Name: Taran Haug Mark _____/50

1. Brief introduction _/3

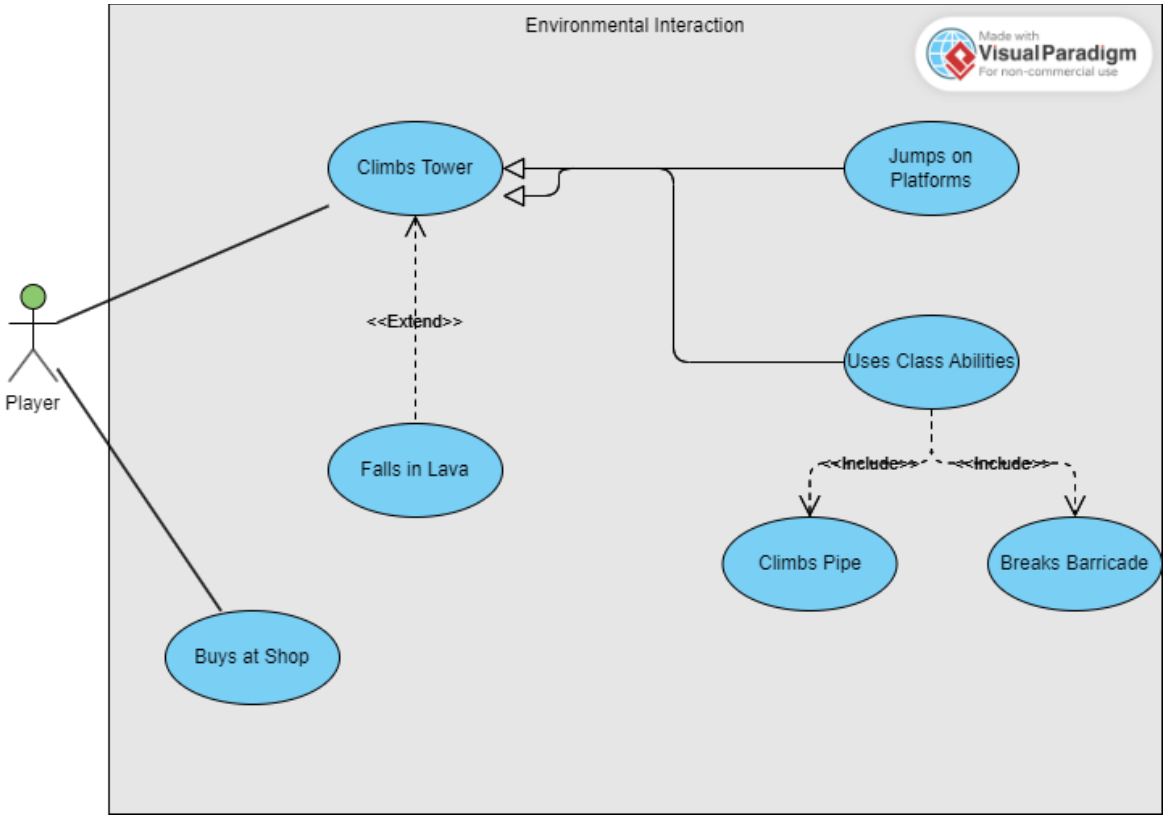
My feature for the Hero Climb video game is the environment generation. Each time the game is played, the environment is procedurally generated.

My job is to ensure that the environment always generates in such a way that the player has at least one route forward, while also allowing for alternative routes available to certain hero classes.

I am also responsible for ensuring that the environment is populated with enemies and collectibles during generation. These must be generated in appropriate amounts at appropriate locations.

2. Use case diagram with scenario _14

Use Case Diagram



Scenarios

Name: Climbs Tower
Summary: The player climbs higher up in the tower.
Actors: Player
Preconditions: Environment generated, player initialized.
Basic sequence:

Step 1: Player climbs up a level of the tower

Step 2: Player reaches a new level of the tower

Exceptions:

Step 1: Player falls lower in tower

Step 2: Rising lava catches player

Step 3: Player Dies

Post conditions: The players current level increases.

Priority: 1

ID: E01

Name: Buys at Shop

Summary: The player spends coins to buy items from the shop.

Actors: Player

Preconditions: Environment generated, player initialized, player reaches shop.

Basic sequence:

Step 1: Player selects item in shop

Step 2: Player loses coins equal to the value of the item

Step 3: Player acquires item

Exceptions:

Step 1: Player has insufficient Coins

Step 2: Player does not lose coins or acquire item

Post conditions: Player has new item.

Priority: 3

ID: E02

Name: Falls in Lava

Summary: The player is caught by the rising lava and dies.

Actors: Player

Preconditions: Environment generated, player initialized, lava has risen to player's level.

Basic sequence:

Step 1: Lava rises

Step 2: Lava makes contact with player

Step 3: Player dies

Exceptions:

Step 1: Player climbs faster than lava rises

Step 2: Player escapes lava

Post conditions: Player loses the game.

Priority: 2

ID: E03

Name: Jumps on Platforms

Summary: The player jumps on platforms in order to progress.

Actors: Player
Preconditions: Environment generated, player initialized.
Basic sequence:
Step 1: Player jumps onto platform
Step 2: Player jumps from platform to platform
Step 3: Player climbs higher to a higher level in tower
Exceptions:
Step 1: Player falls off of platform
Step 2: Player does not progress
Post conditions: Player elevation increases.
Priority: 1
ID: E04

Name: Uses Class Abilities
Summary: Player uses an ability specific to their class.
Actors: Player
Preconditions: Environment generated, player initialized.
Basic sequence:
Step 1: Player presses button to activate their class ability
Step 2: Class ability activates
Step 3: Class ability allows player to progress down a different path
Exceptions:
Step 1: Class ability is pressed while player is already doing an action: ignore input.
Step 2: Class ability is pressed while there is no viable way to use it: ability does not allow the player to progress.
Post conditions: Player accesses new route
Priority: 2
ID: E05

Name: Climbs Pipe
Summary: Player uses rogue class ability to climb a pipe.
Actors: Player
Preconditions: Environment generated, player initialized, player class is rogue, pipe present.
Basic sequence:
Step 1: Player reaches pipe
Step 2: Player uses class ability to climb pipe
Step 3: Player progresses higher
Exceptions:
Step 1: Player jumps while on pipe: player jumps off of pipe
Post conditions: Player reaches area at the top of the pipe.

Priority: 2
ID: E06

Name: Breaks Barricade
Summary: Player uses fighter class ability to break a barricade
Actors: Player
Preconditions: Environment generated, player initialized, player class is fighter, barricade present.
Basic sequence:

Step 1: Player reaches barricade
Step 2: Player uses fighter class ability to destroy barricade
Step 3: Player progresses past the barricade

Exceptions:

Step 1: Player misses barricade with ability: barricade is not destroyed.

