Name\_\_Enoch Myers\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

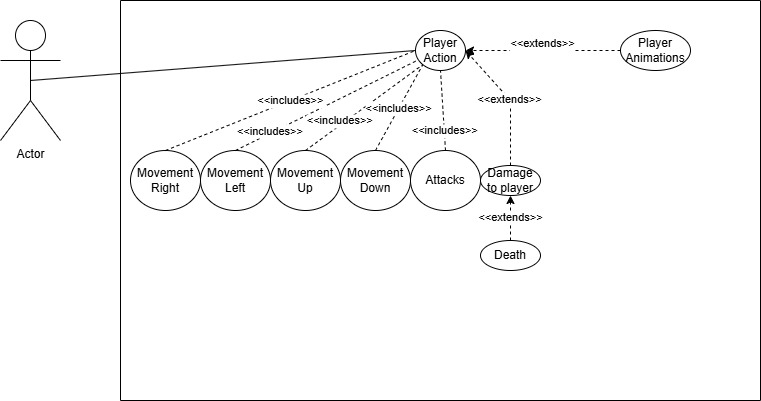
## Brief introduction \_\_/3

The first feature I’ve been tasked with implementing is the player's movement and attack mechanics in the platformer game. This includes, but is not limited to, handling player input for running, jumping, dashing, and any other traversal mechanics, ensuring smooth and responsive controls.

The second feature I’ll be responsible for is the player’s attack system. This involves designing and implementing attack animations, hit detection, damage interactions, and any combo mechanics. The goal is to make combat feel fluid, rewarding, and well-integrated with the movement system.

While my focus is on the player character, these mechanics will interact with other game systems, such as enemy AI and level design, to create a cohesive gameplay experience.

## Use case diagram with scenario \_\_14



### Scenarios

**Name:** Movement

**Summary:** The player moves up, down, left, right, or dashes

**Actors:** Player

**Preconditions:** The player isn’t in the menu and is in a level

**Basic sequence:**

**Step 1:** Player inputs movement

**Step 2:** Character makes that movement

**Exceptions:**

**Step 1:** The player has been hit by an enemy projectile and was knocked back.

**Post conditions:** The player’s movement was executed

**Priority:** 1

**ID:** PM1  
  
**Name:** Attacks

**Summary:** The player fires their weapon to attack.

**Actors:** Player

**Preconditions:** The player isn’t in the menu and is in a level.

**Basic sequence:**

**Step 1:** Player inputs an attack command.

**Step 2:** The character performs the attack animation.

**Step 3:** If an enemy or object is hit, it registers damage or an effect.

**Exceptions:**

**Step 1:** The attack is interrupted because the player is stunned.

**Post conditions:** An attack has been executed, and if it connects, it applies the intended effect (damage, knockback, destruction)

**Priority:** 1

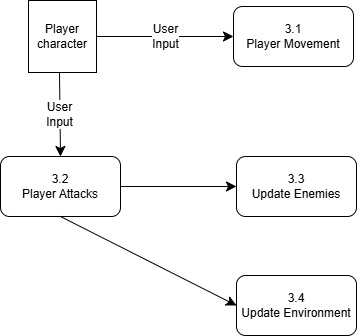
**ID:** ATK1

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

### Data Flow Diagrams

A diagram of a game

AI-generated content may be incorrect.



### Process Descriptions

Player Movement:

function playerMove(player, direction):

if player.isInLevel() and not player.isStunned():

player.move(direction)

player.updatePosition()

else:

ignoreInput()

Player Attacks

function playerAttack(player):

if player.canAttack():

player.playAttackAnimation()

if player.detectHit():

player.applyDamageToTarget()

else:

playInvalidAttackFeedback()

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

Test Case 1: Player Movement

Input: Directional Input (Arrow keys/WASD)

Expected Output: Player character moves in the corresponding direction

Test Case 2: Dashing

Input: Dash button press while moving

Expected Output: Player character moves forward quickly

Test Case 3: Movement Restriction

Input: Attempt to move while stunned

Expected Output: Player input is ignored

Test Case 4: Attack

Input: Attack button press

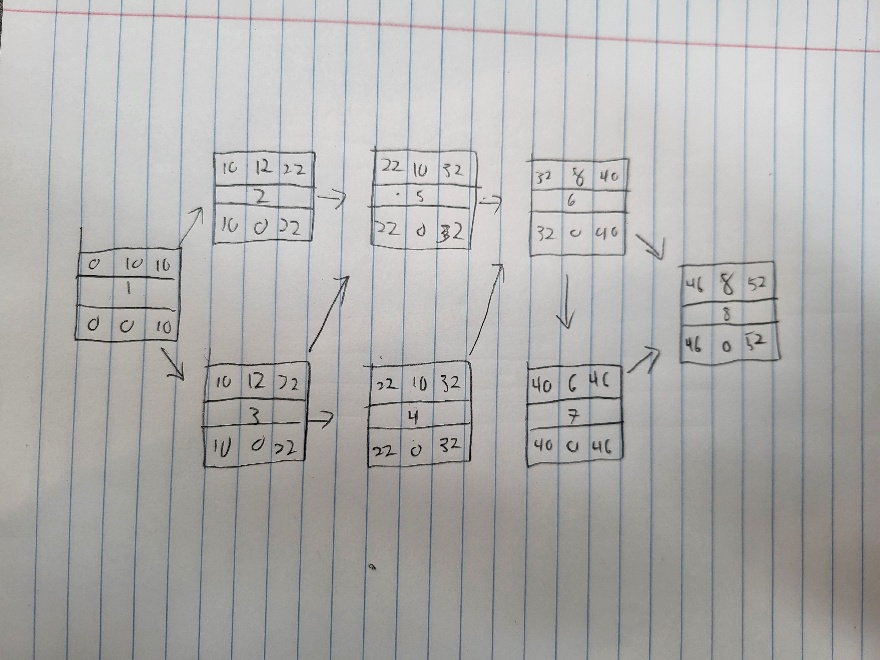
Expected Output: Player performs the attack animation and hitbox is active

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (hours) | Predecessor Task(s) |
| 1. Player State Machine | 10 | - |
| 2. Movement System | 12 | 1 |
| 3. Attack System | 12 | 1 |
| 4. Player sprites & animations | 10 | 3 |
| 5. Hit detection & collistion | 10 | 2,3 |
| 6. Programming | 12 | 4,5 |
| 7. Testing/Debugging | 8 | 6 |
| 8. Integration | 7 | 6, 7 |

### Pert diagram



### Gantt timeline

