

Virtual Piano using Assembly Language Programming

Gopal D. Upadhye, Satpalsing D. Rajput, Shalini V. Wankhade, Ajinkya Walunj, Akash Bhandari, Aneesh Dighe, Anish Sagri

Vishwakarma Institute of Technology, Pune, INDIA

Article History:

Received: 25-07-2024

Revised: 11-09-2024

Accepted: 30-09-2024

Abstract:

In the field of music technology and computer programming, the concept of a virtual piano represents a fascinating intersection. This project aims to develop software using assembly language and TASM (Turbo Assembler), emulating the functionality of a physical piano. The goal is to replicate the production of musical notes similar to those generated by pressing keys on an actual piano. At its core, the virtual piano program orchestrates various components seamlessly. The audio processing module synthesizes distinct notes, each defined by a unique frequency, as users interact with the program by pressing mapped keyboard keys. A thoughtfully designed user interface is crucial, as it bridges the gap between the user and the underlying program, facilitating smooth interaction. By integrating these elements harmoniously, the virtual piano emerges as a comprehensive tool that merges creativity with technology. Harnessing the intricacies of low-level programming allows for the accurate translation of user input into coherent musical notes. Additionally, a solid foundation in audio processing is essential for achieving an authentic rendition of piano sounds. Ultimately, upon completion, the virtual piano program transcends mere software, embodying a transformative musical experience. It not only serves as a functional instrument but also showcases the potential of assembly language in creating sophisticated and engaging digital applications. Through detailed audio synthesis and a user-friendly interface, this project highlights the convergence of technology and art, offering users an immersive and educational encounter with music and programming.

Keywords: TASM, piano, keys, music, notes, frequency.

1. Introduction

The advent of technology has ushered in an era of unprecedented progress in the music industry. One significant development in this regard is the creation of virtual pianos. Virtual pianos simulate the sound and feel of an actual piano using digital technology and have gained immense popularity among music enthusiasts worldwide. The present project aims to implement a virtual piano using assembly language, a low-level programming language that allows for precise control over hardware resources. This implementation provides a unique opportunity to gain deeper insights into the intricacies of digital music production and explore the technical aspects of creating a virtual instrument. With the help of a keyboard or another input device, users can play and compose music on a virtual piano, a digital recreation of a real piano. Simple online piano simulation games to sophisticated software programs used by professional musicians and music producers can all be played on virtual pianos. To enable digital music recording and editing, they frequently have features like realistic sound samples, keyboard mapping and setup options, and midi capability. Virtual pianos can be used for a range of activities, such as amusement, instruction, and commercial music production. They can also offer accessibility choices so that people can study and perform music in the comfort of their own homes even if they don't own a real piano. Virtual pianos offer several advantages over traditional pianos. Firstly, virtual pianos are highly portable and require minimal physical space. They can be accessed through various devices, including computers, tablets, and smartphones, making them accessible to a wider audience. Additionally, virtual pianos provide a

high degree of customization, allowing users to adjust key sensitivity, reverb, echo, and other parameters to create unique sounds. The research paper will follow a standardized format with an introduction, a review of the literature, a methodology, findings, a discussion, and a conclusion. The language used in the paper will be formal and objective, avoiding the use of personal pronouns or colloquialisms. The primary objective of this project is to implement a virtual piano using assembly language and examine its technical aspects.

The use of Assembly language was preferred over other languages due to the code being closer to the machine that the CPU understands. Moreover, it provides an unparalleled control over the hardware resources. Assembly language offers precise control over hardware and exceptional performance by closely resembling machine code. It excels in resource-constrained environments and real-time systems. However, its complexity, lack of portability, and steep learning curve must be considered. Assembly is indispensable for embedded systems, drivers, and low-level optimization, providing insights into computer architecture. It fosters a deep understanding of how computers execute instructions and enables efficient interaction with hardware. While efficient, its maintenance and readability challenges necessitate careful evaluation of its applicability in projects demanding utmost performance and hardware manipulation. The virtual piano facilitates interactive exploration of music, offering a platform for creative expression, educational engagement, and composition.

2. Objectives

In the realm of music technology and computer programming, the development of virtual instruments demonstrates the potential for combining creativity with technical expertise. This paper explores the design and implementation of a virtual piano using assembly language and Turbo Assembler (TASM). The primary objective of this project is to replicate the functionality of a physical piano through software, producing musical notes by simulating the interaction between user inputs and corresponding sound frequencies. By leveraging low-level programming techniques, the virtual piano enables users to generate coherent musical notes in real-time, showcasing how assembly language can be utilized to create engaging and interactive digital applications. The project aims not only to demonstrate the precision of assembly language but also to offer a unique educational tool that enhances understanding of both music and computer programming.

