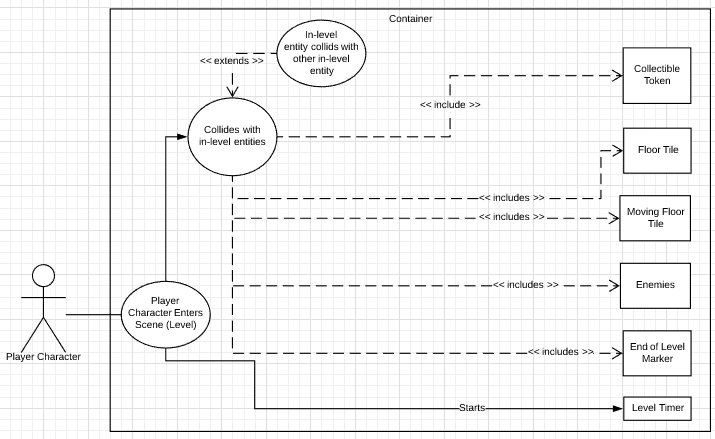
Name Jorge Olivas\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

I will be focusing on level design. The aim is to have it feel like any other platforming game, where your character can move left-right and up-down. There will be moving platforms that move in the same way. There will be parts of the level that affect the character, e.g: slows them down, speeds them up,

