|  |
| --- |
| Object Oriented Programing 5team Final Assignment |
| 이름: 박찬혁, 석제노, 조성찬, 황준호 |

Contents

**1. Project Title, Short Description of Program**

**2. System Requirement for Compilation and Execution**

**3. Functionality of Our Program**

**4. implemented method**

**5. UML Design for our system**

**6. Execution Result**

**7. How we applied Object Oriented Concepts**

**8. Conclusion**

**1. Project Title, Short Description of Program**

-Project Title : Virtual Piano

- Using directX SDK and Midi Class, We make virtual piano that have function to play, record, load

- Player can control volume, octave and play piano by using keyboard.

- Player can record their music and load their music by clicking icon in window.

**2. System Requirement for Compilation and Execution**

-Compiler : Visual Studio 2019

-Language : C++

-Requirement : directX SDK

**3. Functionality of Our Program**

○ Play

- key : ‘q’,’w’,’e’,’r’,’t’,’y’,’u’,’i’,’o’,’p’,’[’,’]’ – whitekey,

‘2’,’3’,5’,’6’,’7’,’9’,’0’,’=‘ – blackkey

- left, right arrow : change the octave

- up, down arrow : change the volume

- F1 : Initialize Octave and Volume

○ Record

- After press ‘녹음시작’ button, we can record what player play.

- we make Record files that have 1)inputted key, 2) octave, 3) input time, 4) key down or up t/f in ‘music’ folder when press ‘녹음종료’ button

○ Load

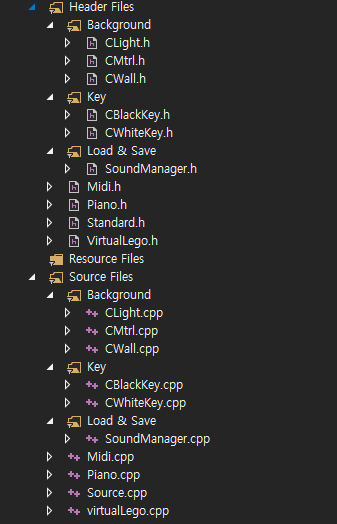
- we can see file list by pressing ‘파일목록’ button

- when we press filename in file list, virtual piano play recorded music automatically.

- stop music by pressing stop button.

**4. Implemented Method & Class**

- Header / Source structure



불빛 클래스 구현부

클래스 구현부

피아노 바탕 클래스 구현부

흰 건반 클래스 구현부

검은 건반 클래스 구현부

파일 녹음 및 재생 클래스 구현부

Midi 소리 클래스 구현부

피아노 연주 클래스 구현부

WinMain 클래스 구현부

총괄 화면 출력 클래스 구현부

불빛 클래스 선언부

클래스 선언부

피아노 바탕 클래스 선언부

흰 건반 클래스 선언부

검은 건반 클래스 선언부

파일 녹음 및 재생 클래스 선언부

Midi 소리 클래스 선언부

피아노 연주 클래스 선언부

라이브러리 포함 클래스 선언부

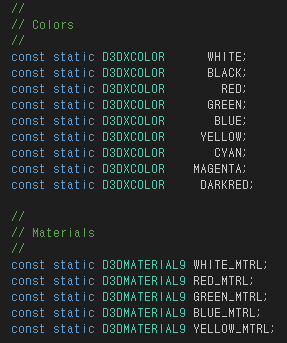
총괄 화면 출력 클래스 선언부

**1) Background Classes**

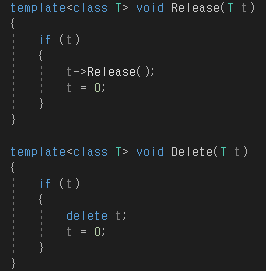
○ CMtrl

- Parant Class of all background, key classes ( interface )