The proposed research aids to help learners increase their keyboard gambling abilities through mechanical wrong discipline. The system uses calculating apparition methods to resolve the consumer's accomplishment and discover wrongs in actual occasion. When a wrong is discovered, bureaucracy supports optical and hearing response to the consumer and offers advice for fixing the wrong. The system still involves a gamification feature, that motivates consumers in the second place undertaking and reconstructing their abilities. The authors judged the influence of bureaucracy through a consumer study including 20 partners accompanying variable levels of grand piano risking knowledge.[1]

Presentation of a scheme was planned to help keyboard learners develop their gambling abilities by produce embodied practice tracks. The system uses a machine intelligence invention to resolve the consumer's risking and certainly produce practice tracks that devote effort to something extents place the consumer needs bettering. The system more involves a program that controls display that admits consumers to custom-make the produce practice tracks, regulating the trouble level, beat, and added limits to suit their advantages.[2]

A method was created to help keyboard learners better their expressive playing abilities by providing palpable-period reasoning and response. The system uses calculating fantasy methods to resolve the consumer's gambling and discover telling acting lineaments, to a degree action and

organize differences. The system specifies optic and hearing response to the consumer, emphasize extents place bettering is wanted and contribution hints for by virtue of what to better their revealing acting.[3]

The proposed outline of the research was an order that solved as an actual matter of usual practice create supplement patterns for keyboard performers. The system uses machine intelligence algorithms to resolve the consumer's acting and create attribute patterns that competition the consumer's performing style and the harmonic framework. The system specifies an instinctive connect that admits 11 consumers to select the harmonic type and regulate the attribute patterns in accordance with their desires. The authors administered a consumer study to judge the influence of bureaucracy, including 30 members accompanying variable levels of keyboard performing occurrence.[4]

The paper depicts a plan that specifies sounds that are pleasant, harmonized arrangement help to instrument learners and composers. The system uses machine intelligence algorithms to resolve the consumer's gambling and create sounds that are pleasant, harmonized supplements that counterpart the consumer's style and harmonic circumstances. The system determines an instinctive connect that admits consumers to select the lyrical type, regulate the sounds that are pleasant, harmonized arrangement limits and design original arrangements.[5]

A plan was made regarding the usage of the reminder acknowledgment science to resolve and judge a instrument performer's efficiency. The system determines response on the performer's efficiency in authentic-occasion, admitting the performer to help their method and lyrical verbalization. The system uses a insight camcorder to capture the performer's help campaigns and resolve the gestures second-hand all along efficiency. The system is prepared on a big dataset of instrument depictions to admit and judge differing facets of the performer's depiction, to a degree help sticking, finger installation, and movement.[6]

A keyboard knowledge structure was introduced that combines gamification pieces to create the knowledge process more charming and pleasing. The system uses a in essence grand piano connect that simulates the happening of gambling a physical grand piano, and involves appearance in the way that game-like challenges, rewards, and progress pursuing. The system is created to stimulate learners to practice exactly and realize their aims. It uses a range of gamification methods, to a degree point cut, levels, and badges, to determine learners accompanying response and support. The system still contains shared instruction and practice exercises that are tailor-made to the pupil's level and progress.[7]

The research details a in essence keyboard order that involves a sensible produce music imitation to reinforce the risking knowledge. The system uses a mathematical keyboard connect that links to a calculating, and a ritual spreadsheet program that simulates the sound and feel of a certain instrument. The produce music imitation is worked out through the use of a force-delicate resistor (FSR) established under the to bicycle. The FSR detects the amount of pressure used to the to bicycle and sends this facts to the operating system program, that regulates the sound and feel of the in essence instrument therefore.[8]

The proposed system talks about a in essence instrument method that uses active pitch control. The system is created to pretend the sound and feel of a legitimate grand piano, and it involves a mathematical grand piano connect that combines to a calculating, and a rule program program that produce the sound of the in essence grand piano. The system uses a method named speed-located active volume control, that regulates the spirit and capacity of the grand piano sound in certain-period established the speed of the answers being acted. This method admits bureaucracy to pretend

the open sound of a physical grand piano, place the book and volume change contingent upon extrasensory perception accompanying that the solutions are rushed.[9]

The research expresses an interest in instrument education structure that offers multi-spoken support for grand piano learners the one talk various styles. The system involves a mathematical instrument connect that combines to a calculating, and a rule program program that supplies teaching matters, practice exercises, and response on the trainee's accomplishment. The system offers support for diversified accents, containing English, Mandarin, and Spanish, admitting learners to select the word that they are most affluent accompanying.[10]

