Name: Rohit Narwal Mark ______________________/50

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

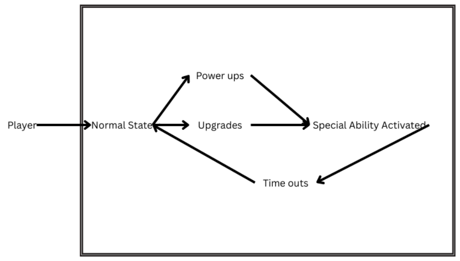
1. Brief introduction __/3

My feature for the Orange Ninja video game is to design and implement the **Interactives and Upgrades** system. This includes ensuring that whenever the player obtains an upgrade or interacts with an in-game object, the functionality is properly executed in the backend.

The goal of this feature is to enhance gameplay variety by providing players with unique powers and tools. Examples include unlimited health mode, different guns, and additional abilities planned for future versions. These upgrades and interactives will ensure that the Orange Ninja feels dynamic and customizable, creating a more engaging experience.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Scenario 1 (Upgrade Acquisition)

- **Name:** Apply Upgrade
- **Summary:** Player acquires or activates an upgrade (e.g., new weapon or power).
- **Actors:** Player

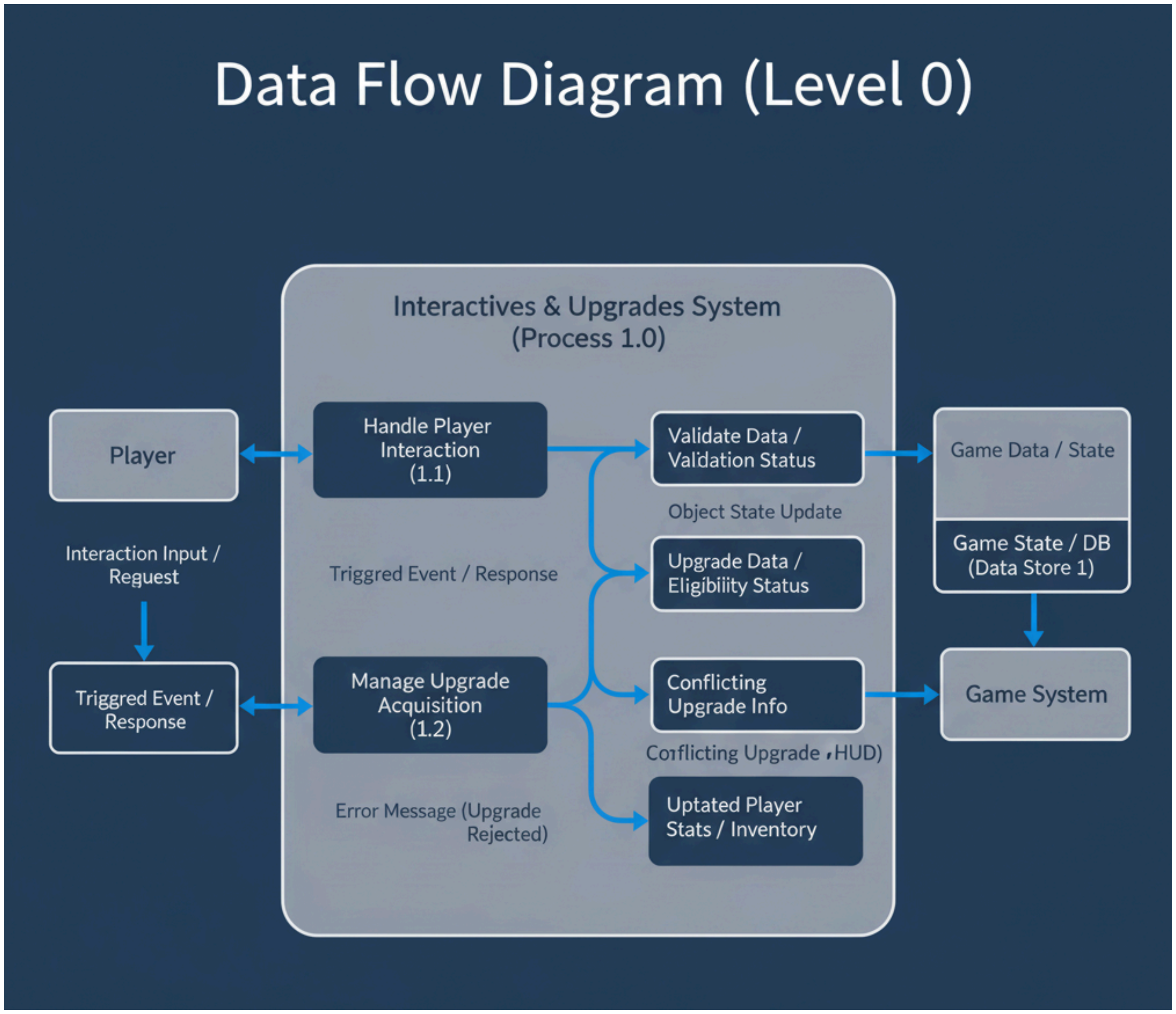
- **Preconditions:** Player has reached a checkpoint, defeated an enemy, or interacted with an object that grants an upgrade.
- **Basic Sequence:**
  - Player picks up or selects upgrade.
  - System validates upgrade eligibility.
  - Upgrade applied → backend updates stats/abilities.
  - HUD updated with new feature (e.g., weapon icon, health boost).
- **Exceptions:**
  - Player already has conflicting upgrade → reject and show message.
  - Upgrade data missing or corrupted → fallback to default.
- **Postconditions:** Player character now has upgraded ability reflected in gameplay.

