Beepiano

Dylan

2001585530

Section I.

1. Description

Beepiano is an amateur piano program made in the programming language, c++. It currently plays notes in the chord of C and Rebemol-C (Black keys). For the program to work, the user is required to enter keys in the keyboard for the program to read and output them in beeps. Keys that have been assigned are : Q to ] and A to K. WindowsAPI is required for the program to work and Beepiano also has hard-coded songs that the user can listen to.

Section II.

1. Design and Plan

iib. Functions

Beepiano currently has 6 functions, 5 which are the code of the program and 1 a function that plays a hard-coded song.

This is the list of all of the functions :

* Interface()
* Exit()
* Choice()
* Piano()
* HardcodePicker()
* Mario()
* Logo()
* main()

The function Interface, would be the first function that will be called. This function outputs all the choices the user can choose and will be asked to input a choice. The choice will then be processed through the function Choice. Choice will then execute another function according to the input of the user. This function uses if to differentiate inputs and what should be outputted accordingly.

The function Exit, is self-explanatory. It exits the program. Then there is the function Piano. This is the backbone of the program. It uses if to output certain beeps according to the user’s input. The function also uses a while loop to constantly clear the screen using CLS giving the program a freezed impression. The point of clearing the screen is so that the terminal would not be cluttered with pressed keys when used. To get input from the user, the program uses getch.

Pressing the key 4 breaks the loop and returns the program to the main menu, calling the function Interface and Choice.

The function HardcodePicker, is a function that prompts the user to pick a song that has been hard-coded to the program and play it. It uses the if function to differentiate choices. currently there is only one song provided which is the Mario Bros. theme song.

The function Mario is where the beep code for the song is stored. When the user picks this song in HardcodePicker function, this function will then be called. When pressing the key ‘4’ the function calls on function Interface and Choice to return to the main menu after the song finishes. The function Logo outputs the Beepiano ascii art shown on the start of the program. It uses raw string to output exactly as typed.

The function main basically calls for Logo(), Interface() and Choice(), it is the ‘main-interface’ of the program.

Section III.

iiia. Problems

Problems that I’ve encountered :

* Which beep makes which note.
* How to make user’s input instantly processed. ( Not requiring ‘ENTER’ to input )
* Going back to the main menu on the Piano() Loop.
* Outputting ASCII logo.

iiib. Solutions

Solving the beep problem isn’t too hard but required some effort. Compare and contrast from the internet and testing the beeps was the way I did it. I managed to make the beeps sound as close as possible to the chord of C in piano and Rebemol C in piano.

I solved the instant input problem by using getch(). It works the way that when the user inputs to a variable, it will instantly process it and output according to the input. By using if, each input would have different beeps according to the keys assigned.

I also solved the loop problem by using break. The main problem was, the CLS command was also inside the loop. After some thinking I managed to use break on the right part of the loop and positioned the CLS command to the right spot. Thus being able to return to the main menu anytime with no problems.

At first, I wasn’t able to output the logo with spaces and new lines. After using many methods such as outputting as a text file, manual cout, I finally found the easiest way to do it, which is using raw string. It outputs the text as stated at the code which is just what I needed.

Section IV.

Code

#include <iostream>

#include <windows.h> // to clear the console when needed using system("cls") and using windowsAPI to use beep and key input commands.

#include <conio.h>

#include <cstdlib>

#include <fstream>

#include <string>

// Main plan : To assign notes on keyboard keys, and using if to execute these notes. Notes will be played automatically when pressing the keys. (No need to press enter.)

using namespace std;

//declarations

int Piano();

int Exit();

int HardcodePicker();

int Mario();

int Interface()

{

cout << "Hello ! Welcome to Beepiano, an amateur piano program." << endl;

cout << "Please enter what would you like to do today." << endl << endl;

cout << "1. Play Piano." << endl;

cout << "2. Play Hardcoded songs." << endl;

cout << "3. Exit." << endl;

}

int Logo()

{

std::cout << R"(

\*paste logo.txt here\*

)";

std::cout<<"\n"<<"\n";

}

int Exit()

{

exit;

cout << "Hope you enjoyed !";

}

int Choice()

{

int number = 0;

cin >> number;

if(number==1)

{

system("cls");

Piano();

}

if(number==2)

{

system("cls");

HardcodePicker();

}

else if(number==3)

{

system("cls");

Exit();

}

else exit;

}

int Piano()

{

{

while(true){

cout << "Press 4 to return to main screen." << endl;

cout << "Enter note : " << endl << endl;

char note = getch();

if(note == 'q') // do c

{

Beep(261,100);

}

if(note == 'w') // re d

{

Beep(293,100);

}

if(note == 'e') // mi e

{

Beep(329,100);

}

if(note == 'r') // fa f

{

Beep(349,100);

}

if(note == 't') // sol g

{

Beep(392,100);

}

if(note == 'y') // la a

{

Beep(440,100);

}

if(note == 'u') // si b

{

Beep(493,100);

}

if(note == 'i') // do c

{

Beep(523,100);

}

if(note == 'o') // re d

{

Beep(587,100);

}

if(note == 'p') // mi e

{

Beep(659,100);

}

if(note == '[') // fa f

{

Beep(698,100);

}

if(note == ']') // sol g

{

Beep(784,100);

}

if(note == 'a') // rebemol

{

Beep(277,100);

}

if(note == 's') // mibemol

{

Beep(311,100);

}

if(note == 'd') // solbemol

{

Beep(370,100);

}

if(note == 'f') // labemol

{

Beep(415,100);

}

if(note == 'g') // sibemol

{

Beep(466,100);

}

if(note == 'h') // rebemol

{

Beep(554,100);

}

if(note == 'j') // mibemol

{

Beep(622,100);

}

if(note == 'k') // solbemol

{

Beep(740,100);

}

if(note == '4')

{

break;

}

system("cls"); // constantly clears screen

}

}

system("cls");

Logo();

Interface();

Choice ();

}

int HardcodePicker()

{

int choose = 0;

cout << "Choose a song." << endl;

cout << "1. Mario Theme song." << endl;

int choice = 0;

cin >> choice;

if(choice == 1)

{

Mario();

}

}

int Mario()

{

Beep (330,100);Sleep(100);

Beep (330,100);Sleep(300);

Beep (330,100);Sleep(300);

Beep (262,100);Sleep(100);

Beep (330,100);Sleep(300);

Beep (392,100);Sleep(700);

Beep (196,100);Sleep(700);

Beep (262,300);Sleep(300);

Beep (196,300);Sleep(300);

Beep (164,300);Sleep(300);

Beep (220,300);Sleep(100);

Beep (246,100);Sleep(300);

Beep (233,200);

Beep (220,100);Sleep(300);

Beep (196,100);Sleep(150);

Beep (330,100);Sleep(150);

Beep (392,100);Sleep(150);

Beep (440,100);Sleep(300);

Beep (349,100);Sleep(100);

Beep (392,100);Sleep(300);

Beep (330,100);Sleep(300);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(100);

Beep (247,100);Sleep(500);

Beep (262,300);Sleep(300);

Beep (196,300);Sleep(300);

Beep (164,300);Sleep(300);

Beep (220,300);Sleep(100);

Beep (246,100);Sleep(300);

Beep (233,200);

Beep (220,100);Sleep(300);

Beep (196,100);Sleep(150);

Beep (330,100);Sleep(150);

Beep (392,100);Sleep(150);

Beep (440,100);Sleep(300);

Beep (349,100);Sleep(100);

Beep (392,100);Sleep(300);

Beep (330,100);Sleep(300);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(100);

Beep (247,100);Sleep(900);

Beep (392,100);Sleep(100);

Beep (370,100);Sleep(100);

Beep (349,100);Sleep(100);

Beep (311,100);Sleep(300);

Beep (330,100);Sleep(300);

Beep (207,100);Sleep(100);

Beep (220,100);Sleep(100);

Beep (262,100);Sleep(300);

Beep (220,100);Sleep(100);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(500);

Beep (392,100);Sleep(100);

Beep (370,100);Sleep(100);

Beep (349,100);Sleep(100);

Beep (311,100);Sleep(300);

Beep (330,100);Sleep(300);

Beep (523,100);Sleep(300);

Beep (523,100);Sleep(100);

Beep (523,100);Sleep(1100);

Beep (392,100);Sleep(100);

Beep (370,100);Sleep(100);

Beep (349,100);Sleep(100);

Beep (311,100);Sleep(300);

Beep (330,100);Sleep(300);

Beep (207,100);Sleep(100);

Beep (220,100);Sleep(100);

Beep (262,100);Sleep(300);

Beep (220,100);Sleep(100);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(500);

Beep (311,300);Sleep(300);

Beep (296,300);Sleep(300);

Beep (262,300);Sleep(1300);

Beep (262,100);Sleep(100);

Beep (262,100);Sleep(300);

Beep (262,100);Sleep(300);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(300);

Beep (330,200);Sleep(50);

Beep (262,200);Sleep(50);

Beep (220,200);Sleep(50);

Beep (196,100);Sleep(700);

Beep (262,100);Sleep(100);

Beep (262,100);Sleep(300);

Beep (262,100);Sleep(300);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(100);

Beep (330,100);Sleep(700);

Beep (440,100);Sleep(300);

Beep (392,100);Sleep(500);

Beep (262,100);Sleep(100);

Beep (262,100);Sleep(300);

Beep (262,100);Sleep(300);

Beep (262,100);Sleep(100);

Beep (294,100);Sleep(300);

Beep (330,200);Sleep(50);

Beep (262,200);Sleep(50);

Beep (220,200);Sleep(50);

Beep (196,100);Sleep(700);

cout << "Press 4 to return to main menu." << endl;

int backout = 0;

cin >> backout;

if(backout == 4);

{

system("cls");

Logo();

Interface();

Choice();

}

}

int main()

{

Logo();

Interface();

Choice ();

}