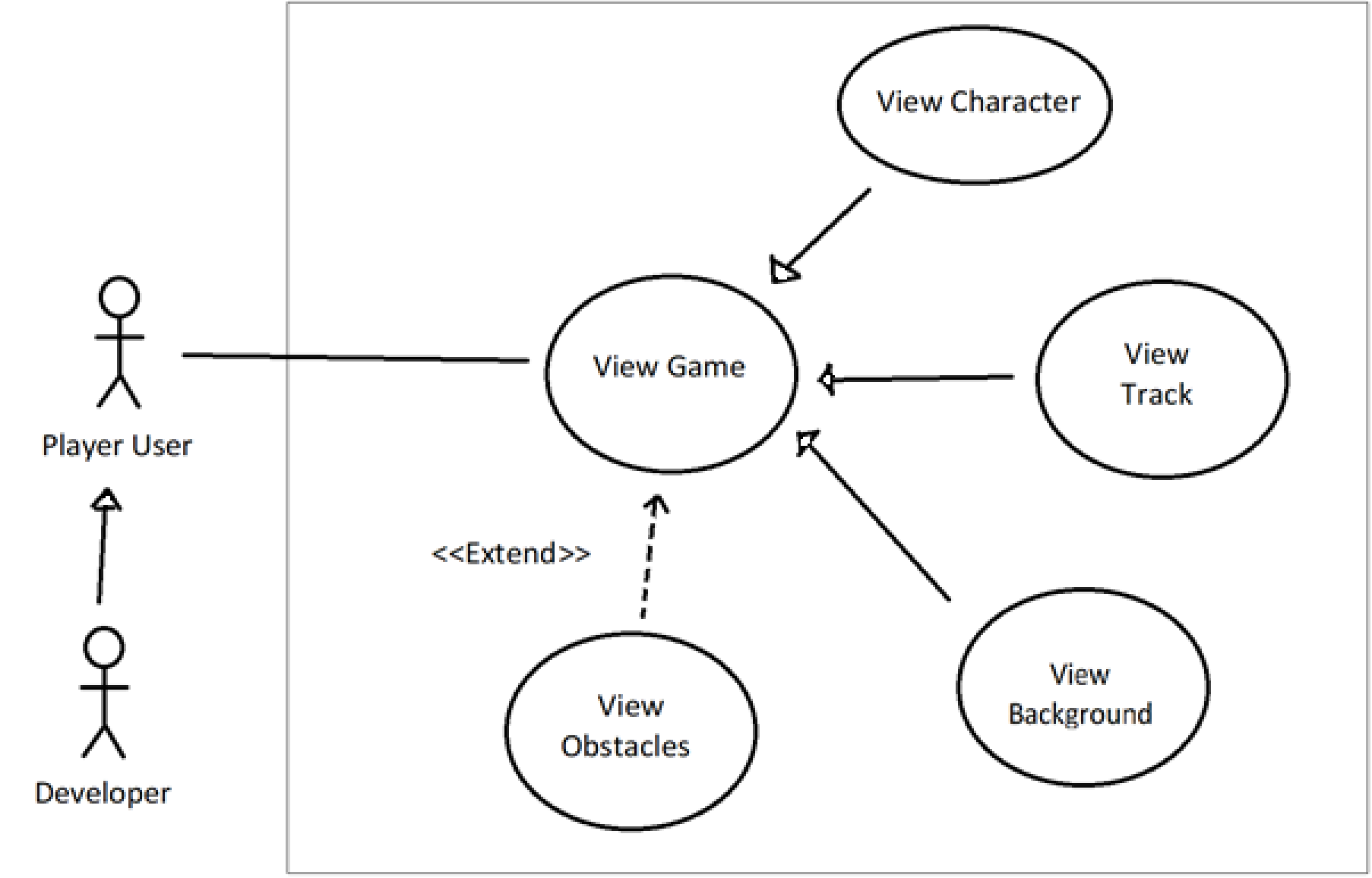
Name\_\_Aaron Ludwig\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature is the Game Manager, which is the process that displays the game in its central gameplay state. It takes information from the Obstacle Manager, Music Manager, and Initialize Game processes, and from the Character and Scenes data stores and uses that information to change what is ultimately displayed to the Player User external agent.

