

Virtual Piano Keyboard: Design, Implementation and Scope

Parth Sharma*, Anshul Wadhwa**, Pronika Chawla***

**(Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies, India)*

***(Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies, India)*

****(Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies, India)*

ABSTRACT

The virtual piano studied in this paper is a 61 key piano keyboard with some extra keys implemented to make it an all-rounder solution rather than being just a plain piano. The layout of this piano is composed of three core containers, these containers are nav-bar, control bar and the actual piano itself. There are two ways to use(or play) this piano - one is ideal for devices with physical keyboards (desktops, laptops, etc) and the other way is ideal for devices with touch screens (smart phones, tablets, etc). The virtual piano keyboard is a web application, so it only needs a web browser and internet availability to run. Light weight techniques are used in the development of this virtual piano to prevent it from being resource-hungry. This software is an attempt to combine multiple functionalities of full fledged software into an online lightweight web application. However, we would like to make it clear that no claim is being made that this can replace a professional music production software. In the upcoming sections, its future scope and potential will be evaluated. In this paper, we will thoroughly study and review the design, implementation and finalizing of this software. In the end, after studying and reviewing Virtual Piano Keyboard's design, implementation, future potential & scope.

Keywords - Virtual Piano Keyboard, Online Piano, Bootstrap, J Query, JavaScript

Date of Submission: 08-05-2020

Date of Acceptance: 22-05-2020

I. INTRODUCTION

A piano is a keyboard musical instrument invented by Bartolomeo Cristofori around the year 1700 in Italy. There are many types of pianos available, the categorization can be done based on factors such as number of keys, digital/analog, size etc. The software being studied in this paper is a 61 key online piano keyboard which can be played on almost any modern-day portable device without any special hardware or any special software. This application is directed towards those who have a side hobby of playing piano but they are unable to do so due to some particular reason. Some of these reasons are lack of willingness to handle a physical object, investment required to buy an actual piano keyboard, need to download and install software, etc. The existing systems available are too basic and plain, they offer a piano, not more than that. The application in review provides some additional features which will be discussed in the upcoming sections. This application has a very high potential if enough resources are invested into this, there will be an upcoming dedicated section in the paper, which will be followed by the conclusion.

II. DESIGN

This section is divided into two sub-sections - User-Interface Design and Core-Logic Design.

2.1 User Interface Design

The application is designed to be responsive, user friendly, cross compatible with multiple devices and operating systems and attractive with fluid and smooth animations. There are three core containers in the user-interface. These containers are nav-bar, controls bar and the actual piano keyboard. The user interface is designed in a way that it pleases the user's eyes. The color selection, font-selection, margins, paddings and the overall layout is well thought out and responsive to various screen sizes. The fact that every component and module of the layout is designed only using HTML and CSS makes it even more impressive. Starting with the very first container i.e. the nav-bar, the nav-bar contains a very fancy heading which states "PIANO" which is also enclosed in an anchor tag with a very fluid zoom-in and zoom-out animation. The animation comes in action when the cursor is hovered over it. The nav-bar also contains the 'Sign Up' and 'Log In' buttons, which when

clicked on opens the sign-up and login windows respectively.

After the nav-bar, it comes to the second container I.e. controls bar. The controls bar contains the supplementary features of this application like dark mode toggle switch, recording features, selecting the bass type, etc. The nav-bar and controls div are separated by a sleek horizontal white colored divider line which gives an unexpected appearance to the whole layout.

The dark mode feature mentioned here can be very helpful for those with eye problems. It simply reverses the color scheme to reduce the eye-strain. The dark mode toggle switch is also animated with the help of bootstrap.

At last, the third and final container is the actual piano itself. The fact is was designed only using HTML and CSS is amazing. It is made using HTML <table> tag and div tag. They are aligned in such a way to look like piano tiles. The black tiles are HTML div tags and the white tiles are div tags inside a single HTML table row. Every tile is designed to be animated with a zoom-in and zoom-out animation when hovered upon. The final layout of the application is shown in Fig. 1.