## Use case diagram with scenario \_\_14

### Use Case Diagrams



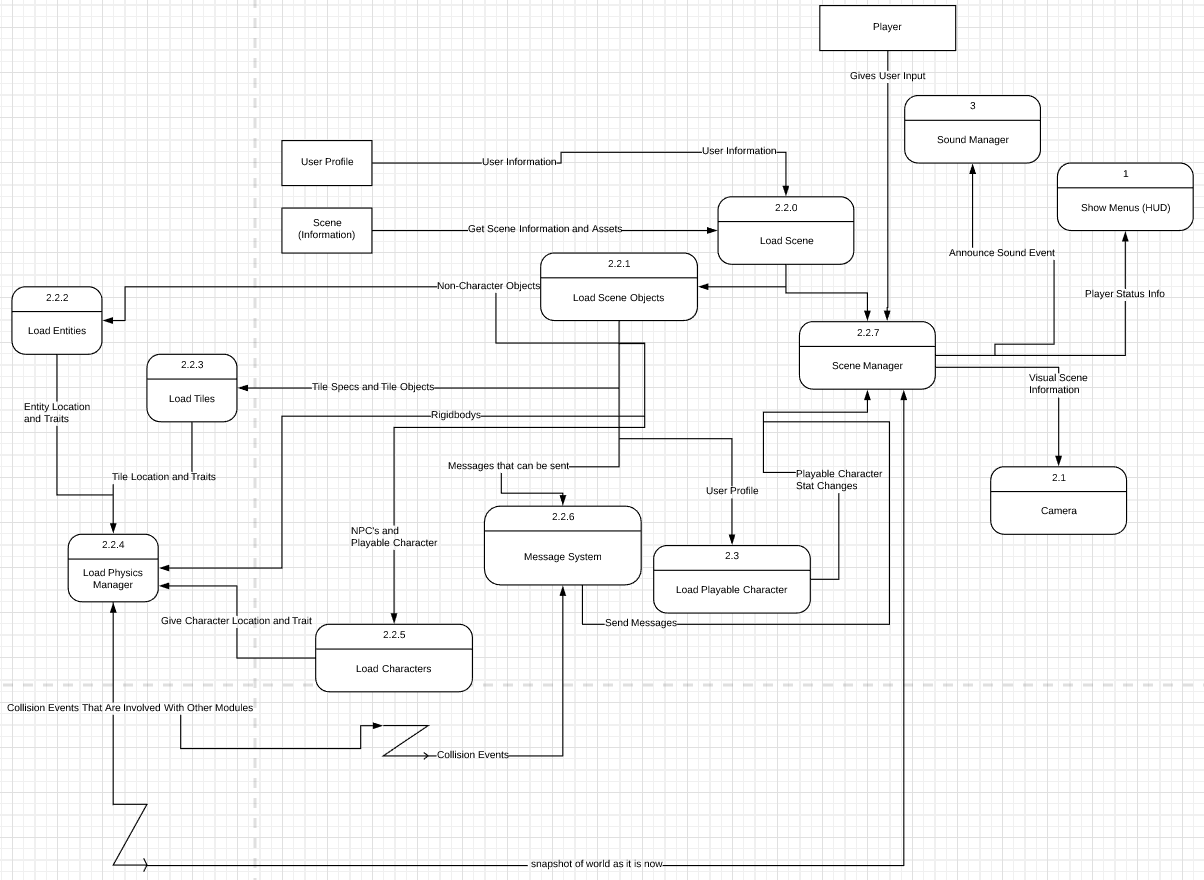
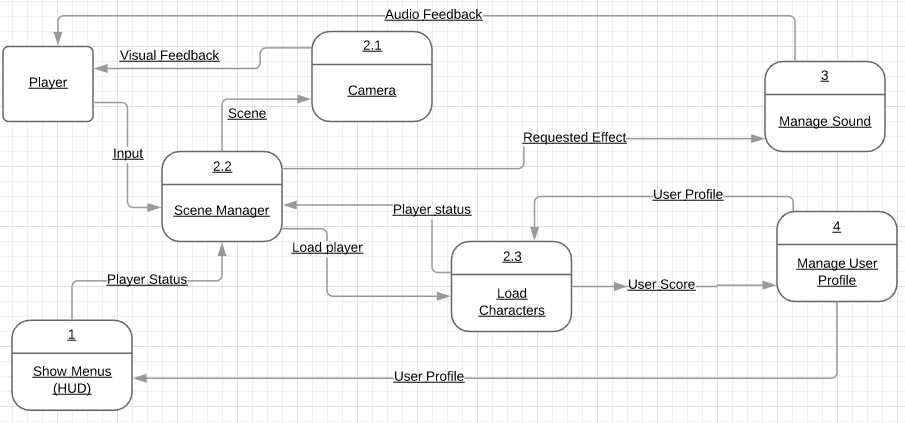
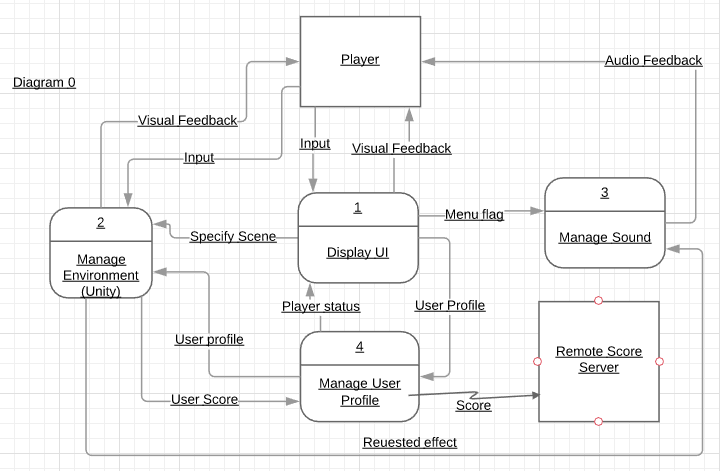
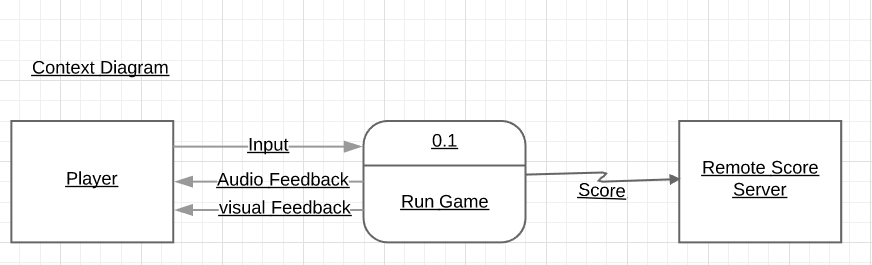
### Scenarios

