Python Programming

Python Final Project

University of Cincinnati

**Magiclash**

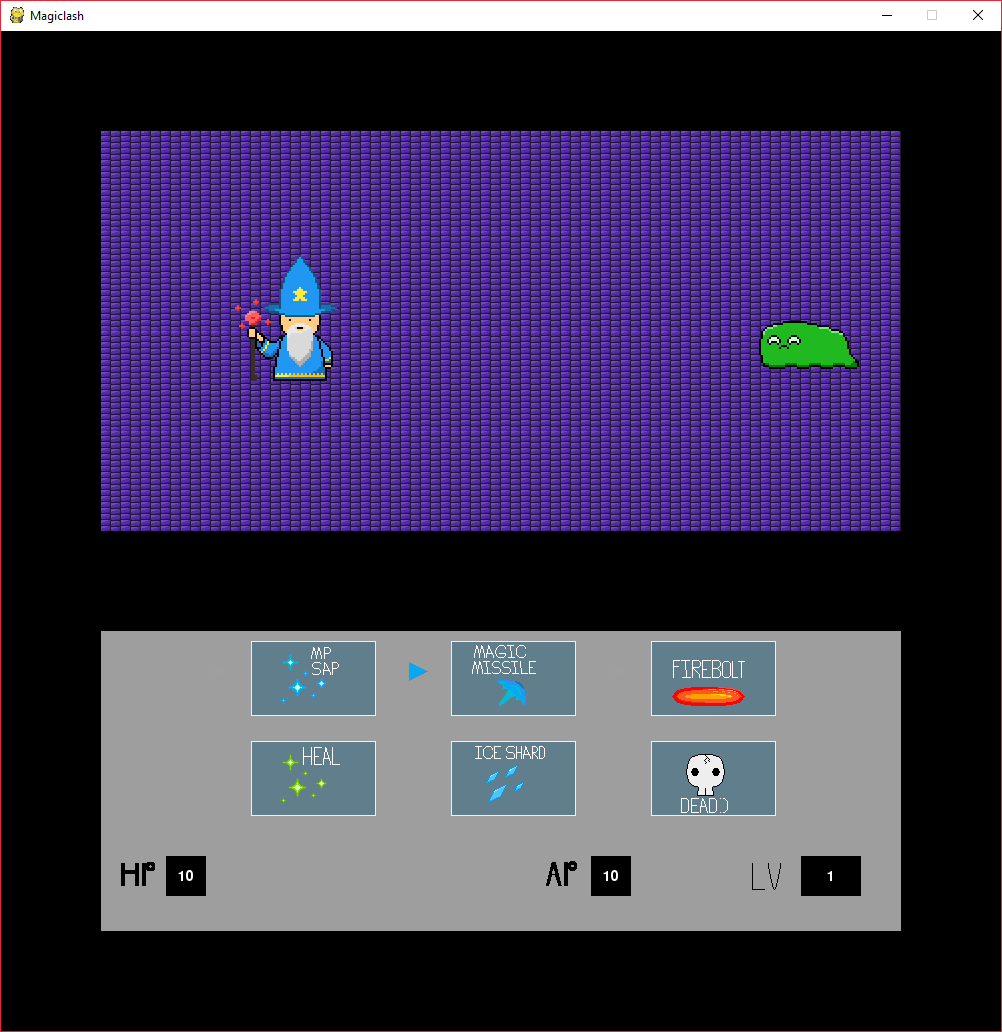
A basic wizardly RPG by Chris Butts, Raymond Gee, and Colt Wells

Magiclash is a basic RPG made with pygame where you play as a wizard and go through 5 floors of an evil wizard’s tower in hopes of reaching the top and defeating him. As you progress upward, there will be enemies that you have to battle to be able to continue the journey upward. The game is divided into two main components: ‘exploration’ and combat. For exploration, the player can move the wizard around the floor map and when they meet an enemy combat begins and the display is switched to the combat layout. For combat, the wizard has an assortment of spells available to himself as he levels up. Starting the game, the wizard starts at level 1 and starts with 2 skills (MP sap and magic missile), and every level up to level 5 the wizard gets another spell at his disposal. The spells are as follow:

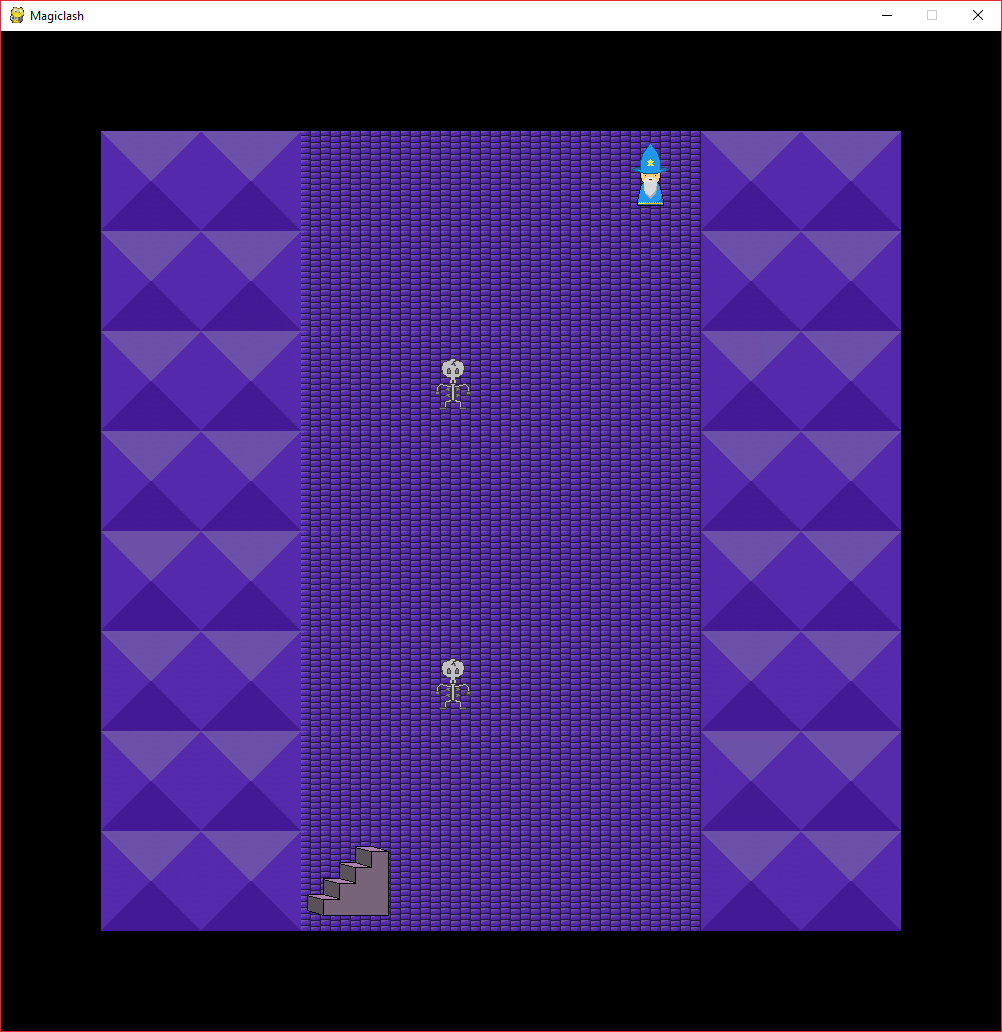
* MP sap (Level 1) – Regenerates some MP to the wizard
* Magic Missile (Level 1) – A basic magic attack, every wizard’s bread and butter
* Firebolt (Level 2) – A small burst of fire aimed at the enemy
* Healing (Level 3) – A curative spell that heals the wizard
* Ice Shard (Level 4) – A multi hitting barrage of ice shards aimed at the enemy. The skill can hit between 1 and 5 times.
* Death :) (Level 5) – The wizard’s most powerful skill. It immediately kills any enemy in his path

Combat works normally like it has in any other game in the sense that you win when the enemy’s HP reaches 0 and you lose when the wizard’s HP reaches 0. When you finish a combat, the wizard gains unseen experience points that when reaching a certain threshold, level the wizard up and fully heals his HP and MP. As well, after every combat the wizard heals a little bit of HP and MP.