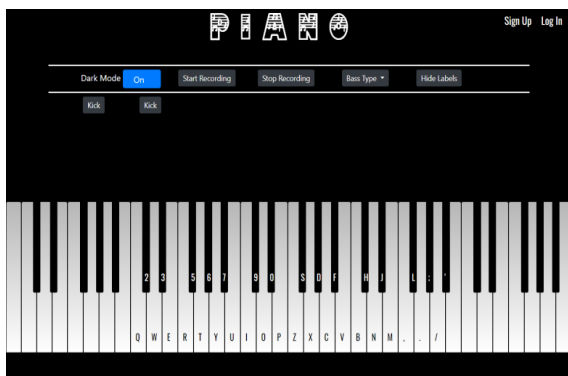


Fig.1 Layout design of the application

2.2 Core Logic Design

When it comes to the core logic behind the functionality, it is all coded in vanilla JavaScript, J Query and Php. The logic is designed to keep the latency and delay at its minimum. When the application loads, all the sounds are first loaded into the memory so that there is no delay between key-press and actual playing of sound. The application is equipped with sign up/login feature using Php and mySQL database. Some proposed features include audio recording feature which will also use the database. This feature is in development as of now and it may or may not be available by the deadline. One of the feature that makes it easy to use is the labeling of piano tiles with the corresponding assigned key on the keyboard. Another feature that

may be implemented is the option to upload the recorded audio to user's Google drive.

2.3 Algorithm

Many algorithms were designed for this application , both for visual and logical elements. Some of these algorithms are discussed below:

2.3.1 Dark/Light Mode Algorithm-The light mode function changes the CSS property of the elements to change their colors according to light mode. J Query is used to manipulate their CSS properties, and the dark mode function changes the CSS property of the elements to change their colors according to darkmode in the same way. The functions to change between two modes first detect whenever the toggle switch on the layout is used. If the toggle switch is on, the darkMode() method is called and if toggle switch is turned off, the lightMode() method is called.

2.3.2 Audio & Sources Combining Algorithm-First, an array called 'Auds' is created to store Audio elements and then the 'Sources' array is created and the directories of the audio files are inserted into 'Sources'. A for loop is used to set the sources of 'Auds' elements as 'Sources' elements. At last, all the audio elements are muted and loaded into memory.

2.3.3 Sign-up & Log-in Design Algorithm-When the application is loaded the sign up and login forms are hidden by default. If the user clicks the sign-up or login button on the nav-bar, then the respective form is shown using J Query method. There is a close button in both forms, if that close button is clicked, the form is hidden again.

III. IMPLEMENTATION

The overall implementation of this application is simple and yet very complex i.e the languages, libraries, APIs or tools used are very basic but they are utilized in a very complex way. The best example that justifies this statement is the fact that each and every component of the user interface is designed using plain HTML5 and CSS3 including the animations. The responsive user interface implies that a lot of time, experimentation and creativity is invested in the user interface. The process of selecting and arranging the HTML components into an interactive piano and then using CSS to transform plain and basic HTML components into what is presented here in itself was very challenging. Starting from the beginning, the very first eye-catching feature is the dark mode toggle switch. The dark mode toggle switch's ui is implemented using bootstrap and its functionality is implemented using J Query and CSS. Two JavaScript method called 'darkMode()' and 'lightMode()' are defined to change the color scheme of the user interface. Then a method called

'changeMode()' is defined, this function checks the current mode and then calls the 'darkMode()' or 'lightMode()' method accordingly. The dark-mode toggle-switch in on-state and off-state is shown in Fig. 2.

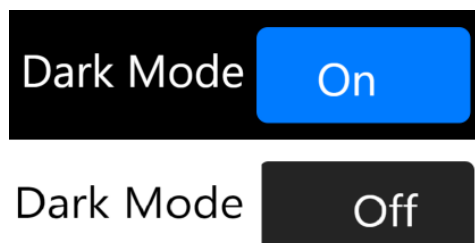


Fig.2 Dark mode toggle switch in on and off state

The user interface of the bass selector in the nav-bar is also implemented using bootstrap dropdown button. The active bass sound is changed using J Query. The audio of all the piano tiles is played and manipulated using HTMLAudioElement.

The actual sounds are managed using the HTMLAudioElement. At first, two arrays called 'Auds' and 'Sources' are created. The directories of the 61 audio files are inserted into the 'Sources' array as sources. Then, a 'for' loop is started in which a new Audio element is created and the audio file from the 'Sources' array is sequentially set as the audio source of the corresponding Audio element. In the current version of Google chrome, a web application is not allowed to play audio without the user interacting first, so in the 'for' loop, the 'muted' attribute is also set to 'true' and after that the 'audio.load()' method is called to load all the audio files into memory to eliminate any delay in audio playing. Now, all the audio objects are ready to be played when needed. The piano can be played by using the keyboard or by just simply touching the piano tiles (on a touch screen device). The touch/click method is implemented by using the JQuery's \$.click() method. Whenever a piano tile is touched or clicked, this method is called and the id of the piano tile is passed as argument, then the currentTime of the corresponding audio is set to zero, muted parameter is set to false and finally the audio is played.

