Name: Jason Culbertson   Mark ______________________/50

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

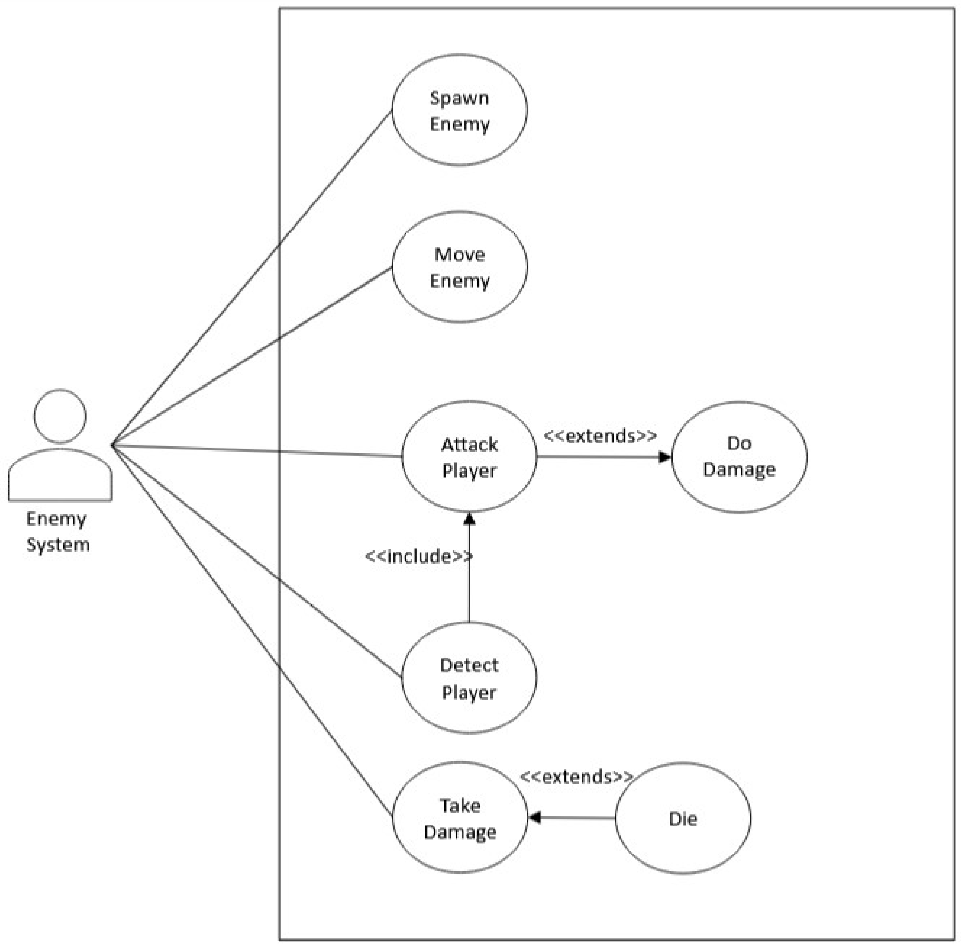
1. Brief introduction __/3

My feature for the Hero Climb video game project is the enemies.

This will include designing and implementing a variety of enemies. Each enemy will have movement, attacks, and pathfinding that will require the use of "AI". I will be implementing this AI using state machines and basic conditionals. I need to make sure that enemies stay where they are supposed to and interact with the level properly.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

[You will need a scenario for each use case]

Name: Create Enemy

Summary: The enemy system will spawn an enemy at the generation of the level

Preconditions: The level has been initialized

Basic sequence:

- **Step 1:** Spawn the enemy at the specific position
- **Step 2:** Begin the movement for the enemy
- **Step 3:** If the player is detected, attack it.
- **Step 4:** Repeat until player is not detected

Exceptions:

- **Step 1:** Player is not detected, just move.
- **Step 2:** Takes damage from player, if health is zero die.

