Software Requirements Specification

for

CS3505 Sprite Editor

Version 1.0

CS3505: Software Practice II

3rice1potato

Thu Ha

Phuc Hoang

Trenton Stratton

Chanphone Visathip

Table of Contents

1. Introduction

1.1 Purpose

This document specifies the software requirements for the Sprite Editor application. It outlines the core functionalities, expected behavior and constraints of the software.

1.2 Document Conventions

This SRS follows standard conventions for formatting and organization. Requirements are identified and prioritized as High, Medium, or Low.

1.3 Intended Audience and Reading Suggestions

The document is intended for CS professors and students.

1.4 Product Scope

The Sprite Editor is a software tool designed for creating and previewing sprite animations. It includes features like animation previews, frame rate adjustments, basic tools for sprite editing, and file management functions for saving and loading projects.

1.5 References

No external references are included.

2. Overall Description

2.1 Product Perspective

The Sprite Editor is developed using Qt framework and provides an environment for creating and animating sprites. It allows users to draw, edit, and animate sprite frames, offering some features like a paint brush, erase, undo functionality, frame management, and setting sprite dimensions. The Sprite Editor is a stand alone application and not part of a larger system.

2.2 Product Functions

- Save the current project to a user-defined file location.
- Load saved animation files from the user's file structure.
- Alter the pixels of the sprite animation with custom picked colors, using various tools.
- Add frames to and remove frames from the sprite animation.
- Display a live preview of animations.
- Adjust the animation preview frame rate.

2.3 User Classes and Characteristics

The software is intended for artists and animators who need an intuitive interface for creating sprite-based animations. Users are expected to have basic computer literacy but do not require advanced technical knowledge.

2.4 Operating Environment

The Sprite Editor will run on modern operating systems, including Windows and macOS. The software should be compatible with standard file management systems.

2.5 Design and Implementation Constraints

- The user interface should be intuitive, responsive, and visually clear.
- The toolset should be basic and lightweight in order to run smoothly on a wide variety of machines

2.6 User Documentation

User documentation will include a feature overview and a simple user guide.

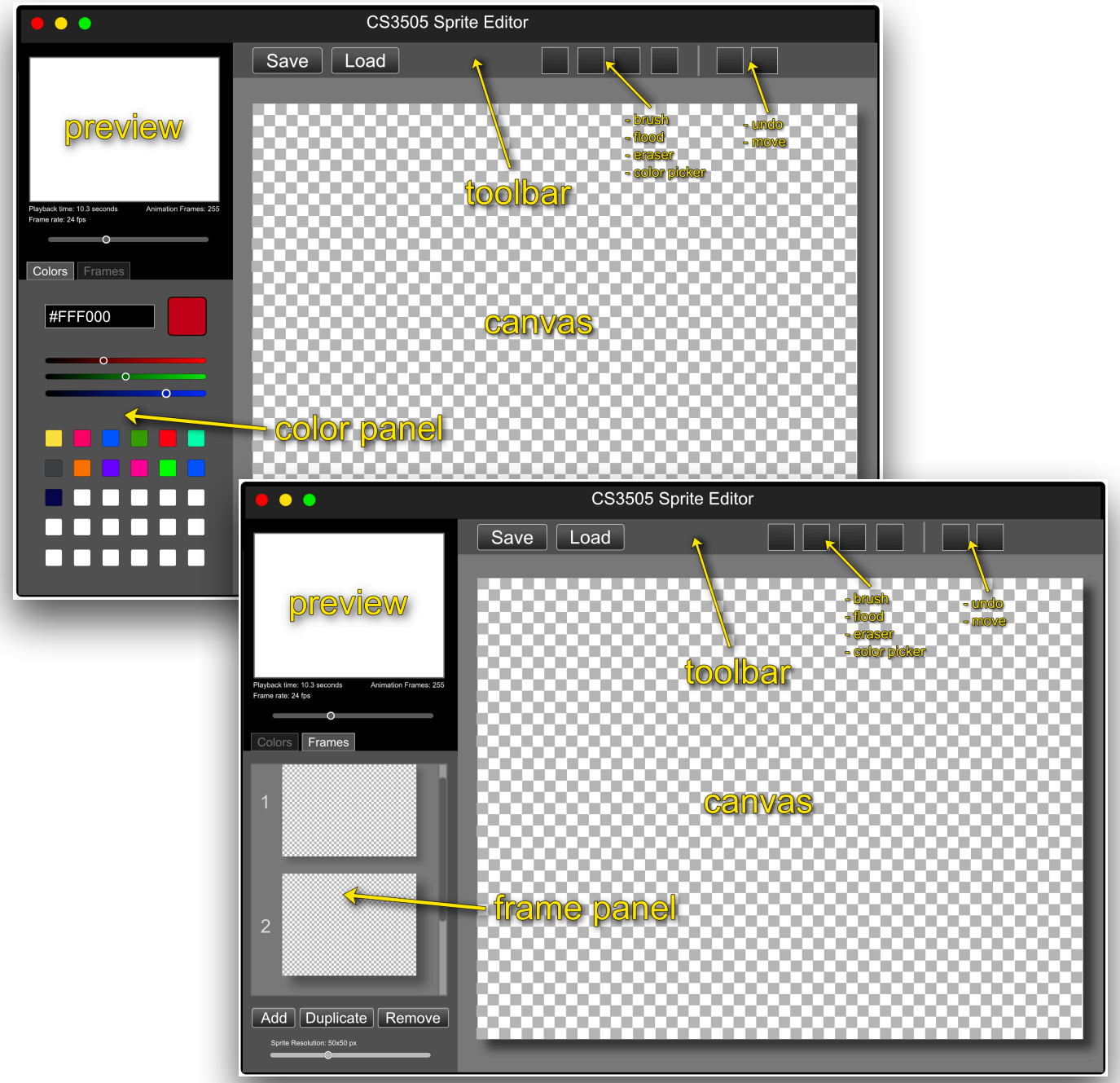
2.7 Assumptions and Dependencies

- Users will have access to a computer with sufficient resources to run the software smoothly.
- The software will rely on the system for file management.

3. External Interface Requirements

3.1 User Interfaces

The CS3505 Sprite Editor provides an intuitive graphical user interface (GUI) designed to facilitate easy and efficient sprite editing for users. The layout comprises key components arranged for clarity and ease of access: a Preview Panel, Toolbar, Color Panel, Frame Panel, and a central Canvas for drawing and editing.



program concept figures (subject to change)

3.2 Hardware Interfaces

Standard computer hardware is required (i.e. mouse, keyboard, and display).

3.3 Software Interfaces

The software will interface with the operating system's file management system for saving and loading files.

3.4 Communications Interfaces

No network communication features are required.

4. System Features

4.1 Save Functionality

4.1.1 Description and Priority

A button to save the current project.
Priority: High

4.1.2 Stimulus/Response Sequences

User clicks "Save" → File dialog opens for saving the project.

4.1.3 Functional Requirements

- **REQ-1**: Must open a file dialog to let the user specify a file location.
- **REQ-2**: Must confirm file save success or failure.

4.2 Load Functionality

4.2.1 Description and Priority

A button to load a saved project.
Priority: High

4.2.2 Stimulus/Response Sequences

User clicks "Load" or “Open” → File dialog opens to select a file.

4.2.3 Functional Requirements

- **REQ-1**: Must open a file dialog to let the user select a file.
- **REQ-2**: Must inform the user of an invalid file type.
- **REQ-3**: Must load the valid selected animation file.

4.3 Set Sprite Dimensions

4.3.1 Description and Priority

Set or adjust the dimensions of the sprite (width and height in pixels).
Priority: Medium

4.3.2 Stimulus/Response Sequences

User adjusts dimensions via a dialog box or setting panel.

4.3.3 Functional Requirements

- **REQ-1**: Users must be able to input dimensions with defined minimum and maximum limits.
- **REQ-2**: Warn users if sprite content may be affected/lost when reducing dimensions.
- **REQ-3**: Add blank canvas space when increasing dimensions.

4.4 Animation Preview

4.4.1 Description and Priority

Displays a real-time preview of sprite animations.
Priority: High

4.4.2 Stimulus/Response Sequences

User adds or modifies frames → Animation preview updates automatically.

4.4.3 Functional Requirements

- **REQ-1**: The animation preview must update in real-time.

4.5 Frame Rate Control

4.5.1 Description and Priority

A slider to control the animation preview's frame rate.
Priority: Medium

4.5.2 Stimulus/Response Sequences

User adjusts the slider → Frame rate changes instantly, and the numeric value updates.

4.5.3 Functional Requirements

- **REQ-1**: Frame rate must be adjustable between predefined values.
- **REQ-2**: Display the current frame rate next to the slider.

4.6 Duplicate Frame

4.6.1 Description and Priority

Create an exact copy of the selected frame and place it after the original.
Priority: Medium

4.6.2 Stimulus/Response Sequences

User clicks "Duplicate Frame" → Copied frame is added, canvas and frame list updates.

4.6.3 Functional Requirements

- **REQ-1**: Ensure an exact copy of the frame.
- **REQ-2**: Update the frame list seamlessly and maintain animation flow.
- **REQ-3**: Provide visual feedback upon duplication.

4.7 Add Frame

4.7.1 Description and Priority

Add a new blank frame after the selected frame in the sequence.
Priority: High

4.7.2 Stimulus/Response Sequences

User clicks "Add Frame" → New frame is inserted, canvas and frame list updates.

4.7.3 Functional Requirements

- **REQ-1**: Ensure the new frame is always blank.
- **REQ-2**: Correctly update and reflect the new sequence.
- **REQ-3**: Provide visual feedback indicating the new frame.

4.8 Remove Frame

4.8.1 Description and Priority

Delete the currently selected frame and shift subsequent frames.
Priority: High

4.8.2 Stimulus/Response Sequences

User clicks "Remove Frame" → Frame is removed, and a warning appears for unsaved changes.

4.8.3 Functional Requirements

- **REQ-1**: Warn with an option to cancel deletion.
- **REQ-2**: Ensure no gaps in the animation sequence.

4.9 Brush Tool

4.9.1 Description and Priority

Allows the user to draw on the sprite canvas with customizable size, color, and opacity.
Priority: High

4.9.2 Stimulus/Response Sequences

User selects the Brush Tool → Cursor changes, options for pixel size adjustment appear. User draws on the canvas.

4.9.3 Functional Requirements

- **REQ-1**: Adjust tool pixel sizes (1x1, 3x3, 5x5, etc.).
- **REQ-2**: Change brush color using a color picker.
- **REQ-3**: Adjust opacity from 0% to 100%.
- **REQ-4**: Paint brushstrokes in real-time with low latency.
- **REQ-5**: Integrated undo function for brush operations.

4.10 Flood Tool

4.10.1 Description and Priority

Fills connected pixel areas with the chosen color.
Priority: Medium

4.10.2 Stimulus/Response Sequences

User selects the Flood Tool and clicks on an area in the canvas → Cursor changes and area is filled.

4.10.3 Functional Requirements

- **REQ-1**: Fill connected pixels with one click ("connected pixels" are of like color and adjacent, but not diagonal).
- **REQ-2**: Allow accuracy control for color matching.
- **REQ-3**: Fill action must complete in under 1 second for a 256x256 canvas.
- **REQ-4**: Integrated undo function for fill actions.

4.11 Eraser Tool

4.11.1 Description and Priority

Removes pixels from the canvas.
Priority: High

4.11.2 Stimulus/Response Sequences

User selects the Eraser Tool and drags it across the canvas → Cursor changes and pixels are removed according to size and shape settings.

4.11.3 Functional Requirements

- **REQ-1**: Adjust tool pixel sizes (1x1, 3x3, 5x5, etc.).
- **REQ-2**: Provide round and square eraser options.
- **REQ-3**: Integrated undo function for eraser actions.

4.12 Move Tool

4.12.1 Description and Priority

Helps users reposition parts of the canvas or selected regions.
Priority: Medium

4.12.2 Stimulus/Response Sequences

User selects a region and drags it → Cursor changes and the region moves in real-time.

4.12.3 Functional Requirements

- **REQ-1**: Select and drag regions on the canvas.
- **REQ-2**: Integrated undo function for movements.
- **REQ-3**: Indicate when an area is selected for movement.

4.13 Undo

4.13.1 Description and Priority

Allows users to revert the most recent change.
Priority: High

4.13.2 Stimulus/Response Sequences

User triggers "Undo" → Previous state is restored.

4.13.3 Functional Requirements

- **REQ-1**: Maintain an undo stack with a limited number of actions.
- **REQ-2**: Group logical consecutive actions to manage the stack efficiently.
- **REQ-3**: Efficiently handle memory usage.

4.14 Color Picker Tool

4.14.1 Description and Priority

Enables users to select a color from the canvas.
Priority: High

4.14.2 Stimulus/Response Sequences

User selects the Color Picker → Cursor changes to an eyedropper. Click captures the color.

4.14.3 Functional Requirements

- **REQ-1**: Activate tool upon icon selection.
- **REQ-2**: Capture pixel color value on click.
- **REQ-3**: Cancel tool when the Brush Tool is selected.

4.15 Color Sliders

4.15.1 Description and Priority

Allows individual color adjustments via RGB sliders.
Priority: High

4.15.2 Stimulus/Response Sequences

User adjusts sliders → Color preview updates in real-time.

4.15.3 Functional Requirements

- **REQ-1**: Provide sliders for each RGB channel.
- **REQ-2**: Update the color preview in real-time.
- **REQ-3**: Display corresponding HEX and/or RGB values.

4.16 Color Palette Customization

4.16.1 Description and Priority

Allows users to create and save custom colors.
Priority: Medium

4.16.2 Stimulus/Response Sequences

User adjusts and adds colors to the palette → System saves and displays the custom palette.

4.16.3 Functional Requirements

- **REQ-1**: Add selected colors to a custom palette.
- **REQ-2**: Delete colors from the palette.
- **REQ-3**: Edit colors in the palette.