SE 3XA3: Test Plan

Group #10
Lab: L03
Aamina Hussain, hussaa54
Jessica Dawson, dawsor1
Fady Morcos, morcof2

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
March 7, 2022	0.0	Initial Document
March 9, 2022	0.1	Added Sections 2, 4
March 11, 2022	0.2	Added Sections 1, 3, 5, 6, 7
April 11, 2022	1.0	Updated Functional Tests Section 3.1

1 General Information

1.1 Purpose

The purpose of this document is to describe the testing, verification and validation procedures that will be used to confirm the correctness of Group 10's *Abstract Art Generator* project. Since the project deals with a graphical user interface, most of the testing involved will be composed of manual tests. The rest of the tests will mainly involve white box testing techniques such as unit testing. Regarding V&V, the product will be considered verified if it functions smoothly and passes all tests, and it will be considered valid if it satisfies all group members.

1.2 Scope

The *Abstract Art Generator* project is a Python program which uses Pygame and Pygame_GUI to display a graphical user interface that allows interaction with the user. It generates random abstract art images that may be exported as PNG files. The scope of testing will cover the various settings of layer generation such as different shapes, styles, sizes, and complexities of each layer. It will also cover the randomization of layer and color palette settings.

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations and Definitions**

Term	**Definition**
V&V	Verification and Validation
Manual Test	Tests which are hand written and involve checking if UI performs how it is supposed to
Automated Test	Tests that run automatically (may involve using a framework such as PyTest)
Unit Test	A set of test designed for methods and functions
Structural Test	Tests based on the internal structure of program
Functional Test	Tests based on a function provided by the program
Nonfunctional Test	Tests based on the qualities of the program
Static Test	Tests where the code is inspected and not executed
Dynamic Test	Tests where the code is executed

1.4 Overview of Document

This document describes the testing methods that will be used to test the first version of this project.

2 Plan

2.1 Software Description

This software will allow users to generate abstract art, which can then be exported by the user as a PNG file. This art can be randomly generated, or can be generated based on the user's choices. The software receives these choices using user input, including both keyboard input and mouse input.

2.2 Test Team

The test team consists of Aamina Hussain, Jessica Dawson, and Fady Morcos. They are all equally responsible for testing the software.

2.3 Testing Tools

The tool we will be using to test the software is Pylint. This tool will allow us to perform some static tests, help us with debugging, and ensure we are following the correct coding guidelines and style.

2.4 Testing Schedule

Please refer to the Gantt Chart at this link.

3 System Test Description

3.1 Functional Tests

3.1.1 User Input: Generate Art

1. FT-GA-1
   Type: Functional, Dynamic, Manual

   Initial State: No art generation options are locked and main program window waiting for input.

   Input: User clicks on `Generate Randomly` button.

   Output: Program generates abstract art piece based on random specification for all options.

   How test will be performed: The program will run and a tester will check the generated art piece. This will be performed a few more times to check that the parameters of the generated art are randomized.

2. FT-GA-2
   Type: Functional, Dynamic, Manual

   Initial State: Some art generation options are locked and main program window waiting for input.

   Input: User clicks on `Generate Randomly` button.

   Output: Program generates abstract art piece based on the specification indicated by the locked options and random specification for the rest of the options that are not locked.

How test will be performed: The program will run and a tester will check the generated art piece.

3. FT-GA-3
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on `Generate` button.

   Output: Program generates abstract art piece based on whatever the current options are.

   How test will be performed: The program will run and a tester will check the generated art piece.

3.1.2 User Input: Use Option Locks

1. FT-OL-1
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on color palette lock and then clicks on the `Generate Randomly` button.

   Output: Program displays lock icon next to color palette and generates an art piece using the color palette chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the color palette. The tester will then check if the generated art uses the chosen color palette.

2. FT-OL-2
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on the background color lock and then clicks on the `Generate Randomly` button.

   Output: Program displays lock icon next to the background color and generates an art piece that has the same background color as the one chosen by the user.

How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the background color. The tester will then check if the background color for the generated art matches the chosen color.

3. FT-OL-3
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on the shape lock for layer one and then clicks on the `Generate Randomly` button.

   Output: Program displays lock icon next to the layer one shape options and generates an art piece that has a layer with the same shape as the one chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the layer one shape options. the tester will then check if the layer one shape in the generated art matches the chosen shape. This will be repeated for layers two and three.