As the progress progressed, we realized that our original ambition was far too much for us to be able to finish and implement correctly in a timely manner. Our original ambition was to have 10 floors in the wizard’s tower with multiple classes to choose from and a system where enemies would potentially drop spell cards which could be used to learn different spells alongside a shop system. While this could have probably been accomplished, with the time provided and our late-ish starting time on the project as well as our inexperience with using pygame we decided to make something basic. In terms of technical pitfalls, whenever we play the game the later, we get the longer it takes for the program to accept an action - in combat specifically. It’s really strange and we’re unsure why it happens because it doesn’t seem to happen consistently in any one spot – what typically happens is it begins to start ‘not responding’ at some point and when it does whenever action is taken whether that be from the enemy slide or the player side it will ‘not respond’ but the code will execute like normal it seems. It is very strange. Finally, the only thing I *really wish* we could’ve figured out was how to separate our code into different files and to import them. We looked up how to separate code into files and import files and folders but none of it would work for us so that’s why we ended up putting everything in one oversized .py file.

On the brighter side of things however, we were able to properly integrate a if checks to be able to save the current layout of the map so that previously defeated enemies wouldn’t show up after finishing a combat. As well, the implementation of a function to allow for custom output during combat and variables so that we could display the wizard’s HP, MP, and level alongside flavor text after using a spell or being attacked. Making all of the sprites was also a lot of fun since Chris really enjoys making pixel art and had to have fun in creating different enemy types, map tiling, and character designs.

Combat window against a slime

Map layout of floor 2

Chris Butts

I was in charge of designing all the sprites and enemies and integrating the map / overworld layout. When approaching this, the first thing I did was design all the maps on a sheet of paper and then design a for loop for printing out all of the different floors without having to code them all individually every time. I also handled some minor combat balancing, since the original numbers we were using made the game extremely difficult to actually…beat. As a minor thing, I also picked out the music used for the overworld as well as in combat. I also wrote up this report and got everything collected together to be turned in.

Raymond Gee

I was in charge of creating the skills and mobs class. This required a class to be created and the members associated with each class was thought up based on what other games would usually have. For the skills, it had attributes for a name, attack, and mana cost. Originally it would have another attribute for hit chance, which would be used for determining whether the players skill would hit, but that was omitted. For the mobs it would have a name, health, experience, and attack. These would be the basis for a mob and would be used during combat. Also, I assisted with creating the combat within the game. We created the combat GUI by allowing the user to select which skill to use with arrow keys. The arrow keys movement had to be restricted to only go to six different spots since there were only six options. The user would press F to use the skill and the spell would be used to determine how much mana was consumed and how much damage was inflicted. The user’s health and mana were displayed within the combat GUI so that the player could keep track of his character’s statistics.

Colt Wells

I was in charge of creating the combat GUI and making it work properly when integrated into the overworld GUI. Originally, I wanted to design the layout to work with pressing the mouse on the appropriate button to use a magic attack, but as time progressed I concluded that using the arrow keys and utilizing an arrow was much easier to implement. The combat function was made to take in the player as well as whatever mob the user was about to start combat with, so that everything could be outputted onto the screen properly. After taking Raymond’s spells class, we implemented it directly into the combat GUI and defined all the spells accordingly and made thresholds for what level was required for the player to use them. Coming into this project, I was nervous about the division of work since groups can end up being one person doing a majority of the work and I wanted to contribute in a meaningful way to the project, so I’m glad I was in charge of creating combat on my own with assistance from Raymond as necessary.

Bibliography:

This project was made utilizing the library pygame. For the most part we utilized the pygame docs section of the website to figure out what functions to use, but for things we couldn’t figure out we referenced stackOverflow.

References used:

<https://www.pygame.org/docs/> (Used for getting a basis of what you could do with pygame and learning what functions we had available to ourselves)

<https://stackoverflow.com/questions/29640685/how-do-i-detect-collision-in-pygame> (Used for figuring out how to make collision detection work properly between entities)

<https://stackoverflow.com/questions/20842801/how-to-display-text-in-pygame> (Used for getting an idea of how to output text to the display window. Colt also referenced Spenser for some help in figuring out how to implement it)