- Select color and materials



- Have Release, Delete function

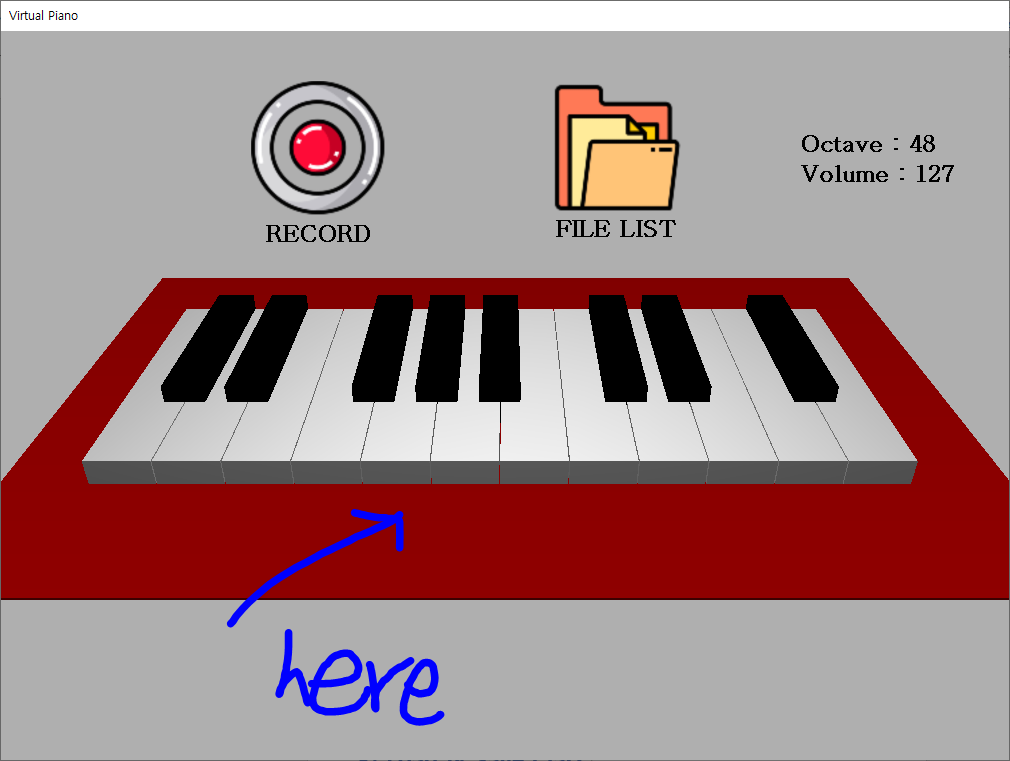


○ CLight

- Create light that enables us to see program

○ CWall

- Create background of piano

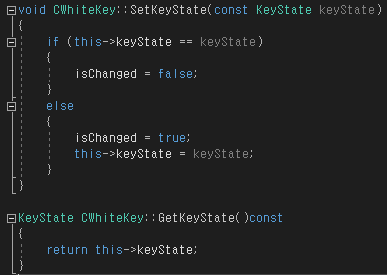


**2) Key Classes**

○ CBlackKey, CWhiteKey

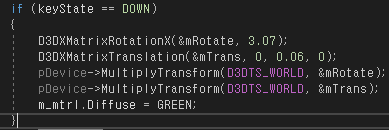
- Create keyboard

- Check key is putted up or down or no change





- pressing a key, it becomes green and rotates



**3) Load & Save Class**

○ SoundManager

- Record, Load file

- File structure

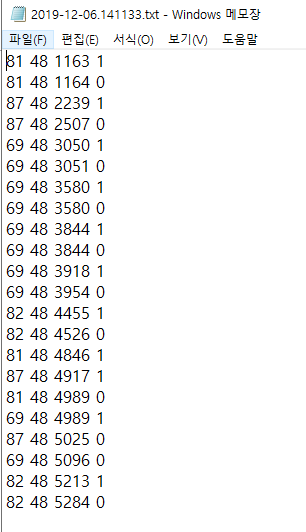
- One key for One line

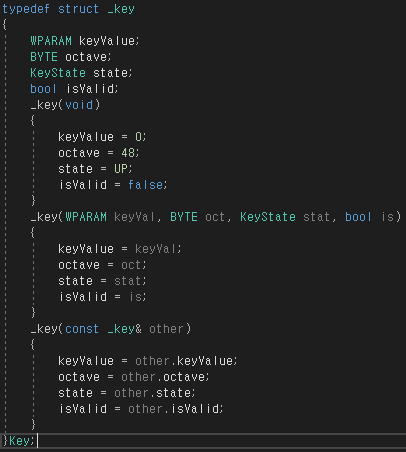
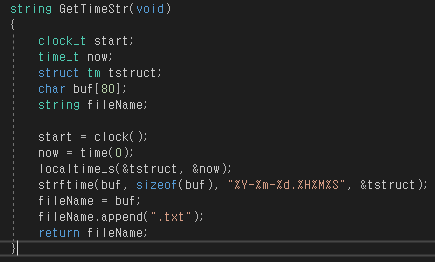
- “KeyCode” “Octave” “Input Time” “DownOrUp”

