

Enemies.EnemyTypes
EnemyDropKernal.cs
EnemyCollisionHandler - HandleObjectCollision
LozHelpers.cs - DropTable

KT's Review

Author: KT Goldstein
Date: 4/7/2021
Sprint #4
Files being reviewed: Listed above
Author: Alan

Enemies.EnemyTypes

Quality: In DeathAnimation.cs, I think the code for Alive = false and Health = 0 should go into the Die() method, and then when lifeSpan == 0 just call Die() to help with cohesion.

Readability: Variable names are good, all makes sense

EnemyDropKernal.cs

Quality: Smart idea to make an abstract kernel class, allows a single point of control for the rest of the enemy code

Readability: Overall I think it'd be a good idea to refactor some of the variable names to make them more descriptive, it could be harder to tell what some variables do otherwise (some examples: "r", "v", "something", "removed")

EnemyCollisionHandler - HandleObjectCollision

Quality: In the future it might make sense to refactor the code to have the enemy AI contained in the abstract class you created for sake of cohesion (rather than having the CollisionHandler handle the enemy AI as well), but that's not really important right now. Other than that, the ChangeDirection() method you created as a part of the abstract class helps the cohesion here, and everything else looks good to me.

Readability: I think it might help readability if you make the "or" logic in the if-statement expand down rather than to the right (not sure how to explain this better, but look at lines 20-22 in this file for an example of what I'm trying to say here. It just gets a bit hard to read when it expands out to the right sometimes).

LozHelpers.cs - DropTable

Quality: I'm wondering if there's an easier way to control the drop rates than just having them take up a certain percentage of a list, like perhaps putting the items and their drop rates in some sort of Dictionary/Map data structure. But if not, this is fine as is, and it's simple enough that it could still be easy to modify if we wanted to add more or different item drops.

Readability: It's a small list, so no problems here.

Summary

Number of minutes taken to complete review: 40 minutes

Overall comments: Good quality code, only common suggestion is having more descriptive variable names, even for intermediate/helper variables

Jaci's Review:

Author: Jaci Taylor

Date: 4/8/2021

Files Author: Alan

Enemies.EnemyTypes:

Everything looks good to me for quality and readability. The only thing I'd say is maybe consider changing the rest of the one line if statements to not include curly braces (this is a personal preference type thing so it's totally up to you).

EnemyDropKernal.cs:

I couldn't find this file, so I'm going to assume you were referring to EnemyKernal.cs! This one looked great! The only things I have are the one line if statement at line 39 being simplified to not take up as much space, and to maybe change the variable names r and v to be a bit more descriptive.

EnemyCollisionHandler.cs:

This mostly looks good! The only thing I'm curious about here is lines 49-51. I'm just curious if there is a way to utilize the collision detection methods to determine this information rather than having to do it again here. It's not a big deal, since it only takes up a couple of lines of code here, but I thought it'd be worth mentioning.

LoZHelpers.cs:

This looks good! The only thing I noticed is that the gold rupees are listed a lot and the flashing rupees aren't used at all. I think it'd be good to utilize the flashing rupees more since they are worth less than the gold and should spawn more than the gold.

Minutes taken: 30