**Software Requirements Specification**

**for**

**<**Sprite Editor**>**

**Prepared by <IG.WXZ>**

**<2018.10.30>**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 3**

1.1 Purpose 3

1.4 Product Scope 3

1.5 References 3

**2. Overall Description 3**

2.1 Product Perspective 3

2.2 Product Functions 3

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

**3. External Interface Requirements 4**

3.1 User Interfaces 4

3.2 Hardware Interfaces 4

3.3 Software Interfaces 4

**4. System Features 5**

4.1 Drawable canvas 5

4.2 Preview Frames 5

4.3 Save/Load/Export 5

# **Introduction**

## **Purpose**

*This Sprite Editor is aimed to help artists to create easy and simple animations under a pixel-based canvas. Most likely to be used in simple 2D game creation, 2D animation, or just expressing crazy thoughts with the editor!*

## **Product Scope**

*The Sprite Editor is build to make animators' life easier, with simple clicks on the mouse to create a well-formed animation. Tools are made to be simple to understand and to use. Exporting is fast, the development area is light, can do creations anytime and anywhere.*

## **References**

*https://www.piskelapp.com/*

# **Overall Description**

## **Product Perspective**

*This editor is aimed to be a overwrite of the existing market editors or at least a supplement of them. With the convention that the program is light-weighted, we want the product to be accessible anytime and anywhere.*

## **Product Functions**

*Able to draw on a pixel-based canvas, able to copy frames, able to adjust frame per second, able to change the color of the brush, able to save/load projects, able to "smooth" the lines, able to draw shapes using shape tools, able to export as a gif.*

## **User Classes and Characteristics**

*This product should be most often used by 2D game creators, 2D animation creators, and any other 2D artists. However, it's not limited to the bounds, everyone who likes drawing and animating should have the ability and accessibility to the product. It's built for everyone.*

## **Operating Environment**

*The first version of the product is supported only on Linux Ubuntu System.*

## **Design and Implementation Constraints**

*Due to that this is an in-semester project and the time and ability limitation, our team might or might not finish all the functions that are listed, but we are sure to finish the basic functions that will support the basic usages.*

## **User Documentation**

*github: https://github.com/kenmingwang/Spirte\_IG.WXZ*

## **Assumptions and Dependencies**

*The editor will NOT automatically generate animations that the user want, the user has to draw themselves. The user should also know that this product is running only on Ubuntu system, the mouse is necessary while the keyboard is not.*

# **External Interface Requirements**

## **User Interfaces**

*There will be a main window at most of the time, where toolkits are all within the main window. Within the main window, there will be a canvas to draw, will be a frame list to adjust, will be a list of toolbar to help drawing, will be a small preview window, will be a save/load buttons.*

## **Hardware Interfaces**

*The product is mostly interacting with the mouse, thus a mouse or a keypad is necessary for using the product.*

## **Software Interfaces**

*Most functions will be gathered in the "FRAME" class, which is the class that represents the canvas that the user is drawing on. There will be a front-end UI class which interacts with the frame class, and possibly sub-functional classes to be used to support other functions of the frame class.*

# **System Features**

## Drawable canvas

4.1.1 Description and Priority

*This is the main feature of the product, the usage frequency is HIGH, and it should be the first thing to implement.*

4.1.2 Stimulus/Response Sequences

*The canvas will respond as soon as the user draws on the canvas, in the way of updating the canvas itself.*

4.1.3 Functional Requirements

*Users can see immediately what they drew and are able to erase what they drew, or start a new canvas, copy a new canvas, change the color of the brush, and to smooth the lines of the canvas.*

## Preview frames

4.2.1 Description and Priority

*This is the main feature of the product, the usage frequency is HIGH, should be implemented right after the DRAWABLE CANVAS.*

4.2.2 Stimulus/Response Sequences

*There should be a auto-play screen whereas the user updates the frames the screen should play automatically based on the user-desired frame.*

4.2.3 Functional Requirements

*The user should not need to click on anything for the preview, if needed to click, then it should be when the user wants to adjust the frame per second, or the full-size play the preview.*

## Save/Load/Export

4.2.1 Description and Priority

*This is the main feature of the product, the usage frequency is MEDIUM.*

4.2.2 Stimulus/Response Sequences

Users interact with the buttons to utilize the save/load/export function.

4.2.3 Functional Requirements

*There should be a recognizable button for users to click on for saving/loading/export, where all of them should pop up a file selection screen asking for where/which place to save/load from/export. Then produce corresponding output.*