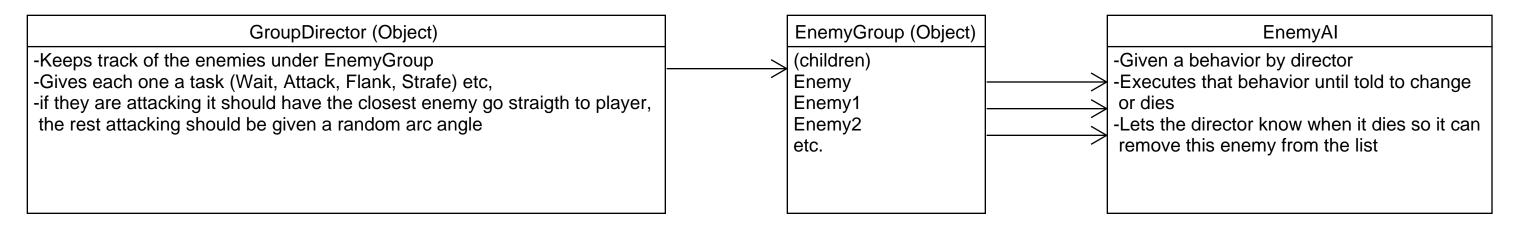
Overview



In-depth



Each enemy is told who the director is when they are added to the list so they can let the director know that they are dead

Basically the Director Uses EnemyGroup to Access a list of enemy gameobjects. EnemyGroup simply calls ChangeStatus with a status [Attacking,Waiting] etc.

If the if this UpdatedStatus is ever different from CurrentStatus, then it tries to set the enemy to that status. There may need to be an intermediate status, so that it can change to that new UpdatedStatus immidiately, instead of waiting for animations and things to finish, or not.

## EnemyGroup.cs Holds a List<EnemyAl> Enemies AddEnemy() RemoveEnemy(EnemyAl e) GetCount() WakeUpEnemies() set each enemy to active SetAttackers(int n){ this will be a good method for now to look for n closest enemies to player and make them attack} AllEnemiesWait() use enemy.ChangeStatus to set enemies to Wait, the enemy will handle itself AllEnemiesAttack() use ChangeStatus to set enemies to attac it allenemies to attac

## GroupDirector.cs

A Gameobject with a trigger collider that wakes up the enemies and closes exits

## Start(){

Gets a list of Enemies from the children of a given EnemyGroupOject and adds them to a list using EnemyGroup

## Update(){

right now, all the enemies we set to wait when the player triggered the collider

we wait 3 seconds to set the enemies to attack

it also checks if all enemies are dead, i.e. the list is empty so it can open the exits.