Name\_\_\_Torry Chmelik \_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

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

## Brief introduction \_\_/3

[Describe your feature briefly]

hud + player health + money count + ability cooldowns

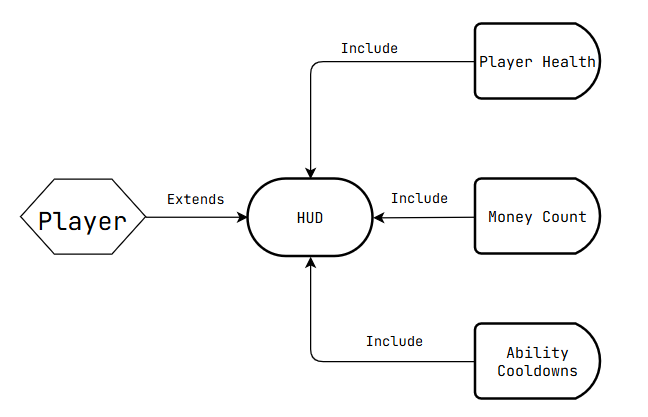
The Player hud and player health will be used to allow the player to see and operate their character while showing them information about their character including what they have equipped and when their abilities come off cooldown, The ability cool downs themselves are a timer that keeps track of when the player can use an ability again after previous use. And the money count will keep track of your economy and the amount of currency you have on your character to spend at the shop.

## Use case diagram with scenario \_\_14

[Use the lecture notes in class.

Ensure you have at least one exception case, and that the <<extend>> matches up with the Exceptions in your scenario, and the Exception step matches your Basic Sequence step.

Also include an <<include>> that is a suitable candidate for dynamic binding]



### Scenarios

**[You will need a scenario for each use case]**

**Name:** HUD

**Summary:** The Player uses the HUD to evaluate information about their Player.

**Actors:** Player

**Preconditions:** Hud Has been set to a base state

**Basic sequence:**

**Step 1:** Hud Takes information about Health, Ability Timers, and Money Count

**Step 2:** When one of these changes, calculate the change

**Step 3:** Update the hud display with the new current values

**Exceptions:**

**Step 1:** a negative value is imputed when the value is at zero : ignore the negative value and stay at zero.

**Step 2:** multiple values try to update the same tracker at once: use a stack to properly organize priority in updating values

**Post conditions:** Calculated value is displayed.

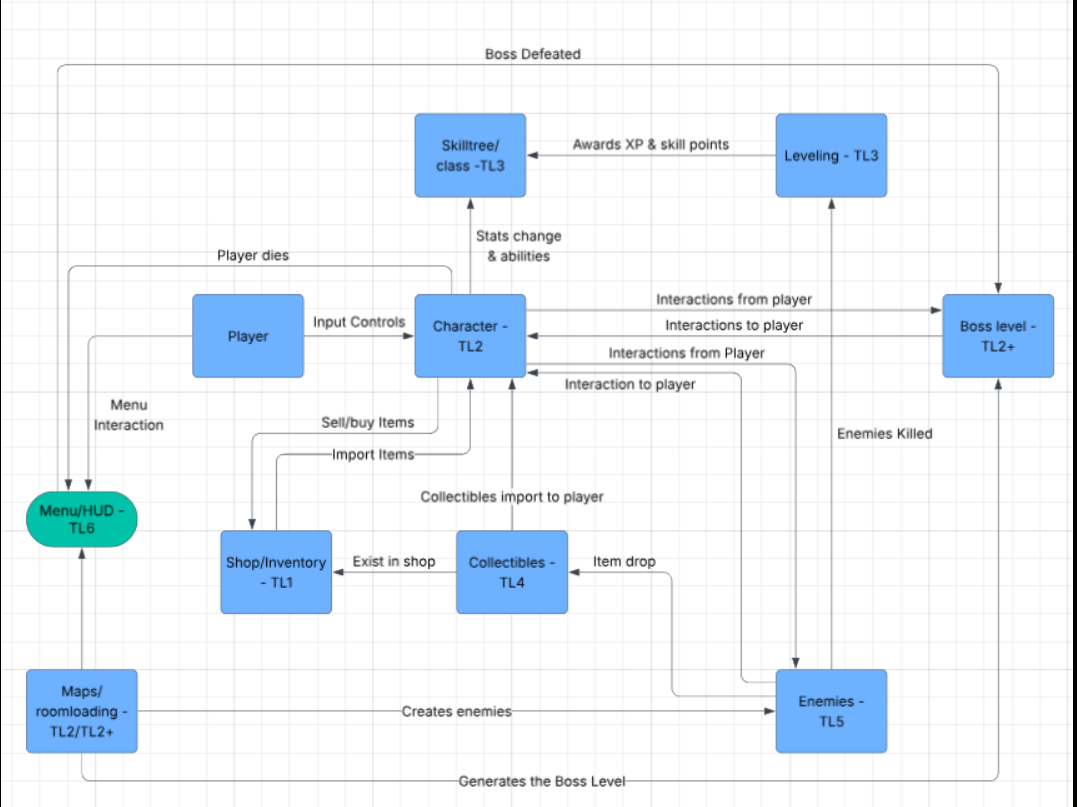
**Priority:** 2\*

**ID:** TC1

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

## 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

A diagram of a software development process

AI-generated content may be incorrect.

### Process Descriptions

If player health <= 0 then enable death hud. While player health > 0 update values for money, health, ability cooldowns, and keep track of state of player. The state of the player should be classified as dead or alive and possibly keep track of status effects that relate to the player. If Ability has been activated the HUD should indicate that the ability is no longer usable until the timer expires. If the player collect money the hud should indicate so and give the ability to check money and inventory. Whenever health is affected the HUD should provide a health bar that indicates the level of health of the player. The HUD should also provide a way to escape the current level and go back to the menu of the game.

## Acceptance Tests \_\_\_\_\_\_\_\_9

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

**Run the HUD and try to update all different values at least 50 times.**

* **Check for negative values**
* **Check for values over the limit**
* **Max Health: 100**
* **Ability Cooldowns: TBT**
* **Max Money Count: 100**

**Example for HUD updates**

|  |  |  |  |
| --- | --- | --- | --- |
| Starting Value | Input | Output | Notes |
| Health: 100 | -10 | 90 | If health is affected, we will update the value |
| Health: 30 | -40 | 0 | If health if affected and willgo below 0, set value to zero and trigger death |
| Money: 50 | 60 | 100 | If money is added and would go over 100, stop at 100 and indicate that inventory is full |
| Money: 25 | -30 | 25 | If money is subtracted by buying something at the shop, if the player cant afford the item, don’t take the money and keep at current value |
| Cooldown:  10 | -10 | 0-10 | If an ability is used, set the cooldown timer back to 10 seconds and begin counting down again |

## Timeline \_\_\_\_\_\_\_\_\_/10

[Figure out the tasks required to complete your feature]

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 1 | - |
| 2. HUD Design | 1 | 1 |
| 3. Report Design | 1 | 1 |
| 4. Value Memory Storage | 1 | 2, 3 |
| 5. HUD Implementation | 1 | 2 |
| 6. Programming | 1 | 3,4 |
| 7. Testing | 1 | 6 |
| 8. Installation | 1 | 7 |

### Pert diagram

### 

### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  | 2,3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  | 3,4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  | 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |