Name: Julia Abdel-Monem Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

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

## Brief introduction \_\_/3

The user interface is a set of menus for the game and a set of controls for both Mobile and Keyboard/Mouse. This includes a main menu, a pause menu, a settings menu, a player HUD, an End Screen, a Shop HUD, a character creation menu, and a Mobile Control HUD.

The main menu and pause menu should let the user start/restart the game (and resume the game in the pause menu), go to the settings screen, and quit the game.

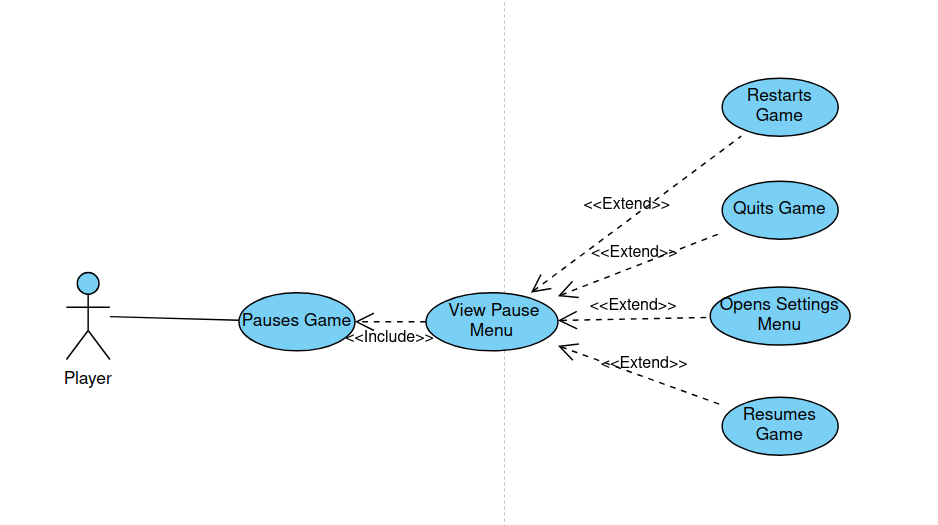
The settings menu should have audio controls, control remapping, and various accessibility settings (color-blind settings, font-accessibility, motion limiting, etc...).

The Player HUD should display various in-game warnings, the player's health and score, and may also shake to add to the tension of the game.

The end screen should let the user restart or quit and should display the player's score. The shop HUD should let the user purchase in-game items using an in-game currency and may change depending on the character class.

The Mobile control HUD should have touch screen controls and may modify the player HUD to optimize for smaller screens.

## Use case diagram with scenario \_\_14



**Name:** Display Pause Menu

**Summary:** The User has opened the Pause Menu

**Actors:** User

**Preconditions:** The game is currently in play

**Basic sequence:**

**Step 1:** The game is paused and any in-game actions are halted

**Step 2:** The Pause menu is displayed

**Step 3:** The user selects one of four options

**Exceptions:**

**A:** The game is resumed

**B**: The game is restarted

**C:** The game is quit and the main menu is shown.

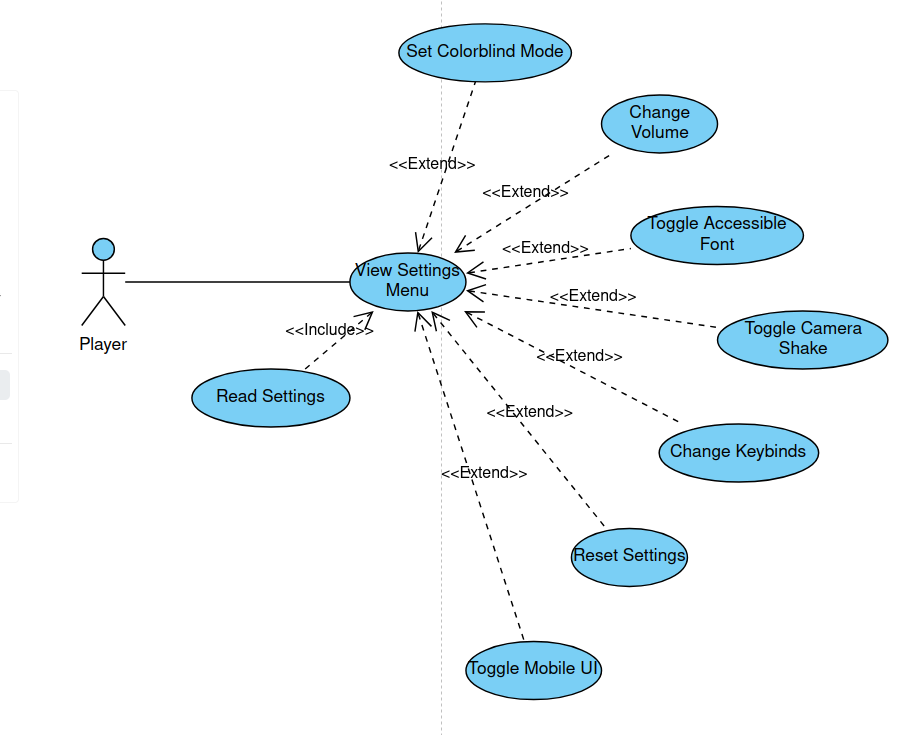
**D:** The settings menu is selected and is then displayed.

