The Painted Orchestra Music Mixer

Dev Notes

- ## Overview of the Music Mixer

We are building an art and orchestral-themed music mixer. It will contain the necessary functions and layout for the web application: a navigation tool (Home, How To, etc.), a logo, a blank canvas with empty scores to add sound in, a palette for the instruments, and the instruments themselves.

- ## Objective

The objective of this project is to create an interactive multimedia application that allows users to mix different types of instruments on an empty score plastered on a blank canvas. Users will be able to drag and drop instruments from the palette to the canvas that will have musical note effects.

- ## Key Features

When the user clicks and drags the instrument icons from the dock in the left bottom corner of the webpage onto the master score hanging on an art easel in the center of the webpage , music will play in a loop. Multiple sounds can play at once. The user can stop, start and remove the audio clips. The navigation menu will be on the top of the webpage.

- ## Architecture

This music mixer will be built using HTML5, CSS and JavaScript. The user interface, drag and drop logic, audio processing and event handling will work together to create the music mixer.

User Interface

The User Interface (UI) will consist of a mixer area, a sound library and global controls and track controls (possibly).The mixer area where the audio clips (that are visually seen on the webpage as an icon, probably an SVG icon) can be dropped and played in the center of the webpage. Visually, it will be at the bottom of the art easel (under the musical score).The audio clips/icons contained in a sound library panel will be on the left of the webpage surrounded by an artist palette. There will be 6 audio clips/icons. We are planning to include a play, pause and volume button on our webpage (global controls). This will be located in the bottom right of the webpage. The navigational menu will be at the top of the webpage. Visual feedback such as a highlight, animation such as a wiggle or colour changes will be used to show the valid drop zones and active sounds to the user. The exact type of visual feedback will be finalized during development. We plan to make our webpage responsive to mobile, tablet and desktop sizes. The art easel (mixer area) will be directly above the art palette (sound panel library) in mobile, a little farther away in tablet and farther still in desktop.

Drag-and-drop Functionality

The mixer will use HTML5 drag and drop. Each audio clip/icon will get a soundId, a file path and a title. When the audio clip/icon is dragged over the mixer area, the area will visually highlight to show that it is a valid area to place the audio clip/icon. On the drop event, Javascript will assign the correct audio clip to the corresponding track(visually its an icon and a title).

Audio Processing

Audio processing will handle how sounds are played and mixed together. Our project will use one unique icon per track so that multiple tracks can play at the same time. Each of the 6 audio clips will be able to play at the same time as the other audio clips. Audio clips will be set to loop while the mixer is running. A reset option will clear the audio clips/icons and stop the audio. Volume controls are also going to be considered in our project.

Event Handling

Every time the user does something on a webpage, it is an event. The mixer will use event listeners to connect user interactions with the UI, state and audio. For example, when an element is marked as draggable in HTML and the user drags it, the browser will trigger the event. An event listener in JavaScript will listen for that specific event.

JavaScript will identify the soundId, title and file path and store it temporarily. The events of the drag will occur until drop takes place (called dragOver in this case).

- Resources/Research (include links)

https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API (Discovering drag and drop functionality)

https://blog.prototypr.io/building-a-responsive-drag-and-drop-ui-5761fd5281d5 (Holistic view of a drag and drop)

https://www.pencilandpaper.io/articles/ux-pattern-drag-and-drop (Discovering drag and drop functionality)

https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/Events (Discovering events in JavaScript)

https://jenkov.com/tutorials/html5/drag-and-drop.html (Discovering drag and drop functionality)

https://www.youtube.com/watch?v=Z-BO6IYTgJg (Discovering UI)

https://developer.mozilla.org/en-US/docs/Web/API/HTMLAudioElement/Audio (Discovering audio handling)

https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener (Understanding event handling)

https://developer.mozilla.org/en-US/docs/Web/API/DragEvent (Understanding event handling)

- Task Delegation (Set up deliverables in sprints)

Designer: Keisha

Developer: Anna

Sprint 1: Setup, Planning, and Research (Feb 9)

In the first sprint, we will create our GitHub repo. We will also create the basic folder structure. We will search for any visual inspo for our UI and also complete any technical research for JavaScript.

Sprint 2: Audio & Drag-and-Drop Implementation (March 9th)

By the middle sprint, we will implement the drag and drop functionality. We will add the finalized audio files and the dev will begin looping and playback logic.

Sprint 3: Final Polish and Optimization - (March 23rd)

In the final sprint, we will refine the UI by making last minute visual changes to the web page. We will also add responsive layout adjustments if needed. Any bugs that occur will also be resolved. This is the time for any last minute tests and clean up.