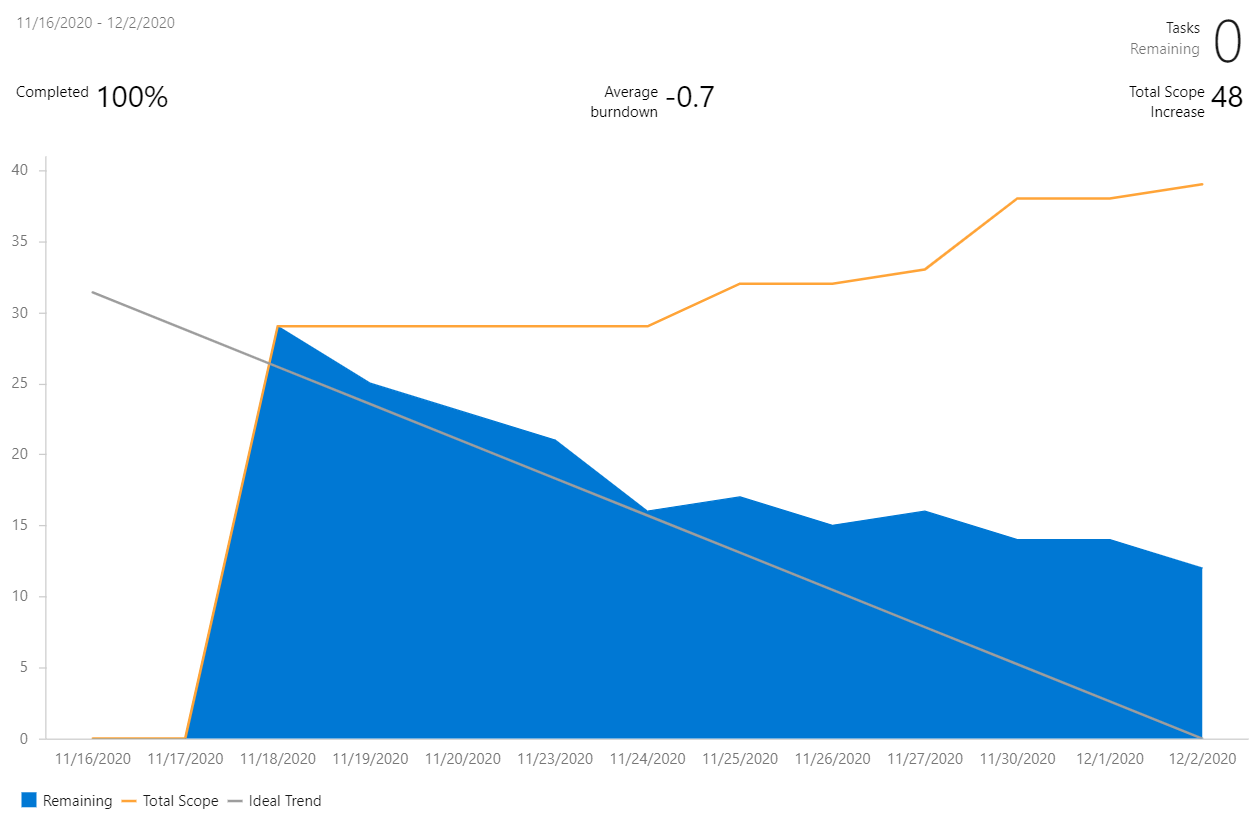
**Sprint 5 Reflection**

**Team Number: 5**

Hunter Figgs, Jared Perkins, Sergei Fedulov, Patrick Haughn, Jeffrey Gaydos

CSE 3902, Professor Boggus

2 December 2020

****

Summary:

We began Sprint 5 by brainstorming new features to add to our game. With the help of the CSE 3902 website we came up with a list of a bunch of ideas including coding an AI to speed run our game, adding more memes, and adding multiple resolution settings. We ended up settling with adding portals, adding additional difficulties and adding local multiplayer functionality. However, before we began implementing these new features, we went back and fixed several bugs we have had in our code from previous sprints. This gave us a much better base to add our new features onto. In addition, we believe our overall code quality also improved as a result of us prioritizing refactoring and bug fixes. We are very pleased with our new features. The multiplayer functionality allows for two players to play different characters at the same time using different sections of the keyboard. If you don’t have a friend nearby then you can opt for the second player to be an AI which follows the player around and hunts monsters on its own. If the original dungeon was too easy for you, then we added two more difficulties for you to put your skills to the test. Last but not least, the portal gun gave us the ability to create as set of Portal-like puzzles outside of the dungeon.

Planning and Documentation:

Our very first meeting for Sprint 5 was mostly a brainstorming session for new ideas. Throughout the semester we had already been brainstorming ideas like multiplayer, so it did not take us too long to come up with a good-sized list. After narrowing down our options we settled on adding portals, adding additional difficulties, adding local multiplayer functionality, and adding an AI player. We felt like these features would be a good amount of work for a group of 5, especially with Thanksgiving break occurring during the middle of the sprint. Like we mentioned in our last reflection, we also decided to start with polishing tasks, like fixing bugs, refactoring classes, and making minor changes to make the game feel more like the real thing. This provided a good base for our new features to be built from and allowed us to stay more focused during the duration of the sprint.

The Burndown chart shown above is fairly inaccurate, as for this sprint we used it partially to brainstorm and get all of our ideas out. The steep drop in tasks at the end correlates not only to work on the final day, but to deleting features that were ideated, but never implemented fully. Additionally, certain bugs related to polishing the functionality of the game were dropped in favor of fixing more important features of the game.

Implementation:

The new features we added in this sprint required the refactoring of many of our previous features. For example, in order to add multiplayer functionality, we had to store IPlayer’s in a list rather than just a single object. This meant that we had to refactor every single class that used Screen’s IPlayer property. In addition, we had to create a new HUD which displays the health and inventory of both players. Also, we decided to add prompts for when a player wants to leave a room so that their partner does not get left behind. The AI player 2 required the implementation of a new utility which calculates the best path to take to either follow player 1 or track an enemy. This utility was also used in order to make certain enemies more difficult for the harder difficulties. The new difficulties that were add were a Medium and a Hard difficulty. The Easy difficulty is the base game. Medium difficulty simply adds more enemies into the room making certain puzzles harder to complete in addition to making it harder to survive. The Hard difficulty not only adds more enemies but also adds harder enemies like skeletons that shoot fireballs or use a pathfinding to track the player down more efficiently. Other enemies have increased health and movement speed. The portal guns required the addition of several more sprites as well as several new environmental blocks, player states and collision handling commands. With the addition of the portals are some awesome new puzzles showcasing the portal gun located to the right of the spawn area.

Code Quality:

The polishing and refactoring we did at the beginning of our sprint had a large positive effect on our code quality. In addition, we continued to use state logic, the command pattern, utility classes and interfaces in order to improve the maintainability and readability of our code. To further improve our code quality from the last sprint, we also refactored audio in order to shorten the class up and to make it more readable and maintainable.