**Post conditions:** Do what the user requests (settings, quit game, restart game, or resume game)

**Priority:** 1

**ID:** C01

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



**Name:** Display Settings Menu

**Summary:** The User has opened the Settings Menu

**Actors:** User

**Preconditions:** The Game is either paused or not in play

**Basic sequence:**

**Step 1:** The player accesses the settings menu through the pause menu or main menu

**Step 2:** The current settings are read from memory  
**Step 3:** The Setting menu is displayed

**Step 4:** The user makes possible changes as desired

**Exceptions:**

**A:** Color Blind Mode is Enabled/Disabled

**B**: The Volume settings are changed

**C:** The use of an accessible font is toggled

**D:** The use of in-game camera shake is toggled

**E:** The keybinds are changed

**F:** The settings are reset to their default options

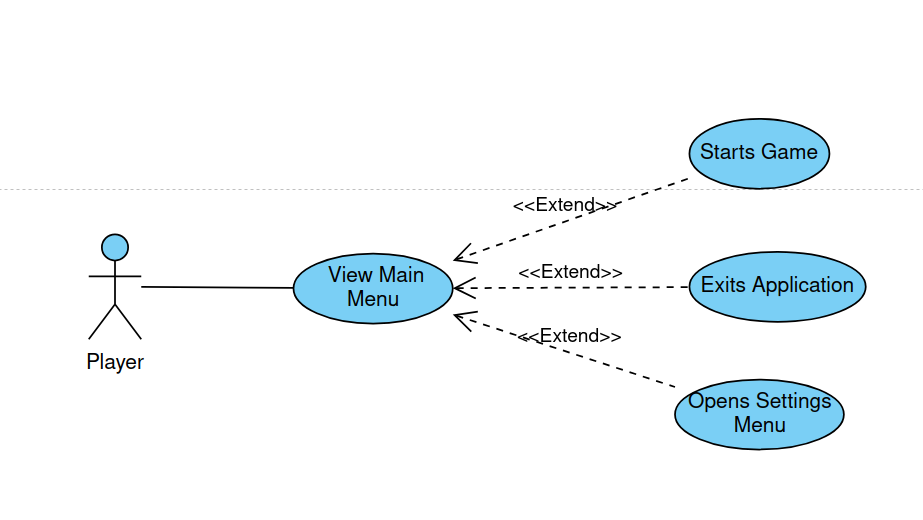
**G:** The Mobile UI (controls) are toggled.

**Post conditions:** Save settings to memory

**Priority:** 2.5 (some options are less essential than others)

**ID:** C02

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



**Name:** Display Main Menu

**Summary:** The User has opened the Main Menu

**Actors:** User

**Preconditions:** The game is not in play

**Basic sequence:**

**Step 1:** The main menu is displayed

**Step 3:** The user selects one of three options

**Exceptions:**

**A:** The game is started

**B**: The settings menu is opened

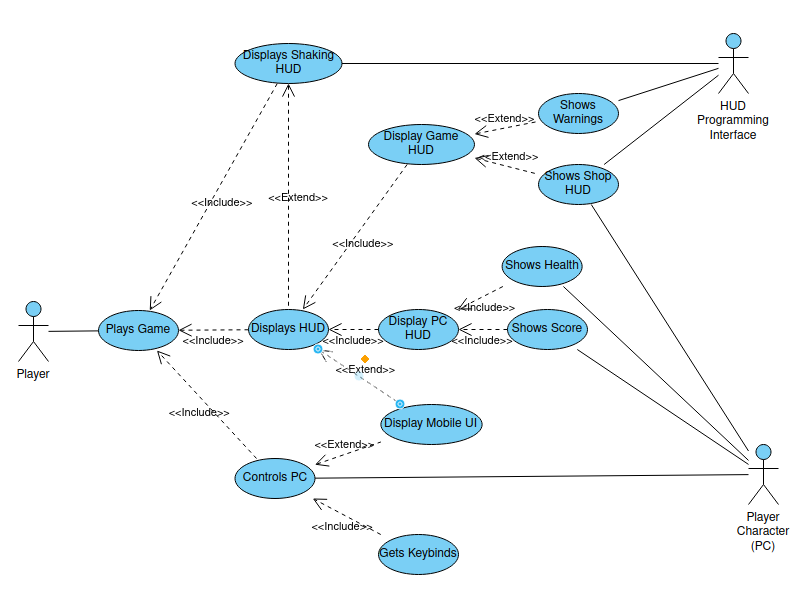
**C:** The application is closed

**Post conditions:**

**Priority:** 1

**ID:** C03

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



**Name:** User Plays Game

**Summary:** The Game is being played

**Actors:** User, HUD programming interface, Player Character

**Preconditions:** The game is currently in play

**Basic sequence:**

**Step 1:** The game is loaded

(Each Frame, Repeat steps 2-

Display the Game HUD

Get the health and score from the player character

Display the Player Character HUD

Display the HUD

Read Keymap from memory

Get User Input to control the PC

**Exceptions:**

**A:** If the mobile UI is toggled, Display that, and use that input to control the PC

**B**: If the HUD API requests the HUD to be shaken, shake the HUD

**C:** If the HUD API requests a warning to be shown, show a warning

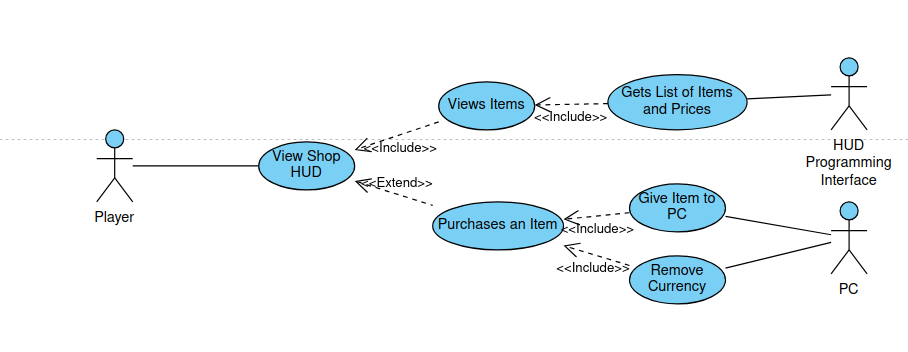
**D:** If the HUD API requests a shop to be shown, show a shop HUD

**Post conditions:**

**Priority:** 1.5 (some elements are less important, such as the shaking HUD)

**ID:** C04

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



**Name:** A shop HUD is being displayed

**Summary:** The Game is being played

**Actors:** User, HUD programming interface, Player Character

**Preconditions:** The game is currently in play

**Basic sequence:**

**Step 1:** The HUD API obtains a list of items and prices to display

**Step 2:** The HUD is displayed

**Step 3:** The HUD displays the list of items

**Exceptions:**

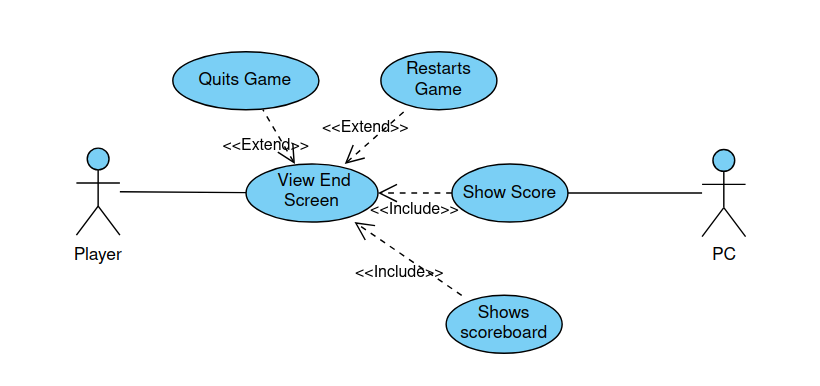
**A:** If the User purchases an item, the item is then given to the PC and currency is removed in exchange

