*\*TP2 Updates found on page 4*

*\*TP3 Updates found on page 5*

15112 – Term Project: Design Proposal

The name of my term project is “Pattern Pal”. It will be a program that introduces the fun art of geometric patterns and tessellations to the user, inspired by geometric patterns found in Islamic and Arabic art and culture. The program begins by showing the user a spread of patterns that they can choose to create their pattern from or the option to create draw their own pattern using the mouse. Then the user has the option to tessellate their pattern into different combinations of a grid using recursion to create an interesting tessellation effect. At the end, the user can export their created pattern in tkinter to a jpeg/dxf file to use for laser cutting.

Competitive Analysis:

Existing projects or programs similar to my project are CAD applications like Rhino and AutoCAD and Adobe Illustrator which allow the user to construct customized shapes and grids of shapes using a variety of commands with the mouse and keyboard. Another similar application would be Microsoft Paint in terms of a desired user interface that starts with the blank canvas and lets users draw and construct shapes of different colors and tones. My project would be very similar to these above applications in terms of interface and features that allow users to draw shapes with the mouse and drag and drop different polygons and in terms of creating background grids and graphics.

My project would be different by having an established emphasis on creating patterns and tessellations that occur when a single grid created by the user is repeated or arranged in some sort of way to make an interesting pattern composition, which would be done using recursive methods. It will also incorporate animation features of patterns opening and closing.

Structural Plan:

The final project will be organized in separate files:

* One main file for containing the overall app interface and app methods for cycling between different modes and features. This file will also contain the user defined grid methods, where the user can draw their own pattern using the mouse.
* Separate files for each of the example pattern templates that will be featured in the early stage of the program.
* A separate file maybe for running the code to export the Tkinter image to a jpeg using a separately installed module.

All these files will be packaged in an appropriate folder that contains cmu 112 graphics package and JSON or JPEG packages.

Algorithmic Plan:

The complex parts of my project include the generation of the tessellation grid of the pattern itself. In order to generate a tessellation from a single design placed on the grid that is either a template or created by the user with the mouse, there will be a basic option that just arranges the pattern in the grid with a nested for loop given user input of columns and rows. Then there will be more complex options of taking the existing pattern and recursively drawing smaller patterns within the shape’s list of points that create the outline of a bounding box. There are multiple possibilities with how the canvas can be divided, so there will need to be multiple recursive methods that subdivide the grid in some pattern and then draw a smaller version of each base pattern in each of the grid cells.

Timeline Plan:

Before Checkpoint 5 (4/20): I hope to create most of the user defined features of drawing shapes and custom patterns with a color palette. I also want to define the recursive methods for subdividing the pattern template into a series of smaller patterns or fractals this weekend.

