TP 1 - Project Proposal

**Project Description:**

My 112 term project is called “Kitchen Nightmare” (not to be confused with the show ‘Kitchen Nightmares’ with Gordon Ramsey). You are a chef trying to survive rogue food monsters trying to attack you. The game is played in a 2-D, top-down style and features increasingly difficult levels with boss stages, randomly generated obstacles/maps, and power-ups. Players can control the chef to dodge incoming attacks and fight for as long as they can survive. Scoring is based off how many levels / rooms / bosses a player can beat. The game can also be played multiplayer using Sockets!

**Competitive Analysis:**

This game structure has based heavily off of a mobile game called ‘Soul Knight’. ‘Soul Knight’ is a 2-D, top-down style game in which you choose a hero class (knight, mage, etc..) and make your way through a dungeon fighting monster/bosses, collecting several weapons, using powerups, and gaining experience to level up your character. You can also play multiplayer across up to 4 phones. My game will attempt to be stylistically similar to ‘Soul Knight’. I am borrowing the 2-D/top-down style, the dungeon-level gameplay and boss level style. My game will be unique in that it will be infinitely long, making the goal to survive for as long as possible, and it more simplified with different themes and only 1 playable character.

Another game that is like mine is one called ‘Realm of the Mad God” (RoMG for short). RoMG is similarly 2-D and top-down, but it is free world and an MMO (Mass-Muliplayer-Online) game. There are optional dungeon missions that players can go through to collect items and points. Most of the game, however, takes place in an endless world filled with increasingly difficult monsters the closer one gets to the ‘center’. All players on a server (up to 100) are fighting on the same world and can work together to kill monsters and bosses. I was able to take many ideas from this game and implement them in my own. The way enemies move, spawn, and attack the players was a lot more relevant to my style of game than that of ‘Soul Knight’s. My game, however, can be played single player and does not have the infinite world aspect. RoMG also has items, armor, and spells, which I am not incorporating in my game. Instead I am implementing my own design of powerups and permanent boosts for beating bosses.

Links to videos of ‘Soul Knight’ and RoMG:

‘Soul Knight’🡪 <https://youtu.be/kUZ-6v-kiLo>

RoMG🡪 <https://youtu.be/_PD45s8Amzo>

**Structural Plan:**

Since I plan to use Pygame, structuring my code using OOP will greatly benefit me. Pygame allows for classes to be defined as sprites which gives them collision detection and several other useful features. I plan to create classes for the player, enemies, projectiles, walls, and bosses. This allows me to create files for each of them and import the several class files into a main file where I control their movements and reactions to collision or user interaction. Since the map must also react to player and enemy collision, I play to create walls as sprites thus easing collision detection with players, enemies, and projectiles. I am also using images for character animation which will be stored in a folder. I have separate files that also manage player animation, splicing of character sprite sheets, and map creation.

**Algorithmic Plan:**

I believe the trickiest part of this game will be creating randomly generated maps with randomly generated obstacles. Another tricky aspect is having the entire map restart after the player defeats the boss and goes deeper into the kitchen-dungeon. I know that each room of the map will be the same size and have walls surrounding them and an entrance from the West and East. Knowing this, I can make an empty room template using a 2-D list of variables representing a wall or ground. I can then create a function that takes an empty room template and, using several pre-determined patterns, place obstacles down in a random pattern at a random position, thus giving users a unique layout every room. With this, I plan on having players start in an empty room as the first room in each level, then when they progress to the side of the screen, they are transported to a new room with monsters and obstacles. Players cannot move to another room while there are still monsters to be defeated in the current room. I will also generate the new room as the player moves to the side of the screen. Players can also move back to previous rooms once they have completed them. I plan to store the created rooms in a list that will be cleared once a player advances a level. I will store variables of which room a player is in and which level they are at so enemy health and power as well as the map can be adjusted accordingly.

**Timeline Plan**:

11/20 (TP1)🡪

I plan to have a prototype dungeon room with working wall collision. There will also be randomly generated obstacles and enemies that respond to player movement and attack players. I also plan on creating a basic menu

During Thanksgiving Break 🡪

I will create more enemies with different attacks and health. I am also planning on expanding the map to including multiple rooms/levels. I will also be refining the menu and creating an instruction page, a pause menu, and instruction page.

