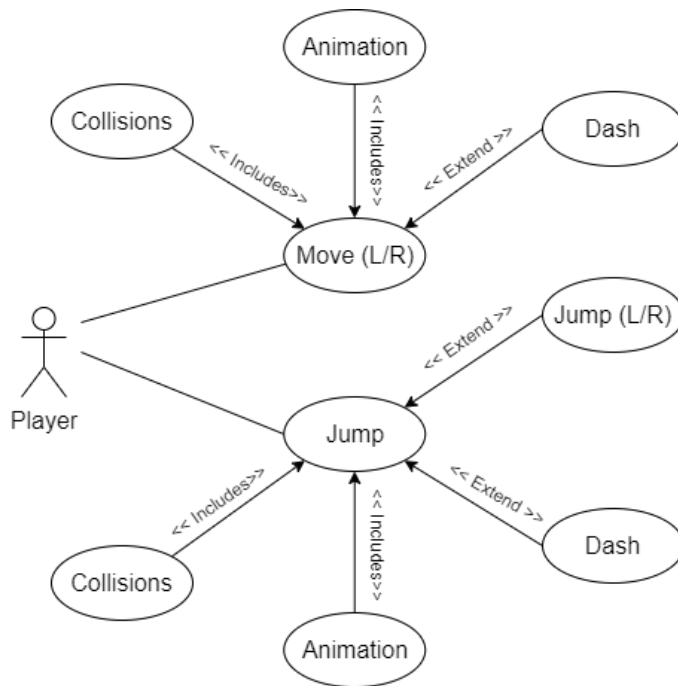# 1. Brief introduction __/3

My feature is player movement. This encapsulates moving the in-game character around.

# 2. Use case diagram with scenario  __14

### Use Case Diagram (Player Movement)



### Scenarios

**Name:** Move (L/R)

**Summary:** The player is able to move the in-game character left and right.

**Actors:** Player (of the game).

**Preconditions:** Player instantiated and level created.

**Basic sequence:**

> **Step 1:** Press A (move left) or press D (move right) on the keyboard
>
> **Step 2:** Pressing Shift and and moving in any direction will dash the player
>
> **Step 3:** Animations will play based on the action taken on move
>
> **Step 4**: Collisions or rather colliders determine what the player is doing (i.e. if they walk off a ledge they fall off).
>
> **Step 5:** In-game character is moved

**Exceptions:**

> **Step 1:** The in-game character will not move if they collide with something (e.g. wall).
>
> **Step 2:**  The player cannot move the character if they die.

**Post conditions:** In-game character changes position or moves.

**Priority:** 1

**ID:** M01

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

**Name:** Jump

**Summary:** The player is able to make the in-game character jump

**Actors:** Player (of the game).

**Preconditions:** Player instantiated and level created.

**Basic sequence:**

> **Step 1:** Pressing spacebar vertically ascends the in-game character.
>
> **Step 2:** The player will jump a different distance depending on how long they hold the spacebar (holding it longer will allow the player to jump higher up to a max height).
>
> **Step 3:** Moving the character left or right while jumping will allow the player to jump left/right (A and D) and anywhere in between (in the upwards direction using a combination of WASD).
>
> **Step 4:** Adding to the previous steps, if the player presses Shift, they will dash in the direction of their choice.
>
> **Step 5:** Animations will play based on the action taken on jump/move/dash.
>
> **Step 6**: Collisions or rather colliders determine what the player is doing (i.e. if they jump off a ledge they fall down towards the bottom of the level).
>
> **Step 7:** In-game character is initially moved upwards and depending on their action they may move in any direction using the move use case or the extension dash.

**Exceptions:**

> **Step 1:** The in-game character will not jump if they collide with something (e.g. a platform).
>
> **Step 2:** The player cannot jump with the character if they die.