**Post conditions:**

**Priority:** 2 (some elements are less important, such as the shaking HUD)

**ID:** C05

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



**Name:** The End Screen is shown

**Summary:** The end screen shows the score, and lets the user quit or restart

**Actors:** User, Player Character

**Preconditions:** The game has ended

**Basic sequence:**

**Step 1:** The final score is obtained from the PC

**Step 2:** The screen is displayed

**Exceptions:**

**A:** The user quits the game

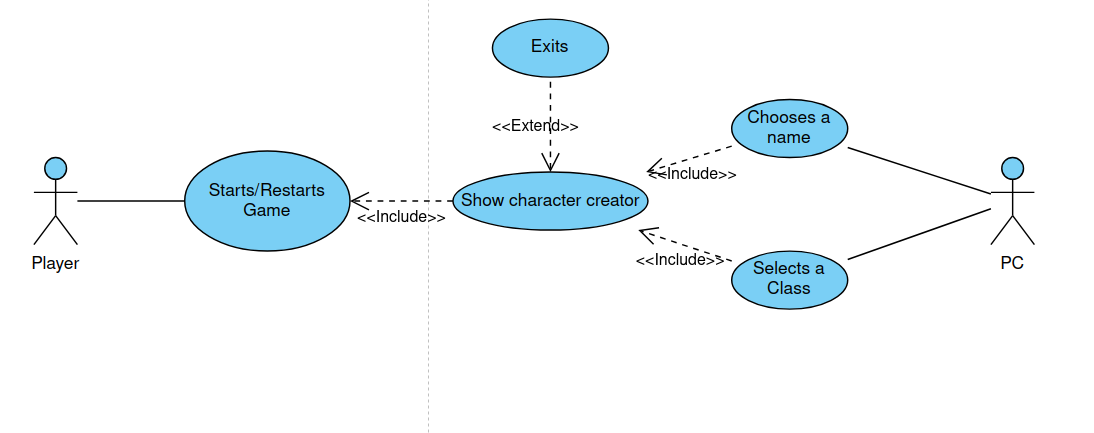
**B:** The user restarts the game

**Post conditions:** Do what the user wants

**Priority:** 1

**ID:** C06

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



**Name:** The Game is started or restarted

**Summary:** The player needs to create a character before playing the game

**Actors:** User, Player Character

**Preconditions:** The game is about to begin

**Basic sequence:**

**Step 1:** The player chooses to start a new game

**Step 2:** The character creator is shown

**Exceptions:**

**A:** The player exits and returns to the main menu

**Post conditions:** either create a PC with the name and class or return to the main menu

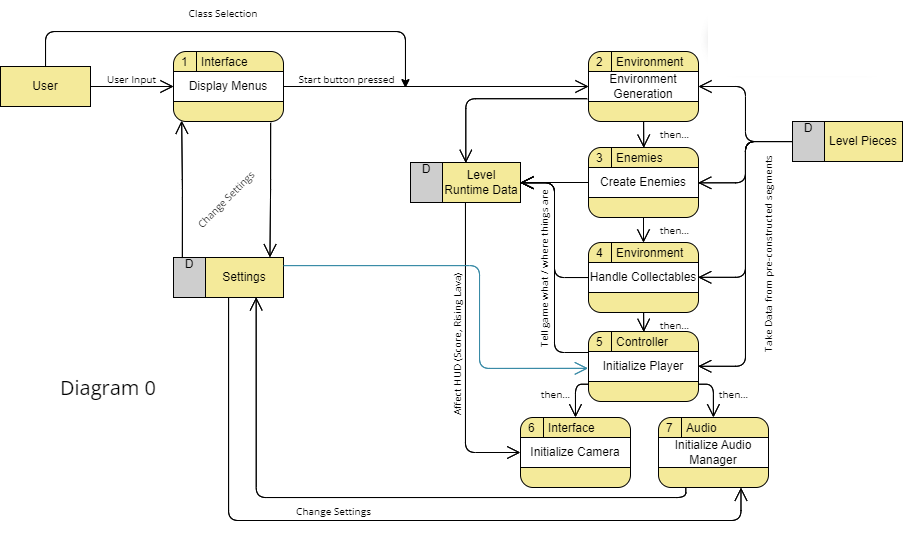
**Priority:** 2

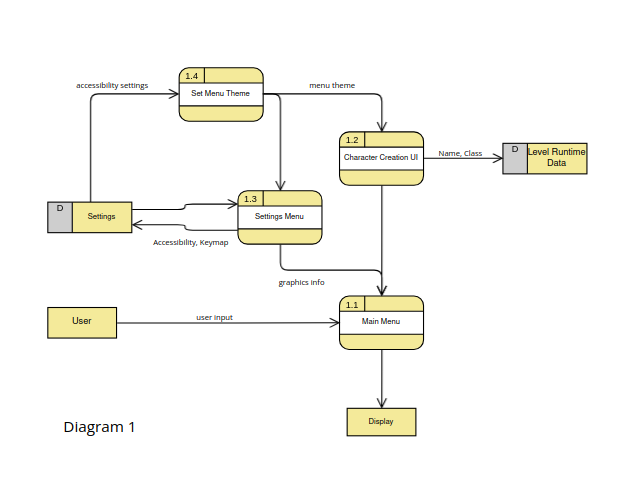
**ID:** C07

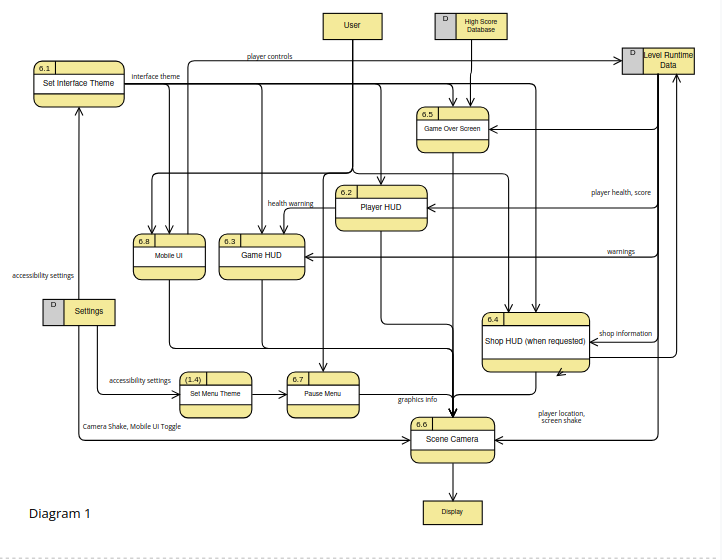
\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

### Data Flow Diagrams







### Process Descriptions

Main Menu:

WHILE option not selected:  
 Display Quit Button

Display Settings Button

Display Start Button

IF settings pressed:

Display Settings

ELIF start pressed:

Display Character Creation UI

ELIF quit pressed