4. FT-OL-4
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on the style lock for layer one and then clicks on the `Generate Randomly` button.

   Output: Program displays lock icon next to the layer one style options and generates an art piece that has a layer with the same style as the one chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the layer one style options. The tester will then check if the layer one style in the generated art matches the chosen style. This will be repeated for layers two and three.

5. FT-OL-5
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

Input: User clicks on the complexity lock for layer one and then clicks on the `Generate Randomly` button.

Output: Program displays lock icon next to the layer one complexity and generates an art piece that has a layer with the same complexity as the one chosen by the user.

How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the layer one complexity. The tester will then check if the layer one complexity in the generated art matches the chosen complexity. This will be repeated for layers two and three.

6. FT-OL-6
   Type: Functional, Dynamic, Manual

   Initial State: Main program window has a previous art piece with a low layer one complexity and is waiting for input.

   Input: User clicks on layer one complexity slider to increase complexity and then clicks on the `Generate` button.

   Output: Program generates an art piece that has a layer with a higher complexity than the previous art piece.

   How test will be performed: The program will run and a tester will check if the layer one complexity in the generated art is greater than the layer one complexity of the previously generated art. This will be repeated for layers two and three.

7. FT-OL-7
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on the size lock for layer one and then clicks on the `Generate Randomly` button.

   Output: Program displays lock icon next to the layer one shape size and generates an art piece that has a layer with the same shape size as the one chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the layer one shape size. The tester will then check if the layer one shape

size in the generated art matches the chosen shape size. This will be repeated for layers two and three.

8. FT-OL-8
   Type: Functional, Dynamic, Manual

   Initial State: Main program window has a previous art piece with a smaller layer one shape size and is waiting for input.

   Input: User clicks on layer one shape size slider to increase the shape size and then clicks on the `Generate` button.

   Output: Program generates an art piece that has a layer with a larger shape size than the previous art piece.

   How test will be performed: The program will run and a tester will check if the layer one shape size in the generated art is larger than the layer one shape size of the previously generated art. This will be repeated for layers two and three.

9. FT-OL-9
   Type: Functional, Dynamic, Manual

   Initial State: Main program window waiting for input.

   Input: User clicks on the transparency lock for layer one and then clicks on the `Generate Randomly` button.

   Output: Program displays lock icon next to the layer one transparency and generates an art piece that has a layer with the same transparency as the one chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the layer one transparency. The tester will then check if the layer one transparency in the generated art matches the chosen transparency. This will be repeated for layers two and three.

10. FT-OL-10
    Type: Functional, Dynamic, Manual

    Initial State: Main program window has a previous art piece with a low layer one transparency and is waiting for input.

    Input: User clicks on layer one transparency slider to increase transparency and then clicks on the `Generate` button.

Output: Program generates an art piece that has a layer with a higher transparency than the previous art piece.

How test will be performed: The program will run and a tester will check if the layer one transparency in the generated art is greater than the layer one transparency of the previously generated art. This will be repeated for layers two and three.

11. FT-OL-11
    Type: Functional, Dynamic, Manual

    Initial State: Main program window waiting for input.

    Input: User clicks on the overlay/border lock and then clicks on the `Generate Randomly` button.

    Output: Program displays lock icon next to the overlay/border and generates an art piece that has the same overlay/border as the one chosen by the user.

    How test will be performed: The program will run and a tester will check the program window to see if there is a lock icon next to the overlay/border. The tester will then check if the overlay/border for the generated art matches the chosen overlay/border.

3.1.3 User Input: Add Text

1. FT-AT-1
   Type: Functional, Dynamic, Manual

   Initial State: Main program window has a generated art piece and is waiting for input.

   Input: User clicks on `Add Text` button and types desired text.

   Output: Program displays the text the user types on top of the previously generated art using the default text size and text font.

   How test will be performed: A tester will type a message in the text box. The the program will run and a tester will check the program window to see if the message typed is displayed on top of the generated art.

   ~~FT-AT-2~~
   ~~Type: Functional, Dynamic, Manual~~

~~Initial State: Main program window has a generated art piece with text and is waiting for input.~~

~~Input: User clicks on a text color option.~~

~~Output: Program changes the text color to the color chosen by the user.~~

~~How test will be performed: The program will run and a tester will check the program window to see if the text color matches the color chosen.~~

2. FT-AT-2
   Type: Functional, Dynamic, Manual

   Initial State: Main program window has a generated art piece with text and is waiting for input.

