

SE 3XA3: Software Requirements
Specification
Title of Project

Team L3 G10, The Python Painters
Aamina Hussain, hussaa54
Jessica Dawson, dawosr1
Fady Morcos, morcof2

February 11, 2022

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
February 9, 2022	1.0	Initial Document; Completed Section 1, 2.1.1, 3.1, 3.2
February 11, 2022	1.1	Completed Section 2.1.2, 2.1.3, 2.2, 3.3-3.9, 4, 5

This document describes the requirements for the Abstract Art Generator. The template for the Software Requirements Specification (SRS) is a subset of the Volere template.

1 Project Drivers

1.1 The Purpose of the Project

The purpose of the proposed project is to enhance an existing art generator. The existing Abstract Art Generator can create a wide array of different images with many customization options. The project will add to these options by increasing the ways the art can be generated and manipulated. The finished product should allow for even more variation in art generation.

1.2 The Stakeholders

1.2.1 The Client

The project's clients will be Dr. Asghar Bokhari as the instructor of SFWRENG 3XA3 and the TAs of the course. They are interested in expanding the Abstract Art Generator and will be determining whether the requirements have been met.

1.2.2 The Customers

The customers of the project are anyone interested in generating abstract art.

1.2.3 Other Stakeholders

The developers of the project are also stakeholders as they will be applying their skills to the project. The developer of the original Abstract Art Generator, Burak Unutmaz, is also a stakeholder as they have an interest in seeing how their project expands.

1.3 Mandated Constraints

1. The project must run on Windows 10 or newer, or Linux Ubuntu 20.04.

2. The project must be completed according to the schedule included in Tasks (Section 4.4).

1.4 Naming Conventions and Terminology

- **Color Palette:** Four different colors that complement each other that the generated abstract art is made up of.
- **Layer:** The Abstract Art Generator contains layers. Each layer is made up of a specified shape laid out in a specified pattern. The colors of these shapes are from the chosen color palette. The layers are then stacked on top of each other to create the finished art piece.

1.5 Relevant Facts and Assumptions

1.5.1 Facts

- N/A

1.5.2 Assumptions

- Users have access to a computer
- Users know how to use a computer
- Users know how to download files
- Users can read English

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

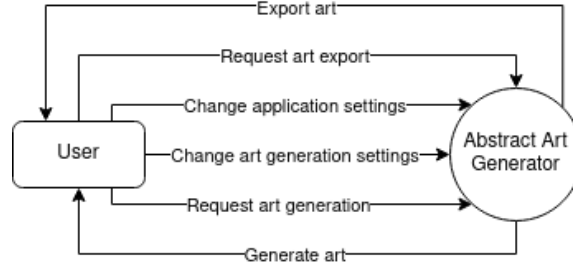


Figure 1: Work Context Diagram

2.1.2 Work Partitioning

Event Number	Event Name	Input	Output
1	Generating a picture	Mouse	Generated Picture
2	Exporting a picture	Keyboard/Mouse	Image File
3	Opening the help overlay	Mouse	Help Overlay
3	Opening the settings menu	Mouse	Settings Menu

Table 2: Work Partitioning Events

Event Number	Summary
1	The user can, through mouse input, choose to manipulate and change generation settings and generate an image. The image will then be displayed to the user in the UI.
2	The user, through mouse and keyboard input, can specify a resolution and file location to export an image file to.
3	The user can open and read the help menu by using mouse input
3	The user can open and manipulate the applications' setting by using mouse input

