**Name: Naga Kadarla** Mark /50

# Brief introduction /3

As a game developer for "Monolith," I shape the player experience, focusing on character animation and movement in our thrilling restaurant management game. In "Monolith," players assume the role of a dedicated restaurant waiter, overseeing bustling dining floors to ensure top-notch customer service. My journey starts with meticulous Character Animation Planning, crafting captivating animations like Idle, Run, and Dish-Drop to breathe life into the restaurant's atmosphere. These animations are seamlessly implemented, immersing players in fast-paced restaurant management.

My responsibilities extend to Final Animation Testing and Debugging, ensuring flawless integration into the game's vibrant narrative. Transitioning to player experience, I design intuitive Player Movement, which evolves into creating foundational Basic Movement Controls. I delve into pathfinding algorithms, enabling efficient character navigation, and add collision detection for precise interactions. Advanced Collision Handling fine-tunes nuanced interactions, and rigorous Testing guarantees a seamless and immersive "Monolith" adventure. I am the artist behind this culinary journey's digital tapestry.

# Use case diagram with scenario.

## Use Case Diagrams

## 

**Scenarios**

**Scenario 1:**

**Name:** Role of Main Player

**Summary:** Take Customer Orders, Serve Food, Collect Payments and Wash The Plates.

Actors: Main Player

**Preconditions:** The game level has been started, and the restaurant environment is set up.

## Basic sequence:

Step 1: Player Moves Towards Customer.

Step 2: Taking Customer’s Orders.

Step 3: Interact with Cook and place the Kitcher Order List.

Step 4: Serving Food to Appropriate Customer.

Step 5: Collecting Payments from Customers.

Step 6: Take All Plats and Clear Table.

Step 6: Put the plates to wash.

## Exceptions:

Step 1: If there is no available path to reach the customer due to obstacles or overcrowding, the player may need to wait for a clear path or adjust their route. Some customers may require more time to decide or move, causing delays in the player's movement.

Step 2: If the player tries to take an order from a customer who is not ready to order, then the player will need to return or wait.

Customers might change their minds after initially placing an order, requiring the player to update the order accordingly.

Step 3: There may be delays in food preparation so wait or do other tasks.

Step 4: Serving the wrong dish or serving food to the wrong customer may occur.

Step 5: The player may need to wait or do other tasks, until customers finish the dish and give signal for payment.

Step 6: If the player misses clearing the plates on the table, then can’t place new customers.

Step 7: The player should move to the dishwasher area properly.

**Post conditions:** The level progresses with customers being served, the restaurant earning money, and customer satisfaction affecting the player's performance.

**Priority**: 1 (must-have)

**ID**: PR1

# 3. Monolith Game Data Flow diagram(s) from Level 0 to process description for your feature 14

## Data Flow Diagrams

## Context Level

## 

## Level 1

## 

## Level 2

**Process Descriptions**

Player Movement Process Descriptions:

1. **Basic Movement:**

* Description: This process handles the fundamental movement of the player character within the restaurant environment. It allows the player to navigate tables, counters, and other obstacles to reach various areas of the restaurant.
* Responsibilities:
* Interpret user input, such as keyboard or touchscreen controls, to move the player character.
* Implement collision detection to ensure the character does not pass through obstacles or occupied areas.
* Update the character's position and animations in real-time.

1. **Navigating Obstacles:**

* Description: The player character must navigate through crowded areas, obstacles like tables and chairs, and spilled items to move efficiently.
* Responsibilities:
* Implement pathfinding algorithms to find optimal routes around obstacles.
* Detect and handle collisions with objects or other characters.
* Ensure smooth navigation in tight spaces.

1. **Interacting with Customers:**

* Description: The player’s character interacts with customers to take orders, serve food, and collect payments. It involves moving the character to customers at various tables.
* Responsibilities:
* Detect customer interactions and trigger relevant actions, such as taking orders or delivering food.
* Handle customer requests and respond appropriately.
* Maintain customer satisfaction and happiness.

1. **Clearing Tables:**

* Description: After customers finish their meals, this process enables the player character to clear tables by removing dirty dishes and resetting them for new customers.
* Responsibilities:
* Move the character to the tables to clear dishes and reset settings.

# 4. Acceptance Tests 9

Acceptance tests for player movement in a game like " Monolith " can help ensure that the player can navigate the game environment effectively. Here are some acceptance tests specifically for player movement in "Monolith ":

1. **Basic Movement Test:**

* Scenario: The player starts the game.
* Action: The player attempts to move the character in different directions using the mouse.
* Expected Result: The player character should move smoothly in the specified direction without any glitches or delays.

1. **Obstacle Navigation Test:**

* Scenario: The game environment includes obstacles like tables and chairs.
* Action: The player tries to navigate the character through a crowded restaurant area with obstacles.
* Expected Result: The player character should intelligently navigate around obstacles, avoiding collisions and getting stuck.

1. **Order Taking Test:**

* Scenario: A customer is ready to place an order player wants to Move the correct table to take the order.
* Action: The player moves to interact with the customer to take their order.
* Expected Result: The player should be able to reach the appropriate customer’s table to take the customer's order.

1. **Kitchen Access Test:**

* Scenario: The player needs to access the kitchen entrance to place orders.
* Action: The player tries to move the character towards the kitchen entrance and place the order.
* Expected Result: The player character should enter the kitchen entrance area smoothly.

1. **Dishwashing Interaction Test:**

* Scenario: The player needs to move and stop while dishwashing area reaches.
* Action: The player moves the character to the dishwashing area.
* Expected Result: The player character should successfully move to the dishwashing area and stop while reaching dishwashing area.

