Madeleine Hagar

15-112 TP1 Proposal

11/30/2020

Updated: 12/05/2020

**Project Proposal:**

My term project will be a tower-defense game consisting of several levels in which the user must build towers and control soldiers to defend against waves of monsters who are attempting to attack the “home base.” There are four different types of towers: Arrows which shoot arrows at enemies, Magic which shoot defensive spells that injure monsters, Explosives which launch bombs, and Barracks which allows users to place soldiers to fight monsters within range. There are also several types of Monsters: Goblins, Ogres, Spiders, Wolves, and Dark Knights. Each monster has a different speed, damage on soldiers, amount of armor, and health level. The user also has a Hero, a soldier with special abilities and extra health, that can be placed anywhere on the path and is meant to be used as the home base’s last line of defense. The user has 10 lives per level and loses a life every time an enemy gets into the home base; if the user runs out of lives, they lose the level.

My project is very heavily inspired by the games Kingdom Rush and Plants vs. Zombies. Like both of those, I have created different types of enemies—goblins, ogres, spiders, etc.—much like the enemies in Kingdom Rush and the different types of zombies in Plants vs. Zombies, which have different attributes like armor, damage, and speed. Much like Kingdom Rush, I have different types of towers that launch different attacks, and the enemies themselves are not attacking the towers but instead trying to reach the homebase. This is dissimilar to Plants vs. Zombies however, since the zombies come into direct contact with the plants that are attacking.

My game is also quite similar to 2018 112 student Alex Isparyan’s term project—112 Tower Defense!—where enemies with different health values travel along a path towards a “home base” while the user launches attacks. However, in my game, the path the enemies must travel along is set by the level, whereas in theirs the user can design the path. I chose to have a set path, as I wanted to have different levels with different challenges, and for the user’s success to rely heavily on the placement of towers, which I feel makes the gameplay more strategic rather than creative.

I will have one file for my custom classes, one file that controls game play, and possibly separate files for each level that specify certain aspects that are unique to that level (path, tower placement, types of enemies that attack, amount of money given to the user at start of level).

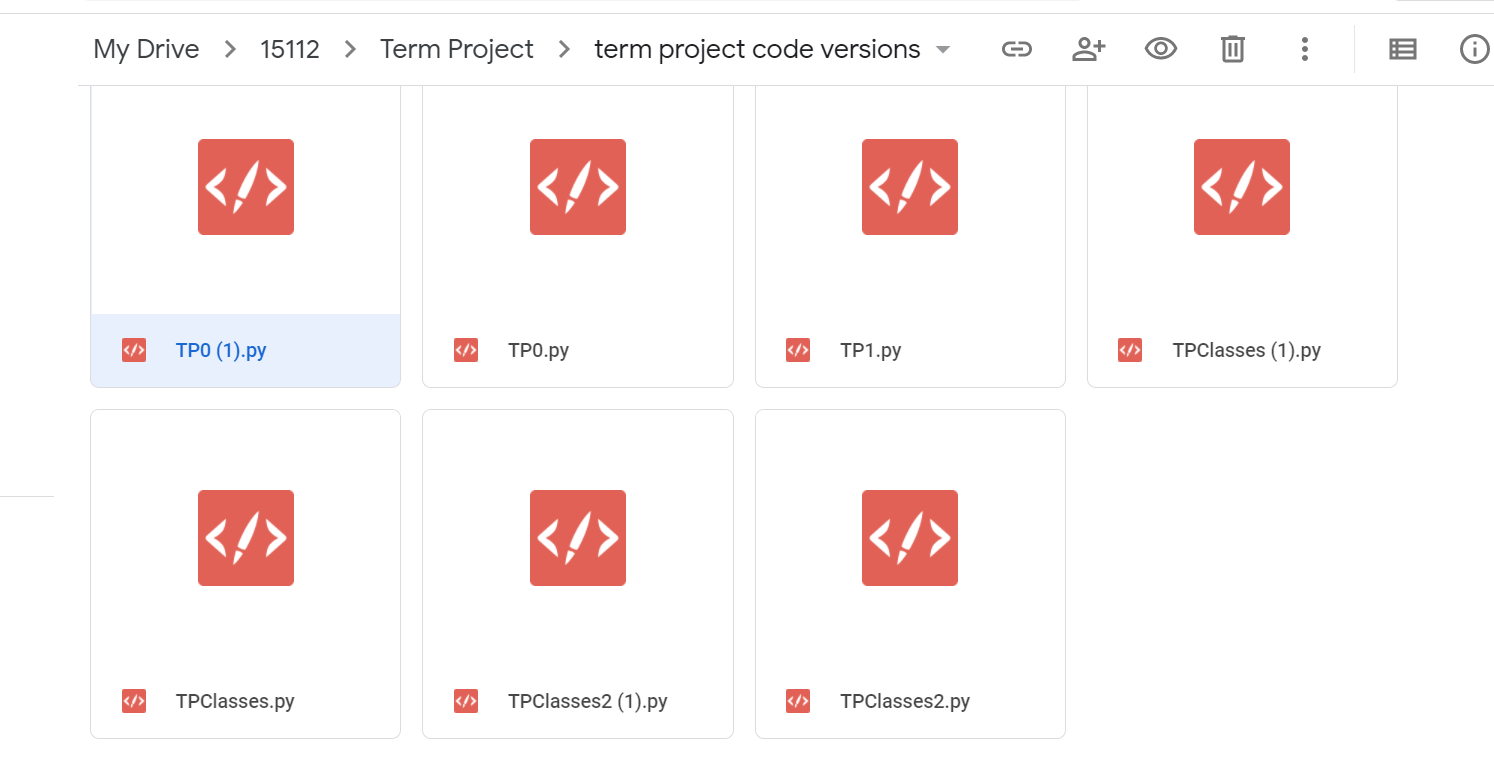
The trickiest parts of my project will be giving the towers the ability to track and attack enemies when in range, and implementing my graph algorithm, which allows enemies to choose the “path of least resistance” (the path to homebase with least active towers and soldiers).

