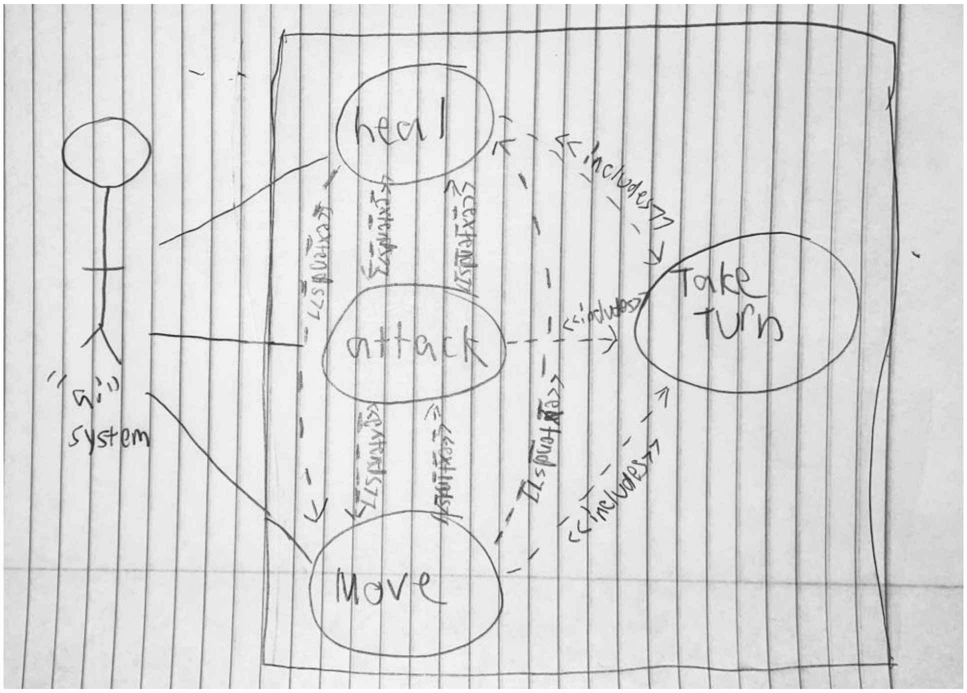
Name_____Damien Miller_________ Mark ______________________/50

1. Brief introduction __/3

For my feature in this game, I will be handling the enemy ai. It should have some form of pathfinding that allows it to determine the best move available to it, using location of itself, allies, and player units. It should also use its health and attack values as well as player units attack and health values to determine its next move, whether it be healing or attacking. There will also be special features for boss ai such as unique movement and attack patterns possibly even "smarter" ai.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Heal

Summary: the ai heals if it needs to

Actors: "ai" System

Preconditions: the value of the health of the unit

Basic sequence:

Step 1: check the health value if it is

Step 2: if health is under threshold, then heal.

Exceptions:

Step 2: if it is above threshhold there is no need to heal

Post conditions: heal and update health

Priority: 2*

ID: C01

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

Name: attack

Summary: the ai system attacks if able to

Actors: ai system

Preconditions: attack stats and health of opponent(player) units

Basic sequence:

- **Step 1:** check attack
- **Step 2:** check health of all player units in range
- **Step 3**: determine if it can kill any units
- **Step 4**: attack

Exceptions:

Step 3: if it can’t kill any units, it will attack the one with the least health/it can do the most damage to.

Post conditions: deal damage

Priority: 2*

ID: C02

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

Name: move

Summary: The ai system will move its units around the grid

Actors: ai system

Preconditions: location of all units/tiles in range

Basic sequence:

- **Step 1:** check units close by
- Step 2: check for special tiles
- Step 3: determine best space to move to
- Step 4: move to space

Exceptions:

Step 3: if it is currently on the best space, it won't move.

