Project Description

* Rogue-like game, with procedurally generated levels. Character tries to survive for as long as possible while exploring dungeons that change according to the player’s playstyle.

Game Controls

* Click to shoot
* Hold shift and drag cursor in a circle to cast spell
* After all enemies are dead, the spawn point will turn pink, standing on the spawn point for more than 2 seconds will allow you to move onto the next level.

Competitive Analysis

* Rogue-like games are extremely popular in the current market and while my project borrows certain ideas from existing games, my rogue-like has the added feature of procedurally generated levels that guarantees interconnectivity in individual levels and also levels that change according to the player’s movements in past levels. E.g. players that tend to explore more will create levels that have more branches, while players that kill more mobs create rooms with more open spaces which allows for more mobs and bosses to spawn.

Structural Plan

* The main game will be placed in the file named “TP”, while all other files, which will mostly be code for different classes and sprites will be placed in the same directory. The contents of each file are quite self-explanatory, e.g. “Map Creator” is in charge of creating the map and “Player” stores the information for the player class.

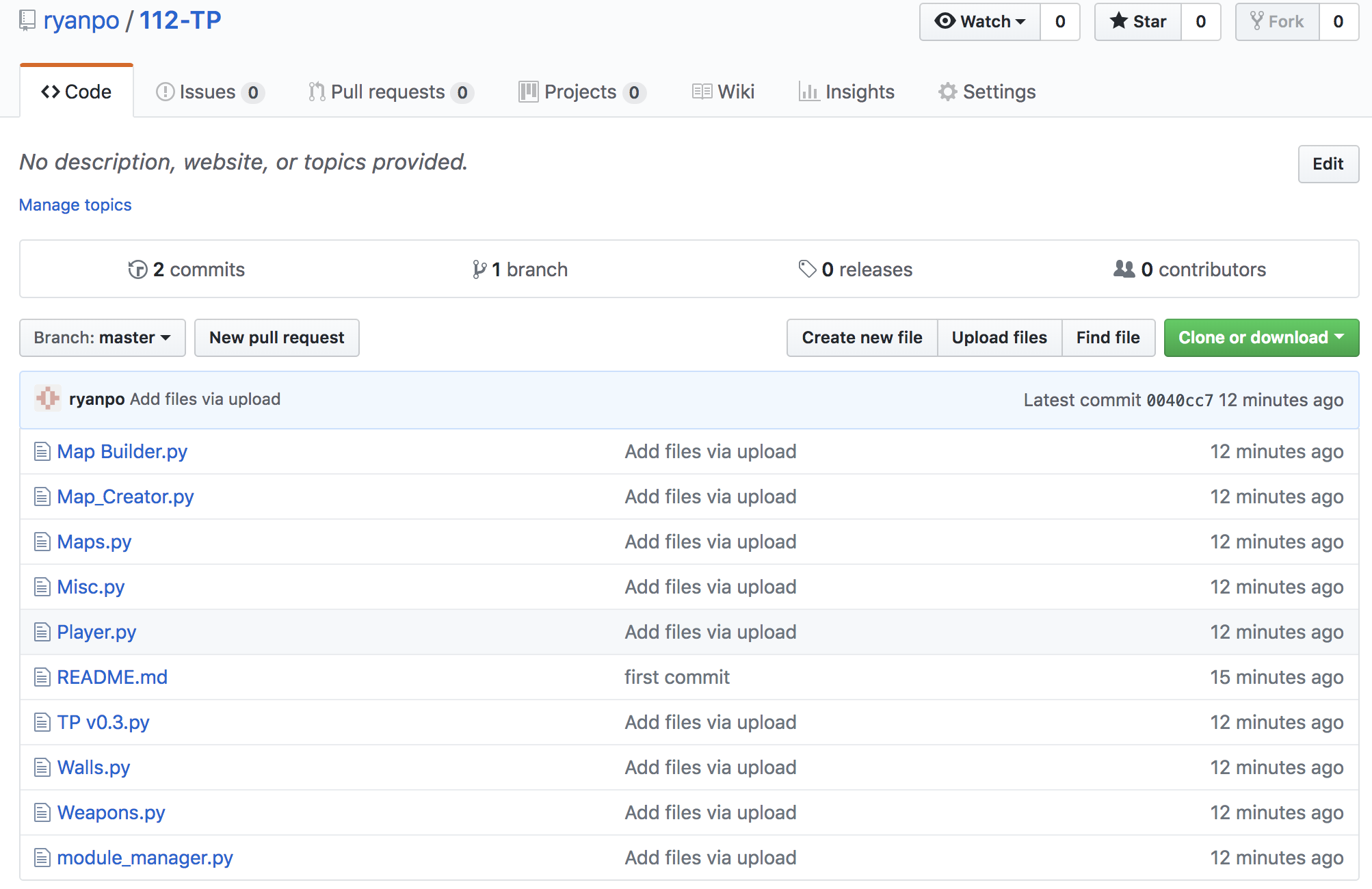
Timeline Plan

* 14/11 – Create Map algorithm
* 16/11 – Create player and monsters
* 20/11 – Create attack animation
* 22/11 – Finish assigning tiles to map
* 23/11 – UI configs
* 27/11 – Added circular spell pattern
* 4/11 – Create algorithm for image classification

Algorithmic Plan

* Major algorithmically challenging components of this project will be the code for map creation and monster AI. Monster AI will be created to react according to the player’s movements. While level generation will be based on a variation of the “drunkard’s walk” method, which may seem quite simple, but certain variations such parameters that allow for more branches/more open spaces adds complexity to the algorithm.

Version Control Plan

* Previous versions of the project are placed within the “Prev. Versions” files, while all of the files will be stored in a repository on GitHub.
* 

Module List

* Pygame

TP2 Update

* Added spell-casting, hold shift to enter spell-casting mode and drag cursor around in a large circle to cast spell
* Different sprites used
* Level size decreased

TP3 Update

* Added 2 more spell-casting patterns, changed spell casting algorithm to one that is based on comparing matrices instead of a hard-coded solution
* Level generation process displayed