

## Players Directory

- ILinkSprite.cs
- ILinkState.cs
- IPlayer.cs
- IPlayerProjectile.cs
- IUsableItem.cs
- LinkSpriteFactory.cs
- PlayerProjectileSpriteFactory.cs
- PlayerProjectileSprites.cs

## Main Directory

- ICommand.cs
  - Lines 26-216
- LoZHelpers.cs
- LegendOfZeldaDungeon.cs
  - Lines 235-245 (only non-standard thing in the file)

---

## Review

**Review Author : Alan Wu**

**Date : 2/21/2021**

**Sprint 2**

**Author : KT & Simon**

ILinkSprite.cs, ILinkState.cs, IPlayer.cs, IPlayerProjectile.cs, IUsableItem.cs

As discussed in the meetings, separate all class into their separate files

## Recap

### Hypothetical Change

I would consider registering commands (84 - 118) in LegendOfZeldaDungeon.cs somewhere other than the main file to reduce clutter and increase readability. I believe a way was suggested during our meetings so if it can be done painlessly I would recommend it.

### Summary

Interfaces and classes have good readability with very precise variable names.

### Time Spent

30 minutes

**Review Author : Kristin Bradshaw**

**Date : 2/21/2021**

**Sprint 2**

**Author : KT & Simon**

#### Code Quality

- Everything looks good quality-wise.
- Since you guys added the new LoZHelpers file, we can start utilizing that more to help get rid of magic numbers.
- All of your variable, class, and interface names help the reader know what's expected.
- Your files are broad enough to add and remove things without breaking a lot. Having one projectile file (**PlayerProjectileSprites.cs**) makes more sense than having a specific file for each enemy implemented. Seeing that made me think about how I can make my own implementation more broad to help with buildability.

#### Readability

- We talked about this in a prior meeting but having all the classes in the same file like in **ILinkState** and **ILinkSprite** makes it hard to comprehend the file as a whole if that makes sense. The files are packed, so breaking up those walls of code would be good.

#### **Recap**

#### Hypothetical changes

- The only thing I can think of has to deal with condensing the code. Alan said it in his review, but there are a lot of commands for Link and that's the bulk of our command section. If it can't be helped that's fine since it doesn't really hurt us that much.

#### Summary

- Overall looks really nice and buildable. Breaking up the files would help with readability and structure.

#### Question

- How did you guys split up the work so that each person had an equal part in the code creation? It was hard for us to figure out how to divide the work equally since we didn't really know what to expect for each task until we started working on it.

#### Time Spent

- Approx. 40 minutes