Game AI Brief

# Entities

## Player

### Behaviours

* KeyboardControlled – can be moved using the keyboard

### Rules

* On collision with walls and other defined obstacles – Move to last valid position

## Enemy

### Behaviours

* Seek – Seeks towards the player or base. If the player enters ‘agro’ range, seek to the player. However, if the player kites the enemy too close to the base, and the base enters agro range, prioritise the base unless the enemy is attacked by the player while the player is within agro range.
* Flee – If the enemy takes too much damage and the player is within agro range, will flee away from the player. However, if the player kites the enemy too close to the base, the enemy will additionally seek toward the base.
* Attack – When within attack distance, can perform an attack at a targeted entity (can only be accessed from seek).
* Wander – If it can’t see either the player or the base, will just wander around the map

### Rules

Wander -> Seek (if target enters agro range)

Seek -> Flee (if target in agro range && health < threshold)

Seek -> Attack (if currently seeking target is within attacking range)

Seek -> wander (if no target in range)

Attack -> Seek (if attacking and target leaves attack range)

Attack -> Flee (if health < threshold && target in agro range – perform another check in case the enemy has a ranged attack)

# TODO

1. Create “Base” class with sprite – no update logic needed
2. Create collection of base objects, update and draw all
3. Add map/background image to project
4. Create player that can be controlled via keyboard
5. Implement camera to follow player
   1. Use “Safe zones” so that player can move freely until they hit the “push” zone that moves the camera – Like Mario. You can move freely until you get close to the right-hand edge
6. Build collision map as discussed
7. If player intersects with collision map, move player to last safe position
8. Create base enemy class that inherits game object
   1. Create “melee” class that inherits from enemy
9. Create collection of enemy objects – update and draw all
10. Implement wander behaviour as default behaviour for enemies
11. Provide enemy with collection of “seekable” objects
    1. Seek towards “best” seekable
12. If enemy is within range of an “Attackable” object
    1. Attack
13. Provide health variable to attackable objects – somehow represent health
14. Enemy flee behaviour
    1. When health < threshold, flee from “aggressors”
    2. You could add events for objects (e.g. onWithinAttackRange)
    3. When enemy is away from aggressor
       1. Seek/find path to health pickups
15. Implement graph, create nodes in grid – except for those on collidable collision map
16. Implement collection of health pack objects that spawn
17. Enemy find path to health pack (see 14-C-i)