

Collisions Folder

- PlayerCollisionHandler.cs
 - Lines 72-75
 - Lines 80-81

HUD Folder

- HUDMap.cs
- LinkOnMiniMap.cs
- MapSprite.cs
- MiniMap.cs
- TriForceOnMap.cs
- PauseMap.cs & PauseMapRoom.cs
 - Worked with with Simon
- LinkOnPauseMap.cs
- PauseMapRoomSprite.cs
- HUDTextureFactory.cs
 - Lines 28-32
 - Lines 36-62

Objects >> Doors Folder

- Added these lines to every file's (except IDoor) ChangeRoom() method:
 - *game.HUDMap.UpdateLinkMapLocation(Direction);*
 - *game.PauseMap.MoveRooms(Direction);*

PlayerInventory.cs

- Added two get set methods and two formal bool parameters to check for map and compass (lines 26-27, 35, 40-41)

GameStateMachine.cs

- HUDMap and PauseMap lines
 - A lot of this was refactored after we completed merging.

LoZHelpers.cs

- Lines 118-136 (Worked with simon)
- Lines 174-183
- Lines 218-221

Yonace's Code Review

Sprint 4

4/8/2021

Approx time 45 mins:

PlayerCollisionHandler, Objects >> Doors, PlayerInventory.cs, GameStateMachine.cs

Quality: Found no issue with the quality of the changes of these files. They are all minor changes that make sense. Like moving links location in the paused map when changing rooms.

Readability: Good code readability, did not find anything out of place.

LoZHelpers.cs

- Lines 118-136 ,Lines 174-183
- Quality: Quality looks good
- Readability: I don't understand the need for values like 0b0000(hexadecimal?) for pause room type values. Maybe too overly complicated?

LoZHelpers.cs, HUD Folder

- LinkOnMiniMap.cs - capitalize method moveLinkOnMiniMap, get rid of (this).linkMiniLocation or use
public LinkOnMiniMap(Vector2 linkMiniLocation, GameStateMachine game)
 - MapSprite.cs **unnneeded casting on row**
 - Quality: No issues, just needs a few minor changes listed above
 - Readability: Good readability
- Random thoughts/ Suggestions
- It kind of seems like a lot of the implementations are separated from each other. An example being HUD and PausedMap. Maybe having a class encapsulate everything might simplify some things?

Simon's Review

Sprint #4

4-8-21

Code Author: Kristin

Collisions/PlayerCollisionHandler.cs

Quality: Simple and coherent.

Readability: Clear and concise.

Refactoring: No suggestions.

HUD

■ HUDMap.cs

Quality: Very solid. Although there is some coupling & repeated code for hasMap and hasCompass assignment.

Readability: Very readable

Refactoring: Might make hasMap and hasCompass private properties like
`private bool hasMap { get { return game.Player.Inventory.HasMap; } }`
so there's a single point of control and we aren't constantly changing the values in Update()

■ LinkOnMiniMap.cs

Quality: Solid, although there are some unneeded components (GameStateMachine reference and Update() method)

Readability: Everything is clear except where the 2 comes from in moveLinkOnMiniMap (and the method name should be capitalized).

Refactoring: See TriForceOnMap.cs

■ MapSprite.cs

Quality: Only issues are in line 45 and line 49. The casting back and forth from int to float to int is redundant, removing all casting would give the same answer and be simpler. In line 49, I believe you should be using LoZHelpers.Scale instead of multiplying by 4.

Readability: Love the readability!

Refactoring: Just what's noted in Quality.

■ MiniMap.cs

I'm confused on what this class does. It seems like an unneeded layer of abstraction from the ISprite gotten from HUDTextureFactory.Instance.CreateMiniMap().

■ TriForceOnMap.cs

Quality: GameStateMachine reference is unused.

Readability: Straightforward

Refactoring: I feel that this and LinkOnMiniMap could be combined with just more variables passed in the constructor. Alternatively, this could be removed entirely since it's an extra layer of abstraction on HUDTextureFactory.Instance.CreateTriForceIndicator();

■ PauseMap.cs & PauseMapRoom.cs

Worked on it together.

Tempted to say we should just use

■ LinkOnPauseMap.cs

Quality: GameStateMachine is unused, moveLinkOnPauseMap() should be capitalized, Update() can be removed entirely

Readability: Easy to read

Refactoring: See Quality, otherwise no big ideas.

■ PauseMapRoomSprite.cs

This class is never used...

■ HUDTextureFactory.cs

Solid work, what I expect from a SpriteFactory. Might want to change magic numbers to be formulaic (if using an atlas), but we've gotta put the constants somewhere, so it's not an important suggestion.

LoZHelpers.cs

Quality: Occasionally you have some unscaled values mixed with scaled values, is there reasoning behind this?

Readability: Whenever you add a vectors X and Y value, it's cleaner to just add the vector as a whole (takes half the line space).

Refactoring: Nothing beyond what's mentioned in the other sections.

Closing

Time Spent:

Final Comments: The following files add abstraction that I believe add more confusion & difficulty than clarity

- LinkOnMiniMap.cs
 - You can just hold a vector2 location and ISprite in HUDMap
- TriForceOnMap.cs
 - Same as LinkOnMiniMap.cs, just move moveLinkOnMiniMap functionality to UpdateLinkMapLocation
- PauseMapRoomSprite.cs
 - This class is never used

Otherwise, there are a bunch of smaller value topics I brought up in the above sections.