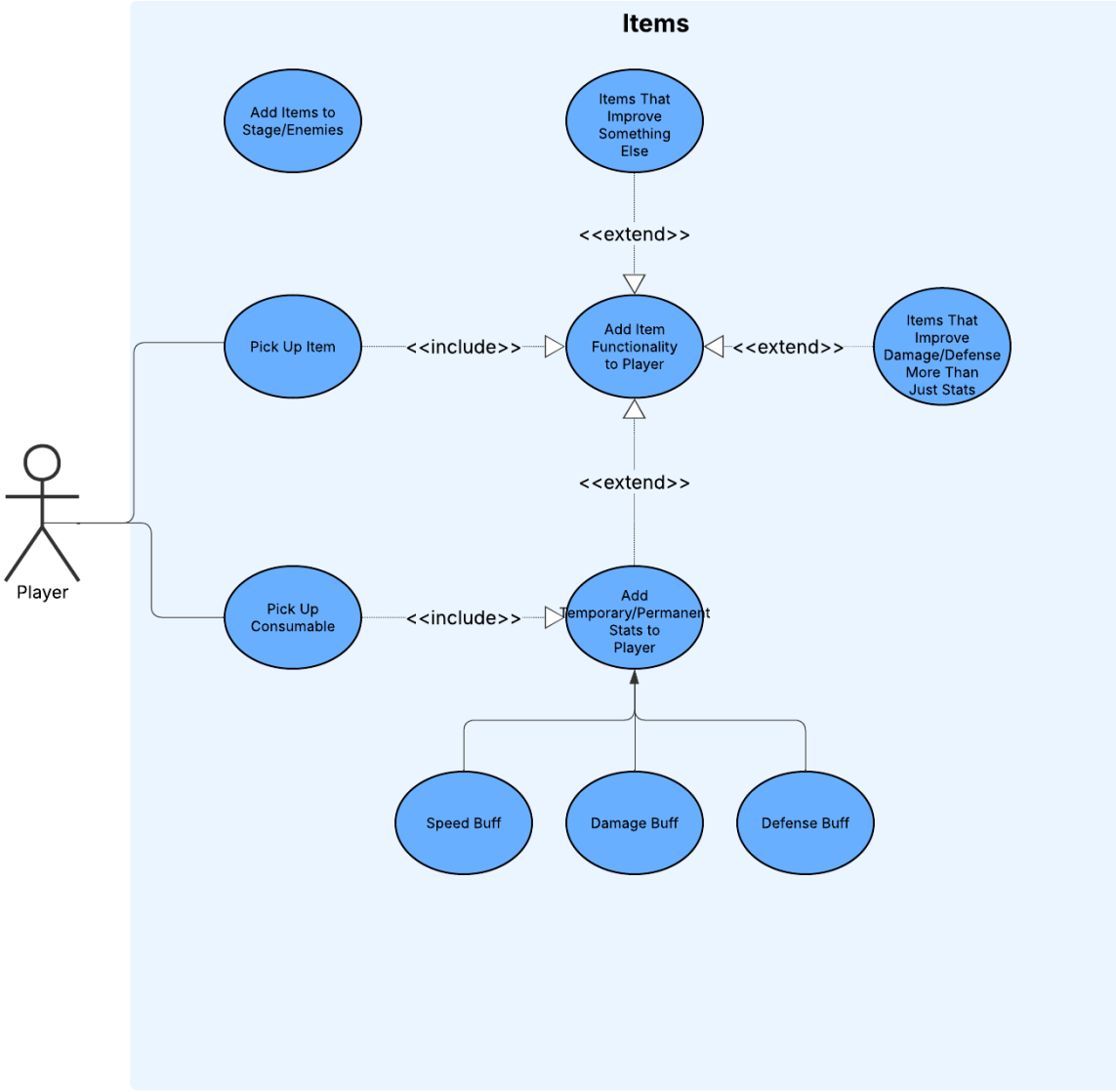
[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

1. Brief introduction __/3

I'm going to be doing the items and consumables for our game. I'm planning on adding various items that buff the player permanently and some that will provide temporary buffs.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

[You will need a scenario for each use case]

Name: Pick up item

Summary: The player picks up an item.

Actors: Player

Preconditions: Item has spawned

Basic sequence:

Step 1: Figure out which item has been picked up

Step 2: Add stats to player.

Exceptions:

N/A

Post conditions: Player has added stats.