The second method to play the piano is by using the keyboard. This method is implemented by using JQuery's keypress() method. In this method, switch-case is implemented, the key-code of the key pressed is passed as the switch() argument and the audio from the Auds array is played accordingly. Another feature that helps users to use this application optimally on a mobile device is the notice it shows when loaded on a mobile device. Whenever the application is loaded onto a mobile device, it shows a notice saying "Hold your device

in landscape mode to play properly.". To implement this feature, a variable named 'isMobile' is declared and a boolean value is assigned to the variable using the userAgent. Then an if statement checks if the value is true or false and if it is true, it shows the alert box.

IV. SCOPE

Virtual Piano Keyboard is an online piano that covers a lot of features but there is a lot of room for improvement and new features in the future. There are many new features that can be added to make it a dependable and reliable online platform. Some of these features include the ability to change frequency, pitch, internal volume and other properties of the sound in real-time, the option to sign in using different social media platforms like twitter, Facebook, Google, Microsoft, Sound Cloud, etc and using their APIs, the option to upload the recorded tune to online storage like Google Drive, Microsoft One Drive, etc can also be implemented. Other features like more instruments, loops, patterns, etc and the ability to share the played tune on social media can be implemented if enough resources are invested into this project. As of now, this project can not be used as a dependable and reliable piano however if enough resources are invested into this, it can come very close to emulating a real electronic piano with some features that even a real piano does not provide. If enough time and resources is invested in this application, it can be transformed into an actual dependable and reliable piano keyboard. Moreover, there is also a possibility that it can be transformed into a professional and reliable music production online platform.

V. CHALLENGES FACED

There were many challenges faced during the development of this web-application. From optimizing the user-interface to reducing the audio delay. Some of the challenges faced are discussed below:

5.1 We faced some issues in the coloring, sizing and alignment of the dark-mode toggle switch and mapping its events to change the CSS of the whole page without reloading the page.

5.2 Another minor issue we faced with the page-layout is with the the sign-up and log-in boxes. Both the boxes were appearing in different places and we wanted to overlap them. With the help of CSS, it was resolved.

5.3 A major issue faced was the delay between the timing of key-press and playing of audio. We figured out that it was happening because the audio files were not loaded in the memory. With the help of the 'load()' method, it was resolved.

VI. CONCLUSION

Virtual Piano Keyboard is a web-application planned, designed and implemented by two computer science students as their final semester project. It emulates a 61 key electronic piano with some additional features. In conclusion, it can be an effective application to fulfill people's side hobby or curiosity to play an actual piano, however in its current state we can not say that it is a complete package that is reliable and dependable to be used professionally. It is discussed in the previous sections that it is not very far away from being reliable and professional. We can definitely say that if enough time resources are invested to optimize, polish and to add more features, it can come very very close to emulating a real 61 key electronic piano.

ACKNOWLEDGEMENTS

We would like to express our gratitude to our primary supervisor, Pronika Chawla, who guided us throughout this project.

We wish to acknowledge the help provided by the technical and support staff in the Computer Science department of Manav Rachna International Institute of Research and Studies. We would also like to show our deep appreciation to our supervisor who helped us finalize our project.

REFERENCES

Journal Papers:

- [1]. Brent Gillepsie, *The Virtual Piano Action: Design and Action*, 1994.
- [2]. YI Zhang, Lifeng Wang, Wei Luo, *Web Design Style Based on Digital Technology*, 2017.
- [3]. Xiaosheng Yu, Cai Yi, *Design and Implementation of the Website Based on PHP & MYSQL*, 2010.
- [4]. Sarah Lim, Joshua Hibschan, Haoqi Zhang, Eleanor O'Rourkei, *Ply: A Visual Web Inspector for Learning from Professional Webpages*, 2018.
- [5]. Pierre Genevès, Nabil Layaida, Vincent Quint, *On the Analysis of Cascading Style Sheets*, 2012.
- [6]. Punam Kumari, Rainu Nandal, *OnWebsite Development using XAMPP/PHP*, 2017.
- [7]. Wei Jiang, Meng Zhang, Bin Zhou, Yujian Jiang, *Responsive web design mode and application*, 2014.