Post conditions: Enemy is despawned out of range

Priority: 1*

ID: C01

*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

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

Data Flow Diagrams

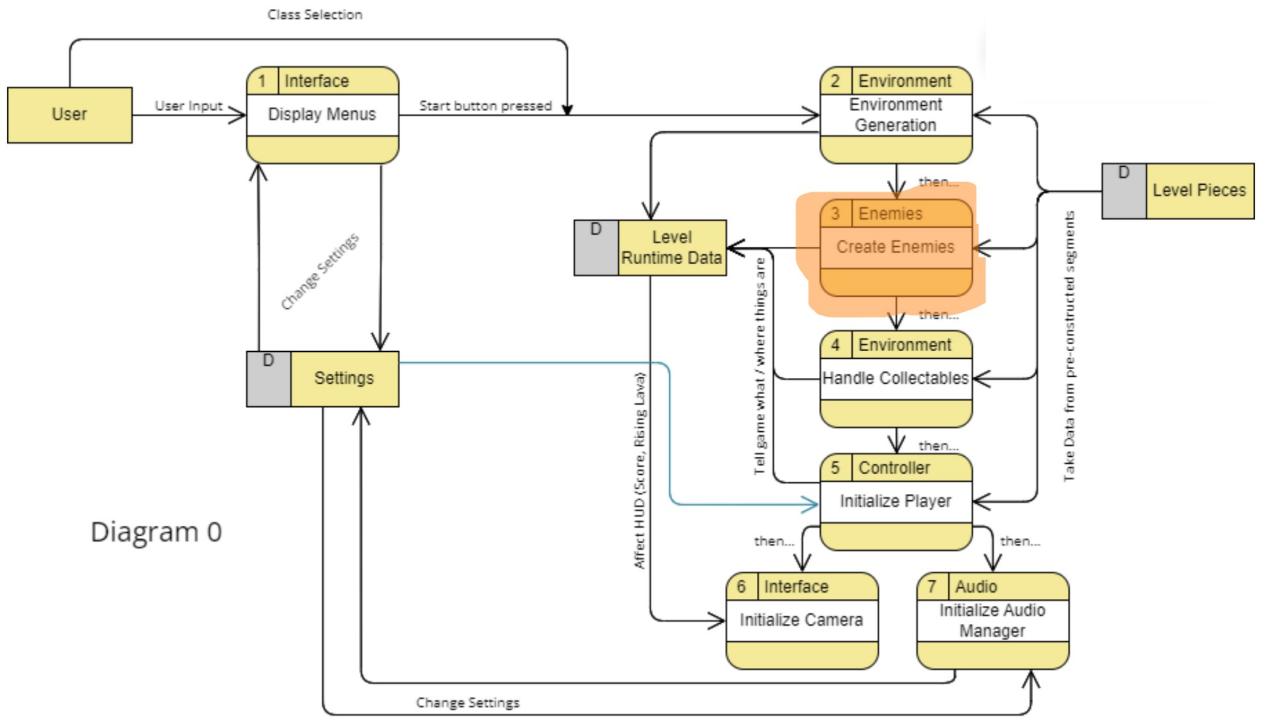
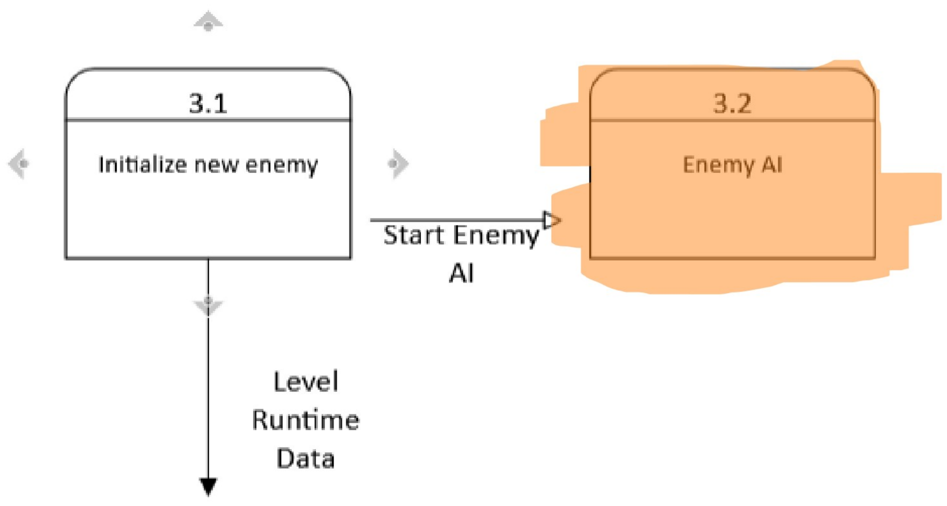
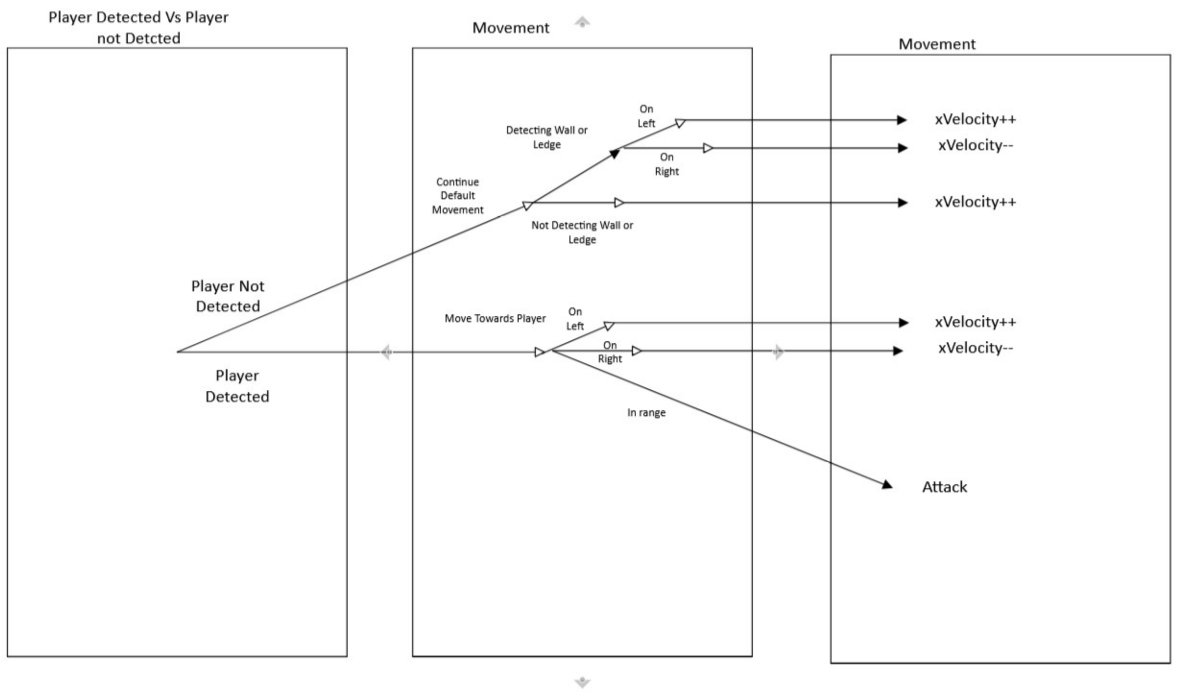