Priority: 2*

ID: C01

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

Name: Pick up consumable

Summary: The player picks up a consumable.

Actors: Player

Preconditions: Consumable has spawned

Basic sequence:

Step 1: Figure out which consumable has been picked up

Step 2: Add stats to player for a set duration.

Exceptions:

1: Added stats exceed duration.

Post conditions: Player has added stats for a set duration.

Priority: 2*

ID: C01

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

Name: Spawn items

Summary: Items spawn on the map.

Actors: N/A

Preconditions: Stage is generated.

Basic sequence:

Step 1: Figure out which items should be spawned

Step 2: Figure out where they are spawned

Step 3: Spawn the items

Exceptions:

N/A

Post conditions: Player has added stats.

Priority: 2*

ID: C01

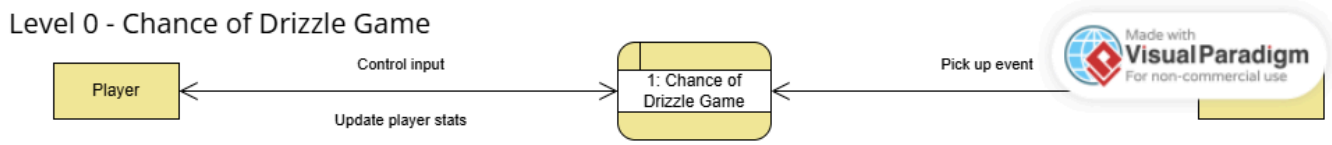
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

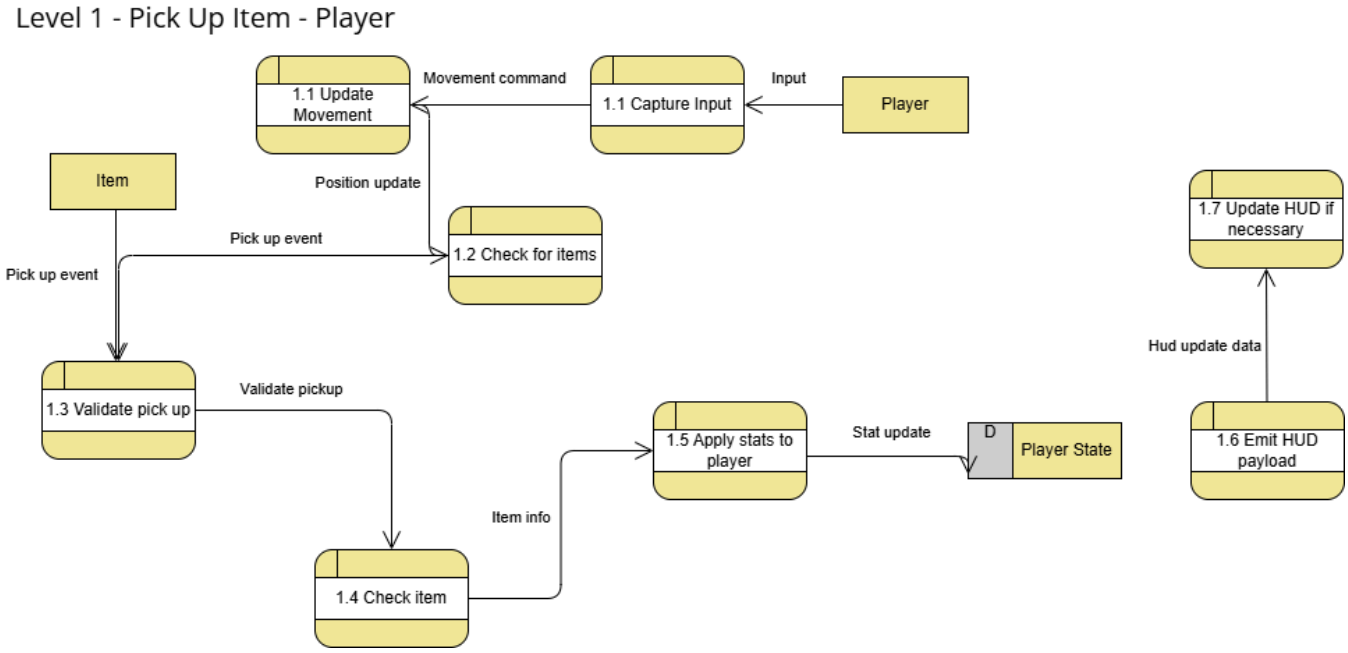
3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