   Input: User clicks on a text size option.

   Output: Program changes the text size to the size chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if the text size matches the size chosen.

3. FT-AT-3
   Type: Functional, Dynamic, Manual

   Initial State: Main program window has a generated art piece with text and is waiting for input.

   Input: User clicks on a text font option.

   Output: Program changes the text font to the font chosen by the user.

   How test will be performed: The program will run and a tester will check the program window to see if the text font matches the font chosen.

3.1.4 User Input: Export Art

1. FT-EA-1
   Type: Functional, Dynamic, Manual

   Initial State: Main program window has a generated art piece and is waiting for input.

Input: User clicks on `Export` button.

Output: Program opens file explorer window.

How test will be performed: The program will run and a tester will check to see if the file explorer window opens.

2. FT-EA-2
   Type: Functional, Dynamic, Manual

   Initial State: Main program has the file explorer window open and is waiting for input.

   Input: User clicks on directory they want to save the art and names the file.

   Output: Program names the file and saves it in the chosen directory.

   How test will be performed: The program will run and a tester will open the file explorer to check if the file is saved in the correct directory and that the file is named correctly.

3.1.5 User Input: Change Display Settings

1. FT-DS-1
   Type: Functional, Dynamic, Manual

   Initial State: Main program window is currently in dark mode and is waiting for input.

   Input: User clicks on `THEME` button.

   Output: Program switches to light mode.

   How test will be performed: The program will run and a tester will check to see if the program window switches to light mode. The tester will check if the background color of the main program window is a light color and the UI text is a dark color.

2. FT-DS-2
   Type: Functional, Dynamic, Manual

   Initial State: Main program window is currently in light mode and is waiting for input.

   Input: User clicks on `THEME` button.

Output: Program switches to dark mode.

How test will be performed: The program will run and a tester will check to see if the program window switches to dark mode. The tester will check if the background color of the main program window is a dark color and the UI text is a light color.

~~FT-DS-3~~
~~Type: Functional, Dynamic, Manual~~

~~Initial State: Main program window is currently in dark mode and is waiting for input.~~

~~Input: User clicks on `Dark Mode` button.~~

~~Output: Program stays in dark mode.~~

~~How test will be performed: The program will run and a tester will check to see if the program window remains in dark mode. The tester will check if the background color of the main program window is black and the UI text is white.~~

~~FT-DS-4~~
~~Type: Functional, Dynamic, Manual~~

~~Initial State: Main program window is currently in light mode and is waiting for input.~~

~~Input: User clicks on `Light Mode` button.~~

~~Output: Program stays in light mode.~~

~~How test will be performed: The program will run and a tester will check to see if the program window remains in light mode. The tester will check if the background color of the main program window is white and the UI text is black.~~

3. FT-DS-3
   Type: Functional, Dynamic, Manual

   Initial State: Main program window is waiting for input.

   Input: User clicks on `HELP` button.

   Output: Program displays the instructions.

How test will be performed: The program will run and a tester will check to see if the program displays the instructions.

3.2 Nonfunctional Tests

3.2.1 Performance Requirements

1. NFT-PR-1
   Type: Non-Functional, Dynamic, Manual

   Initial State: Program is idling waiting for input.

   Input: The user changes a setting.

   Output: The time it takes for the program to return to idling. Should be less than MAX_PARAM_TIME.

   How test will be performed: A tester will change a setting and time how long it takes before the program accepts input again. This will be repeated ten times and an average will be taken and compared to MAX_PARAM_TIME.

2. NFT-PR-2
   Type: Non-Functional, Dynamic, Manual

   Initial State: Program is idling waiting for input.

   Input: The user presses the `Generate` button.

   Output: The time it takes for the program to display an image. Should be less than MAX_GENERATION_TIME.

   How test will be performed: A tester will press the generate button and time how long it takes before the program to display an image. This will be repeated ten times and an average will be taken and compared to MAX_GENERATION_TIME.

3.2.2 Operational and Environmental Requirements

1. NFT-OE-1
   Type: Non-Functional, Dynamic, Manual

   Initial State: Two laptops, one running Windows 10, the other Ubuntu 20.04.

   Input: The user attempts to run the program.

Output: The program runs.

How test will be performed: A tester will install the program on each of the laptops and try running it on each. The program should run in both environments.

3.2.3 Usability and Humanity Requirements

1. NFT-UH-1
   Type: Non-Functional, Dynamic, Manual

   Initial State: 10 third party users who have no experience with the program.