1. **Speed and Responsiveness Test:**

* Scenario: The player needs to respond quickly to various in-game events.
* Action: The player attempts to move the character rapidly in different directions.
* Expected Result: The player character should respond promptly to user input, allowing the player to react to changing situations in the game.

1. **Pathfinding Test:**

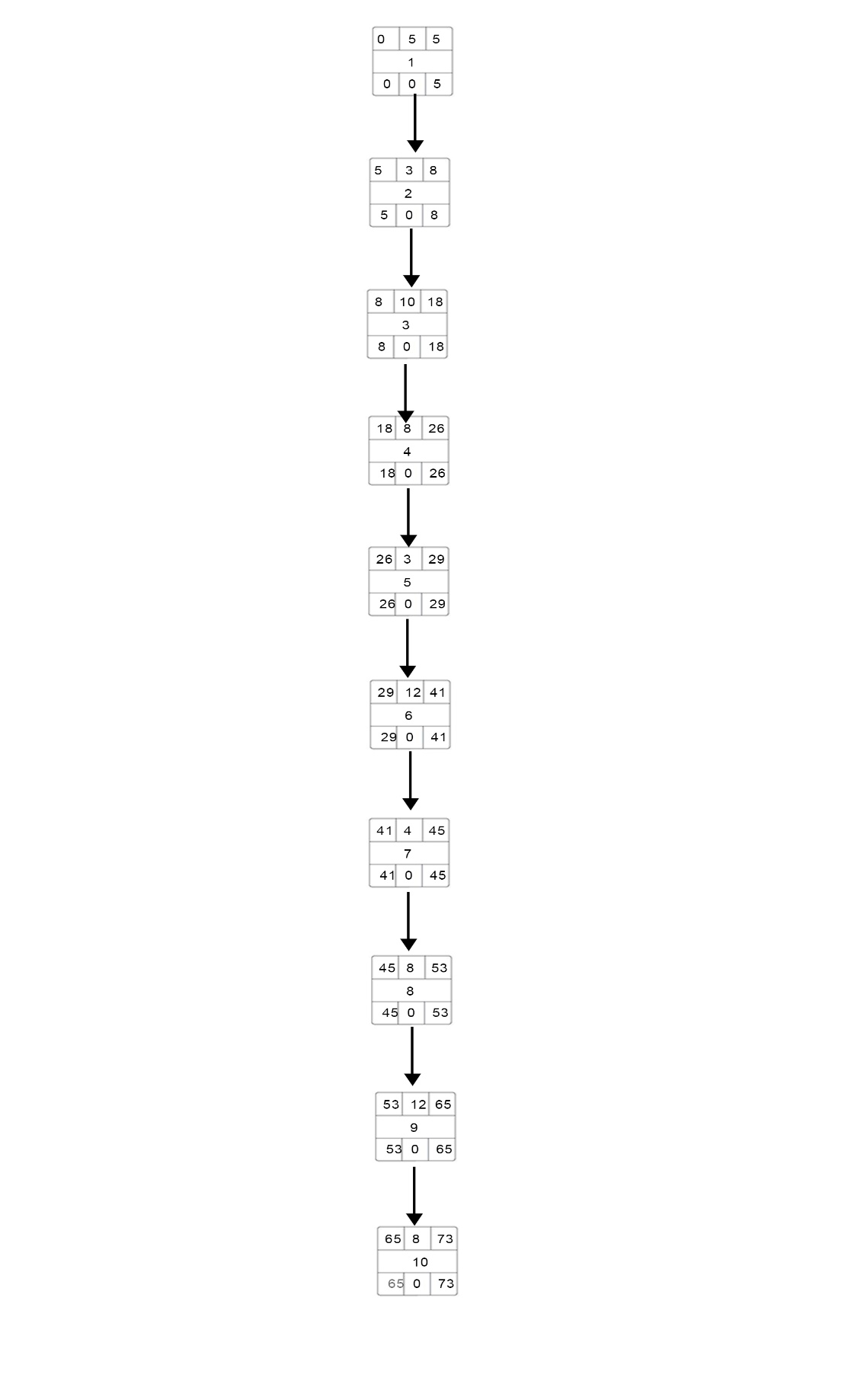
* Scenario: The game has complex layouts with multiple tables, kitchen entrance, dish washer area and customer’s table.
* Action: The player moves the character to a distant location.
* Expected Result: The player character should navigate efficiently using a pathfinding algorithm, avoiding obstacles, and reaching the destination.

**5. Timeline /10**

## Work items

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Task** | **Duration (PWks)** | **Predecessor Task(s)** |
| 1 | Study Unity Character Design | 5 |  |
| 2 | Player Locomotion and Asset Design | 3 | 1 |
| 3 | Player Control Development | 10 | 2 |
| 4 | Pathfinding Algorithm Implementation | 8 | 3 |
| 5 | Animation Asset Creation | 3 | 2 |
| 6 | Animation Integration and Synchronization | 12 | 5 |
| 7 | Rigorous Animation Quality Assurance | 4 | 6 |
| 8 | Comprehensive Detection Systems | 8 | 4 |
| 9 | Complex Collision Resolution | 12 | 8 |
| 10 | Holistic Movement and Collision Testing | 8 | 9 |

**Pert Diagram**

****

**Gantt Timeline**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ` |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 3 |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 9 |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 10 |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 6263 64 65 66 67 68 69 70 71 72 73 |