**Post conditions:** In-game character changes position or moves (upwards initially and any direction afterwards).
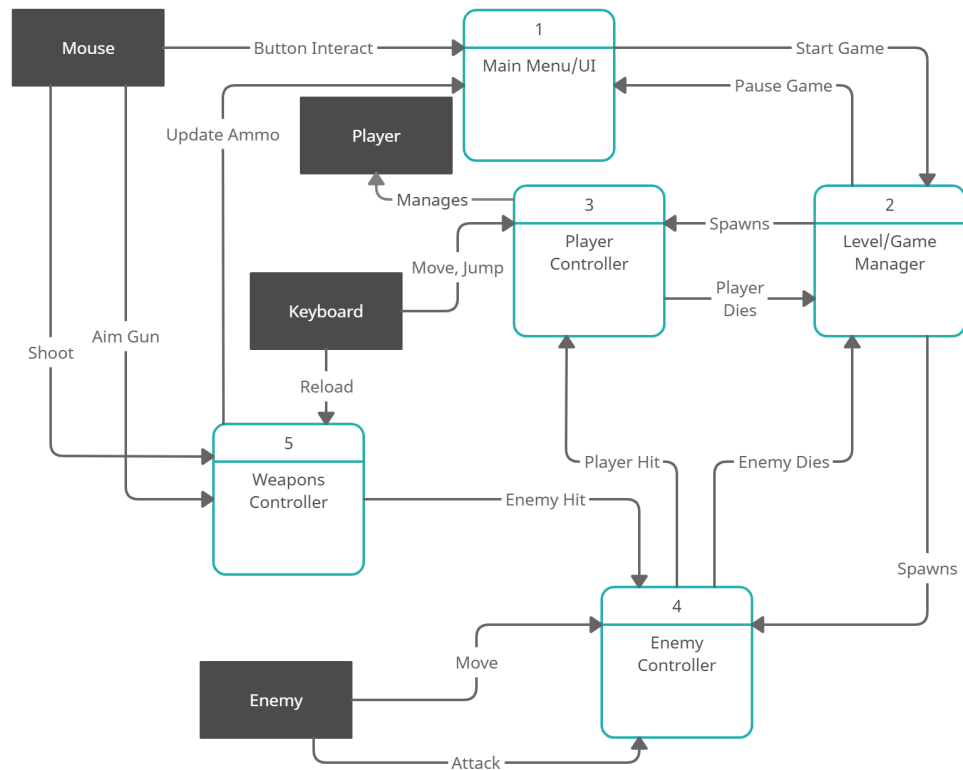
**Priority:** 2

**ID:** M02

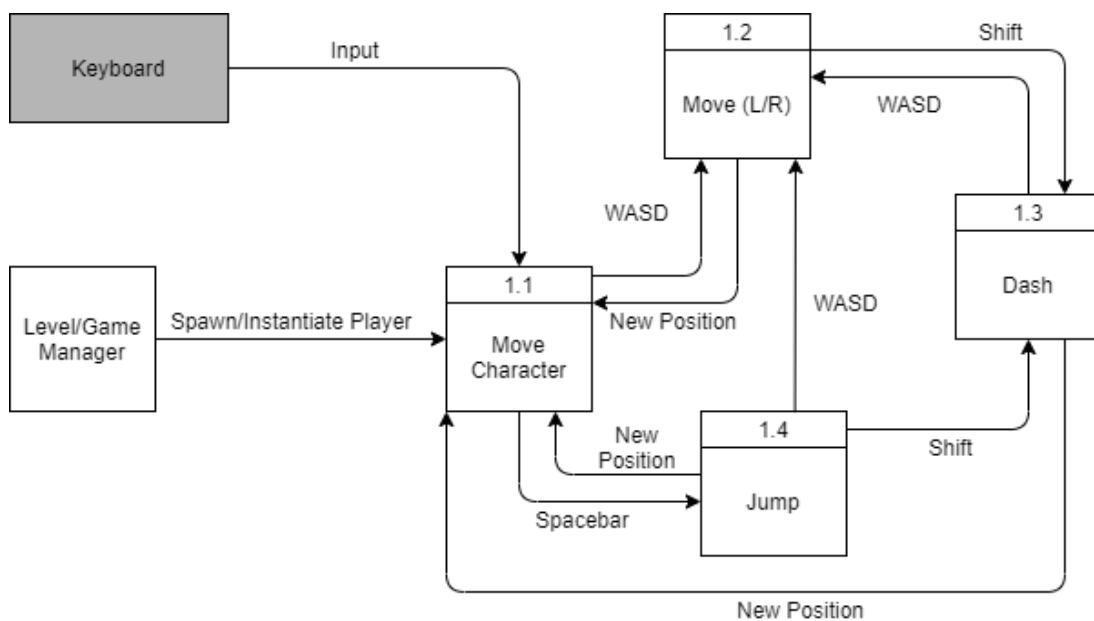# 3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

## Data Flow Diagrams                                                      Level 0



Level 1

**Process Descriptions**

Move Character:

WHILE input from keyboard change position of character

END WHILE

Move (L/R):

WHILE A key or D key, change horizontal position of character

IF Shift key is pressed: dash

return new position, set hasDashed (must touch ground before another dash)