   Input: Each user will attempt to operate the program.

   Output: The opinions of the users on how easy the program is to learn and operate.

   How test will be performed: A tester will ask each of the 10 users to create certain art pieces, and manipulate certain settings. The tester will then record the users' opinions on how easy these tasks were to accomplish. The opinions will be recorded as a rating of 1 to 5, 1 being not satisfied, and 5 being fully satisfied. An average of these ratings will be taken and the test will pass if the average is greater than 3.5.

4 Tests for Proof of Concept

This section contains tests we ran to confirm our program was functional for the Proof of Concept Demonstration.

4.1 Third Layer Tests

4.1.1 Third Layer Existence

1. POC-3L-1
   Type: Functional, Dynamic, Manual

   Initial State: Each layer has unique settings that can be visually distinguished from each other and the program is idling waiting for input.

   Input: The user presses the `Generate` button.

Output: An art piece will be displayed with three layers, each with their selected settings.

How test will be performed: A tester will manually create the initial state, where settings being visually distinct is based on their own discretion, and press the `Generate` button. They will then check that three layers are generated in the image.

4.1.2 Third Layer Settings

2. POC-3L-2
   Type: Functional, Dynamic, Manual

   Initial State: The program has a previous art piece displayed and is idling waiting for input.

   Input: The user changes the layer three pattern drop down and presses the `Generate` button.

   Output: A new art piece will be displayed with the new pattern option for the third layer.

   How test will be performed: A tester will manually create the initial state, change the pattern setting, and press the `Generate` button. They will then check that the new pattern appears in the third layer.

3. POC-3L-3
   Type: Functional, Dynamic, Manual

   Initial State: The program has a previous art piece displayed and is idling waiting for input.

   Input: The user changes the layer three shape drop down and presses the `Generate` button.

   Output: A new art piece will be displayed with the new shape option for the third layer.

   How test will be performed: A tester will manually create the initial state, change the shape setting, and press the `Generate` button. They will then check that the new shape appears in the third layer.

4. POC-3L-4
   Type: Functional, Dynamic, Manual

Initial State: The program has a previous art piece displayed and is idling waiting for input.

Input: The user changes the layer three complexity slider and presses the `Generate` button.

Output: A new art piece will be displayed with the new complexity option for the third layer.

How test will be performed: A tester will manually create the initial state, change the complexity slider, and press the `Generate` button. They will then check that the new third layer displays the increased complexity.

5. POC-3L-5
   Type: Functional, Dynamic, Manual

   Initial State: The program has a previous art piece displayed and is idling waiting for input.

   Input: The user changes the layer three size slider and presses the `Generate` button.

   Output: A new art piece will be displayed with the new size option for the third layer.

   How test will be performed: A tester will manually create the initial state, change the size slider, and press the `Generate` button. They will then check that the new third layer displays the increased sized of shaped.

4.1.3 Third Layer Randomization

6. POC-3L-6

   Type: Functional, Dynamic, Manual

   Initial State: No art generation option are locked and the program is idling waiting for input.

   Input: The user presses the `Generate Randomly` button.

   Output: All options, including the layer three options, will be randomized and an art piece will be generated with the resulting settings.

   How test will be performed: A tester will manually create the initial state and press the `Generate Randomly` button. They will then check

whether the layer three options have been randomized and that this is reflected in the art.

7. POC-3L-7

   Type: Functional, Dynamic, Manual

   Initial State: All third layer options are locked and the program is idling waiting for input.

   Input: The user presses the `Generate Randomly` button.

   Output: The third layer settings will stay the same and an art piece will be generated with these settings.

   How test will be performed: A tester will manually create the initial state and press the `Generate Randomly` button. They will then check whether the layer three options have been randomized and that this is reflected in the art.

5 Comparison to Existing Implementation

N/A

6 Unit Testing Plan

6.1 Unit testing of internal functions

N/A
##COMMENT: Due to the nature of the program, its randomization, and its output being a randomized image, we decided to not perform automated testing. Unit tests are not extensive enough to thoroughly test the changes made to the UI display as a result of some user input. All of the functionality of the program can be tested using manual testing, which is what we will use.

6.2 Unit testing of output files

N/A

7 Appendix

7.1 Symbolic Parameters

MAX_PARAM_TIME: 1 sec
MAX_GENERATION_TIME: 10 sec

7.2 Usability Survey Questions

N/A