I plan on approaching the “tower attack” by having a monster class method (inRange(tower)) that returns True when it is in range of any tower along the path. I will then have a tower class method (attack(target)) that attacks a certain target, whether that means shooting an arrow, sending soldiers to battle, etc. The attack method will cause that tower’s given weapon to move along a path towards the target and inflict damage on the first monster it hits; if it does not hit any monsters, the tower will continue to attack. This is especially tricky because the arrow, magic and explosives towers must select targets that do not move, since its weapons, once launched, have no intelligence and must only follow the path they are given. Therefore I plan on having a weaponPath() method that determines a straight-line path (not reliant on current position of monster) from tower to target for the weapon to move along.

Dissimilarly, the soldiers and monsters *can* track each other. So soldiers must determine their own paths to monsters, while monsters attempt to avoid soldiers and active towers. This is done with the graph algorithm which will give a resistance value to each possible path based on the towers the user builds, the placement of the soldiers, and the other enemies already on that path. The enemies will then choose to take the path with the lowest resistance value. However, once an enemy chooses a path, it must stay on it, so the movement of soldiers only affects the monsters who have yet to decide between paths.

I plan on designing all my custom classes and having a sample animation for my game working by TP1, so that I can visualize the monsters moving along the path and the towers “attacking” (the attack methods will be designed but not necessarily functional at this point). By TP2, I must have a working demo, as noted in the course notes, so I will have at least 1 level completely working, ready to play (and all my MVP requirements met). By TP3, I hope to have several more levels, and possibly have more advanced graphical features. For example, I might have arrows and bombs follow parabolic instead of straight paths to look more realistic to users. I may further animate the monsters and soldiers, so they look more realistic while moving. I also hope to add features such as power-ups the user can buy with coins earned while playing and more options for tower level-ups (for example the user could choose if their level 4 arrow towers shoot rifles or spears).

I have been uploading my code to a folder in my google drive whenever I make significant changes. This way, if I lose code, everything major will be saved, and it will be accessible from any device. This also allows me to look back at previous versions if need be.



I am using no external modules.

**TP2 Update:**

I finalized my graph algorithm concept so that there are certain “nodes” on the path; these nodes occur when two paths converge to one point. Before a monster takes one of these paths that lead to a node, I call a function bestPathToNode that predicts each step the monster would take along the path and calculates a “path weight” based on the possible damage that could be done to the monster. For instance, if the monster will take 20 steps in range of an arrow tower while on a path to node 1, the pathweight for that monster would be 20 times the arrow’s damage level. Similarly, if the monster is in range of a Barracks tower, the weight will be increased by the average of each soldier’s damage and health. Since the monsters start with different heights along the path, one monster might be in range of a tower while another will not be therefore the pathweight can be different for each monster, which is why some monsters may follow different paths. This algorithm is something that is not incorporated into Kingdom Rush, which I took a lot of inspiration from, and actually has the potential to make the game a lot more difficult for the user.

I also have given my soldiers the ability to detect and attack monsters in range! They simultaneously subtract their damage from the other’s health, and whoever loses all their health first “dies” :( . The hero fights similarly to the soldiers, but has a higher range, higher health, greater damage, can be moved anywhere on the path (not bounded by towers) and can fight two enemies at once.

I now have incorporated “waves” of monsters into the game, so you can keep playing and playing after some of the monsters either all die or leave the screen. These waves start with the easiest enemies, goblins, and progress to the hardest monsters, the dark knights, which have high levels of damage and armor.

I also improved visual aspects of the game since TP1! I downloaded images from the game Kingdom Rush and redrew them to fit my game; I had to scale them down quite a bit though, so they look a bit lower definition in the game than they did in my drawings.

**TP 3 Update:**

For TP3, I added some new smaller features to improve the user experience, and also worked on making my code less repetitive and more efficient. I added a highscore feature, which saves the user’s top score in a separate text file so that they keep their same high score even after closing the app. I also created a levelUp method for each tower type, that allows users to upgrade existing towers to have higher range, higher damage, and other things like extra soldiers! I have also given my soldiers and my hero the ability to resurrect with full health after they are dead for a certain amount of time. Another feature I improved was the different difficulties; before the difference in difficulty was a matter of how many enemies came in each wave. Now however, increased difficulty also means increased monster damage and speed! I also added an instructions screen, that gives the user a guide on what key and mouse presses to use in order to play. Lastly, I fixed my tower-shooting and hero-fighting bugs from TP2.