ELSE

change position of character (can change after dash)

END WHILE and return new position if no shift key pressed

Dash:

If Shift key is pressed after Move (L/R) or Jump:

return new position, set hasDashed (must touch ground before another dash)

Jump:

Record spacebar press to determine jump

Move vertical character position

IF Shift key is pressed in the air: dash

return new position, set hasDashed (must touch ground before another dash)

able to move using WASD before touching ground

ELSE IF WASD pressed without shift

return new position after vertical ascension

ELSE return position (should be the exact same as before the jump)

## 4. Acceptance Tests _____9

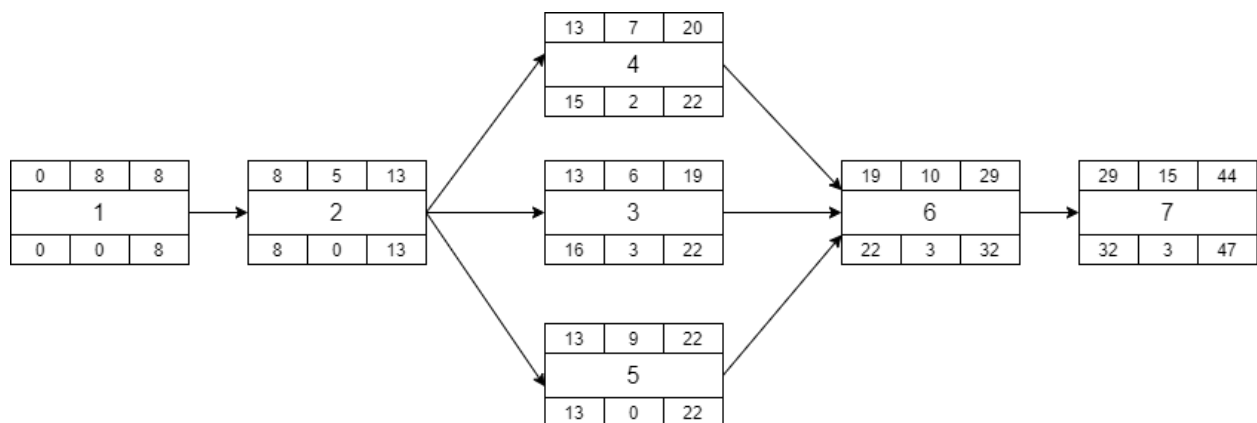| INPUT | OUTPUT |
| --- | --- |
| W | Nothing |
| A | Character moves left |
| S | Character moves right |
| D | Nothing |
| Spacebar | Variable jump, depending on how long spacebar is held; up to a max height |

| Spacebar + WASD | Jump in specified direction upwards (W and S have no effect) |
|---|---|
| Spacebar + Shift | Nothing |
| Spacebar + Shift + WASD | Jump in direction (W and S have no effect) and shift in direction (WASD have effect) |
| Shift | Nothing |
| Shift + WASD | Shift in direction (WASD have effect) |
| Shift + WASD + Spacebar | Shift in direction (WASD have effect but spacebar does not) |

## 5. Timeline _____/10

### Work items

| Task | Duration (Hours) | Predecessor Task(s) |
|---|---|---|
| 1. Watch Tutorials | 8 | - |
| 2. First Implementation | 5 | 1 |
| 3. Work on Collision Cases | 6 | 2 |
| 4. Add Dash | 7 | 2 |
| 5. Animation and Effects | 9 | 2 |
| 6. Second Rendition | 10 | 3, 4, 5 |
| 7. Final Rendition | 15 | 6 |

### Pert diagram

# Gantt timeline



| | duration | slack |

Left chart rows (1–7) plotted over time axis 1–56.

| | predicted time(hrs) | time spent(hrs) | key | | completed | | this week | | planned | |
|---|---|---|---|---|---|---|---|---|---|---|
| Miguel | | | hours | | 3 6 9 | 12 15 | 18 21 | 24 27 | 30 33 36 | 39 42 45 48 51 |
| Watch Tutorials | 8 | 5 | | | | | | | | |
| First Implementation | 5 | 3 | | | | | | | | |
| Work on Collision Cases | 6 | 3 | | | | | | | | |
| Add Dash | 7 | 3 | | | | | | | | |
| Animation and Effects | 9 | 5 | | | | | | | | |
| Second Rendition | 10 | 0 | | | | | | | | |
| Final Renditon | 15 | 0 | | | | | | | | |
| totals | 60 | 19 | | | | | | | | |