Post conditions:

Priority: 1*

ID: C03

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

Name: Take turn

Summary: The ai system will perform the actions it deems best for the turn

Actors: none

Preconditions: heal, attack, and move

Basic sequence:

Step 1: Heal

Step 2: move

Step 3: attack

Exceptions: none

Post conditions:

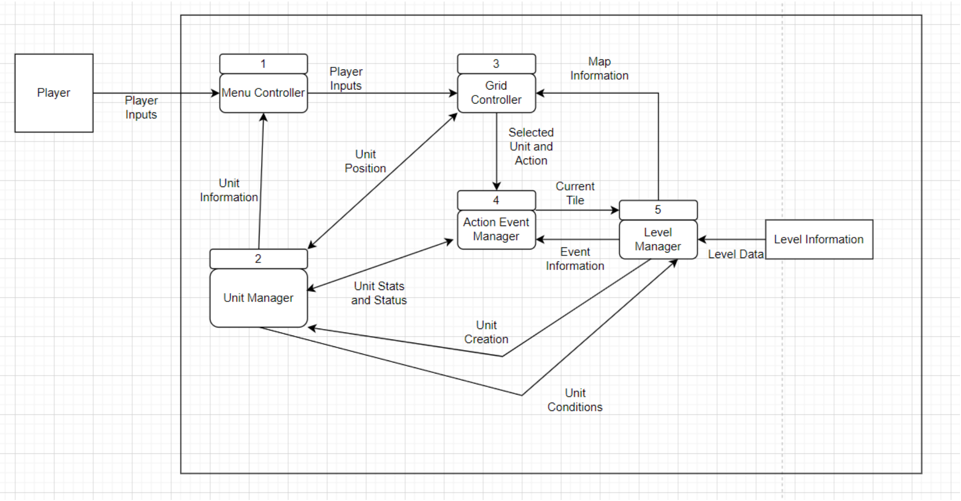
Priority: 1*

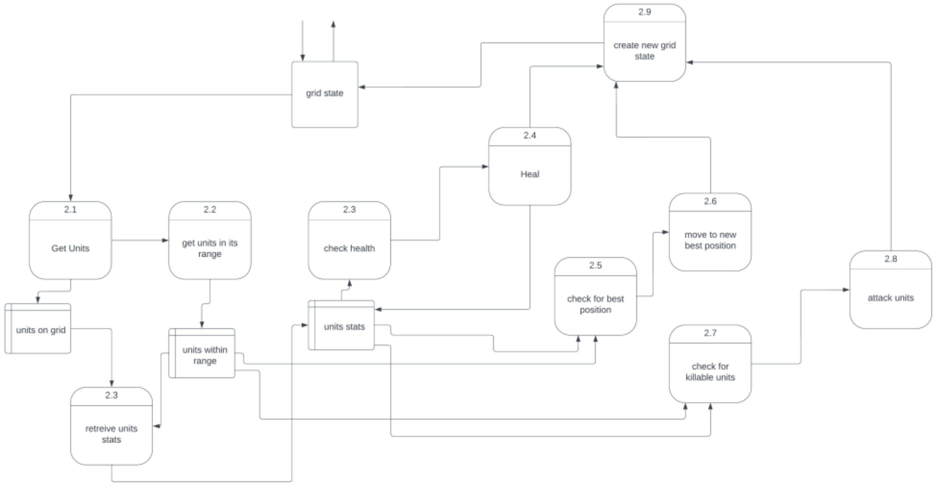
ID: C04

*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





Process Descriptions

```
Get units:
    WHILE units haven't been visited, visit and store them in file
    END WHILE
Retrieve unit's stats:
Takes in a unit and returns its stats.
get units in its range:
Removes all units that aren't in the currents units range. Returns all units left.
Check health:
If(health < threshold) { heal; );
Heal:
    Determine how it will heal(potions/tile), health ++
    Returns new health value
check for best position:
    Check if unit can move into attacking range of any player units
    Returns chosen location
Move:
    Move unit to new best position, if it is the same it stays put
    Returns new position
check for killable units:
Check units in range for any units it can kill this turn if there aren't any choose the one with the least health
Returns chosen unit
attack units
Deal damage to chosen unit.
create new grid state:
Update what has changed on the grid and return the new grid state
```