Table 3: Work Partitioning Summaries

2.1.3 Individual Product Use Cases

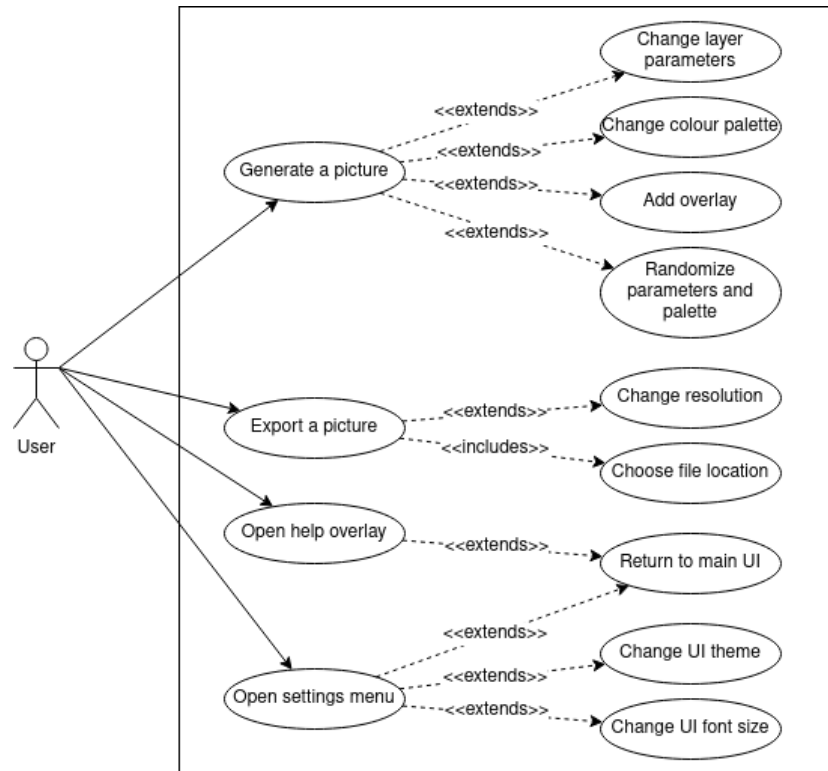


Figure 2: Use Case Diagram

The above diagram specifies the product use cases. Generating a picture is the most complex as it involves changing generation parameters. Exporting a picture requires a resolution and file location to be specified. Opening the help overlay just involves the user accessing a help button. Opening the settings menu allows the user to change aspects of the UI and program itself.

2.2 Functional Requirements

FR1. The system shall have 30 unique color palette options.

- Rationale: The user will have a wider range of color options to choose from.

FR2. The system shall allow the user to choose the background color of the art from the chosen color palette.

- Rationale: The user may want to choose the exact background color instead of it being randomly chosen from the color palette.

FR3. The system shall display which color the user chose as the background.

FR4. The system shall allow the user to have three generated layers.

- Rationale: Currently, the program only allows for two generated layers. This will increase the variety of art the user can generate.

FR5. The system shall allow the user to change the transparency of the layers.

- Rationale: This will increase the variety of art the user can generate.

FR6. The system shall allow the user to have text on top of the art.

- Rationale: This will increase the variety of art the user can generate.

FR7. The system shall allow the user to change the font of the text on top of the art.

FR8. The system shall allow the user to change the size of the text on top of the art.

FR9. The system shall allow the user to change the color of the text on top of the art.

FR10. The system shall allow the user to add a border to the art.

- Rationale: This will increase the variety of art the user can generate.

FR11. The system shall display the border options.

FR12. The system shall display the display settings.

FR13. The system shall allow the user to configure the display settings, including choosing light or dark mode, the text size of the UI, etc.

- Rationale: This will increase accessibility and allow users to adjust the UI settings to their comfort level.

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.1.1 Appearance Requirements

LF1. The menu of the program must be formatted neatly and not be cluttered.

- Rationale: The user should find the appearance of the program menu attractive to further their interest in using the program.

3.1.2 Style Requirements

LF2. The program background must be a simple and solid color.

- Rationale: If the program background is not simple, it may clash with the generated art and end up distracting the user.

3.2 Usability and Humanity Requirements

3.2.1 Ease-Of-Use Requirements

UH1. The options of the program shall be easy for the user to access via the program menu.

- Rationale: The user should be able to select any option efficiently and should not struggle to find the options.

3.2.2 Accessibility Requirements

UH2. The UI text shall be legible for the average user.

- Rationale: The average user cannot make full use of the features if they find it difficult to read the instructions or option descriptions.

3.2.3 Learning Requirements

UH3. The instructions for the program shall be clearly available to the user via the program menu.

- Rationale: The instructions are necessary for the user to be able to use the program.

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

PR1. The system shall respond to the user's parameter input or selection within MAX_PARAM_TIME seconds.

