Name: Nathan Zander Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

I will be creating the design of the level. I will handle the placement of objects in the level, and program them to move towards the player in time with the music. I will control how and when they enter the scene, and finish line.

Use case diagram with scenario \_\_14

Example:

### Use Case Diagrams

### Scenarios

**Name:** Level select screen

**Summary:** Player uses this to select the level.

**Actors:** Player.

**Preconditions:** Game has been started.

**Basic sequence:**

**Step 1:** Start Game.

**Step 2:** Select level.

**Exceptions:**

**Step 1:** The game is quit: exit game.

**Post conditions:** The game begins

**Priority:** 2

**ID:** C01

**Name:** Level begin

**Summary:** This will instantiate the level.

**Actors:** Player.

**Preconditions:** Level has been selected.

**Basic sequence:**

**Step 1:** Place meteors.

**Step 2:** Place rings.

**Step 3:** Place power ups.

**Step 4:** Begin moving.

**Exceptions:**

**Step 1:** The game is quit: exit game.

**Post conditions:** Gameplay has been started.

**Priority:** 1

**ID:** C02

**Name:** Meteor

**Summary:** An object that must be avoided or else a game over is applied.

**Actors:** Player.

**Preconditions:** Level has begun.

**Basic sequence:**

**Step 1:** Meteor is placed at the start of level.

**Step 2:** Meteor must be avoided. If it is not, player is given a game over.

**Step 3:** Meteor is removed at end of level.

**Exceptions:**

**Step 1:** The game is quit: exit game.

**Post conditions:** The game is lost unless it was avoided

**Priority:** 2

**ID:** C03

**Name:** Power up

**Summary:** Applies a score multiplier to the player.

**Actors:** Player.

**Preconditions:** Level has begun.

**Basic sequence:**

**Step 1:** Power up is placed at beginning of level.

**Step 2:** If the power up is collided with, add a temporary score multiplier.

**Step 3:** Power up is removed at end of level.

**Exceptions:**

**Step 1:** The game is quit: exit game.

**Post conditions:** A score multiplier is added, unless it was avoided.

**Priority:** 3

**ID:** C04

**Name:** Ring

**Summary:** Adds to the score board when collided with.

**Actors:** Player.

**Preconditions:** Level has begun.

**Basic sequence:**

**Step 1:** Ring is placed at start of level.

**Step 2:** Collision with ring increases the score.

**Step 3:** Ring is removed at end of level.

**Exceptions:**

**Step 1:** The game is quit: exit game.

**Post conditions:** Player’s score has been incremented

**Priority:** 2

**ID:** C05

**Name:** Score board

**Summary:** Keeps track of how man points the player has.

**Actors:** Player.

**Preconditions:** Level has begun.

**Basic sequence:**

**Step 1:** Score board starts at 0 when level begins.

**Step 2:** Score board increments at a slow pace as you survive.

**Step 3:** Rings will add to the scoreboard.

**Step 4:** Multipliers will multiply the incoming score.

**Step 5:** Level is completed or lost and score is read out to the player.

**Exceptions:**

**Step 1:** The game is quit: exit game.

**Post conditions:** The player is given a rating of how well they did.

**Priority:** 2

**ID:** C06

**Name:** Finish line

**Summary:** Completes the level and unloads everything.

**Actors:** Player.

**Preconditions:** Level has been completed.

**Basic sequence:**

**Step 1:** Finish line is placed at end of level.

**Step 2:** Player completes the level without hitting any obstacles and crosses the finish line.

**Step 3:** Meteors are unloaded.

**Step 4:** Power ups are unloaded.

**Step 5:** Rings are unloaded.

**Step 6:** Final score is read out to the player.

**Step 7:** Return player to level up screen.

**Exceptions:**

**Step 1:** The game is quit: exit game.

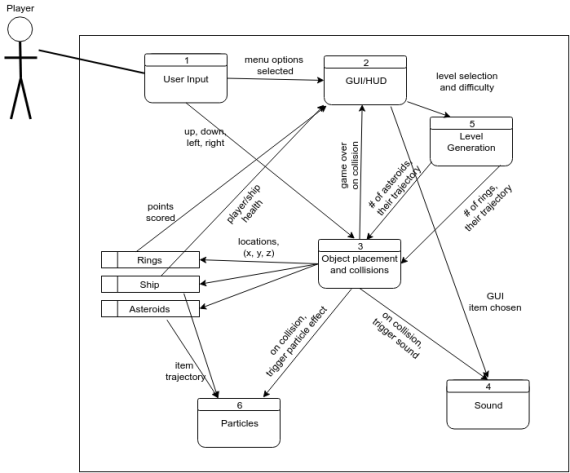
**Post conditions:** Player is returned to level select screen.

**Priority:** 1

**ID:** C07

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

### Data Flow Diagrams



### Process Descriptions

If the level has been selected, load the arrangement of obstacles specified in that level’s scene object.

Begin moving objects toward the player. Listen for collisions with objects.

If the player collides with a meteor, display the game over screen and unload the level.

If the player crosses the finish line, display the level complete screen and unload the level.

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

Create a short test level with pre-recorded player movements.

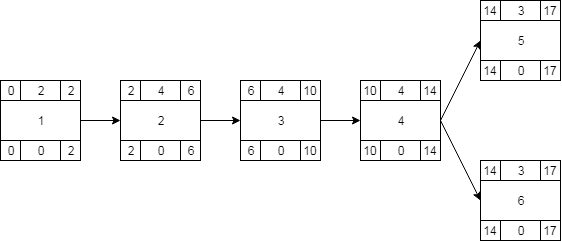
* Fly into a ring and make sure the score increases.
* Fly into a power up and then a ring to ensure the score increases by a higher amount.
* Fly into a meteor and check to see if it is in the game over state.
* Complete the level and make sure it displays the level complete screen and returns to level select.

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Days) | Predecessor Task(s) |
| 1. Create scene with camera and boundaries | 2 | - |
| 2. Add skybox and stand-ins for meteors, powerups, rings | 4 | 1 |
| 3. Program Level movement | 4 | 2 |
| 4. Create finish line | 4 | 3 |
| 5. Program Game Over | 3 | 4 |
| 6. Program Level Complete | 3 | 4 |

### Pert diagram



### Gantt timeline

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