SE 3XA3: Project Approval

Group 10
Lab: L03
Team Name: The Python Painters
Aamina Hussain, hussaa54
Jessica Dawson, dawsor1
Fady Morcos, morcof2

Original Project Name: Abstract Art Generator
URL: https://github.com/Burakcoli/Abstract-Art-Generator
License: GNU General Public License v3.0 (allows redevelopment)
Language: Python (is feasible for our team)
Lines of code: 1400
Can compile: Yes

The project randomly generates abstract art. It does this by generating *layers*, with each layer having a set of parameters that specify what shapes to draw on it, what size to draw them, how many to draw, and their distribution pattern. The layers can be placed on top of one another in an order requested by the user. The art can then be exported as a png file. The program currently has two layers that draw and layer themselves over each other. A color palette of four colors is used to color the art, with one color being randomly chosen as a background and the others being assigned at random to the layers' shapes. Pygame is used to generate the art and a mix of pygame and tkinter is used for the GUI. Learning the project and necessary libraries should not be difficult.

Intended changes:

- Modularize/Clean-up code (it is currently one singular file and can definitely be broken down into multiple)
- Add additional layers (allow for more than two generated art layers)
- Allow user to include text on top of the art
- Allow user to choose the background color (instead of it being randomly chosen from the color palette)
- Add new generators and algorithms:
  - Layout shapes as a border
  - Change the transparency of the layers
  - Add new shape options (i.e. triangles, stars, spiky balls, jagged lines, etc.)
  - Add new shape layout options (i.e. zig-zag, aforementioned border, spiral, etc.)

Testing:

- Pytest: We will run portions of the code with expected outputs and test whether the actual outputs match the outputs we expected. This will be done using the automated unit testing framework of Pytest. This can be done to test whether the color palette chosen by the user matches the color palette displayed, or whether the correct shape is used for the layer.
- Manual Testing: We will test some of the visual components of this project manually by having a user actually using and interacting with the program and physically checking if those components are producing the required response.