Data Flow Diagrams





Process Descriptions

Validate pickup:

IF item collision detects player AND player collision detects item, validate the pickup and go to check item

Check item:

IF item is an item AND item can be applied, apply the item to the player and update any player stats or abilities accordingly

Apply stats to player:

IF item was checked to be valid apply item buffs

Update movement:

IF input was captured AND it's a valid move (ie not into a wall), then move player model and camera accordingly

4. Acceptance Tests _____9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Run feature 100 times with different item inputs sending output (buffs to player) to a file if possible

The output file will have the following characteristics:

-Test with items that don't exist, shouldn't add anything to player

-Test all implemented items with successful results in the output file

-Test exceeding the limit of stat buffs. Etc

-No results that don't align with feature intentions

Example for item feature

Output	GetItem(item)	Notes
speed += 1	"speedBoots"	
speed += 1 for 1 minute	"speedBerry"	Only add buffs for 1 minute
null	"notanitem"	Discard all input that isn't a valid item

if(speed + 1 > 100) return null	“speedBoots”	If we have some cap on stats, make sure player doesn’t exceed those caps

5. Timeline _____/10

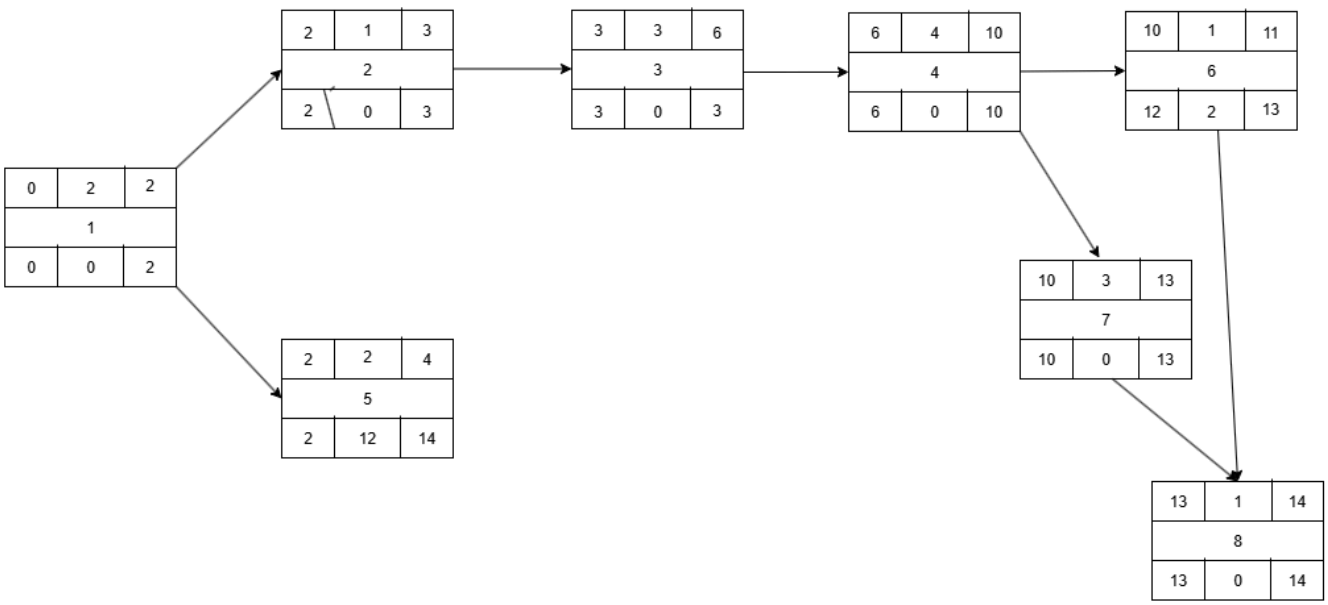
[Figure out the tasks required to complete your feature]

Example:

Work items

Task	Duration (Weeks)	Predecessor-Tasks
1. Item Creation and Design	2	-
2. Item Collision Programming	1	1
3. Item Pickup Programming	3	2
4. Item Stats/Upgrades Programming	4	1, 2, 3
5. Item Art	2	1
6. Documentation	1	2, 3, 4
7. Testing	3	1, 2, 3, 4
8. Installation	1	6, 7

Pert diagram



Gantt timeline

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1															
2			1												
3				2											
4							3								
5			1												
6											4				
7											4				
8															