Scenario 2 (Interactive Object Use)

- **Name:** Use Interactive Object
- **Summary:** Player interacts with map objects like levers, chests, or portals.
- **Actors:** Player
- **Preconditions:** Player is near an interactive object.
- **Basic Sequence:**
  - Player triggers interaction button.
  - System validates interaction availability.
  - Object triggers event (e.g., open chest → give item, lever → unlock door).
- **Exceptions:**
  - Object locked or unavailable → error message.
- **Postconditions:** Object responds to interaction, player inventory or map state updated.

3. Data Flow diagram(s) from Level 0 to process description for your feature ______14

Data Flow Diagrams


Data Flow Diagram (Level 0)
Interactives & Upgrades System (Process 1.0)
Player
Handle Player Interaction (1.1)
Validate Data / Validation Status
Game Data / State
Object State Update
Interaction Input / Request
Triggred Event / Response
Upgrade Data / Eligibility Status
Game State / DB (Data Store 1)
Triggred Event / Response
Manage Upgrade Acquisition (1.2)
Conflicting Upgrade Info
Game System
Conflicting Upgrade ,HUD)
Error Message (Upgrade Rejected)
Uptated Player Stats / Inventory

Process Descriptions

```
WHILE game session active:
  IF player interacts with object:
    Validate interaction
    Trigger object event (reward, unlock, etc.)
    Update game state and HUD
  ENDIF

  IF player acquires upgrade:
    Validate upgrade eligibility
    IF valid:
      Apply upgrade effect
      Update stats and HUD
      Save progress
    ELSE:
      Reject upgrade with error message
    ENDIF
  ENDIF
END WHILE
```

4. Acceptance Tests _______9

Upgrade Test

- Description: Verify upgrades are applied correctly.
- Steps: Acquire an upgrade.
- Expected Output: Player stats updated, HUD reflects change.

Conflicting Upgrade Test

- Description: Prevent incompatible upgrades.
- Steps: Attempt to apply two conflicting upgrades.
- Expected Output: System rejects second upgrade with error message.

Interactive Object Test

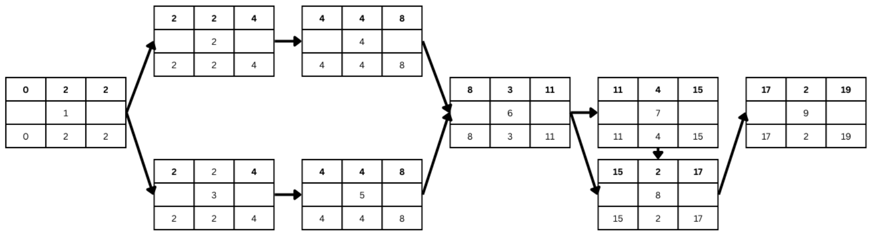
- Description: Verify player can interact with objects.
- Steps: Approach chest, press interaction key.
- Expected Output: Chest opens, item added to inventory.

5. Timeline ________/10

Task	Duration (days)	Predecessor Task
1. Requirements Collection	2	
2. Design Interactives	2	1
3. Upgrade System Design	2	1
4. Programming Backend (Interactives)	4	2
5. Programming Backend	4	3

(Upgrades)		
6. Conflict/Error Handling	3	4, 5
7. Testing	4	6
8. Documentation	2	6, 7
9. Final Integration	2	8

6. Pert diagram



7. Gantt timeline

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	Requirements Collection	■	■																	
3	Design Interactives			■	■															
4	Upgrade System Design			🟦	🟦	🟦	🟦													
5	Programming Backend (Interactives)					■	■	■	■											
6	Programming Backend (Upgrades)					■	■	■	■											
7	Conflict/Error Handling					🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦					
8	Testing												■	■	■	■				
9	Documentation																■	■		
10	Final Integration																		■	■