Software Requirements Specification

for

A7: Sprite Editor

Version 1.0 approved

Prepared by Cole Hanson, Harry Kim, Cory Theel, Braden Morfin, and Joshua Taylor

Team Dolphins

November 9, 2021

Table of Contents

Table of Contents ii

1. Introduction 1
- 1.1 Purpose 1
- 1.2 Document Conventions 1
- 1.3 Intended Audience and Reading Suggestions 1
- 1.4 Product Scope 1
- 1.5 References 1

2. Overall Description 1
- 2.1 Product Perspective 1
- 2.2 Product Functions 2
- 2.3 User Classes and Characteristics 2
- 2.4 Operating Environment 2
- 2.5 Design and Implementation Constraints 2
- 2.6 User Documentation 2
- 2.7 Assumptions and Dependencies 2

3. External Interface Requirements 3
- 3.1 User Interfaces 3
- 3.2 Hardware Interfaces 4
- 3.3 Software Interfaces 4

4. System Features 4
- 4.1 Sprite Display 4
- 4.2 Save and Load 5
- 4.3 Frames 5
- 4.4 Color Picker 5
- 4.5 Draw Tool 6
- 4.6 Eraser 6
- 4.7 Eyedrop Tool 6
- 4.8 Zoom Tool 7
- 4.9 Negative Tool 7

1. Introduction

1.1 Purpose

The product this document goes over is a sprite editor program. This document covers the initial release of the product. This document also covers the entirety of the program, and this is version 1.0 of this document.

1.2 Document Conventions

Section 2 of this document (overall description) uses bullet points to list features, with the order of the bullets having no significance. Section 3 (external interface requirements) uses numbered lists, with sublists ordered by letter. Lists are numbered by letters belonging to the number that precedes it; this is also apparent through indentation.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by developers as well as university staff. The rest of this document contains the overall description of the product, external interface requirements, and additional system features. It is suggested to read the document from start to end (section 1, section 2, etc.).

1.4 Product Scope

The purpose of the product in this document is to allow users to create and edit sprites, as well as save and load them. This includes the requirements given in the assignment handout.The required features from the handout include a brush, sprite preview animation, multiple frames, save and load, eyedrop tool, eraser, negative tool, and zoom function.

1.5 References

Assignment handout page -
https://utah.instructure.com/courses/718555/assignments/9944968?module_item_id=16126017

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained product that will be used to create and edit sprites. This product is not related to, or is a member of, any other currently existing product.

2.2 Product Functions

This product's main and most important function is to be able to create and edit sprites. Users must be able to:

- Set the size in pixels of the sprite.
- Adjust the number of frames for the sprite animation.
- Modify the pixels of a sprite.
- Set a pixel to a specified color.
- View a preview of the sprite animation cycle.
- Adjust the frames per second of playback.
- Have an option of seeing the playback at the actual size of the sprite.
- Save and load a project.

2.3 User Classes and Characteristics

The user class for this product will most likely be hobbyists that only require a simple sprite editor in order to fulfil their needs as opposed to more professional users that require complex and advanced tools most users would not have use for. This product will be designed for ease of use and comprehensibility. Users are expected to be at least familiar with any form of drawing program in order to make effective use of our product.

2.4 Operating Environment

The product will be expected to be able to operate on operating systems that can run executables of code such as, but not limited to, Windows, Linux, and macOS. The product should be able to run without any need of a co-running program or application.

2.5 Design and Implementation Constraints

Language and Technology Requirements:

- C++ 11
- QT Creator 5.0.2
- Windows, Linux, or macOS Computer

2.6 User Documentation

No user documentation will need to be delivered with software.

2.7 Assumptions and Dependencies

Assumed factors that may affect the requirements stated in the SRS:

- Motivation
- Time constraints
- Scheduling conflicts
- Unexpected update to the system(s) currently being used to build this product

- Unexpected natural disasters

3. External Interface Requirements

3.1 User Interfaces

The behavior of the GUI for the sprite editor is detailed in this section.

1. A Panel with a frame for where the sprite is edited/created
    a. Allows user to manipulate the sprite with current selected tool
2. Scrollable panel which allows users to select which frame in the current working space they are editing. Include the following buttons:
    a. Add frame button
        i. Adds a new editable frame to current working space
    b. Delete frame button
        i. Removes current frame
    c. Duplicate frame button
        i. Copy current frame and adds it as a new frame
3. Toolbar with frame manipulation tools as selectable buttons
    a. Brush Tool
        i. Add/replaces a pixel(s) of a user defined color and opacity values
        ii. Users can change the amount of pixels that the brush added/replaces on the frame with tool
    b. Erase Tool
        i. Replaces a pixel(s) with a fully transparent pixel
        ii. Users can change the amount of pixels that are replaced on the frame with this tool
    c. Negative Tool
        i. Inverts selected pixel(s)
        ii. Users can change the amount of pixels that are inverted on the frame with this tool
    d. Zoom Tool
        i. Allows user to move and zoom around the frame
4. Animation Playback of the frames
    a. Cycles through the frames in current working space at user set speed
    b. A slider which users can change the speed at which frames are cycled through
5. File Options
    a. Save file button:
        i. Saves frames in the current working space as a .ssp file
    b. Load file button:
        i. Loads frames from a selected .ssp file
    c. Create a new frame:
        i. Creates a new frame from user inputed height and width in pixels, overriding current working space
    d. A brush size and color options

3.2 Hardware Interfaces

In this section covers the details of hardware interfaces that can interact with the editor

1. Supported devices:
    a. Devices running macOS, windows, or linux and able to run C++ 11

3.3 Software Interfaces

Section pertaining to the softwares that are to be used.

1. C++ 11
    a. Classes and methods are to be written in this language
2. QT creator 5.0.2 used to create, connect, and edit the GUI
    a. Includes QT libraries for C++
        i. Allows manipulation of QT UI objects
    b. Allows manipulation of C++ scripts
    c. Has an interface which allows GUI manipulation/setup
3. The editor should run on the following operating systems:
    a. Windows
    b. Linux
    c. MacOS
4. Supports .ssp files, which are encoded with json data structure

4. System Features

4.1 Sprite Display

4.1.1 Priority/Description

High. Provides a visual display of the current sprite that is being edited. Also is used to view a sprite's animation.

4.1.2 Stimulus/Response Sequences

Sprite Updated > Updates view to reflect the current state of the sprite.
Play Animation Request > Loops continually loops through frames and disallows the user to edit the sprites during the loop.
Stop Animation Request > Stops the loop and reverts the sprite to the frame that was selected prior to the animation start.

4.1.3 Functional Requirements

1) Displays a sprite on the screen.

2) Allows for easy editing of the sprite.

4.2 Save/Load

4.2.1 Priority/Description

High. Provides the user the ability to save and load sprites.

4.2.2 Stimulus/Response Sequences

Save Request > Saves the current sprite (all frames) into a .ssp file, which uses json data structure.
Load Request > Loads a .ssp file into the viewer (with all frames).

4.2.3 Functional Requirements

1) Stores all information about the sprite that is relevant to viewing eg. pixel values and all frames.

4.3 Frames

4.3.1 Priority/Description

High. Provides the user the ability to have multiple “states” of a sprite. These could be used in an animation or as different states of the sprite.

4.3.2 Stimulus/Response Sequences

Add Frame > Adds a frame to the list of sprite frames.
Delete Frame > Deletes the frame that is currently selected.
Copy Frame > Adds a copy of the current frame to the list of frames.

4.3.3 Functional Requirements

1) Frames should be distinct (edits to one do not affect another).
2) Include a copy frame button that copies the currently selected frame.

4.4 Color Picker

4.4.1 Priority/Description

Medium. Allows the user to specify what color of pixel they want to paint with.

4.4.2 Stimulus/Response Sequences

Choose color > Any painting will be with that color.

4.4.3 Functional Requirements

1) Allows the user to specify the R, G, B, and A values of a color.

4.5 Draw Tool

4.5.1 Priority/Description

High. Allows the user to edit the value of pixels in the view.

4.5.2 Stimulus/Response Sequences

Click > changes a pixel in the view.

4.5.3 Functional Requirements

1) Button in view to select this tool.

4.6 Eraser Tool

4.6.1 Priority/Description

Medium. Allows users to erase pixels in the view.

4.6.2 Stimulus/Response Sequences

Click > changes a pixel in the view to be transparent.

4.6.3 Functional Requirements

1) Button in view to select this tool.

4.7 Eyedrop Tool

4.7.1 Priority/Description

Low. Allows users to select a color that is already present in the sprite.

4.7.2 Stimulus/Response Sequences

Click > sets the color of the draw tool to be that of the clicked pixel.

4.7.3 Functional Requirements

1) Button in view to select this tool.

4.8 Zoom Tool

4.8.1 Priority/Description

Low. Allows the user to zoom into or out of a sprite so it is easier to edit.

4.8.2 Stimulus/Response Sequences

Click > toggle zoomed into sprite
Mouse Wheel Up > Zoom In.
Mouse Wheel Down > Zoom Down.

4.8.3 Functional Requirements

1) Minimum zoom of 10%.
2) Maximum zoom of 500%.
3) Button in view to select this tool.

4.9 Negative Tool

4.9.1 Priority/Description

Low. Similar to the draw tool, except that it will cause the pixel to become the negative of itself.

4.9.2 Stimulus/Response Sequences

Click > changes a pixel in the view.

4.9.3 Functional Requirements

1) Button in view to select this tool.