Project description:

Title: *112 Galaga*

Description: This program will be a remake of the classic arcade game *Galaga* with Python. It consists of the player being a spaceship that is defending himself against aliens. It is a level base game and every time you kill all the aliens you pass on to the next level. The goal is to try to get as far as possible in levels.

Competitive analysis:

The online versions of *Galaga* are all fairly similar. Some of them are remakes of the original 1980s *Galaga* and other are newer versions. In all of them, the player is the spaceship trying to defend itself from incoming aliens (flying-in in a loopy path) and trying to get as far as possible in levels. These versions don't include scoreboards and some of them have just 20 levels. My version of *Galaga* will be like these. However, I will implement a scoreboard and potentially add some type of AI that can generate random levels. In addition, I would like to incorporate power-ups into the gameplay to make it more fun and interesting.

Structural Plan:

- I will use OOPy animation to make classes for elements of the game like the ship, aliens, bullets, and paths.
- The program will be divided into files where every state (start, game, game over, etc.) and character (ship, aliens, bullets) will have its own file. Each file will contain a class that element of the game.

Algorithmic plan:

- To generate the random loopy paths aliens come in, I will use functions for graphing parabolas, circles, sin, cos, etc. to generate a list of x,y coordinates for the aliens to follow.
- For example:
    - To fly in a circle: $(x - cx)^2 + (y - cy)^2 = r^2$ and solve for y given a radius, center coordinates and a range of x's
    - To move in a curved direction, I will use even/odd functions on a range of x's
    - For wavy paths, use sin and cos
- I will use equations/functions like these in a range of x's and connect them to trace loopy paths for the enemies to follow
- Once I have created the paths I want to use, I will assign them randomly to enemies

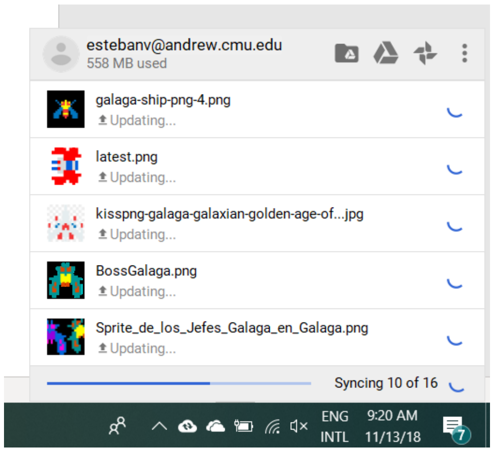
Plan of attack:

- By TP1 (Nov 20)
    - Minimum: Level 1 with enemies appearing without fancy path
- By TP2 (Nov 28)
    - Finish levels
    - Randomized loopy paths for enemies
    - If there is time:

        - Start audio
        - Very basic AI
- By TP3 (Dec 6)
    - Finished UI
    - AI: multiple levels of AI
    - Scoreboard

Version Control Plan:

- Use Google Backup & Sync to upload TP folder constantly to Google Drive



estebanv@andrew.cmu.edu
558 MB used

galaga-ship-png-4.png
Updating...

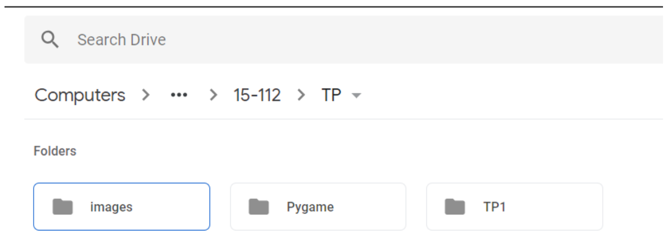
latest.png
Updating...

kisspng-galaga-galaxian-golden-age-of...jpg
Updating...

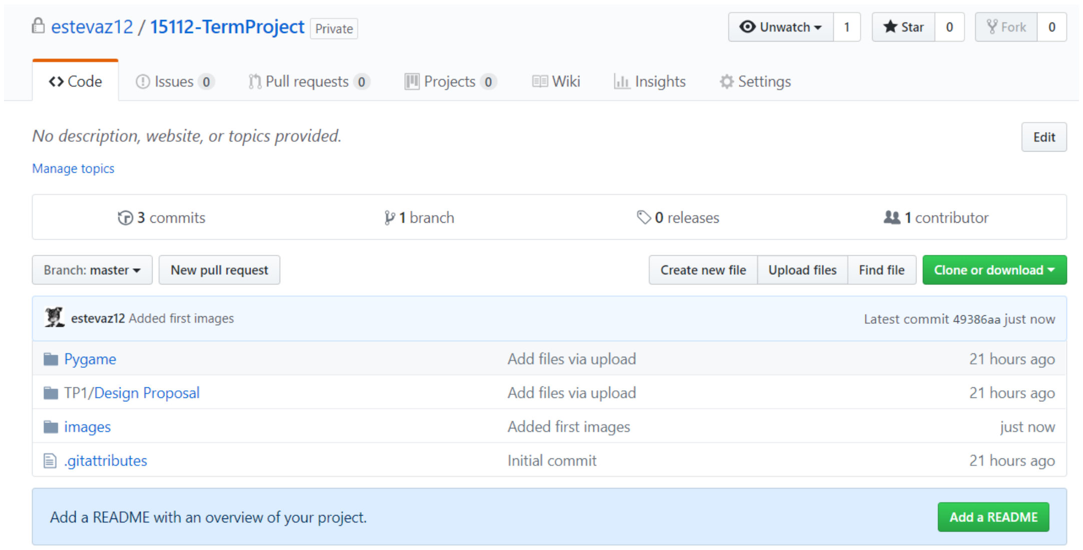
BossGalaga.png
Updating...

Sprite_de_los_Jefes_Galaga_en_Galaga.png
Updating...

Syncing 10 of 16



- Push my daily changes into a private GitHub repository



No description, website, or topics provided.

3 commits | 1 branch | 0 releases | 1 contributor

estevaz12 Added first images — Latest commit 49386aa just now

Pygame	Add files via upload	21 hours ago
TP1/Design Proposal	Add files via upload	21 hours ago
images	Added first images	just now
.gitattributes	Initial commit	21 hours ago

Add a README with an overview of your project.

Module list:

- Pygame