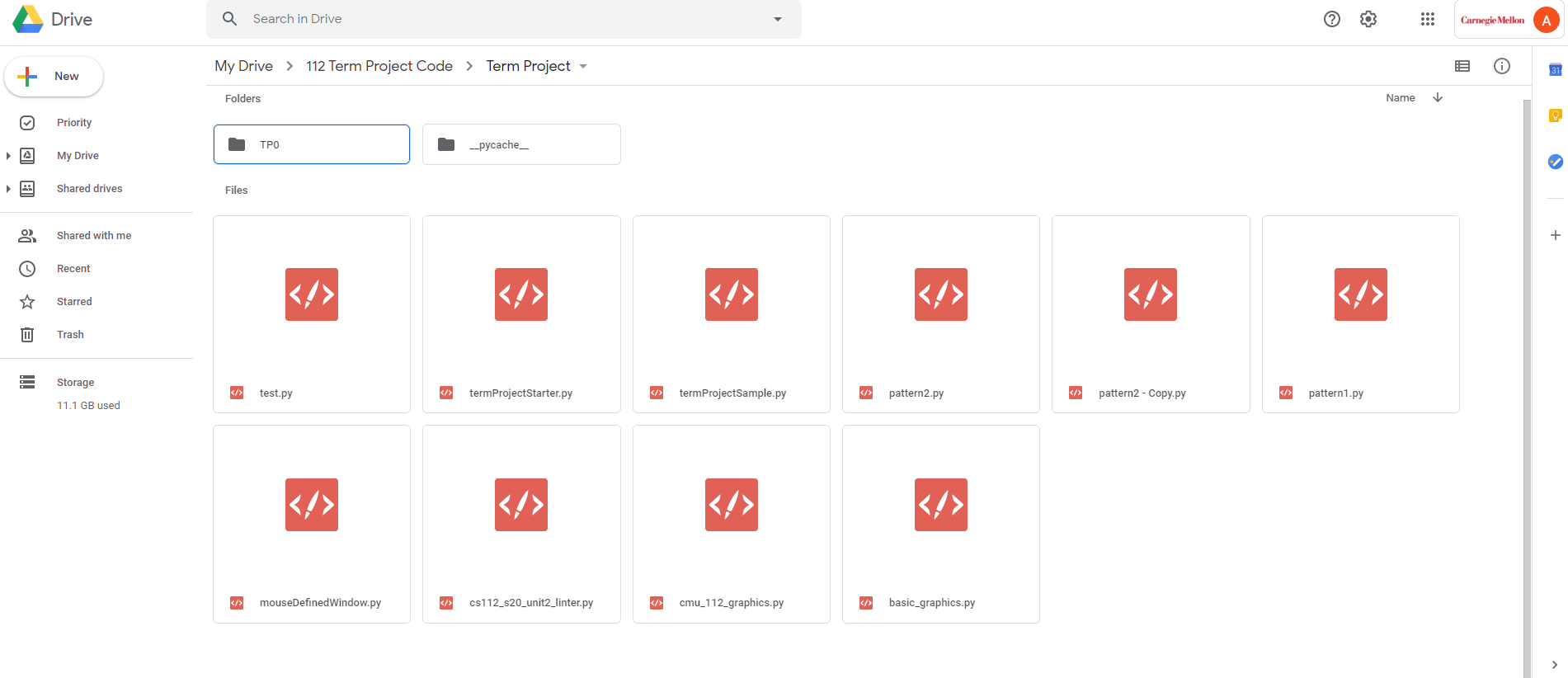
Before TP2 (4/23): I am going to add the JSON feature for storing and saving projects by implementing all the user defined objects into a list that can be accessed later. I also will organize my code more efficiently into the desired structure above.

Before Checkpoint 6 (4/27): I am going to incorporate the external feature of exporting the tkinter image into a jpeg. I am also going to use advanced Tkinter objects for creating canvas objects that can hold a pattern and additional features on the window as well.

Final Push (4/27 to 4/29): Debugging and fixing minor details in the project as well as trying to make it more aesthetic if possible. If there is time, I’d like to use Django to make a web application out of my project. I will also make the project description video in this time.

Version Control:

I am backing up all of my code to a folder on Google Drive. I update this folder with different versions of each file at the end of every day.



Module List:

The approved external module I am going to use is advanced Tkinter objects containing widgets. If there is time, I’d be interested in using Django as a web application.

*TP2 Design Documentation Updates:*

Structural Plan:

The final project will be organized in separate files:

* One main file for containing the overall app interface and app methods for cycling between different modes and features. This file will also contain the user defined grid methods, where the user can draw their own pattern using the mouse.
* Separate files for each of the example pattern templates that will be featured in the early stage of the program. Each separate template file has the accompanying functions for recursively dividing that base pattern into a grid.
* In total there is one main file containing the app interface, and 4 separate template files.
* All drawing functions for the template patterns are found in a separate file
* All objects created to construct the geometry including the point class, line segment class, circle class, and polygon class, and helper functions with mapping value ranges are included a separate file as well

All these files will be packaged in an appropriate folder that contains cmu 112 graphics package and JSON or JPEG packages.

Algorithmic Plan:

Only change is that the basic grid will no longer use a nested for loop, but will also be done recursively.

Timeline Plan:

Before Checkpoint 6 (4/27): I am going to fix any bugs associated with capturing the Tkinter screen and saving it as an image. I am going to improve the User interface features for the buttons and saving/opening existing files. I am going to try and implement the drawFace feature that takes an image of a user’s face and tries to draw a corresponding pattern from that using image processing.

Final Push (4/27 to 4/29): Debugging and fixing minor details in the project as well as trying to make it more aesthetic if possible. I will also make the project description video in this time.

Module List:

The approved external module I am going to use is advanced Tkinter objects containing widgets. If there is time, I’d be interested in using Django as a web application or OpenCV for facial recognition.

*TP3 Design Documentation Updates:*

Structural Plan:

No changes

Algorithmic Plan:

No changes besides introducing a method that recursively divides the pattern into a spiral. I also included a new method that draws a user’s self-portrait based on an image that they may have given.

Timeline Plan:

N/A

Module List:

The approved external module I am going to use is advanced Tkinter objects containing widgets.