4. Acceptance Tests ______9

To run tests on my feature of the game I will check that my ai can move, heal, and attack, while still following the rules of the game. I think the best method to test it would be to play against the ai and debug any unwanted feature. For edge cases I will run a simulation of the ai playing against itself and flag any illegal moves

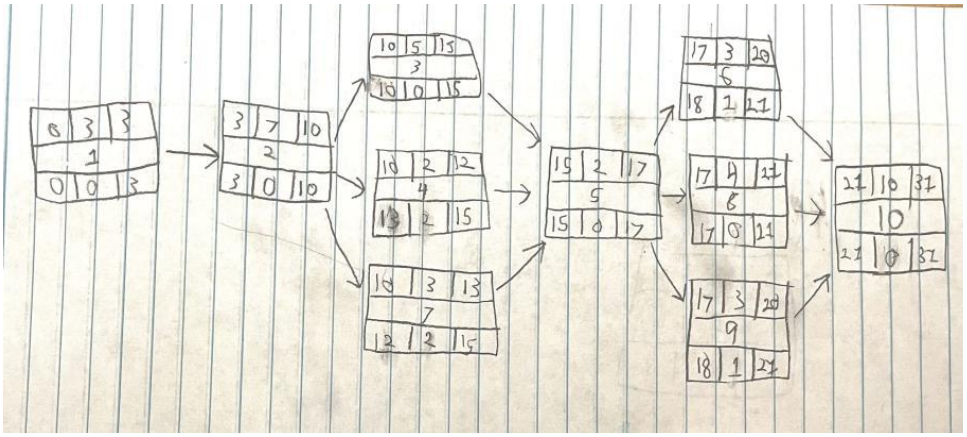
5. Timeline _______/10

Work items

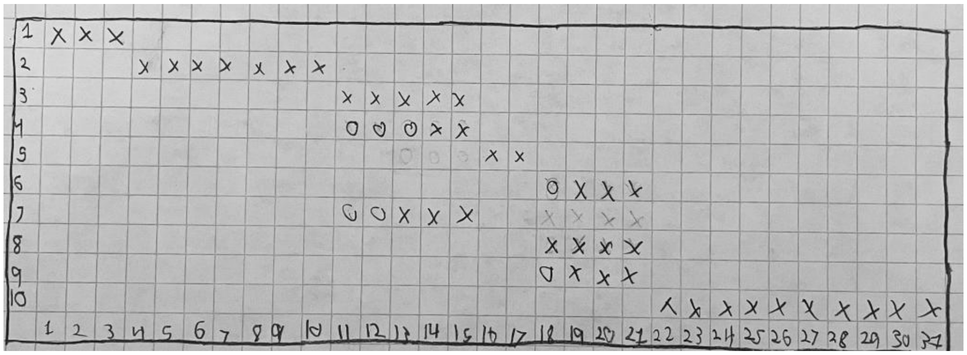
Task	Duration (PWks)	Predecessor Task(s)

1.General design	3	-
2.pathfinding (location, "vision", distance from player unit, etc.)	7	1
3.interaction with tiles	5	2
4.attacking the player	2	2
5.special ai "general setup"	2	2, 4
6.boss ai	3	2, 4, 5
7.healing	3	2
8.special movement patterns	4	2, 4, 5
9.Smarter attacks	3	2, 4, 5
10.Testing/debugging	10	1-9

Pert diagram



Gantt timeline



1	X	X	X																												
2				X	X	X	X	X	X	X																					
3											X	X	X	X	X																
4											O	O	O	X	X																
5													O	O	O	X	X														
6																		O	X	X	X										
7											O	O	X	X	X			X	X	X	X										
8																		X	X	X	X										
9																		O	X	X	X										
10																						X	X	X	X	X	X	X	X	X	X
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31