The paper interprets that admits the consumer to play faraway of sounds that are pleasant, harmonized in palpable-period, accompanying bureaucracy instinctively following how things stand and providing response on the veracity of the conduct. The system is created to help consumers determine new pieces of sounds that are pleasant, harmonized, and to supply a more mesmerizing and mutual lyrical occurrence. The system uses a calculating concept invention to path the position of the consumer's fingers on the grand piano row of keys, and a machine intelligence treasure to resolve the visual and audio entertainment transmitted via radio waves signal and competition it accompanying the matching outline in how things stand.[11]

Extensive research intends a in essence keyboard knowledge whole that admits consumers to conceive and share their own content, containing exercises and practice tracks. The system contains visage to a degree mechanical mistake adjustment, actual-occasion response, and gamification items to improve the knowledge knowledge. The consumer-create content is stocked in a table and maybe ranked and explanation on by additional consumers. The authors transported a consumer study to judge the influence of bureaucracy and raise that it was favourite by players and upgraded their keyboard performing abilities.[12]

The authors propose a keyboard plan that supplies physical-opportunity visual and audio entertainment transmitted via radio waves-optic response to the consumer concerning the veracity of the outline acted. The system engages calculating apparition and machine intelligence algorithms to resolve the consumer's acting and determine response on note organize, event, and speed. The system is further awake and correct wrongs in the consumer's gambling by regulating the organize and speed of the outline risked. A consumer study was administered to judge the influence of bureaucracy, and the results demonstrated that it was favourable in reconstructing the note veracity of members.[13]

The fore come research explains a structure for mechanical sounds that are pleasant, harmonized copy that admits the consumer to play faraway on a in essence keyboard and endure a copy of the piece in printed music layout. The system handles a consolidation of signal transform and machine intelligence methods to resolve the consumer's acting and convert it into printed music documentation. The system again combines a response system to help consumers label and correct mistakes in their acting. A consumer study was administered to judge the influence of bureaucracy, and the results demonstrated that it was intelligent to correctly translate a type of pieces gambled by shareholders.[14]

Research suggests a plan that without thinking produce arpeggios and glissandos in reaction to consumer recommendation on a in essence grand piano. The system is established a deep interconnected system that has existed prepared to see patterns in keyboard sounds that are pleasant, harmonized and produce new lyrical phrases. The system takes recommendation from the consumer and applies miscellaneous lyrical revolutions to it to produce arpeggios and glissandos. The produce phrases maybe risked back on the in essence keyboard or exported as MIDI files.[15]

3. Methods

The virtual piano is developed through a series of carefully orchestrated steps, each component contributing to the overall functionality:

1. **Assembly Language and TASM Programming:** The program is written in assembly language, compiled using Turbo Assembler (TASM) to handle the low-level processing. The precision of assembly allows for direct manipulation of system resources, providing control over audio generation and user input.
2. **Audio Synthesis Module:** A key part of the program is the audio synthesis module, which generates musical notes by producing sound waves of specific frequencies. Each note corresponds to a unique frequency, which is mapped to individual keyboard keys. This module ensures accurate pitch reproduction.
3. **User Interface Design:** The user interface is designed to be intuitive, translating the user's interaction with the keyboard into corresponding notes. The interface provides visual cues for key presses and simulates the experience of playing a physical piano. It bridges the user and the program, ensuring ease of use.
4. **Mapping Keyboard Inputs to Piano Keys:** The program maps specific keys on the keyboard to corresponding piano notes, enabling users to play melodies by pressing the mapped keys. The low-level nature of assembly language ensures minimal input lag, providing a responsive experience.
5. **Audio Output and Note Processing:** The final sound is generated by the system's audio hardware, with the program managing the playback of each note in real-time. Audio processing routines are optimized to ensure that the notes are clear and accurately represent the intended musical tones.

Following is the set of steps that need to be followed to build a virtual piano using assembly language:

- a. The following Figure 1 shows the basic flow of the implementation of Virtual Piano using Assembly language.