11/28 (TP2/MVP) 🡪

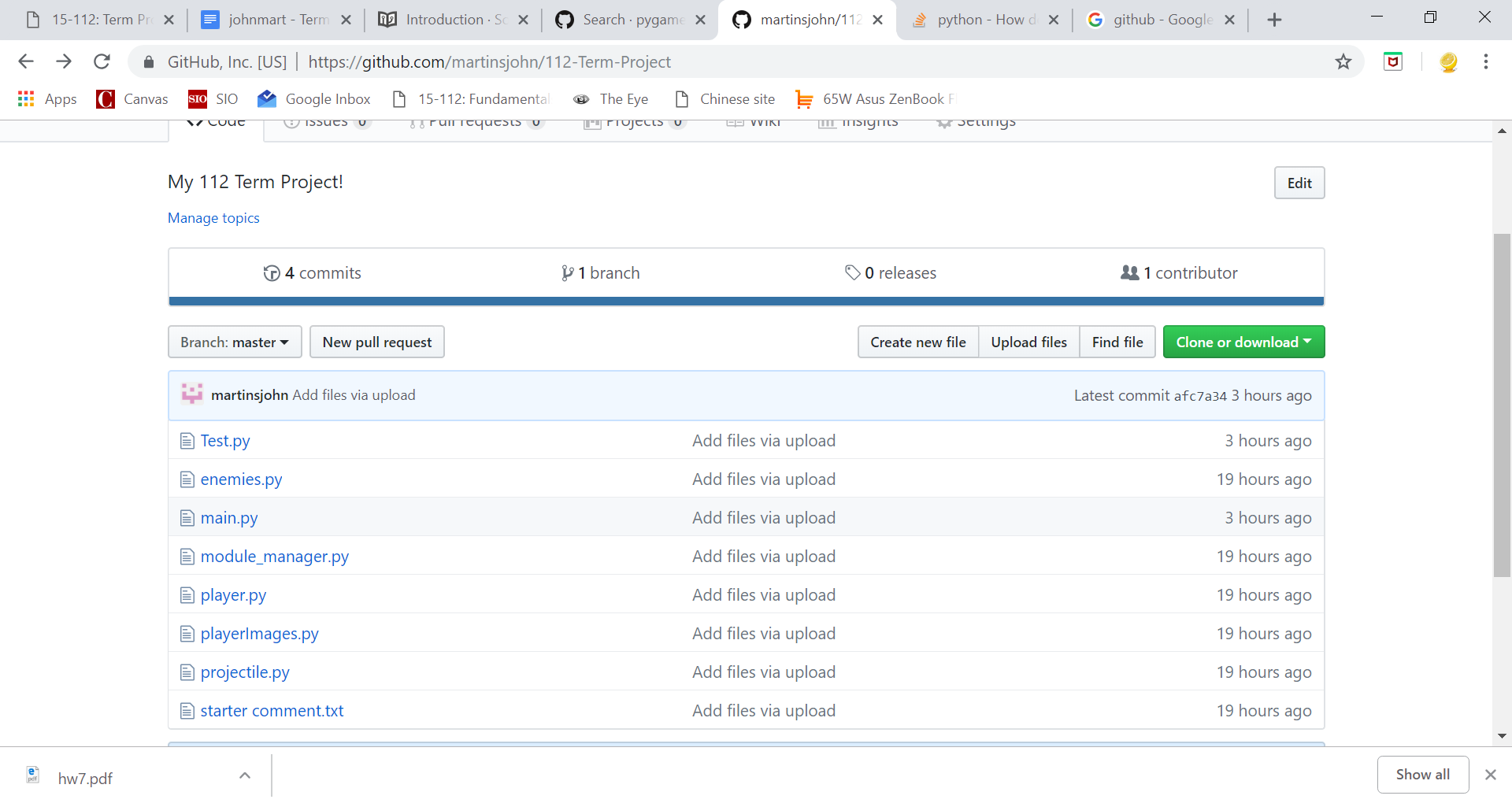
I plan on having the maps/levels working properly with enemies getting increasingly difficult. I also plan on having powerups, and more advanced enemy movement (incorporating AI features).

12/6 (TP3)🡪

I plan on having multiple bosses that have several features and attacks. I also plan on refining gameplay style to incorporate more player weapons/attacks. I will also incorporate multiplayer into my game using Sockets

**Version Control Plan:**

I am using GitHub as version control. I am saving my work regularly and whenever I make moderate or large changes, I upload/update the files to GitHub.



**Module List:**

I am using Pygame for my project. I have also switched IDEs to PyCharm since it is more organized for project management. I am currently planning on using Sockets for a multiplayer experience.