## Use case diagram with scenario \_\_14

### Use Case Diagram:



### Scenarios

**Name:** View Character

**Summary:** The Player User uses the output of the game manager to view the position of the Character entity.

**Actors:** Player User

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Initialize Game process indicates main view should be shown

**Step 2:** Game Manager receives current position and status of the character from the Character data store

**Step 3:** Game Manager displays Character entity on the screen

**Exceptions:**

**Step 1:** Game is paused: do not display character

**Step 2:** Game ends: do not display character

**Post conditions:** Game is currently in the gameplay state and no menus are being displayed.

**Priority:** 2\*

**ID:** C01

**Name:** View Track

**Summary:** The Player User uses the output of the game manager to view the track on which the Character entity and Obstacles are placed on.

**Actors:** Player User

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Initialize Game process indicates main view should be shown.

**Step 2:** Game Manager receives track entity information from Scenes data store

**Step 3:** Game Manager displays track entity on the screen

**Exceptions:**

**Step 1:** Game is paused: do not display character

**Step 2:** Game ends: do not display character

**Post conditions:** Game is currently in the gameplay state and no menus are being displayed.

**Priority:** 1\*

**ID:** C02

**Name:** View Background

**Summary:** The Player User uses the output of the game manager to view the background of the game.

**Actors:** Player User

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Initialize Game process indicates main view should be shown.

**Step 2:** Game Manager receives background entity information from Scenes data store

**Step 3:** Game Manager displays game background on the screen

**Exceptions:**

**Step 1:** Game is paused: do not display gameplay background

**Step 2:** Game ends: do not display gameplay background

**Post conditions:** Game is currently in the gameplay state and no menus are being displayed.

**Priority:** 3\*

**ID:** C03

**Name:** View Obstacles

**Summary:** The Player User uses the output of the game manager to view the Obstacles on the track.

**Actors:** Player User

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Initialize Game process indicates main view should be shown.

**Step 2:** Game Manager receives indication from Music Manager that an Obstacle should appear.

**Step 3:** Game Manager receives Obstacle data from Obstacles Manager.

**Step 4:** Game Manager displays Obstacle entity on the screen

**Exceptions:**

**Step 1:** Game is paused: do not display Obstacle entities

**Step 2:** Game ends: do not display Obstacle entities

**Step 3:** Character entity passes Obstacle entity: hide passed Obstacle entity

**Post conditions:** Game is currently in the gameplay state, no menus are being displayed, and Music Manager has indicated an Obstacle should appear.