Exit Application

END

END

END  
Character Creation UI:

WHILE name not entered OR class not selected:

Display name entry element and class selector element

END

END

Settings Menu:

WHILE back button not pressed:

Display various settings set to their current value

END

END

Set Interface/Menu Theme:

Read accessibility settings

Configure interface/menu theme to input

END

Player HUD:

WHILE playing

Get player health and score

display player health and score

END

END

Game HUD:

WHILE playing

IF warning received: display warning

END

END

Shop HUD:

WHILE current shop != null

Get current shop information

Display current shop information

IF item clicked:

Give Item to player

Remove Item from shop

END

END

END  
Mobile Controls:

Display Player Controls

WHILE button pressed:

send action to player

END

END

Scene Camera

WHILE playing

IF game paused:

Display Pause menu

Else:

Set position to current player position

Display Shop HUD IF current shop != null

Display Game HUD

Display Player HUD

IF screen shake requested

Shake HUD

END

IF Mobile UI Toggle

Display Mobile UI

END

END

END

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

To determine if the interface works as designed, test data for health and score will be fed into the game and visual confirmation on all platforms will be necessary.

Unit testing the various external calls for screen shaking can also be done to ensure that they work properly. visual confirmation is necessary.

To determine if the settings menu works as designed, each individual setter can be fed test data and then the game can be tested to ensure that each one works as designed. visual confirmation.