Post conditions: Player accesses area behind the barricade.
Priority: 2
ID: E07

*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

Data Flow Diagrams

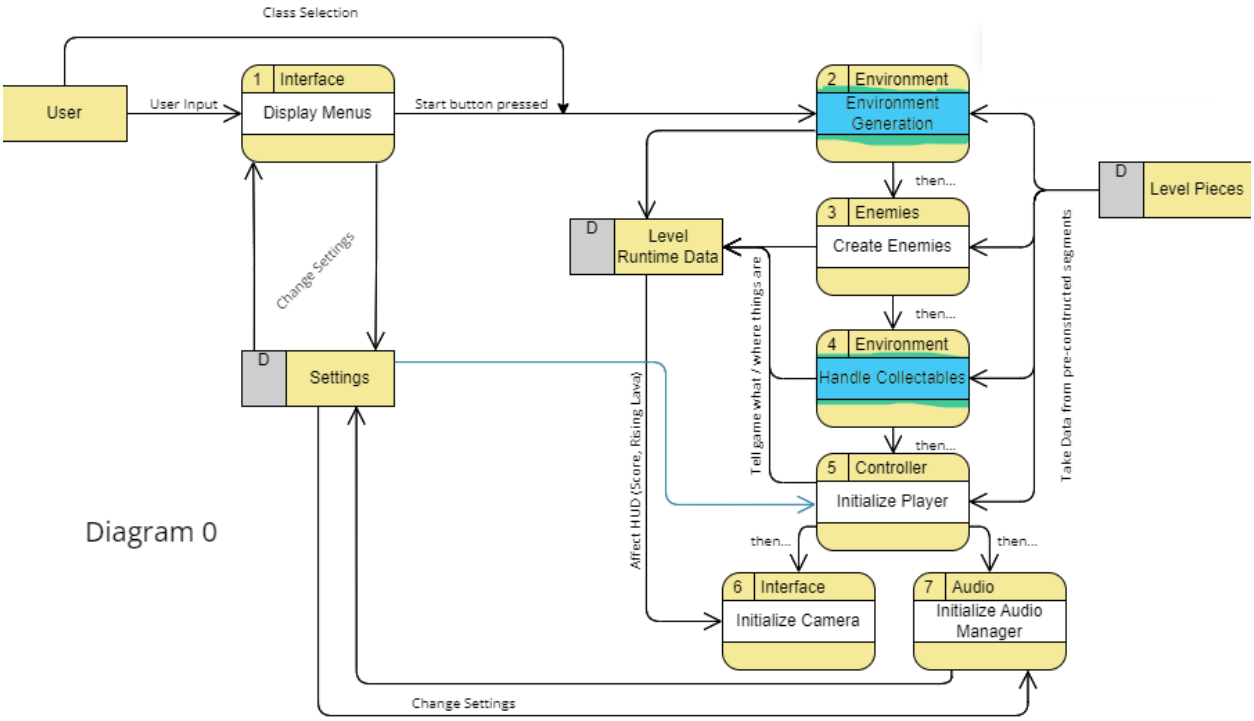
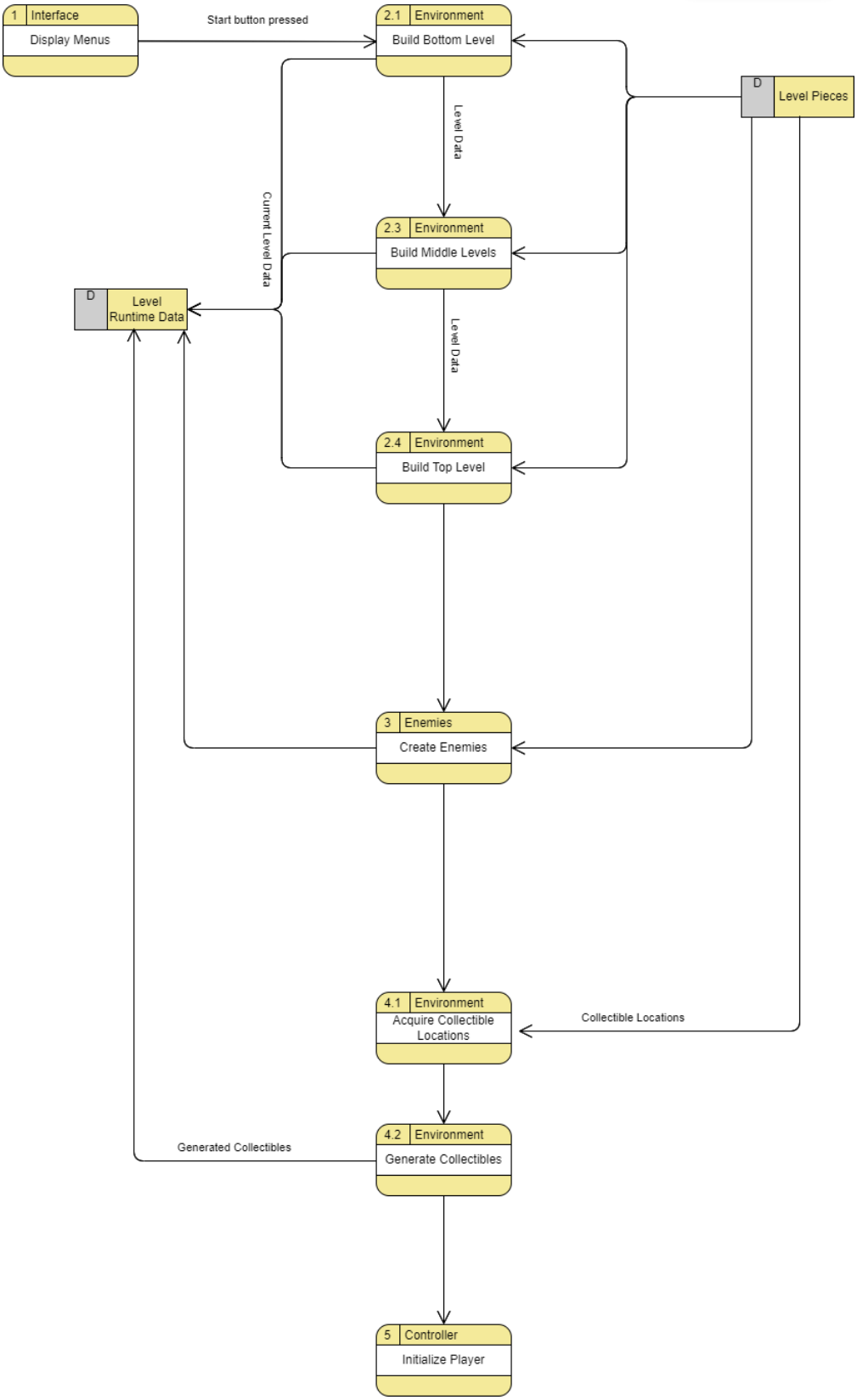