Diagram 0

Diagram for Enemies (3)



Process Descriptions



4. Acceptance Tests ______9

Example for enemy position test feature

Run feature for each test case, asserting expected outcomes.

The test suite will have the following characteristics:

- Initial position set to (0, 0)
- Position can be set to new coordinates
- Enemy can move relative to current position
- Enemy cannot move outside game boundaries
- Collision detection works with other game objects

Example for position and movement tests

Test Case	Initial Position	Action	Expected Outcome
Set Position	(0,0)	Set to (10,20)	Position = (10,20)
Move	(5,5)	Move by (3,-2)	Position = (8,3)
Boundary Check	(98,98)	Move by (5,5)	Position = (100,100)
Collision	(5,5)	Check collision	Collision Detected

		with object at (5,5)	
No Collision	(5,5)	Check collision with object at (20,20)	No Collision Detected

5. Timeline ________/10

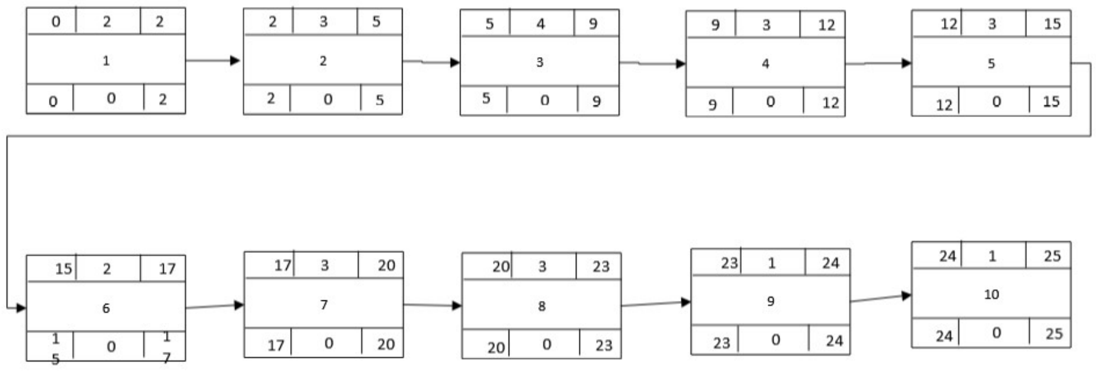
[Figure out the tasks required to complete your feature]

Example:

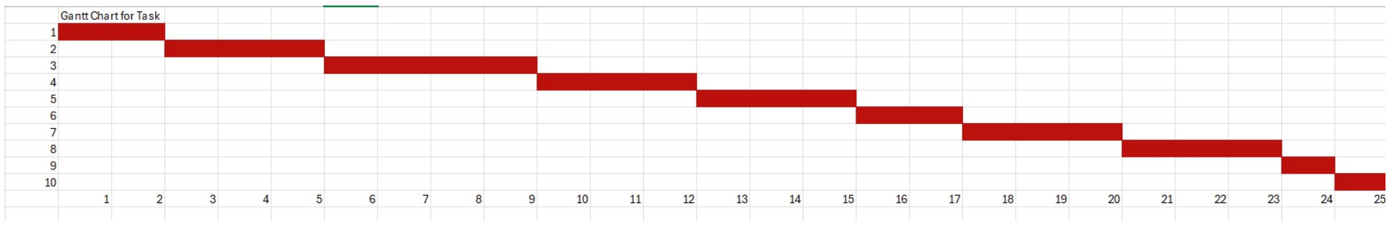
Work items

Task	Duration (Hours)	Predecessor Task(s)
1. Research AI patterns for 2D platformers	2	None
2. Design basic enemy behavior (e.g., patrolling)	3	Research AI patterns
3. Implement pathfinding algorithms	4	Design basic enemy behavior
4. Create state machine for enemy states (e.g., idle, alert, chase)	3	Implement pathfinding algorithms
5. Program enemy attack patterns	3	Create state machine for enemy states
6. Develop collision detection for enemies	2	Program enemy attack patterns
7. Integrate AI with level design elements (e.g., platforms, obstacles)	3	Develop collision detection for enemies
8. Test and debug enemy AI interactions	3	Integrate AI with level design elements
9. Optimize performance of the AI code	1	Test and debug enemy AI interactions
10. Final review and adjustments based on playtesting feedback	1	Optimize performance of the AI code

Pert diagram



Gantt timeline


Gantt Chart for Task