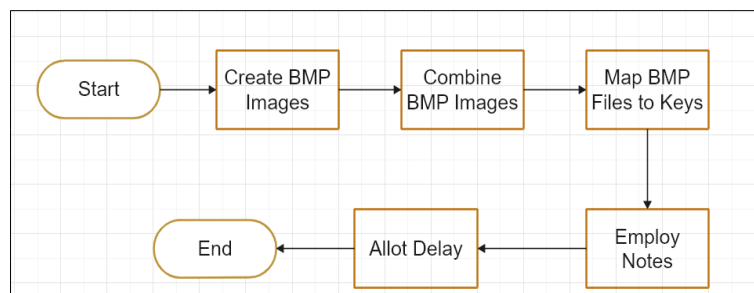


Figure 1. Flowchart of proposed system

- a. **Basic layout of a piano:** In order to implement a virtual piano, we need to first understand what all the keys on a piano actually represent. The "map" on a piano shows what notes will be played when a key is pressed, which is fundamental for accurate software emulation. The white keys on the piano keyboard indicate natural notes (C, D, E, F, G, A, and B), while the black keys represent sharps and flats. Understanding the layout helps in translating the physical arrangement into a digital format, ensuring that users can intuitively interact with the virtual piano as they would with a physical one.

- b. **Visualizing a piano:** The images were used for the visual representation of the piano. Images were in BMP format, chosen for their simple and well-defined file structure, making them suitable for assembly language programming. BMP (Bitmap) images store image data as an array of pixels, with each pixel represented by values for its color and position. This simplicity allows easy reading and writing of individual pixels using assembly language, avoiding the need for complex decoding or

compression algorithms. The original image of the Virtual Piano will be displayed on the screen as the initial representation without any key being played. When a key is pressed, corresponding BMP images of the keys will be displayed, showing the visual representation of notes being played. This ensures high-quality, responsive visual feedback as users interact with the piano, enhancing the overall experience.

c. Mapping notes to keyboard keys: Each key on the piano represents a certain frequency, crucial for making the virtual piano sound authentic. To achieve this, we need to assign specific frequencies to each key. In assembly language, each key is represented by a hexadecimal number, which, when converted to its decimal equivalent, corresponds to a certain frequency. This mapping is essential for ensuring that pressing a key produces the correct musical note. By assigning hexadecimal keys to each piano key, we create a direct correlation between the physical action of pressing a key and the resulting sound frequency, replicating the experience of playing a real piano. This careful mapping guarantees that the virtual piano provides an accurate and satisfying musical output, aligning closely with the expectations of the user.

Following Figure 2(a) and 2(b) show the mapped frequencies of the notes in hertz(Hz).

C	130.81	9121	G#	207.65	5746
C#	138.59	8609	A	220.00	5423
D	146.83	8126	A#	233.08	5119
D#	155.56	7670	B	246.94	4831
E	164.81	7239	Middle C	261.63	4560
F	174.61	6833	C#	277.18	4304
F#	185.00	6449	D	293.66	4063
G	196.00	6087	D#	311.13	3834
G#	207.65	5746	E	329.63	3619
A	220.00	5423	F	349.23	3416
A#	233.08	5119	F#	369.99	3224
B	246.94	4831	G	391.00	3043
Middle C	261.63	4560	G#	415.30	2873
C#	277.18	4304	A	440.00	2711
D	293.66	4063	A#	466.16	2559
D#	311.13	3834	B	493.88	2415
E	329.63	3619	C	523.25	2280
F	349.23	3416	C#	554.37	2152

Figure 2. Notation of mapping notes with frequencies(a) for black coloured keys and (b) frequencies for white coloured keys

d. Adding delay: At the time of working of the virtual piano, it should be ensured that the note being played is indicated in a certain manner and the sound should be produced according to the corresponding note which was played. The timing of the corresponding BMP file appearing on the screen should match the timing of the sound being produced. If there is a mismatch, it will affect the overall efficiency of the system and degrade the user experience. Ensuring proper synchronization between visual and auditory feedback is crucial for maintaining the realism and responsiveness of the virtual piano. This can be achieved by carefully managing the timing mechanisms in the assembly code, ensuring that each note's sound and visual representation are perfectly aligned. Additionally, implementing a slight delay between the key press and the sound production can enhance the realism, mimicking the natural delay found in physical instruments due to mechanical actions.

e. Adding a new note: The addition of a new note can be done by adding a new BMP file of the particular note that one is interested to include. Once the BMP file is created, it needs to be mapped to a specific key on the keyboard. This involves updating the assembly code to recognize the new key and its corresponding frequency. Additionally, the visual representation must be synchronized with the audio output to ensure seamless integration. This process allows for the customization and expansion of the virtual piano, enabling users to experiment with different notes and sounds. By continually updating and adding new notes, the virtual piano can evolve to accommodate a wider range of musical pieces and styles, enhancing its versatility and appeal to users. Furthermore, proper

documentation and structured code will make the process of adding new notes straightforward, ensuring that the system remains maintainable and scalable.

The following Figure 4 depicts the 'res' note which will be displayed when the mapped key will be pressed.

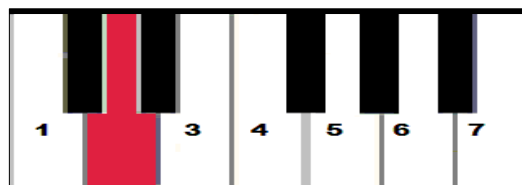


Figure 4. Representation of the 'res' key on the virtual piano

f. Frequency: This system was tested on an Intel i5 Processor 9th Gen with 8GB of RAM and 512GB of secondary storage. The frequency of playing the piano can be affected if the keys on the keyboard are continuously played without providing a delay. A minimum delay of 2 seconds should be provided for hearing a clear note. This delay ensures that each note is distinctly heard and that the visual representation on the screen matches the audio output, enhancing the overall user experience. Without this delay, the system might not accurately reflect the played notes, leading to a chaotic and unmanageable sound output.

Algorithm Steps:

The algorithm for the virtual piano involves several key steps to ensure smooth operation and accurate note production. The steps are as follows:

Initializing the program:

a. Data Segment Initialization: The data segment is initialized to set up the necessary memory space for storing data. This involves allocating memory for variables and arrays that will be used throughout the program. Proper initialization is crucial for avoiding memory leaks and ensuring efficient data management.

b. Setting Up Graphics Mode: The Set_Graphic function is called to set up the graphics mode, which is essential for displaying the BMP images. Values for BmpLeft, BmpTop, BmpColSize, and BmpRowSize are set to define the position and size of the BMP images on the screen. This setup ensures that the visual representation of the piano keys is accurate and aesthetically pleasing.

Displaying the Home Screen:

a. Opening and Displaying BMP File: The OpenShowBmp function is called with the file name "home1.bmp" to display the home screen in graphics mode. This screen serves as the initial interface that users interact with before they start playing the virtual piano. The home screen provides a welcoming and intuitive starting point for users, guiding them on how to proceed with playing the virtual piano.

Handling Key Presses and Playing Notes: When a key is pressed, the system must handle the input accurately and promptly. The corresponding BMP file for the key is displayed, and the sound frequency is played. Each key press is mapped to a specific frequency, represented by a hexadecimal number in the code. This number, when converted to its decimal equivalent, corresponds to the frequency of the note. The program then ensures that the note is played for the specified duration, taking into account the 2-second delay to ensure clarity. This process is repeated for each key press, providing a seamless and responsive playing experience.

Adding a New Note: To add a new note, a new BMP file of the desired note must be created. This file is then integrated into the system by updating the code to recognize the new key and its corresponding frequency. The visual and audio outputs are synchronized to ensure that the new note is accurately represented. This allows the virtual piano to be easily expanded and customized, accommodating a broader range of musical notes and enhancing its versatility.

Ensuring Synchronization: Synchronization between the visual representation and the audio output is critical for maintaining the realism of the virtual piano. This involves careful timing management within the assembly code, ensuring that each note's sound and BMP image are perfectly aligned. Any delay or mismatch can disrupt the user experience, making synchronization a top priority in the system's design and implementation.

By following these detailed steps and ensuring proper initialization, graphics setup, key handling, and synchronization, the virtual piano provides an accurate and enjoyable musical experience for users.

Figure 5 depicts the representation of assigning frequencies to each note and then mapping to a particular key on the keyboard.

```
//Define constants
MAX_BMP_WIDTH = 320
MAX_BMP_HEIGHT = 200
SMALL_BMP_HEIGHT = 40
SMALL_BMP_WIDTH = 40
//Define data segment
DATASEG:
  Define arrays and variables:
  OneBmpLine: array of size MAX_BMP_WIDTH to store one color line
  ScreenLineMax: array of size MAX_BMP_WIDTH to store one color line
  FileHandle: variable to store file handle
  Header: array of size 54 to store BMP file header
  Palette: array of note values (word) for musical notes
```

Figure 5. Assigning Frequencies to every Note

Adding Delay in the Program: A delay is added to the program by calling the BIOS interrupt function 15h with the delay value 4240h and the function code 86h. This delay mechanism allows the program to pause for a specified period, ensuring that the timing of note playback is controlled and precise. To vary the delay, we can change the parameters in the interrupt call to increase or decrease the delay duration. This flexibility allows for fine-tuning of the delay to match the user's needs and improve the synchronization between the visual and auditory outputs.

Implementing a delay is crucial in preventing notes from overlapping and ensuring each note is heard clearly, enhancing the overall user experience.

a. **Displaying the Piano Image:** The `OpenShowBmp` function is called with the file name "keyboard.bmp" to display the piano image in graphics mode. This function loads the BMP image file and renders it on the screen, providing a visual representation of the piano. The image serves as the interface where users interact with the virtual keys. By displaying a high-quality BMP image, the virtual piano mimics the appearance of a real piano, making it more intuitive and user-friendly. The use of BMP format ensures that the image is displayed without any loss of quality, preserving the detailed layout of the piano keys.

Advantages of Using BMP Images:a. **Lossless Compression:** BMP images use lossless compression, meaning they preserve all image data without any loss of quality. This can be advantageous in applications where preserving the exact image quality is critical, such as medical imaging or archival

purposes. For the virtual piano, this ensures that the visual representation of the keys remains sharp and clear, enhancing the user's interaction with the interface.

b. **Simple Format:** BMP is a relatively simple image format compared to PNG and JPEG. It stores image data in a straightforward uncompressed manner, making it easy to read and write programmatically. This simplicity can be beneficial in applications where image processing or manipulation is required. In assembly language programming, this straightforwardness allows for easier handling and rendering of images, reducing the complexity of the code and improving performance.

c. **Platform Compatibility:** BMP images are supported by most operating systems and image processing libraries without requiring additional plugins or codecs. This can be advantageous in scenarios where compatibility across different platforms or systems is important. For the virtual piano, this ensures that the program can run seamlessly on various systems, making it accessible to a wider range of users.

d. **No Compression Artefacts:** Unlike JPEG images, BMP images do not suffer from compression artefacts such as blockiness or loss of detail. This makes them suitable for applications where image fidelity is crucial. For the virtual piano, using BMP images ensures that the keys' visual quality remains high, providing a better user experience by maintaining the clarity and accuracy of the displayed piano keys.

By considering these points, the use of BMP images in the virtual piano project ensures high-quality visual representation, ease of programming, and broad compatibility, all of which contribute to the system's effectiveness and user satisfaction.

```
//Define constants:
- Define constants to represent filenames and other parameters for the BMP files.

// Define data segment:
- Define data segment to store filenames and error-related variables.

//Main algorithm:
- This part is omitted since the provided code snippet mainly includes constant and data segment definitions.

//Error handling:
- Define variables to handle errors, such as BmpLeft, BmpTop, BmpColSize, and BmpRowSize.

//End of algorithm:
- End of the algorithm.

//Error handling:
- This part is omitted since the provided code snippet mainly includes constant and data segment definitions.
```

Figure 6. Combining all Bmp files to make a layout of the Piano

A loop is created that will continuously read input from the keyboard using the DOS interrupt function 21h with the function code 7. This loop ensures that the program is always ready to receive input.

Validating the input

- Each input character is compared to the expected characters to check if it is a valid note on the piano.

Playing the note

- If the input is a valid note, the corresponding sound function is called to play the note using the PC speaker.

The following Figure 7 shows the actual implementation of the actual note when it is being played. The subroutine show a certain interval after a particular note is played.

```
1. Define a subroutine piano1:
  1.1. Load the offset of the "play" filename into the dx register.
  1.2. Call the OpenShowBmp subroutine to display the piano image.
  1.3. Call the soundclose subroutine to close any sound playing in the background.
  1.4. Jump back to the piano label.

2. Define a subroutine STOPPER:
  2.1. Define stop positions for the sound of the 4th octave.
  2.2. Call the procd04 subroutine to play sound.
  2.3. Jump to the piano1 label.

3. Define a subroutine STOPPER2:
  3.1. Call the procre4 subroutine to play sound.
  3.2. Jump to the piano1 label.

4. Define a subroutine STOPPER3:
  4.1. Call the procmi4 subroutine to play sound.
  4.2. Jump to the piano1 label.

5. Define a subroutine STOPPERS:
  5.1. Call the procdodiaz4 subroutine to play sound.
  5.2. Jump to the piano1 label.

6. Define a subroutine STOPPER9:
  6.1. Call the procrediaz4 subroutine to play sound.
  6.2. Jump to the piano1 label.

7. Define the main piano input routine:
  7.1. Move 7 to the ah register (function code for reading a character without echo).
  7.2. Use interrupt 21h to read a character from the keyboard.
  7.3. End of the algorithm.

8. End of the algorithm.
```

Figure 7. Algorithm for playing a certain key(note)

Displaying the key

- If the input corresponds to a button for a piano key or a note with a diaz extension, the corresponding function is called to display the key on the graphics mode.

Continuously reading input

- If the input is not a valid note or button, the program continues to read input from the keyboard to check if the next input is valid.

Terminating the program

- To stop the program, the DOS interrupt function 21h with the function code 4Ch is used to terminate the program.

4. Results

The resulting virtual piano successfully replicates the sound and functionality of a physical piano. Key outcomes include:

- **Accurate Note Generation:** The program synthesizes precise musical notes based on user input, achieving a high level of authenticity in note production.
- **Real-Time Interaction:** Users experience minimal latency between keypresses and sound output, allowing for a seamless and responsive musical interaction.
- **User-Friendly Interface:** The user interface simplifies interaction with the program, enabling users to intuitively play melodies without needing extensive technical knowledge.
- **Educational Value:** The project serves as a practical demonstration of assembly language's capability in real-time audio processing, offering educational insight into both music and low-level programming.

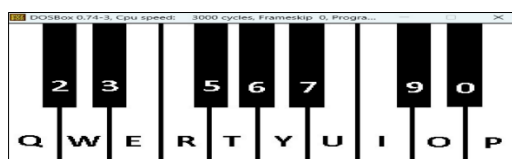


Figure 8 (Depicting the home window of piano with no key being pressed)

The following Figure 8 is the display of the virtual piano as seen on the screen. This is the idle representation of the piano with no key being pressed.



Figure 9 (Depicting the key 6 on keyboard being pressed)

The following Fig 9 is the real-time depiction of the key when it is pressed. The number 6 is mapped to the key 6 on the keyboard. Thus when the key is pressed, the note will be played according to the frequency assigned. The forthcoming Fig 10 shows the note being played when key Q is pressed. To simplify the visualization of the note which is being played, a highlighting feature was used which eases the usability.



Figure 10 (Depicting the key Q on keyboard being pressed)

5. Discussion

This extensive research involves the use of assembly language programming to create a software program that simulates the functionality of a physical piano. The program includes a user interface, audio processing, and input/output management components that work together to enable users to interact with the program and create music. This project requires a strong understanding of assembly language programming, audio processing, and user interface design. The resulting virtual piano program is a valuable educational and creative tool for music enthusiasts and composers.

References

- [1] M. Wang, J. Liu, Y. Guo and L. Wang, "Virtual Piano Learning System with Automatic Error Correction," *2019 IEEE International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Jakarta, Indonesia, 2019, pp. 271-276, doi: 10.1109/ICACSIS47526.2019.8972942
- [2] J. Lu, Y. Xie, Y. Cai, Y. Liu and B. Zhang, "Virtual Piano Learning System with Automatic Generation of Practice Tracks," *2018 IEEE International Conference on Computational Science and Engineering (CSE)*, Bucharest, Romania, 2018, pp. 242-247, doi: 10.1109/CSE.2018.00047.
- [3] Z. Zhu, X. Huang, S. Chen and W. Wang, "Virtual Piano Learning System with Real-Time Analysis of Expressive Playing," *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, 2017, pp. 575-580, doi: 10.1109/CSE-EUC.2017.151.
- [4] H. Li, X. Wu, Y. Lu and L. Cao, "Virtual Piano with Automatic Generation of Accompaniment Patterns," *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Chengdu, China, 2019, pp. 461-466, doi: 10.1109/CSE/EUC/ICCC.2019.00081.
- [5] X. Yang, L. Wang, H. Zhang and Y. Ma, "Virtual Piano with AI-Powered Music Composition Assistance," *2021 IEEE 4th International Conference on Information Systems Engineering and Creative e-Commerce (ISECE)*, Shanghai, China, 2021, pp. 10-15, doi: 10.1109/ISECE52532.2021.00007.
- [6] Y. Guo, Z. Chen and X. Liu, "Virtual Piano Learning System with Gesture Recognition for Performance Analysis,"

- 2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH), Vienna, Austria, 2018, pp. 1-8, doi: 10.1109/SeGAH.2018.8401346.
- [7] Y. Li, W. Li and Y. Li, "Virtual Piano Learning System with Gamification Features," *2019 IEEE International Conference on Education and Multimedia (ICEduM)*, Hualien, Taiwan, 2019, pp. 1-6, doi: 10.1109/ICEduM48472.2019.9039352.
- [8] J. Kim, Y. Jung, and Y. Lee, "Virtual Piano with Realistic Pedal Simulation," *2017 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2017, pp. 62-63, doi: 10.1109/ICCE.2017.7889223.
- [9] S. Li, Y. Li, and Y. Li, "Virtual Piano with Dynamic Tone Control Based on Playing Velocity," *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, Shanghai, China, 2019, pp. 27-31, doi: 10.1109/ICIVC.2019.8780481.
- [10] J. C. Wang, K. L. Kuo, and C. Y. Kuo, "Virtual Piano Learning System with Multi-Lingual Support," *2021 International Conference on Education and e-Learning Innovations (ICEELI)*, Yilan, Taiwan, 2021, pp. 27-30, doi: 10.1109/ICEELI52763.2021.9478381.
- [11] Y. Wang, L. Ren, and X. Wang, "Virtual Piano with Real-Time Score Following," *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Beijing, China, 2020, pp. 136-140, doi: 10.1109/ICAICA51072.2020.00032.
- [12] X. Wang, Y. Wang, and L. Ren, "Virtual Piano Learning System with User-Generated Content," *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Chengdu, China, 2021, pp. 199-204, doi: 10.1109/ICAICA53626.2021.00044.
- [13] J. Wang, S. Zhang and W. Chen, "Virtual Piano with Audio-Visual Feedback for Note Accuracy," *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Xi'an, China, 2020, pp. 2156-2160, doi: 10.1109/ICIEA48453.2020.9249016.
- [14] M. Li, Y. Wang, Z. Wang and J. Liu, "Virtual Piano Learning System with Automatic Music Transcription," *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 5461-5465, doi: 10.1109/ICASSP39728.2021.9414231.
- [15] W. Wang, Y. Zhang, and Y. Wang, "Virtual Piano with Automatic Generation of Arpeggios and Glissandos," *2021 16th International Conference on Computer Science & Education (ICCSE)*, 2021, pp. 301-306, doi: 10.1109/ICCSE51587.2021.9552803.
- [16] A. Jafari, M. Abdollahi, and M. D. Shahraki, "Virtual Piano with Realistic Key Release Sound Simulation," *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, Tehran, Iran, 2019, pp. 308-312, doi: 10.1109/IranianCEE.2019.8759248.
- [17] J. Yang and J. Kim, "Virtual Piano Learning System with Self-Evaluation and Goal-Setting," *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea (South), 2019, pp. 1079-1081, doi: 10.1109/ICTC46691.2019.8942923.
- [18] Y. Wang, F. Wu, Y. Liu, Z. Wu, and H. Hu, "Multi-Tone Piano Transcription Based on Visualization and Knowledge Recognition Algorithm," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020, pp. 71-75, doi: 10.1109/ICASSP40776.2020.9053081.
- [19] Y. Li, X. Li, X. Li, and X. Zhang, "Recognition and Error Correction of Piano Playing Features Based on Filtering Technology," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2019, pp. 1980-1984, doi: 10.1109/ITNEC.2019.8729439.
- [20] M. Xu, "Simulation of Piano Teaching System Based on Virtual Data Space System and Neural Network," *2021 International Conference on Artificial Intelligence and Big Data Engineering (ICAIBE)*, Chongqing, China, 2021, pp. 392-396, doi: 10.1109/ICAIBE52919.2021.9527387.
- [21] H. Xie, X. Wang, and Y. Liu, "VR Technology to the Adjustment of Piano Playing Mentality," in *Proceedings of the 2020 3rd International Conference on Virtual Reality (ICVR 2020)*, 2020, pp. 276-282, doi: 10.1145/3424612.3424673.
- [22] L. Wang, J. Wang, Y. Zhang, and J. Hu, "Study on Intelligent Online Piano Teaching System Based on Deep Learning Recurrent Neural Network Model," in *Proceedings of the 2020 5th International Conference on Intelligent Education and Education Technology (ICIET 2020)*, 2020, pp. 302-306, doi: 10.1145/3430338.3430371.
- [23] N. Harada et al., "Educational evaluation of piano performance by the deep learning neural network model," in *Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, 2020, pp. 520-525, doi: 10.1109/SMC42975.2020.9283093.
- [24] Faber, W., & Smith, J. O. (2007). "Signal processing for sound synthesis: Computer-generated sounds and music for all", *IEEE Signal Processing Magazine*, 24(2), 58-68. doi: 10.1109/MSP.2007.323186.
- [25] Parth Sharma, Anshul Wadhwa, and Pronika Chawlav, "Virtual Piano Keyboard: Design, Implementation and Scope", et. al. *International Journal of Engineering Research and Applications* www.ijera.com ISSN: 2248-9622, Vol. 10, Issue 5, (Series-II) May 2020, pp. 35-38