Diagram 0

Diagram 1


1 Interface
Display Menus
Start button pressed
2.1 Environment
Build Bottom Level
D Level Pieces
Level Data
Current Level Data
2.3 Environment
Build Middle Levels
D Level Runtime Data
Level Data
2.4 Environment
Build Top Level
3 Enemies
Create Enemies
4.1 Environment
Acquire Collectible Locations
Collectible Locations
4.2 Environment
Generate Collectibles
Generated Collectibles
5 Controller
Initialize Player

Process Descriptions

Build Bottom Level:

Instantiate a bottom level piece into the scene at the base of the tower.

Build Middle Levels:

WHILE the number of levels on the tower is less than its maximum:

Randomly determine whether each section of the next level will have a path up, a path accessed by a class ability, or no path up.

IF x levels have been added since the last shop:

Replace a one section with a shop

END IF

Ensure that at least one section has a path up. If none do, change one section to add a path up.

Randomly select a corresponding level piece for each section.

Add the selected level pieces as a new level at the top of the tower.

END WHILE

Build Top Level:

Instantiate a top level piece into the scene at the top of the tower.

Acquire Collectible Location:

FOR each level in the tower:

Store potential locations on that level where collectibles may spawn.

END FOR

Generate Collectibles:

FOR each collectible spawn location:

Randomly select a collectible to add to the level (none is valid)

END FOR

4. Acceptance Tests _____9

To determine whether the function for determining the kind of level piece used in each section of a level works properly, we run the following test:

Generate values for 1000 levels and write these values to a file. Separate the values for each level with a newline. A value of 0 is no path, 1 is a path, 2 is a class-specific path, 3 is a shop.

This output file will have the follow characteristics:

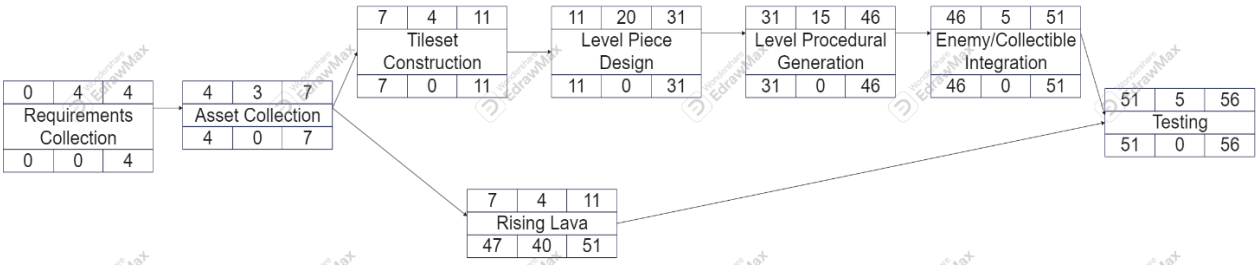
- Each line will have the same number of values
- Each line will have at least one value of 1.
- Each value will be between 0 and 3.
- Each line will have at most one value of 3.

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	4	-
2. Asset Collection	3	1
3. Tileset Construction	4	2
4. Rising Lava	4	2
5. Level Piece Design	20	3
6. Level Procedural Generation	15	5
7. Enemy and Collectible Integration	5	5, 6
8. Testing	5	6, 4

Pert diagram



Gantt timeline