# CSE3209Project Sprint 2

Authors Aidan Haskett, Hemil Amin, Abhi Malisetty, Matthew Landwehr, and Johnathan Browning

README Code Description

Our code currently hits all Sprint 2 specifications. The arrow and "WASD" keys control Link and change his direction while the "Z" and "N" keys cause him to swing his sword. Each number key in the top row will cause Link to use a specific item, and the "E" key will damage Link. Blocks and obstacles appear fixed on screen and can be switched with the "T" and "Y" keys. An item pickup is shown on screen, and the displayed item can be changed by pressing the "U" and "I" keys. Items are also animated. Enemies also appear on screen and are animated. Enemies can be swapped using the "O" and "P" keys. Finally, "Q" is used to quit, and "R" will reset the screen to the initial loaded version of the game. Our code is organized with interfaces and then underlying classes which implement said interfaces. We have created interfaces for commands, controllers, game objects, and sprites. Beneath those we have separate classes for individual link, items, obstacles, enemies, and keyboard controls. The Link.cs class is under the PlayerStuff folder and controls all of Link's movements and animation. In the future, we may need to resize some sprites and edit the spritesheets to make the animations look cleaner. All enemy classes are in the EnemySpriteClasses folder and handle each individual’s animation and location data. The same is true for obstacles and items which fall under the Environment and Items folders respectively. All keyboard controls are housed under the CommandStuff folder and call methods to execute each needed function. These folder names are mostly temporary and may be subject to change.

Known Issues and Roadmap

The code for NPCs is not currently implemented in an ideal way in terms of organization. The current behavior has been hardcoded in the sprite classes itself and moving forward the plan is to redesign the way that NPC sprites are implemented and make better use of the INPC interface to achieve this.

Code Analysis

Sprint Reflection

The team performed well this sprint, however there is definite room for improvement. We did a good job of meeting and organizing our self at the start of the sprint by creating a discord server to communicate, a GitHub repository for our code, as well as a Trello board to manage tasks and decide who would be working on what. After that everyone mostly worked on their own to complete their tasks and add it to the GitHub. I think we could have done a better job of communicating when we each got our tasks done in order to start combining them sooner. We waited until the last weekend to start combining all of our code which was fine, but we definitely could have done so sooner which would have relieved a little bit of stress and have been a better use of the 3 weeks allotted for sprint 2. We noticed we are a lot more productive when all 5 members of the group can meet all at one time. So, for future sprints we are going to make a defined schedule for meeting and working instead of just seeing when everyone is free day of.