Name\_\_\_\_\_\_\_\_\_Yiqing\_\_\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

For this project, I am a member of team Low Effort Game and I am documentation specialist. For our game, I try to design the character controller part. It is the important part of the game. I will decide how to control the character and it changes during the game. The character needs to change his position when meet obstacles and wrong path so that keep the character continue running. I will try to give more characters into the game so that to enrich the game.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

**Scenarios 1**

**Name:** Character controller- movement

**Summary:** To control the movement of the character

**Actors:** Player

**Preconditions:** Start the game

**Basic sequence:**

**Step 1:** The player opens the game program and chooses to start the game

**Step 2:** Input a character name for the first time

**Step 3:** Use keyboard to control the character movement

**Step 4:** Control character inside path and not hit obstacle

**Exceptions:**

**Step 3:** If jump twice, player can get more score

**Step 4:** If hit with obstacles or outside path end game

**Post conditions:** Player model is transitioned to a new location in the scene

**Priority:** 2\*

**ID:** C01

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

**Scenarios 2**

**Name:** Character control - block

**Summary:** To stop character moving / stop the game

**Actors:** Player

**Preconditions:** Start the game

**Basic sequence:**

**Step 1:** start the game

**Step 2:** Use keyboard to control the character stop moving

**Exceptions:**

**Step 3:** Use keyboard to control the character continue/restart moving

**Post conditions:** stop game and turn on menu

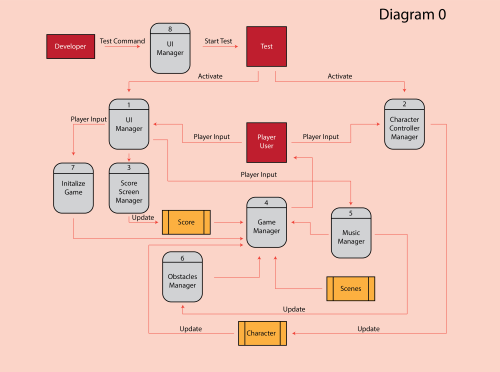
**Priority:** 2\*

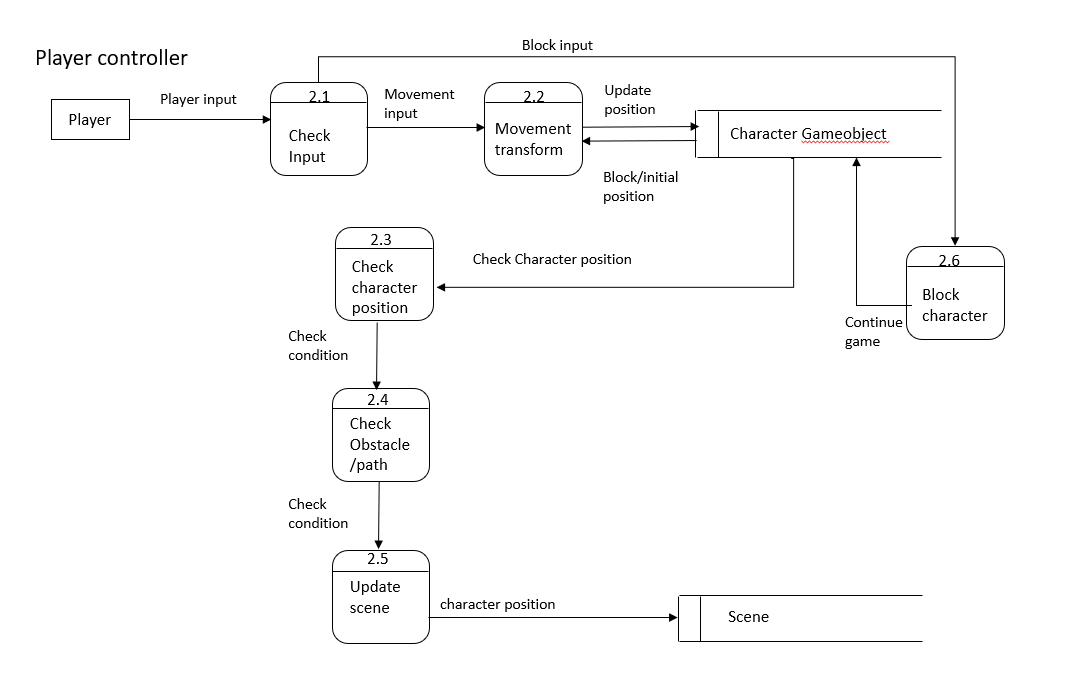
**ID:** C01

\*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

2.1 Check Input (Parameters: Player Input):

IF Player Input is movement control input

Call the Transform Character process

2.2 Transform Character (Parameters: Movement Input):

