Software Requirements Specification

for

Acies

Version 1.1 approved

3/6/22

Prepared by Sam Waggoner, Will Cunningham, Dylan Haughton, Michael Wilkinson, and Callen Shaeffer

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Sam, Will, Dylan, Callen	2/18/22	Original creation	1.0
Sam, Will, Dylan, Callen	3/4/22	Updated according to feedback. Functional requirements reworked.	1.1

1. Introduction

1.1 Purpose

This software requirements specification is for the desktop application Acies. This SRS describes the features of Acies, and is version 1.1.

1.2 Document Conventions

Concerning Sections 4 and 5:

> Every requirement has its own priority; high-level requirements' priorities are not the same priority as subsidiary detailed requirements.
> Indented requirements are subsidiary requirements of the previous non-indented requirement.

1.3 Intended Audience and Reading Suggestions

Since all people working on the project are developers, this document is intended primarily for developers and those grading the assignment. The SRS contains an introduction, description, summary of external interfaces, functional requirements, non-functional requirements, and other requirements in that order. Reading this document sequentially by section will give the reader a complete introduction and overview of the product.

1.4 Product Scope

Acies is a desktop game primarily for kids. In the game, an orb will follow along a line until that line intersects with another, at which point the game will make a sound. The user can modify the pitch and quality of each line's sound, and draw and arrange lines in order to make a melody or rhythm. The app is meant to be an entertaining game for music creation. In addition, the audio-visual connection is aimed at providing a platform that can develop musical learning. The secondary objective of this project is to create a functional, finished product. The primary objective is to use sound methodologies in the process of creation, and to familiarize ourselves with established and effective practices in software development. This includes strong documentation and artifact preservation. This project has no corporate goals. If this product were to be launched, we would want the product to be successful enough to yield a profit. However, the purpose of this project is not to create a profit; it is for our learning. Thus, we have no business objectives or strategies as we will not do any marketing or implement any paid features.

1.5 References

There are no external references contained in this document.

2. Overall Description

2.1 Product Perspective

Acies is a standalone, self-contained product. There is therefore no larger system to which to relate it.

2.2 Product Functions

Concerning the essentials, the product will be able to:

- Detect when the object passes a point to make noise.
- Draw lines on the grid.
- Allow the editing of line properties

2.3 User Classes and Characteristics

The product is intended for all users but mainly directed towards children with a basic understanding of technology.

Children (most important):

- No understanding of music theory.
- Little to no experience using music creation apps.
- Enjoy vibrant colors and pretty sounds.

Adults (less important):

- May have some understanding of music theory
- Will appreciate hotkeys and finer controls

2.4 Operating Environment

The software will target the Windows 10 operating system as a downloadable application.

2.5 Design and Implementation Constraints

N/A

2.6 User Documentation

N/A

2.7 Assumptions and Dependencies

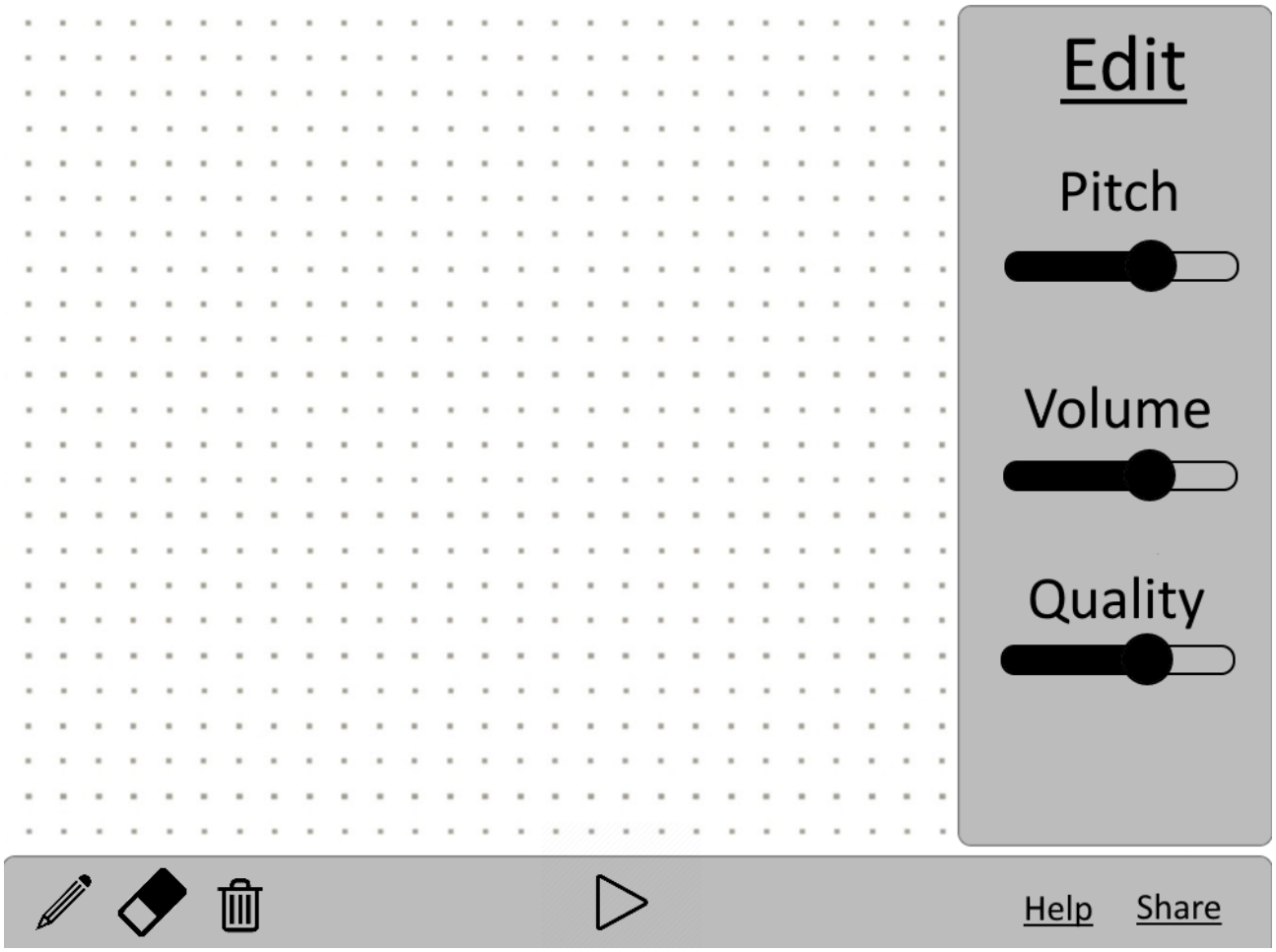
It is assumed that:

1. It is possible to play sounds with PyGame.
2. PyGame apps can be exported to desktop.

3. External Interface Requirements

3.1 User Interfaces

The interface will consist of a main grid on which lines will be drawn, with a toolbar on the bottom of the screen and an information/editing panel that will contextually appear on the right side of the screen when a line is being edited. An example of what this may look like is seen below:



- The toolbar (bottom) will control the playing of music, selection of tools, and contain miscellaneous buttons such as the help or export buttons.
- The edit/info panel (side) will display information stored in the selected line so that it can be edited. <parameter> is being used as a placeholder in the image above.
- The grid (center) will contain the lines and orbs that the user has created.

The UI should generally have rounded edges and follow a simple color palette. That palette used in the reference image is simply a placeholder. With window scaling, the toolbar and edit panel will remain constant in size, while the grid expands or contracts. Keyboard shortcuts will be based on the conventions used by most popular editing tools, such as L for line and Delete/Backspace for delete. Error messages will appear in pop-up notification windows.

3.2 Hardware Interfaces

The hardware that the application will interface with will be the screens, keyboards, speakers, and hard drives of the device. All such interactions will be handled by PyGame. The supported devices will be Windows, Mac, and Linux.

3.3 Software Interfaces

The application will use the PyGame python library. There will be no other connections between this product and other components.

3.4 Communications Interfaces

The application will not communicate with external interfaces.

4. System Features

UI

FR-1.0: The application shall allow the user to see where they can place lines on a grid.

FR-1.1: The application shall allow the user to see a toolbar with all of the functionalities.

FR-1.2: The application shall allow the user to incrementally adjust orb speed.

FR-1.3: The application shall allow the user to adjust line qualities[1] when lines are selected.

FR-1.3.1: The application shall allow the user to incrementally adjust sound pitch.

FR-1.3.2: The application shall allow the user to incrementally adjust sound loudness.

FR-1.3.3: The application shall allow the user to incrementally adjust sound compression.

Tools and Buttons

FR-2.0: The application shall allow the user to draw lines.

FR-2.1: The application shall allow the user to remove orbs.

FR-2.2: The application shall allow the user to draw lines with identical values to existing lines.

FR-2.3: The application shall allow the user to duplicate blocks of grid.

FR-2.4: The application shall allow the user to create a new orb on a line.

FR-2.5: The application shall allow the user to draw lines when the user holds left click.

FR-2.6: The application shall allow the user to erase lines when the user holds right click.

FR-2.7: The application shall allow the user to remove all existing lines and orbs.

[1]Line qualities: Line qualities include pitch, loudness, and quality. Those three characteristics correspond to color, line width, and dottedness respectively.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

NFR-1.0: The application shall install onto a device within one (1) minute 99% of the time.

NFR-1.1: The application shall launch within six (6) seconds 90% of the time.

NFR-1.2: The application's response time shall not exceed 1 second 95% of the time.

NFR-1.3: The application shall allow a user to upload a 2MB mp3 file in 5 seconds 95% of the time.

NFR-1.4: The application shall restart after encountering an error 99% of the time.

NFR-1.5: The application shall crash once out of 120 or more minutes of typical usage.

5.2 Safety Requirements

Since the app is completely standalone, there are no relevant safety requirements.

5.3 Security Requirements

Since the app is completely standalone, there are no servers and no data to secure.

5.4 Software Quality Attributes

NFR-2.0: The software shall be downloadable for free from Itch.io

NFR-2.1: The application shall be compatible with Mac, Windows, and Linux.

NFR-2.2: The software shall take up less than four (4) GB of space.

5.5 Business Rules

NFR-3.0: The system shall include an FAQ such that the user's question will be answered 80% of the time before having to send an email.

6. Other Requirements

We expect no database, internalization, or legal requirements.

Appendix A: Glossary

Orb references the traveling point on each line that will create a sound when encountering a corner.
Line references the lines the user can draw on the grid which orbs follow .
GB refers to gigabytes.
FAQ refers to a list of frequently asked questions and their answers.
UI refers to user interface.

Appendix B: Analysis Models

None

Appendix C: To Be Determined List

There are no ambiguous external references in this document.