- Rationale: The user will not appreciate having to wait too long for the system to respond to their change.

PR2. The system shall generate an image within MAX_GENERATION_TIME seconds.

- Rationale: The user will not appreciate having to wait too long for the system to respond to their change.

3.3.2 Reliability and Availability Requirements

PR3. The program shall be available for the user to run whenever they wish.

- Rationale: The program does not require any external resources since it is downloaded and run on the user's machine. So the program should be designed such that as long as the user's PC is working, they should be able to run the program.

3.4 Operational and Environmental Requirements

3.4.1 System Requirements

OE1. The program shall be able to run on Windows 10 or newer, or Linux Ubuntu 20.04.

- Rationale: The program should be available to user's with common operating systems.

3.4.2 Productization Requirements

OE2. The program shall be available as an executable file for the user to download.

- Rationale: Having the program like this will ensure it is not difficult for the user to install.

3.5 Maintainability and Support Requirements

3.5.1 Maintenance Requirements

MS1. The source code shall be fully documented and consistent.

- Rationale: This will make it simpler and more straightforward for the current and future developers to understand.

3.5.2 Support Requirements

MS2. The program's repository will be public.

- Rationale: A public repository for the program will allow users to raise issues and access the most updated information regarding the program's development.

3.6 Security Requirements

N/A

3.7 Cultural and Political Requirements

CP1. The program instructions or feature descriptions shall not have any culturally offensive or politically sensitive words, phrases, or symbols.

3.8 Legal Requirements

LR1. The program shall adhere to and not violate any Canadian or US laws.

3.9 Health and Safety Requirements

N/A

4 Project Issues

4.1 Open Issues

Issue 1. Currently all the code for the original Abstract Art Generator program is written and contained within one single module composed of 1400

lines of code. This creates an issue for understandability and maintainability. Having poor modularity makes future adjustments and modification difficult. The next developer to try and update or make adjustments to the program may run into trouble comprehending it and thus have a hard time altering the program.

4.2 Off-the-Shelf Solutions

There are many websites and applications which also generate random art similar to what our project does (for example, randomart.co.uk). However, there are not many open-source random art generators, and it is unlikely we will be using any other sources of code. Most of the adapted code will be coming from the original Abstract Art Generator we are enhancing.

4.3 New Problems

N/A

4.4 Tasks

1. Begin modularizing the code.
2. Conduct first code review and debugging.
3. Add additional layers (allow for more than two generated art layers).
4. Add text overlay feature to the abstract art generator.
5. Add feature that allows user to set preferred background colour.
6. Conduct second code review and debugging.
7. Add feature that allows user to change the display settings.
8. Add new generators and algorithms.
9. Conduct testing of the code.
10. Conduct third code review and debugging.

11. Conduct an analysis of the program against the functional and non-functional requirements.

For a more detailed overview refer to the project [Gantt Chart](#)

4.5 Migration to the New Product

N/A. The new product will be a standalone version of the original Abstract Art Generator.

4.6 Risks

There is minimal risk associated with this project since it is an independent program that does not rely on any external resources. The one main risk is that since the code was not well managed originally, it may be difficult to understand and work with. This could cause delays and difficulty modifying the program.

4.7 Costs

There is no monetary cost related to this project. This is because we are working on an open sourced project and the tools we are using are all free of cost. The only cost associated with this project is the time and effort our team will be spending on brainstorming, development, and testing, which is roughly 5-8 hours weekly for each team member.

4.8 User Documentation and Training

The program will include a help tab that the user can select to access the instructions. The GUI will be kept clean and simple so the user should not have any issue navigating the program menu.

4.9 Waiting Room

Possible future features to be added:

1. Create a more appealing GUI (i.e. better choice of fonts, more attractive front-end, etc.).

2. Add layer effects like distortion and noise.
3. Add a feature that allows the program to design and generate 3-dimensional abstract art pieces.

4.10 Ideas for Solutions

Issue 1. A restructuring of the code may help in modularizing it to allow for better understandability, maintainability, and modification.

5 Appendix

N/A

5.1 Symbolic Parameters

MAX_PARAM_TIME: 1 sec

MAX_GENERATION_TIME: 10 sec