Collaboration with Simon/ Jaci
MovableBlock.cs
MovableBlockGoal.cs
GameStateMachine.cs 132-144
LegendOfZeldaDungeon.cs changes in InitializeRooms method
BlockCollisionHander.cs 54-62

Collaboration with Jaci
OrbSprite.cs
OrbSwitch.cs
BlockCollisionHander.cs 37-50

Minor Changes (probably not worth mentioning)

- Room.cs additional cases
- .csv room design changes

Simon’s Review
Sprint #5
4-21-2021
Code Author: Yonace

Blocks

- ### MovableBlock.cs & MovableBlockGoal.cs

I collaborated on these, so I don’t have much to add.

- ### OrbSprite.cs & OrbSwitch.cs

Quality: It seems the OrbSwitch is tightly bound to the values of the OrbSprite. I feel it’d be better to encapsulate ALL the sprite information within the OrbSprite.

Readability: No issues with the readability.

Refactoring: I’d remove the set from WasHit and replace it with a Hit() method. This method would change a private wasHit bool and Update the OrbSprite, causing it to switch the frame of animation it draws. This would prevent the OrbSwitch from having to know about the sprite sheet used by OrbSprite and make it so the OrbSwitch only needs a reference to an ISprite, not an OrbSprite.

Collisions/BlockCollisionHandler.cs

Only commenting on 37-50 since I helped with the other section.

To go along with the other suggestion, you’d just change line 43 to have

```
CurrentBlock is OrbSwitch orbSwitch
```

and lines 44-48 to

```
orbSwitch.Hit();
```

GameStateMachine.cs

Solid integration of OrbSwitch into checking if doors should be opened.

And kudos for the great idea of making multiple lists in InitializeRooms() for readability!

Time Spent: 20 minutes

Alan's Review
Sprint #5
4-21-2021
Code Author: Yonace

Blocks

- ### MovableBlock.cs & MovableBlockGoal.cs

  Quality - I'm assuming these are the blocks associated with the puzzle room. I would just make sure that they can't be pushed to the perimeter of the room just so the game doesn't soft lock.
  Readability - Looks good to me.

- ### OrbSprite.cs & OrbSwitch.cs

Quality: Quality looks good.
Readability: Readability looks good.

Collisions/BlockCollisionHandler.cs

Quality - Just standard collision detection and handling so nothing to really change.
Readability - Everything is concise so readability is good.

GameStateMachine.cs

Quality - Quality looks good.
Readability - Separation of rooms into multiple lists makes code easy to read and follow.

Time Spent: 20 minutes