Swift Art: An Auto-Generated 2D Raster Logo Editor Software

A Thesis
Presented to the Faculty of
Mobile Application and Web-Development Program
STI College - Davao

In Partial Fulfilment
Of the Requirements for the Program
Of Mobile Application and Web Development

Agang, Ralph Vincent
Antega, Miguelito
Aritalia, Shebul Rey
Balista, Carol Clare
Dulatre, Kurt Mathew
Grande, Janela
Ledesma, Ezra
Lotero, William Oliver
Ricablanca, Belle Katherine
Subu, Cedrick

June 2022

INTRODUCTION

Introduction

Swift Art makes logo design simple for everyone. It saves the user from hassle in coming up with a logo idea and actual design. The app automatically generates a logo according to the preference of the end-users and also allows them to modify the generated design**.** The proponents will employ minimalism art as its core foundation for the logo design and the overall graphics of the application interfaces. The generated logo is under non-exclusive right and can be used freely for any purposes (personal or commercial) without any license restrictions from third-party elements.

Background of the Problem

The proponents of this study have gathered and reviewed the Content License Agreement or the Terms of Service (TOS) of 5 free logo design applications. According to Rathnayaka (2021), "*A software licensing is creating a legally binding agreement through a software license, between the owner or developer of a software program and the user.*" We found out that 5 of these logo design applications contain third-party elements that also have their Content License Agreement / TOS. We like to describe this circumstance as a "Tree-branch" structure because of its parent-child relationship. This "Tree-Branch" structure of content license may be convoluted for the end-users. According to Patar (2019), "*Two law professors analyzed the sign-in terms and conditions of 500 popular US websites, including Google and Facebook, and found that more than 99 percent of them were "unreadable," far exceeding the level most American adults read at, but are still enforced.*" According to Guynn

(2020), “*A Deloitte survey of 2,000 U.S. consumers in 2017 found that 91% of people consent to terms of service without reading them. For younger people, ages 18-34, that rate was even higher: 97% did so.*” 2 out of 5 free logo design applications that we have gathered and reviewed have a license that is limited to personal use only and is strictly not allowed for commercial use unless the end-user buys a subscription. The remaining 3 out of 5 applications have a license that is allowed for commercial use, but the application has no way of knowing whether the other license from third-party elements is accurate.

Overview of the current state of the technology

Swift Art will be deployed for the Android platform. Its Minimum API Level is Android 4.4 ‘Kitkat’ (API Level 19) and the target API Level follows Google Play's target API level requirement, which is Android 10 (API level 29) or higher. To reduce the application file size, it will be published in Google Play Store and uploaded as AAB (Android App Bundle); *“a new upload format that includes all your app’s compiled code and resources”* (Reduce your app size (n.d.). The proponents will use Unity Engine for developing the application. “*Unity is a cross-platform game engine used by many games on the Google Play Store.*”(Build your game in Unity (n.d.). Unity allows developers to create scripts that can interact with objects by using C# alone. “*Unity employs automatic memory management for your user-generated code and scripts*.” (Jacobsen (2021). Swift Art’s graphical assets will be created in Affinity Designer, “*a vector graphics editor developed by Serif for macOS, iPad, and Microsoft Windows (Affinity (n.d.).*”

Objectives of the study

Main Objectives

To make Swift Art accessible to everyone and can be used for any purposes (excluding trademark) without any convoluted license agreement.

Specific Objectives

- Swift Art aims to have user-friendly UX; “*User Experience (UX) is how a person feels when interfacing with a system*” (Gube (2021)).
- Swift Art aims to be accessible to anyone.
- Swift Art aims to be used for any purpose (personal or commercial).
- Swift Art aims to employ minimalism art as its foundation in making logos.

Scope and limitation of the study

Scope:

- Create Graphic application-specific in logo design.
- App optimization via Unity’s profiler
- Unity Framework
- Employ of Minimalism Art

Limitation:

- The end-users can use the generated logo under a non-exclusive right since any users can recreate similar designs.
- End-users can use the generated logo for personal, commercial, or small projects but it cannot be trademarked as the users do not have exclusive rights to the design
- Specific to one (1) art style only and that is Minimalism Art. "*Minimalism art is an extreme type of abstract art that usually is depicted through simplistic shapes and hard edges*" (Wolfe (n.d.)).

Review of Related Literature

All software that are publicize contain a unique terms of condition to avoid infringements. "*Terms and Conditions" (TOR) is the document governing the contractual relationship between the provider of a service and its user*." (What Are the Terms and Conditions, (n.d.)). Some end-users do not read TOR as most of it contains complex languages. According to Harrar (2018), "*Research by communication professors Jonathon Obar of York University in Toronto and Anne Oeldorf-Hirsch of the University of Connecticut showed that only a quarter of people bother to look at Terms & Conditions (T&Cs), with the vast majority spending less than a minute poring over the complex language and conditions set out within. This is concerning given people blindly trust there is nothing unfavourable in the T&Cs, but it's hardly surprising*." Some applications or software services have convoluted TOR / T&Cs. A convoluted T&Cs can potentially cause trouble to the end-user because it demoralizes the end-user to read thousands of complex languages. According to Patar (2019), "*Two law professors analyzed the sign-in terms and conditions of 500 popular US websites, including Google and Facebook, and found that more than 99 percent of them were "unreadable," far exceeding the level most American adults read at, but are still enforced.*" Additionally, he stated that "*Take Fteja v. Facebook, a 2012 New York district court case, in which a Facebook user alleged that his account was closed because of his religion and ethnicity. What the claimant didn't know, because he didn't read the contract, was that Facebook has a clause that states any claim against the company must be resolved in Santa Clara County, California*. *Ultimately, the judge transferred the case, stating that "failure to read a contract before agreeing to its terms does not relieve a party of its obligations under the contract.*" If an end-user fails to comply with the TOR of a specific application or software, then end-user may face copyright infringement penalties. According to Kenton (2022), "*Copyright infringement is the use or production of copyright-protected material without the permission of the copyright holder*." If an end-user uses a copyrighted graphic asset and sells or offer it as a service then the end-user's action can be considered illegal. According to Cass (2009), "*Characters out of a copyrighted typeface (scalable fonts are copyrightable) is classified as "mere lettering" and it is "independent of the overall utilitarian shape" which means that it exists separately from the typeface itself, therefore making it illegal to sell*." Fonts and other graphic design assets that are publicize as free, doesn't mean an end-user can use it for any purposes. "*If an image is labeled as 'royalty free,' that does not mean it's free for anyone to use. This is because the 'free' in a royalty-free means that there is a one-time paid license to use the image. Royalty-free means that those who pay for the license can use the image without also paying any additional fees (or royalties) based on the number of times it is used*." (Copyright Infringement Penalties (n.d.)). According to Graham (2017), "*Artist Richard Prince used dozens of unlicensed photos for his project "Canal Zone". The photographer, Patrick Cariou, filed suit for copyright, which Prince countered with a "fair use" defense. The Southern District of New York said the photos were an infringement, as Prince wasn't "commenting upon" the original works*." According to Peterson (2021), "*A writer for the online publication BuzzFeed used a photo, captured by photographer Gregory Mango for the New York Post, without his permission, and then replaced Mango's photo credit with the name of a law firm related to the story itself*."

Review of Related Studies

1. Intellectual Property for Graphic Designers and Visual Artists

Copyright law for graphic design is not properly regulated, anyone may use a graphical asset without giving attribution to the owner. **“***Graphic designers and visual artists can leverage intellectual property (IP) rights to protect and maximize the value of their creative works. With the vast amounts of digital content being shared online today, there’s a high likelihood that others can use your work without giving you proper credit.***”** (IP for graphic designers (2021)).

2. Licensing in Product Design

Graphical assets that are marketed as free don’t mean it is totally free. All of them can be one of these types of licenses: non-exclusive and exclusive license rights. According to Dettoni (n.d.), “*Non-exclusive License A non-exclusive license is one that it is not exclusive to you or anyone else. Everyone has the right to use that design on any product and in any market. Exclusive License An exclusive license is one in which the design is unique to you. You have proprietary rights to use the design on any product and in any market, but you do not own the copyrights*.”

3. Minimalism in logo design

Most of the logo designs of companies nowadays utilize minimalism as their art style. According to Crutchfield (2018), “*many companies eventually modernized their logos, which really means giving them a minimalist makeover. Minimalism is a design approach that utilizes existing elements, maximizing simplicity and capitalizing on space.*

Review of Related Systems

1. Canva

"*Canva is a free-to-use online graphical tool that combines design, photo-editing, and layout to help teachers and students create beautifully finished projects. Canva can be used by teachers to generate posters, guidance, and projects, but it can also be accessed by students for tasks*." (Edwards (2021)).

2. The Adobe Creative Cloud Express Logo Maker

"*The Adobe Creative Cloud Express logo maker is instant, intuitive, and intelligent. With a dash of science and a hint of magic, our logo maker brings you AI-generated logos with countless customization possibilities.*" (Make logos for free online (n.d.))

3. Logo Maker

"Logo Maker started in the early 2000's as a means to help business owners design a company logo**.** With online logo maker you can create, edit, and save as many logos as you like**"** (Make a logo design in minutes (n.d.))

4. Tailor Brands Logo Maker

"*Tailor Brands is such an easy-to-use tool, anyone can create their own logo and branding with it. Best part, it incorporates current trends in their logo designs and gives great inspiration*." (Easily Make a Unique Logo (n.d.))

5. Hatchful

"*Using Hatchful's free logo generator, you can create a custom logo by choosing from hundreds of professionally-designed templates. The logo creator will help you choose from countless industries, each with their own specialized logo style. Once you create a design with the logo maker tool, you'll get instant access to your own library of logo files.*" (Make a logo for free (n.d.))

Synthesis

What makes Swift Art different from other free logo maker applications is that it doesn't contain any third-party elements like images and fonts from other resources. It is a standalone application that has all the assets needed in doing its task. The main objective in developing Swift Art is to make it accessible for anyone to use in any purposes (excluding trademark) without any convoluted licensing agreements. The core mechanic of Swift Art is its auto-generation of logo designs. This mechanic saves the end-user from the hassle of making a logo systematically. Another mechanic of the application is its editor feature. The editor feature gives creative freedom to the end-user as it allows them to modify the generated design. The proponents will employ minimalism art as its core foundation for the logo design and the overall graphics of the application interfaces. The generated logo is under non-exclusive rights, therefore the end-user can use it for personal, commercial, and small projects but it cannot be trademarked.

Swift Art: An Auto-Generated 2D Raster Logo Editor Software

Overview of the Project

Swift Art is a 2D graphic design mobile application, specifically made for logo designing. The proponents develop the said application for people who wants to create a logo quickly without worrying about convoluted license restrictions. The activities of the application is divided into two (2) phases; inquiry and processing phase. The inquiry phase is where the application asks the end-users for information about preferred sector and design name. In the processing phase, the application generates a logo design according to the inputted value from inquiry phase. Also, the processing phase allows the end-user to modify the generated design. After satisfaction is met, the end-user can export the design as PNG or JPG image format.

Hardware/Software

Hardware

1. The proponents will use a laptop to utilize the said software below

Software

1. The proponents will use Unity Engine for developing and building the application.
2. The proponents will use Affinity Designer for making the graphical assets.
3. The proponents will use Visual Studio Code for writing the code.

Methodology .

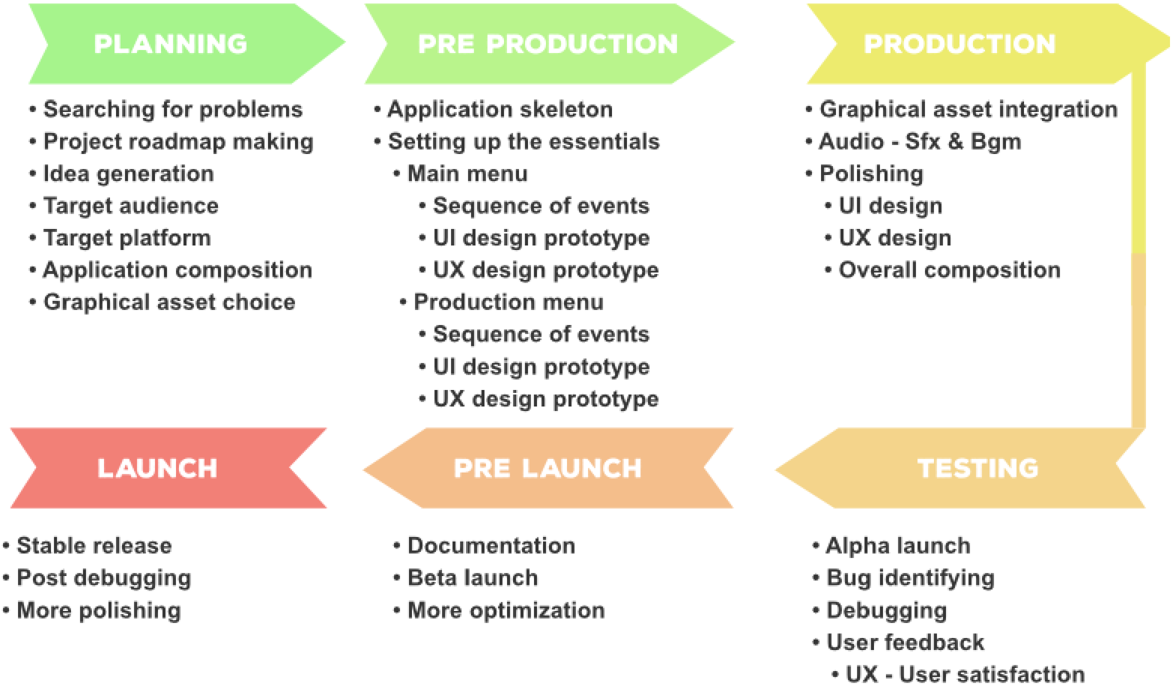


Figure 1.0. Development Phases

The proponents will utilize the Waterfall Development Model. The said model provides a comprehensible development guideline and is not difficult to follow and execute. According to Lewis (n.d.) "*The waterfall model is a linear, sequential approach to the software development life cycle (SDLC) that is popular in software engineering and product development. The waterfall model emphasizes a logical progression of steps. Similar to the direction water flows over the edge of a cliff, distinct endpoints or goals are set for each phase of development and cannot be revisited after completion*."

The Waterfall Development Model is divided into six (6) phases; planning, pre-production, production, testing, pre-launch, and launch. The first phase (planning) is where the proponents finalize an idea and create a development roadmap according to that idea. In this phase, the proponents have chosen their target audience & platform, composition for the application, and graphic assets. The second phase (pre-production) is where the proponents apply or execute the plans from the first phase. In this phase, the proponents will deal with setting up the backbone of the application or its main functionality. The goal in this phase is to build a prototype as early as possible. The third phase (production) is where the proponents further develop the application, specifically adding audio effects, graphical assets, and animation. This phase is where the overall polishing occurs, like optimizing the application performance and cleaning up the composition. The fourth phase (testing) is where the proponents identify known bugs in the application and seek feedback from end-users. The purpose of this phase is to improve the performance and user experience of the application. The fifth phase (pre-launch) is where the proponents release the application to a limited audience. The goal in this phase is to ensure that the improvements done in the testing or fourth phase are applied. The sixth phase (launch) is where the proponents build and deploy the stable version of the application in the selected platform.

System Design Specification

Swift Art's logical architecture is called Sequence of Events (SOE). A sequence of events (SOE) is a series of actions called states. The application consists of four (4) states that are organized into two (2) phases: Inquiry Phase and Processing Phase. The inquiry phase is responsible for listening and taking the inputs from the end-user. The processing phase is responsible for the data taken from the inquiry phase and using it to generate a design according to the preference of the user.

The proponents have also added one (1) feature for the processing phase, and that is the editor feature. The editor allows the end-user to modify the generated design. This feature allows creative freedom for the end-user. Below is a flowchart of how the SOE of the application works. The proponents have recommended to the readers to first understand the Detailed Sequence of Events (DSOE) of the application. The DSOE is a separate file, attached in the zip file.

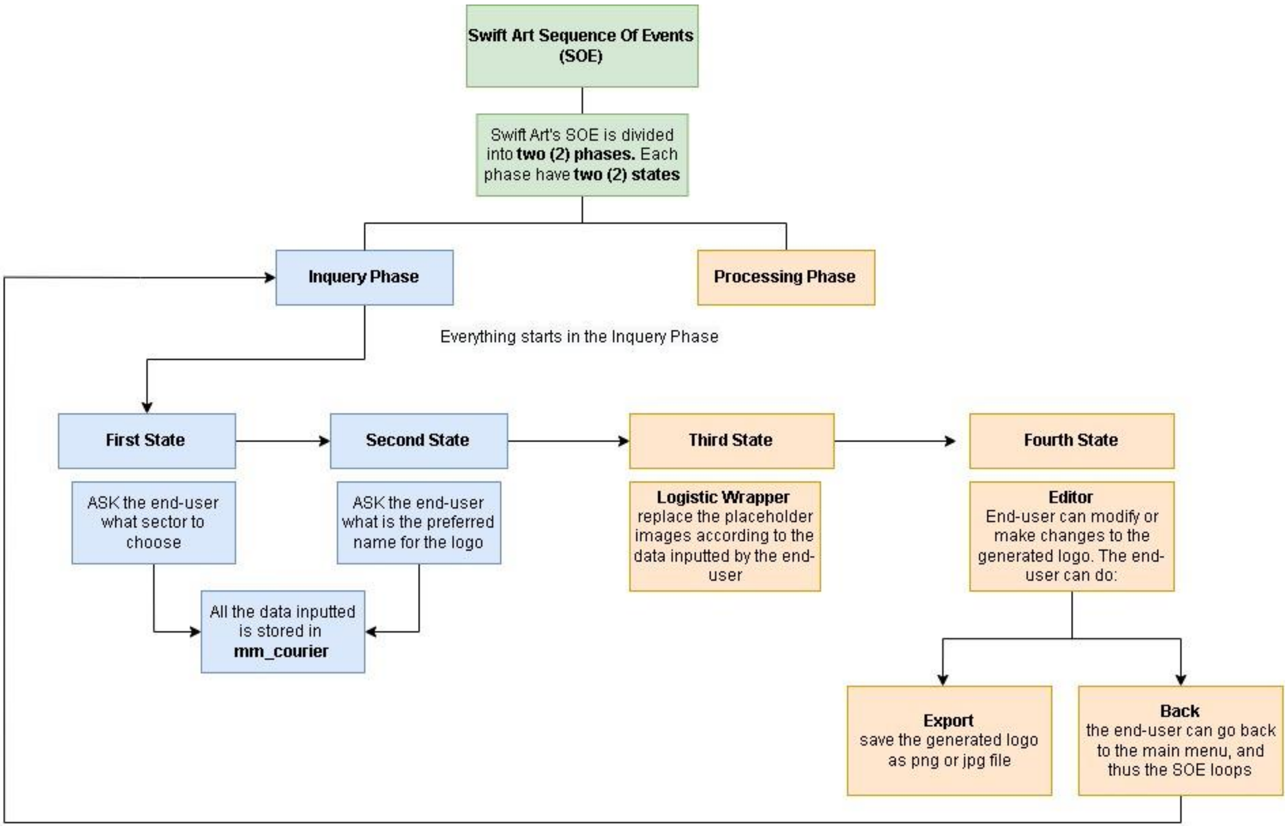

Swift Art Sequence Of Events (SOE)
Swift Art's SOE is divided into two (2) phases. Each phase have two (2) states
Inquery Phase
Processing Phase
Everything starts in the Inquery Phase
First State
ASK the end-user what sector to choose
Second State
ASK the end-user what is the preferred name for the logo
All the data inputted is stored in mm_courier
Third State
Logistic Wrapper replace the placeholder images according to the data inputted by the end-user
Fourth State
Editor End-user can modify or make changes to the generated logo. The end-user can do:
Export save the generated logo as png or jpg file
Back the end-user can go back to the main menu, and thus the SOE loops

Figure 2.0. Sequence Of Events

Swift Art will be built for the Android platform and will be deployed in the Google Play Store. The application's minimum API level is API Level 19 (Android 4.4) and the target API level is API Level 29 (Android 10). The proponents chose this API level because it is according to the recommended or required level of the Google Play Store. Below is a flowchart about the flow of the build and deployment of the application.

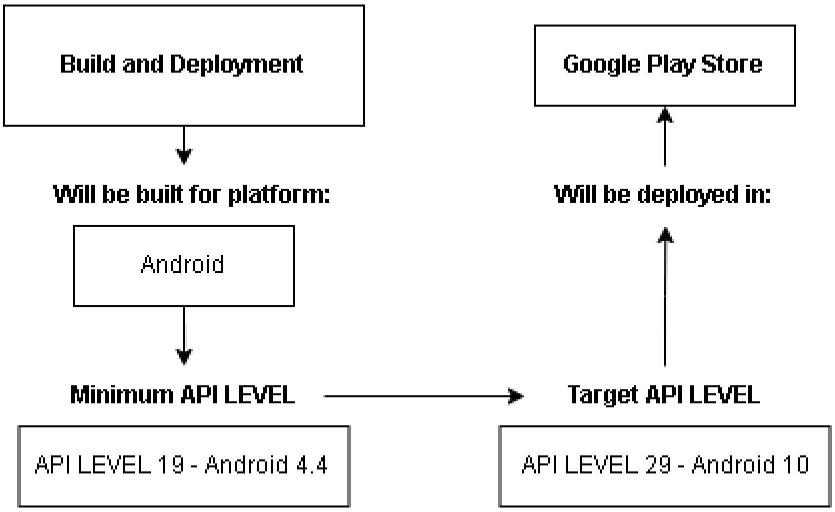


Figure 3.0. Build and Deployment

Data Flow Diagram

The proponents will use a non-relational database, specifically JSON. "*A JavaScript Object Notation (JSON) database is arguably the most popular category in the NoSQL family of databases.*" (Morris, (2021)). The relational database is not suitable or practical for the application because it doesn't need to store large and complex data. The application will only store one data, and that is the **FirstTime** object. FirstTime is an object from class Authenticator. The class authenticator contains one property, and that is **isTrue**. The figure in the next page explains the flow of how the application stores data.

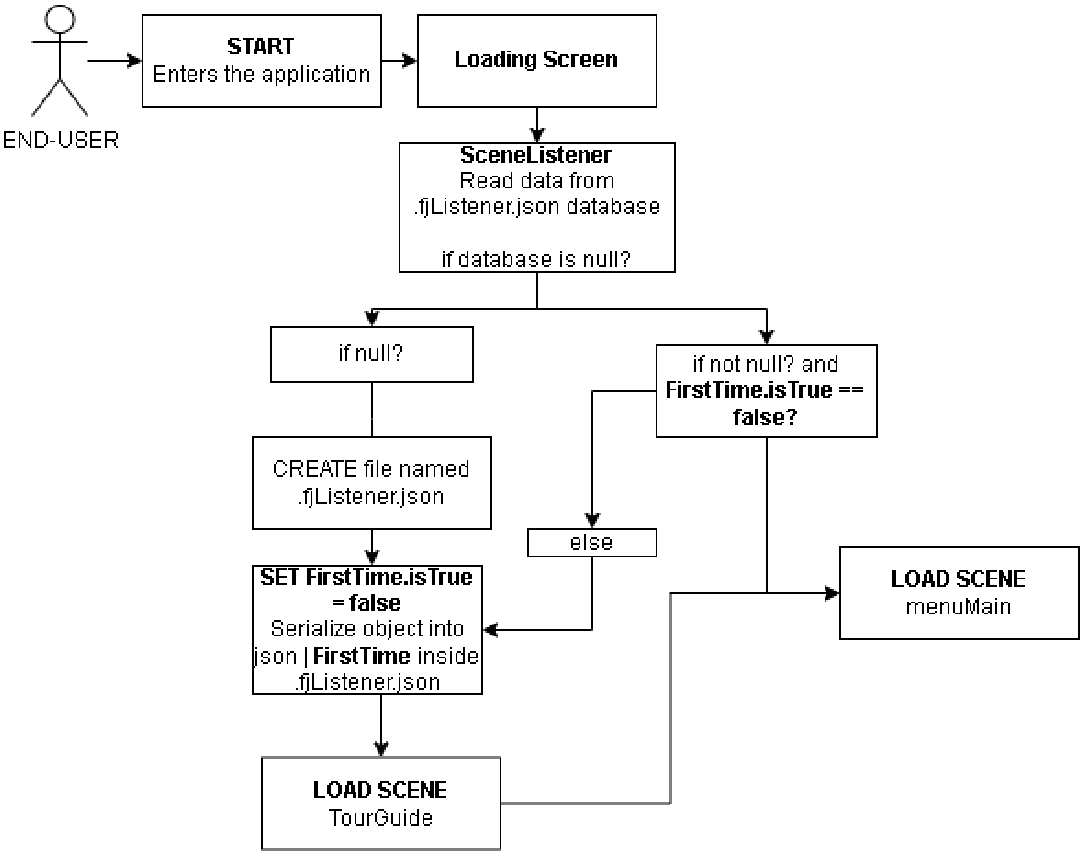
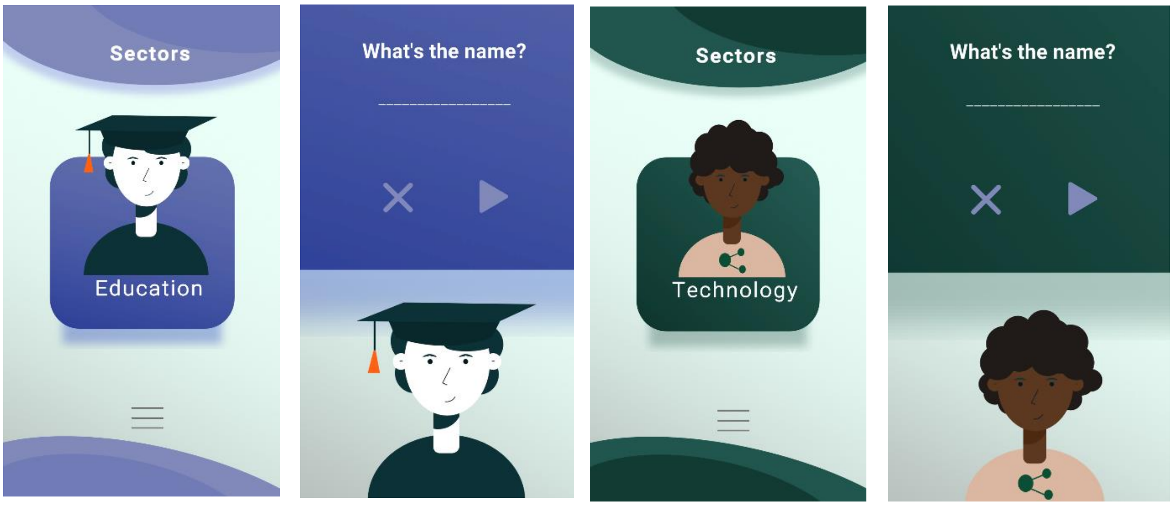


Figure 4.0 Scene Listener Data

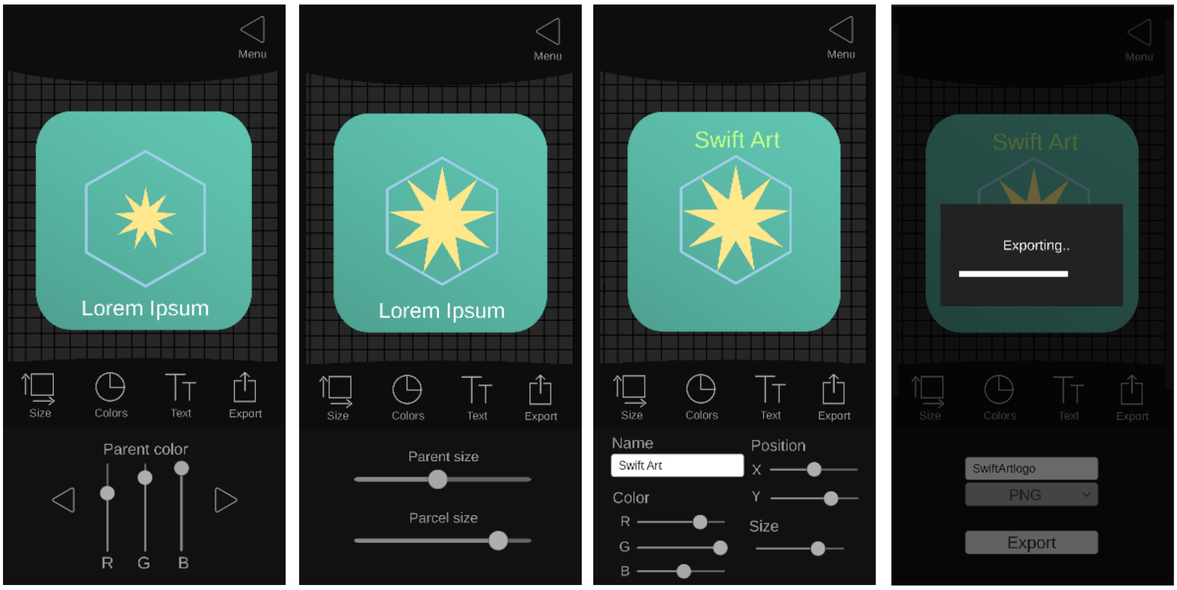
Swift Art Graphical User Interface Architecture

Swift Art's graphical user interface (GUI) consists of two (2) modules. Similar to the phases of the sequence of events (SOE) that is mentioned earlier, the GUI is divided into two (2) modules, the inquiry module and the processing module. The said modules have the same function as the phases from SOE, the only difference is that the modules are the execution or practical use from those phases.

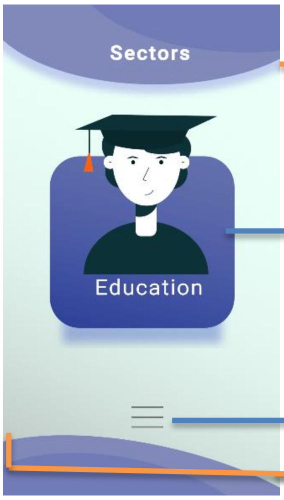
Inquiry Module



Processing Module



Inquiry Module



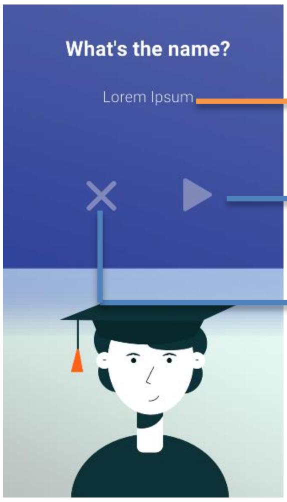
First State

Listener Area| This area listens for user inputs. This is responsible for the swipe mechanic

Button | Executes a void method that changes the animation state of the overall GUI objects

Button | Executes a void method that makes the bottom-bar appear.

The first state allows end-users to select their preferred sector. Currently, Swift Art offers 6 sectors. These are: Agriculture, Animals, Business, Education, Technology, and Transport



Second State

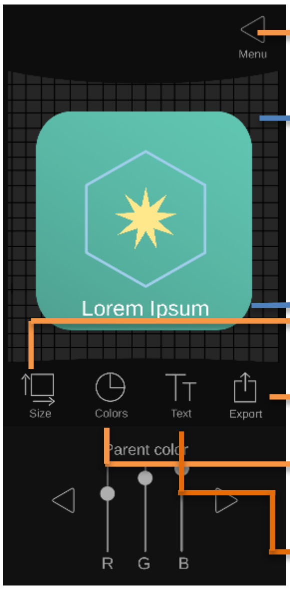
Input Field | Allow end-users to enter any string value. This is used for getting the end-user's preferred name for the design

Button | Executes a void method that stores the inputted values and transitions the scene to the processing method.

Button | Executes a void method that reverts back to the first state

The second state allows end-users to input their preferred name for the design. Also, if the end-user want to change the sector, there is a revert button that assists the end-user for that.

Processing Module



Third State

Button | Executes a void method that reverts the scene back to the first state of the Inquiry Division

The Logo Object | This is the generated logo design. Its properties are modifiable.

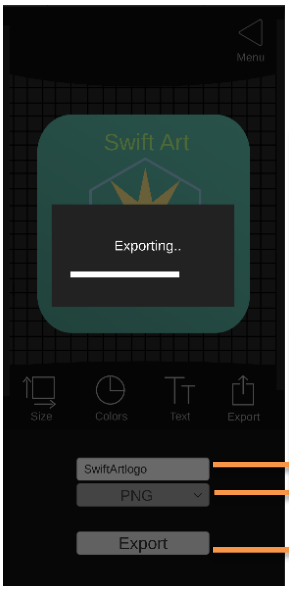
Button | Opens up the user interface for the sizing options.

Button | Opens up the user interface for exporting the design as png or jpg.

Button | Opens up the user interface for the color options.

Button | Opens up the user interface for the text options.

The third state reads the data taken from the inquiry phase and uses it for generating a design that is under the preference of the end-user. Third state isn't the state where an end-user can modify the design. It is in the fourth state.



Fourth State

Dropdown | Shows a list of image types that an end-user can select

Input Field | Allows end-user to input any string value for the filename.

Button | Executes a void method that exports the design as image file type.

The fourth state allows the end-users to modify the generated design. The sizes, position, and color can be modified using the tools highlighted in the third state. The fourth state is also where an end-user can export the design and can revert to the first state of the Inquiry phase.