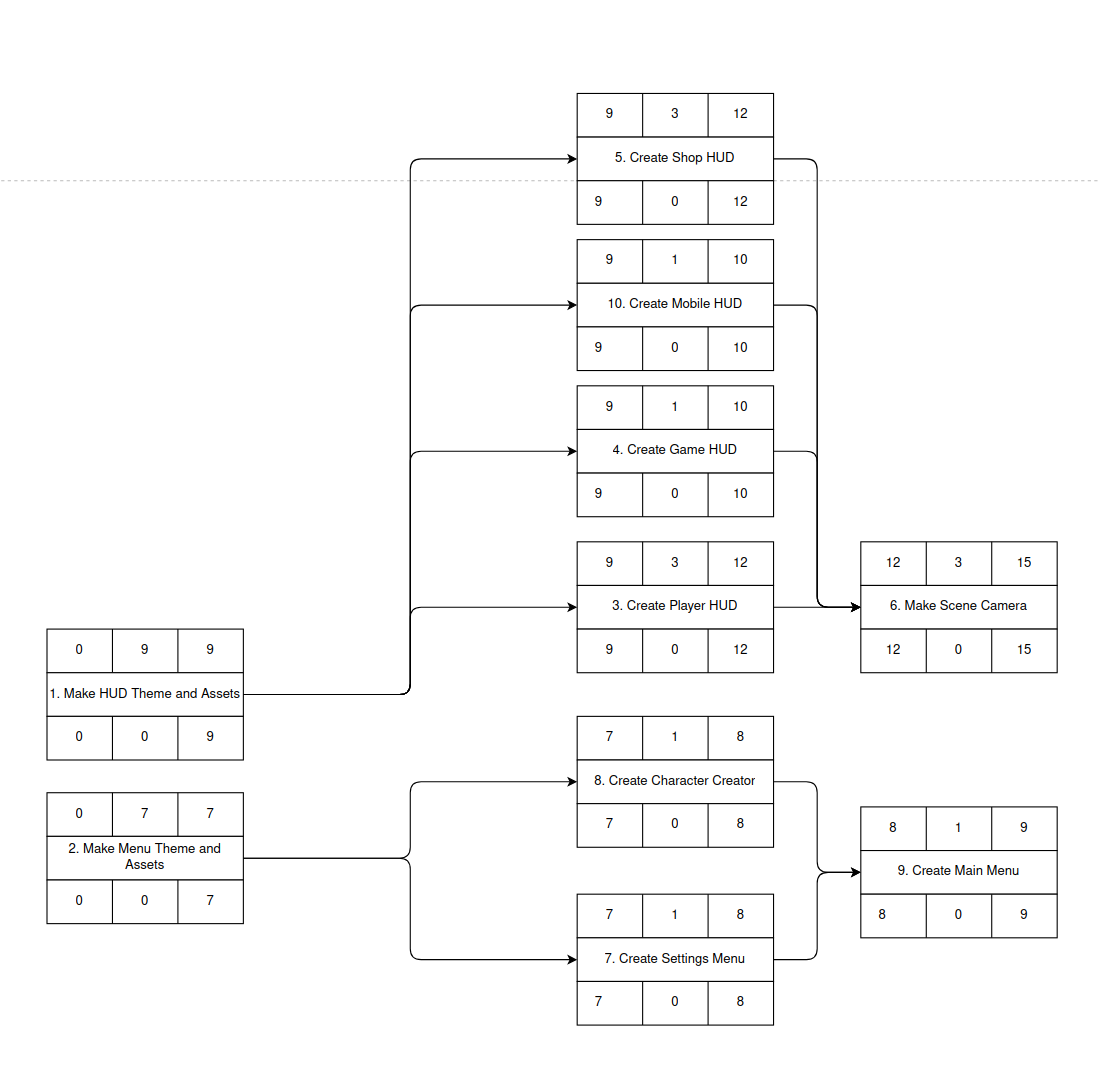
## Timeline \_\_\_\_\_\_\_\_\_/10

(create in this case means either find or make the required asset)

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours (out of 30)) | Predecessor Task(s) |
| 1. Make Interface/HUD Theme and Create Assets | 9 | 0 |
| 2. Make Menu Theme and Create Menu Elements (Toggles, Sliders, Keyboard Input, Text Buttons, Key Select) | 7 | 1 |
| 3. Create Player HUD (health) | 3 | 1 |
| 4. Create Game HUD (warnings) | 3 | 1 |
| 5. Create Shop HUD | 3 | 1 |
| 6. Create Scene Camera | 1 | 3, 4, 5, 10 |
| 7. Create Settings Menu | 1 | 2 |
| 8. Create Character Creator | 1 | 2 |
| 9. Create Main Menu | 1 | 2, 7, 8 |
| 10. Create Mobile HUD | 1 | 1 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  | 345 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  | 78 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 11 | 22 | 33 | 44 | 55 | 66 | 77 | 811 | 91 | 1111 | 2 2 | 33 | 1 | 2 | 3 |  |  |  |  |  |  |  |  |  |