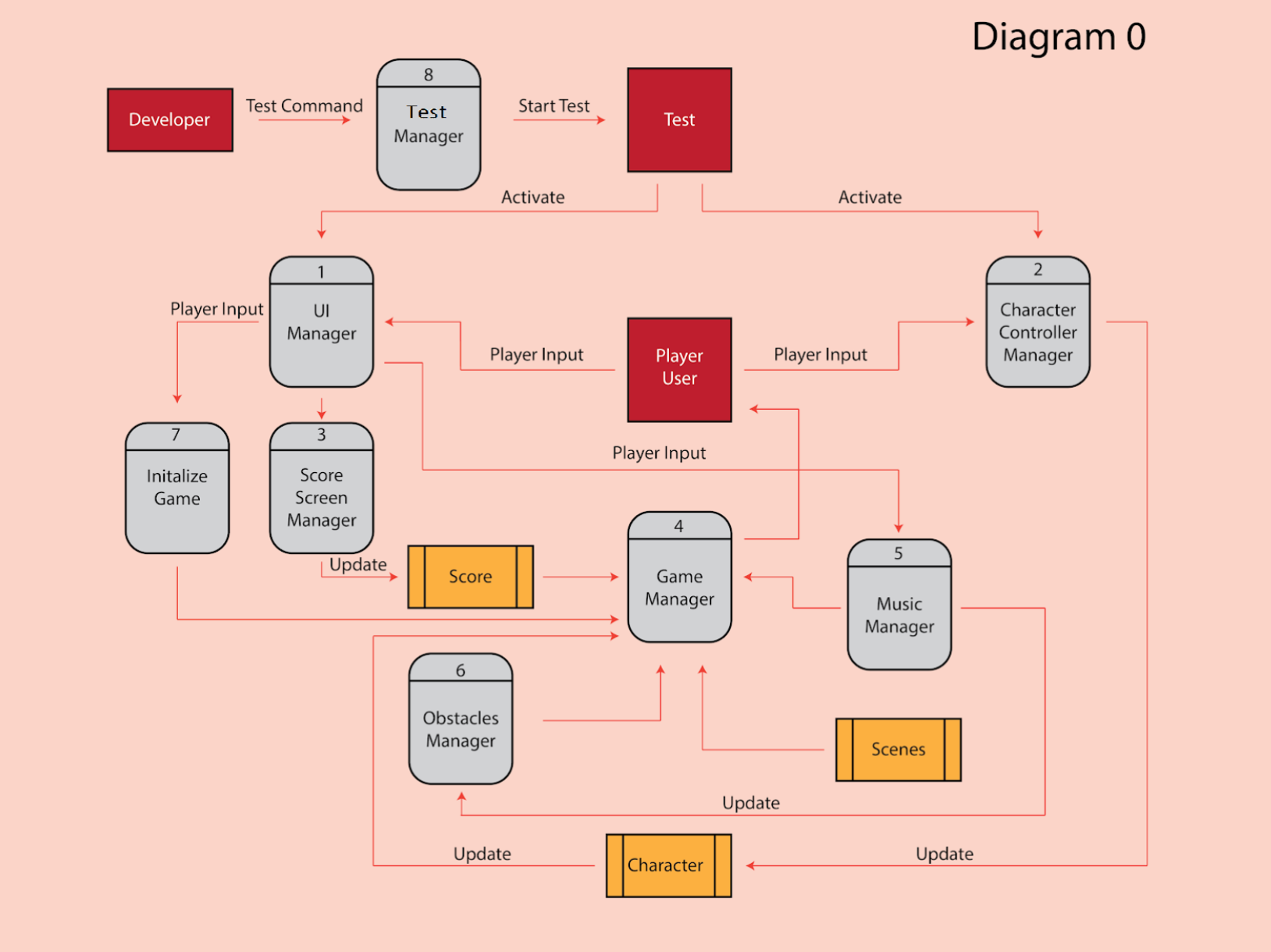
**Priority:** 2\*

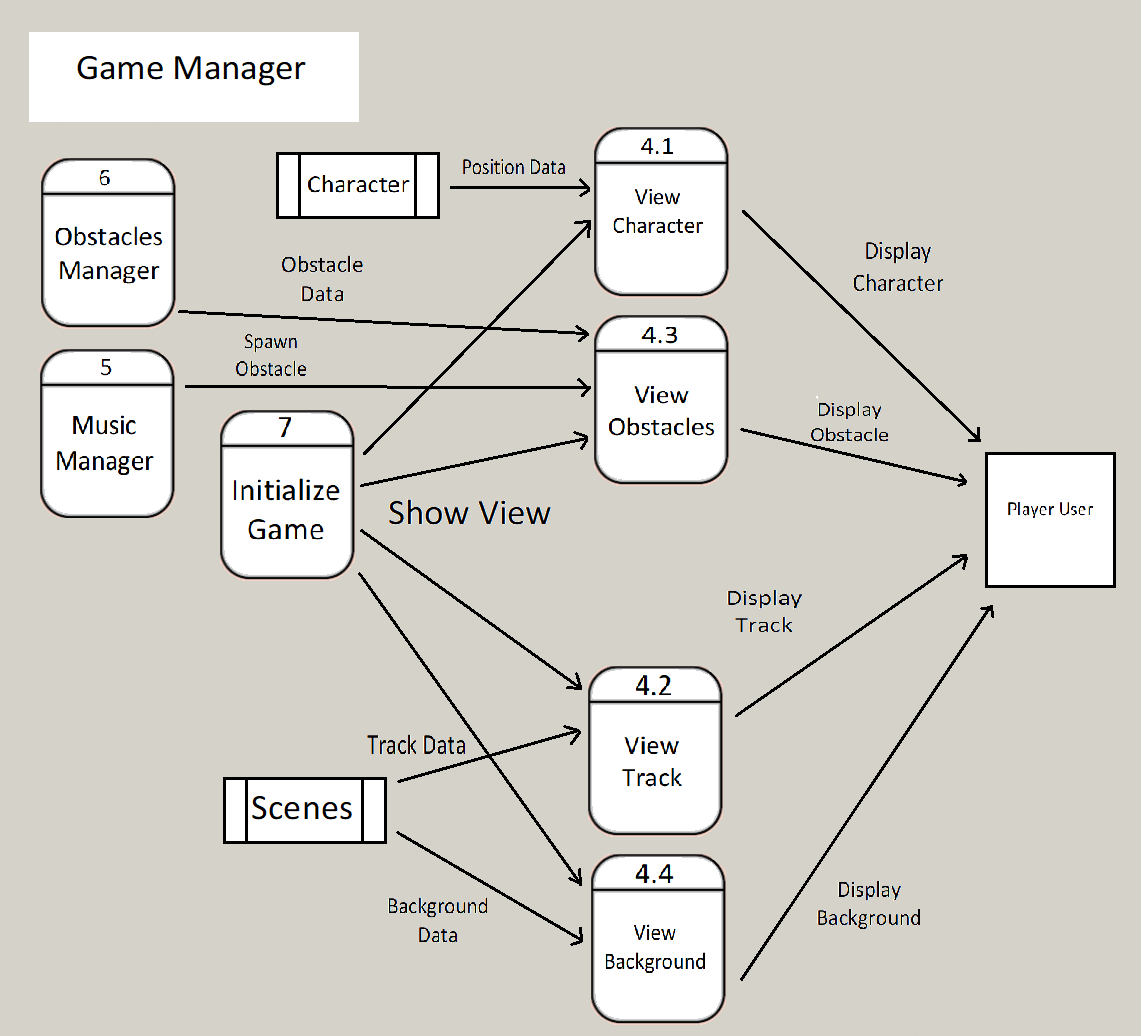
**ID:** C04

\*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

View Character\*:

WHILE Game is running

IF Game has been initialized AND Menu is not displayed

Display Character

ELSE

Hide Character

Freeze Character movement

END WHILE

View Obstacles\*:

WHILE Game is running

IF Game has been initialized AND Menu is not displayed

IF Obstacle is requested

Place given Obstacle on Track

Display given obstacle

Unfreeze Obstacle movement

IF Obstacle has passed Character

Hide Obstacle

Freeze Obstacle movement

ELSE

Hide Obstacle

Freeze Obstacle movement

END WHILE

View Track\*:

WHILE Game is running

IF Game has been initialized AND Menu is not displayed

Display given Track

ELSE

Hide Track

END WHILE

View Background\*:

WHILE Game is running

IF Game has been initialized AND Menu is not displayed

Display given Background

ELSE

Hide Background

END WHILE

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

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Start instance of a Game

* Is the Character entity displayed when a menu is not present?
* Is the Character entity not displayed when a menu is present?
* Is the Character entity’s displayed position updating based on what’s stored in its data store when a menu is not present?
* Does the Character entity’s position not update when a menu is present?
* Is the Track entity displayed when a menu is not present?
* Is the Track entity not displayed when a menu is present?
* Is Background entity displayed when a menu is not present?
* Is Background entity not displayed when a menu is present?

Request Instance of an Obstacle

* Is the Obstacle displayed when a menu is not present?
* Is the Obstacle not displayed when a menu is present?
* Is the Obstacle entity’s displayed position updating based on what’s stored in the provided data store when a menu is not present?
* Does the Obstacle entity’s position not update when a menu is present?
* Is Obstacle no longer displayed once character entity has passed?

If all tests pass, the Game Management system should be working.