|  |  |
| --- | --- |
| **Name** | Player Character Enters Scene |
| **Summary** | The player begins playing the level |
| **Actors** | Player |
| **Preconditions** | In game scene and all contents therein have been initialized. Managers of all modules involved in the game have been initialized. |
| **Basic Sequence** | Step 1: Receive character information and scene requested to load  Step 2: Spawn in Characters and in-game entities, physics, etc. Start in-game timer. |
| **Exceptions** | None |
| **Post Conditions** | Scene is active and constantly refreshing. |
| **Priority** | 1 |
| **ID** | C01 |

|  |  |
| --- | --- |
| **Name** | Player Character collides with other in-level entities |
| **Summary** | Depending on the collision, information will be sent about the event to the messaging system. |
| **Actors** | Player, other in-game entities |
| **Preconditions** | In game scene and all contents therein have been initialized. Managers of all modules involved in the game have been initialized. Character is in level. |
| **Basic Sequence** | Step 1: Check player character location and size  Step 2: Check all entity’s size and location  Step 3: See if they collide  Step 4: Report collision information to message system |
| **Exceptions** | Step 1: Compare each in-game entity with every other entity to see if they collide  Step 2: Report collision data to messaging system |
| **Post Conditions** | In-game events can now communicate with other modules. |
| **Priority** | 1 |
| **ID** | C02 |

## Data Flow diagram(s) from Level 0D to process description for your feature \_\_\_\_\_\_/14

### Data Flow Diagrams



Focus on 2.2: Scene Manager Below

### Process Descriptions

* Load Scene
  + Load Non Character Entities (NCE) with List of NCE’s
  + Load Tiles with List of Tiles
  + Load Characters
  + Load Message System
  + Load Playable Character
  + Load Physics System – Insert
    - NCE’s
    - Tiles
    - Characters
    - Make physics system talk to message system
    - Make physics system talk to scene manager
    - Make Message System listen to Physics System
  + Start Scene Manager – Give
    - Snapshot of world as it is
    - Make listen to message system
    - Take in Player input
    - While level not complete / while not game over
      * Refresh physics system
        + Send message about collision to message system if necessary

Have message system send the collision information to the scene manager so it can prompt whichever module it needs to go to

* + - * Refresh snapshot of world sent to scene manager
    - If level complete
      * Give user profile manager new player stats for the level
    - Else
      * Hand over control to the UI manager

## Acceptance Tests \_\_\_\_\_\_\_\_/9

**Load Scene**

* When a scene is loaded, it is loaded
  + Check that player character is colliding with ground to make sure collision data was loaded
* Scenes that are a work in progress cannot be loaded
  + Try to load level that isn’t on list of completed levels
* A player can only choose from scenes specified by the UI manager

**Communicate with User Profile manager**

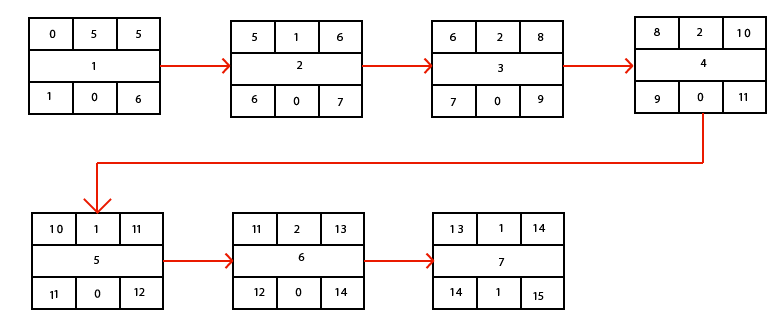
* Can only send valid user data to user profile manager (all values initialized.)
  + Test that user data values are initialized before they are sent

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

## Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 5 |  |
| 2. Make Placeholder Art | 1 | 1 |
| 3. Create Tile Template System | 2 | 2 |
| 4. Create Message System | 2 | 3 |
| 5. Communicate Message System and Template System | 1 | 4 |
| 6. Design Levels | 2 | 5 |
| 7. Testing | 1 | 6 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |