

---

# **Software Requirements Specification**

**for**

## **PixGunga Master**

**Version 0.1**

**Prepared by Andrei Tokhtuev, Daniel Wong, Brendon Li, Seojun Chung, Max  
Sone**

**GungaLLC**

**11-2-2023**

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>1. Introduction</b>	<b>2</b>
1.1 Purpose	2
1.2 Intended Audience and Reading Suggestions	2
1.3 Product Scope	2
1.4 References	2
<b>2. Overall Description</b>	<b>3</b>
2.1 Product Perspective	3
2.2 Product Functions	3
2.3 Operating Environment	3
2.4 Design and Implementation Constraints	3
2.5 User Documentation	3
<b>3. External Interface Requirements</b>	<b>4</b>
3.1 User Interfaces	4
3.2 Hardware Interfaces	4
3.3 Software Interfaces	4
<b>4. System Features</b>	<b>5</b>
4.1 Sprite Pixel Placement	5
4.2 Erasing	5
4.3 Color Selection	6
4.4 Save/Load	7
4.5 Animation Cycle Preview	7
4.6 Frame Selector	8
4.7 Onion Skin	9
4.8 Drawing Shapes	9
4.9 Brush Size	10

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of the PixGunga Master sprite editor. It will explain the application's purposes, features, interfaces, and constraints under which it must operate and how the system will react to user input. This document is intended to direct the design and implementation of the application in C++ and Qt.

## 1.2 Intended Audience and Reading Suggestions

This document is intended to be used as a contract for the developers to follow, and a rubric for the course staff. The rest of this document contains specifics concerning the functional requirements with use cases and capabilities of PixGunga Master such as user tools and special features. Course staff is recommended to read in sequence from section 1 to 4 while developers are recommended to read starting from section 2.

## 1.3 Product Scope

PixGunga Master is a pixel editing application. It is designed to be beginner-friendly, providing a small but robust array of tools for basic pixel art composition and editing. The application is also intended to support frame-by-frame animation.

## 1.4 References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

<https://bobsprite.com/>

<https://www.piskelapp.com/>

David, Pat. "GIMP - About GIMP." GIMP - GNU Image Manipulation Program,

<https://www.gimp.org/about/introduction.html>. Accessed 2 Nov. 2023.

## **2. Overall Description**

### **2.1 Product Perspective**

PixGunga Master is a new, self-contained application being developed for an assignment in the University of Utah FALL 2023 CS3505 class.

### **2.2 Product Functions**

- Basic sprite composition and editing:
  - Erasing pixels.
  - Adjusting the color being drawn to the pixel canvas.
- Adjusting the number of frames for the sprite animation.
- Preview of the sprite animation cycle.
- Save and load project.

### **2.3 Operating Environment**

PixGunga Master is developed with Qt Creator in C++ as it is intended to run on Windows and MacOS devices. The user requires Qt creator to run the application. No external hardware is required.

### **2.4 Design and Implementation Constraints**

The main constraint in terms of implementation is time. The application must be developed within two weeks.

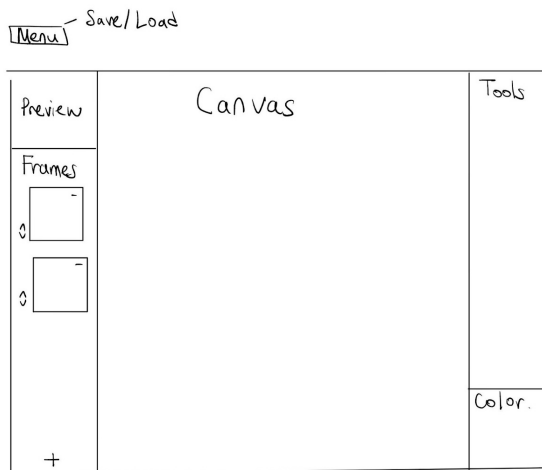
### **2.5 User Documentation**

A README and video manual will be provided as user documentation components alongside the main software product. These will be in standard .md and .mp4 files.

### 3. External Interface Requirements

#### 3.1 User Interfaces

The primary User Interface will be a single window composed of multiple docked widgets. In the center of the UI will be the sprite canvas, with tools and colors management to the right. In the top left of the UI, the user can see a preview of their sprite animation cycle and below that, they can adjust the frame they are working on. A save/load drop-down will be obscured behind a button in the top toolbar. An example is illustrated below.



#### 3.2 Hardware Interfaces

The user requires several hardware interfaces for this application. Primarily, the software requires some form of digital display, such as a monitor, with a minimum resolution of 1024 x 768. Some type of pointing device for moving the cursor, such as a mouse or touchpad, is required to interact with the widgets located in different parts of the user interface. The point device must also be able to detect primary mouse functionality, such as right-click and left-click, to select functions within the application.

#### 3.3 Software Interfaces

The application is designed to run on a Windows or MacOS device with Qt Creator support.

## 4. System Features

### 4.1 Sprite Pixel Placement

#### 4.1.1 Description and Priority

HIGH PRIORITY

Allows users to individually manipulate pixels. The user is provided with a pixel canvas in the form of a grid-based interface and the application will allow them to manipulate the contents of each of these cells.

#### 4.1.2 Stimulus/Response Sequences

- **Stimulus:** The user clicks on a grid cell within the sprite canvas.
  - **Response:** If brush is the currently selected tool, the application changes the color of the targeted pixel to the currently selected color. If the user has a larger brush size selected, the targeted pixel will be the center of the brush stroke.
- **Stimulus:** User clicks brush button.
  - **Response:** Brush is loaded as the currently selected tool.

#### 4.1.3 Functional Requirements

- REQ-1: **User Interface:** Must provide a user-friendly interface that clearly displays the pixel canvas.
- REQ-2: **Grid-Based Pixel Canvas:** Must display a grid that represents pixel boundaries.
- REQ-3: **Color Selection:** Must provide a color palette from which the user can select the color to paint the targeted cell.
- REQ-4: **Brush Size Selection:** Slider to allow the user to change brush size, brushes place/remove pixels based on this value.
- REQ-5: **Input Detection:** The application must detect user actions, such as clicking the pixel canvas or click-and-dragging.

### 4.2 Erasing

#### 4.2.1 Description and Priority

HIGH PRIORITY

The user can remove pixels by swapping to the eraser tool and clicking the area to erase, pixels will be removed from the currently selected frame based on the current brush size. This should be functionally identical to placing pixels on the screen except pixels are removed instead of placed.

#### 4.2.2 Stimulus/Response Sequences

- **Stimulus:** The user clicks the eraser button.
  - **Response:** Eraser is loaded as the currently selected tool.
- **Stimulus:** The user clicks the canvas while eraser is the currently selected tool.
  - **Response:** A circle of pixels centered at the location of the click are removed based on the current brush size.

#### 4.2.3 Functional Requirements

- REQ-1: **Eraser Toolbar Button:** Button on toolbar that switches currently selected tool to eraser.
- REQ-2: **Grid-Based Pixel Canvas:** Visible grid of pixels that the user can click to perform various actions.
- REQ-3: **Brush Size Selection:** Slider to allow the user to change brush size, brushes place/remove pixels based on this value.
- REQ-4: **User Interface:** Clearly displays the pixel canvas and eraser button.
- REQ-5: **Input Detection:** The application should detect a click on a targeted cell in the grid and the eraser button should detect clicks.

### 4.3 Color Selection

#### 4.3.1 Description and Priority

HIGH PRIORITY

The application will allow the user to select a custom color to draw with.

#### 4.3.2 Stimulus/Response Sequences

- **Stimulus:** The user is able to select a color in an option menu.
  - **Response:** The selected color is applied to the brush.

#### 4.3.3 Functional Requirements

- REQ-1: **Input Detection:** After the user selects *Color Button* a pop up menu appears, displaying available colors. User selects a color from the menu, or specifies custom color values.
- REQ-2: **Error Case:** If the user inputs invalid color values the application will not change the color of the brush.
- REQ-3: **User Interface:** Displays the selected color.



## 4.4 Save/Load

### 4.4.1 Description and Priority

HIGH PRIORITY

The application will allow a user to save the sprite to a custom .JSON file. Users will be able to load these files into the editor and generate a sprite based on it.

### 4.4.2 Stimulus/Response Sequences

- **Stimulus:** User clicks the save button.
  - **Response:** The current state of the sprite (including all frames) will be converted into a file and saved in a user specified file path.
- **Stimulus:** User clicks the load button.
  - **Response:** User will be prompted to select a file. If a valid file is given, the state of the sprite editor will change to represent the image (frames included) specified by the file.

### 4.4.3 Functional Requirements

- REQ-1: **Save Button:** The current state of the sprite (including all frames) will be converted into a file, and the user will be prompted to name it.
- REQ-2: **Load Button:** A pop up will appear, prompting the user to select a file to load.
- REQ-3: **Error Case:** If the user attempts to load an invalid file type, an error message is displayed, and the state of the editor is unchanged.
- REQ-4: **User Interface:** Drop down menu containing two buttons, one for load and another for save.

## 4.5 Animation Cycle Preview

### 4.5.1 Description and Priority

MEDIUM PRIORITY

This feature allows the user to be provided with a preview of the sprite animation cycle with the adjustment of frames per second of playback and at the scale of the original sprite.

### 4.5.2 Stimulus/Response Sequences

- **Stimulus:** Adjust Frames per Second (FPS) Incrementers
  - **Response:** The animation playback speed is updated in real-time in the preview box according to the selected FPS, allowing the user to see the animation at different speeds.

### 4.5.3 Functional Requirements

- REQ-1: **Graphical User Interface:** The software must provide a user-friendly graphical user interface for the animation preview panel that includes an incrementer for adjusting FPS and a toggle button to view animation preview at actual size. In case of an empty canvas, nothing will show up on the preview.
- REQ-2: **Input Detections:** The software should detect user actions, such as editing on the canvas or the incrementer and toggle box to control animation preview. It should respond in real-time to changes in user inputs, including adjustments to FPS incrementer and toggle box. Incrementer will only take FPS values that are not too small or too high that may cause errors.
- REQ-3: **Frame Playback:** The software must be capable of playing animation cycles at different speeds, as specified by the FPS incrementer. The software should allow smooth transitions between frames, allowing users to preview animation at various speeds without discomfort.

## 4.6 Frame Selector

### 4.6.1 Description and Priority

HIGH PRIORITY

The sprite will be represented as a sequence of images, referred to as frames, that can each be selected and edited individually. This will allow the user to create animations.

### 4.6.2 Stimulus/Response Sequences

- **Stimulus:** A frame is hovered over and left clicked on.
  - **Response:** The selected frame is displayed on the canvas and becomes editable.
- **Stimulus:** The addition symbol is clicked.
  - **Response:** A frame is added to the bottom of the list of frames.
- **Stimulus:** The subtraction symbol on a frame is clicked.
  - **Response:** The frame is deleted from the list and all frames after it move down one position to fill in the resulting void.
- **Stimulus:** The scroll-wheel or scroll bar is moved up or down.
  - **Response:** The vertical display of frames is moved up or down.

### 4.6.3 Functional Requirements

- REQ-1: **Frame Graphical User Interface:** A scrollable window on the left of the canvas that displays all the frames of the sprite. The user can scroll through the window to access all the sprites and click on a frame to display it on the canvas and edit it.
- REQ-2: **Frame Addition:** An addition symbol on the frame window that adds a frame to the bottom of the list of frames.

- REQ-3: **Frame Removal:** A subtraction symbol on each frame in the window that, if left-clicked, removes the frame from the list.

## 4.7 Onion Skin

### 4.7.1 Description and Priority

LOW PRIORITY

A toggleable button that allows a translucent overlay from the previous frame to appear on the current frame. Intended for convenience in sprite animation.

### 4.7.2 Stimulus/Response Sequences

- **Stimulus:** Onion skin toggle is activated.
  - **Response:** A translucent version of the previous frame is overlaid over the current frame.
- **Stimulus:** Onion skin toggle is deactivated.
  - **Response:** The translucent overlay is removed, showing the current frame without any layering.

### 4.7.3 Functional Requirements

REQ-1: **Toggle Button:** There must be a button that can be toggled on and off.

REQ-2: **Onion Skin Graphic:** When the button is on, a translucent overlay of the previous frame must appear on the current frame. When the button is off, the overlay must disappear.

REQ-3: **Previous Frame Missing Error Handling:** When the first frame is selected, the frame must be unaffected and should have no translucent overlay as there is no frame before it.

## 4.8 Drawing Shapes

### 4.8.1 Description and Priority

LOW PRIORITY

This feature allows the user to select and place a shape from a given set of choices, with the user's input of size and color onto the canvas.

### 4.8.2 Stimulus/Response Sequences

- **Stimulus:** Selecting a Shape from the set of shape choices on toolbar
  - **Response:** The software switches the currently selected tool to the selected shape
- **Stimulus:** User clicks canvas with a shape selected as current tool
  - **Response:** The software draws selected shape at location based on size and color

- **Stimulus:** Adjust Shape Size through slider
  - **Response:** The software changes the size of the shape using a slider in real-time, allowing the user to customize the size to their preference
- **Stimulus:** Customizing Shape Color through color palette and/or rgb input box
  - **Response:** The software displays a color palette and an input box for rgb values for the user to fill the shape with the user's desired color. The software changes the color of the shape in real-time.
- **Stimulus:** Erasing the Shape with eraser
  - **Response:** The software erases the part of the shape that was erased using the eraser while maintaining the rest of its shape on the canvas.

#### 4.8.3 Functional Requirements

- REQ-1: **Shape Selection:** The software must provide a selection of predefined shapes (e.g., square, circle, triangle) for the user to choose from. The user may not choose a shape outside of the given selection.
- REQ-2: **Size Adjustment:** The software should allow users to adjust the size of the shape using a slider or input field with real-time feedback. The size of the shape is limited.
- REQ-3: **Color Customization:** The software must offer a color selection that is applied to the shape when drawn on the canvas.
- REQ-4: **Eraser Functionality:** The software should allow the user to erase parts of the shape using the general eraser.

## 4.9 Brush Size

### 4.9.1 Description and Priority

MEDIUM PRIORITY

A slider on the UI that allows the user to change brush size, the value of brush size is used to determine the amount of pixels placed/removed by brushes.

### 4.9.2 Stimulus/Response Sequences

- **Stimulus:** User drags size bar.
  - **Response:** Brush size value changes.

### 4.9.3 Functional Requirements

- REQ-1: **Brush Size Selection** - Slider to allow the user to change brush size, brushes place/remove pixels based on this value.
- REQ-2: **User Interface:** Must provide a user-friendly interface that clearly displays the pixel canvas along with the user's available tools and options.