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

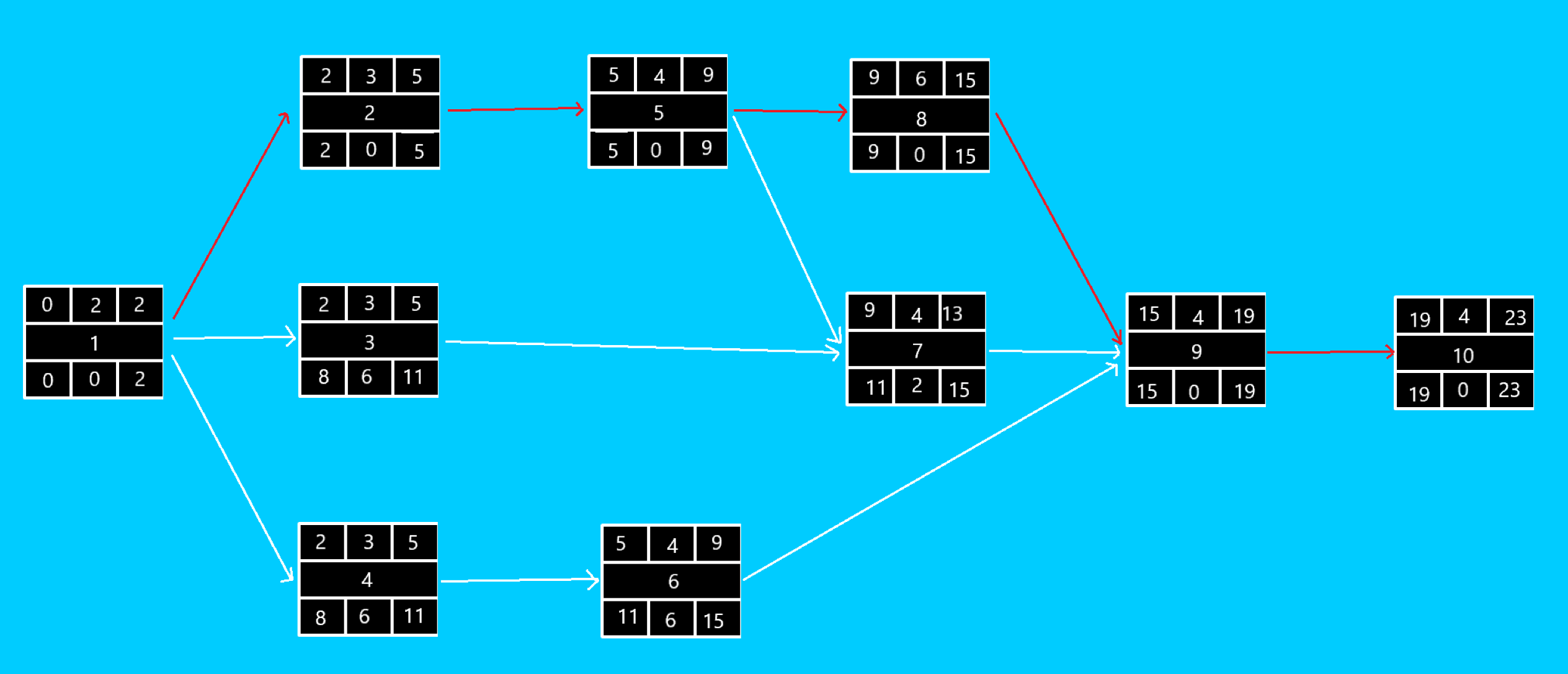
[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 2 | - |
| 2. Track Design | 3 | 1 |
| 3. Character Design | 3 | 1 |
| 4. Background Design | 3 | 1 |
| 5. View Track Functionality | 4 | 2 |
| 6. View Background Functionality | 4 | 4 |
| 7. View Character Functionality | 4 | 3, 5 |
| 8. View Obstacles Functionality | 6 | 5 |
| 9. Testing | 4 | 6, 7, 8 |
| 10. Integration With other Components | 4 | 9 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  | 3,5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6,7,8 |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 9 |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |