Enemies

29/7/17

For now, enemies spawn if there are fewer than 2 enemies on the screen. So when one dies, a new one appears.

There are some problems; the death detection only counts for the last made enemy. (line 302, if enemy.dead)

Enemies check if they’re alive themselves – this is not how it should work. If an enemy is dead, it shouldn’t know it’s dead, it should have already been deleted before it can check. It’s allowed a death function, but it doesn’t check its own death; something else triggers it.

Do we need an enemy manager? Or can we check, for all enemies, to take damage/kill?

# Collisions

|  |  |
| --- | --- |
| Class | Collides with |
| Player | Enemy |
| Bullet | Enemy |
| Enemy | Player |
| Enemy Bullet | Player, Bullet |
|  |  |

How about there’s a group for collision type – is this called collision groups? Basically, each class puts objects into a group for that object AND a group for collision types. Eg. Enemies collide with bullets and player – there’s a group called enemyColliders that contains players and bullets.

Or, make a collision function that takes the sprite input and a list of groups to check collisions for.

I think there needs to be a priority system. Bullets collide with the enemies, or enemies collide with the bullets? If you have code for both, you could end up with an exception if you kill the objects before trying to then reach some of their code. Also you are doing the same thing twice.

The best way to think about it is the attacker checks if it reached the target. It can then kill the other object – as I said, the target should not have to check for its own death. So each individual bullet checks for a collision with an enemy. Enemies check for collisions with the player. Player checks nothing.