WHILE there is Movement Input

Update the position of the character gameobject

2.3 Check Character (Parameters: Character position):

IF Character Position is in range of gameobject

RETURN True

ELSE

RETURN False

2.4 Check Obstacle (Parameters: Path and Obstacle position)

IF find Obstacle

Return False

ELSEIF find path inside range and no obstacle around

Return truth

2.5 Update Scene (Parameters: condition return by 2.3 and 2.4):

IF True

Continue game

ELSE

End game

2.6 Block Character (Parameters: player input)

IF Player input block game

Stop game

ELSE

Continue game

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

**Character controller**

Player control the character with keyboard that keep character moving

**Test:** Verify “Jump” control input transforms character to cross a solid obstacle in front of him

**Parameters/Input:** space bar

**Output:** Character continue to move towards

**Test:** Verify “Left” control input transforms character to the left path

**Parameters/Input:** A / left arrow

**Output:** Character continue to move to left

**Test:** Verify “Right” control input transforms character to the right path

**Parameters/Input:** D / right arrow

**Output:** Character continue to move to right

**Test:** Verify “Duck” control input transforms character to cross a path under the obstacle

**Parameters/Input:** S / down arrow

**Output:** Player continue to move under an obstacle

**Test:** Verify “Squeeze” control input transforms character to be thin or fat to cross the path

**Parameters/Input:** W / up arrow

**Output:** Character shape will change to thin or fat

**Test:** Verify “Block” control to stop the game

**Parameters/Input:** Block control

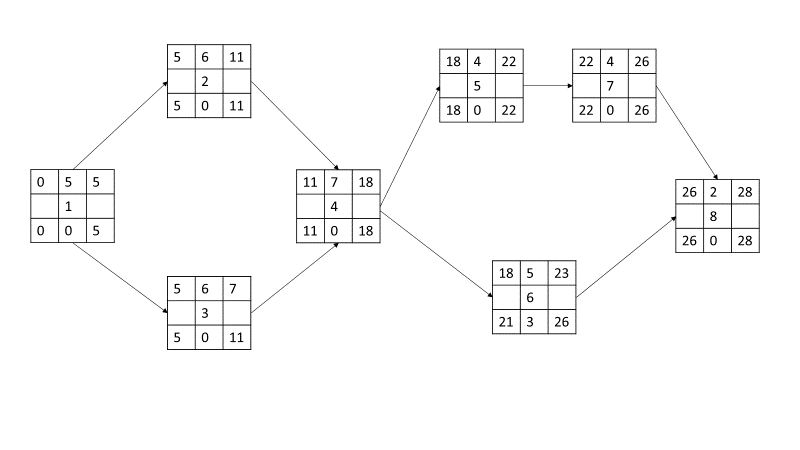
**Output:** Game stop and turn on menu

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 5 | - |
| 2. Character control design | 6 | 1 |
| 3. Report design | 6 | 1 |
| 4. Programming | 7 | 2, 3 |
| 5. Testing individual | 4 | 4 |
| 6. User Documentation | 5 | 4 |
| 7. Combine coding testing | 4 | 5 |
| 8. Installation | 2 | 6, 7 |

### Pert diagram



### Gantt timeline