- Input Time is in millisecond(ms)

- DownOrUp : 1 is down, 0 is up

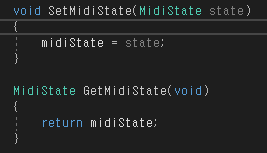
- File name : Time when pushing Record button

Ex) 



- Check now state ( play, record, load )

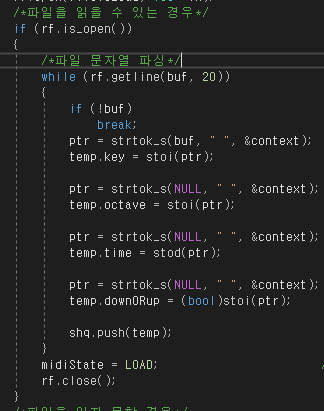
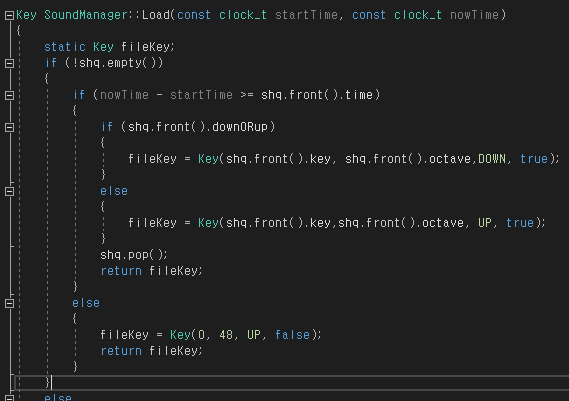




- Recording & Load method

Calculate the difference between when key puts and start time, make/check key’s state up or down

+) Load : make queue of contents of the file. Read file and push queue, next pop and play it.



**4) Overall implementation Class**

○ Midi

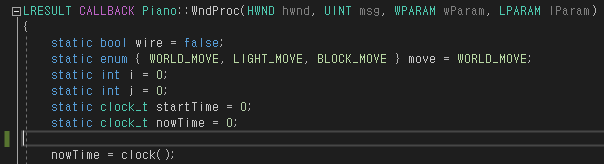
- Core of this program, Make a sound

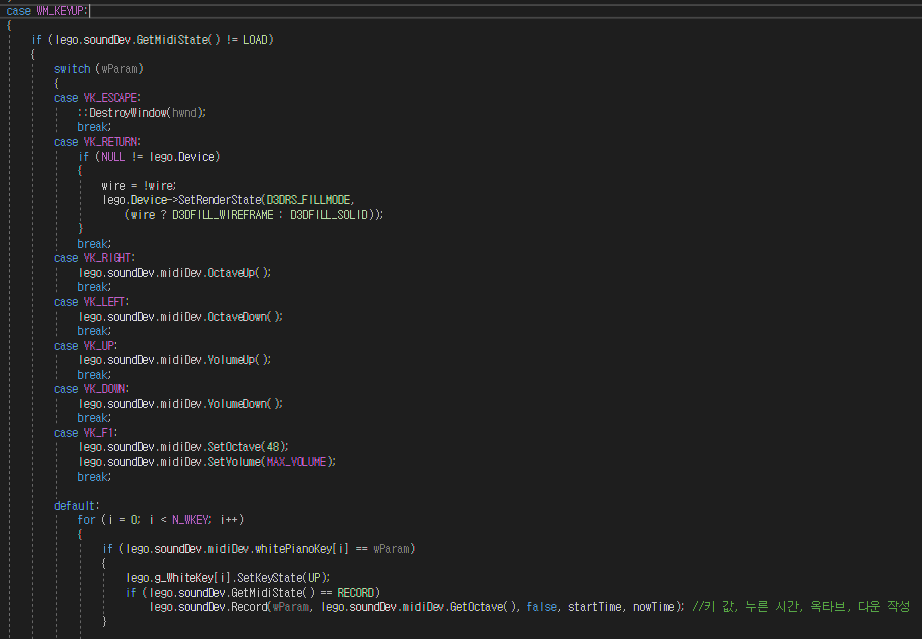
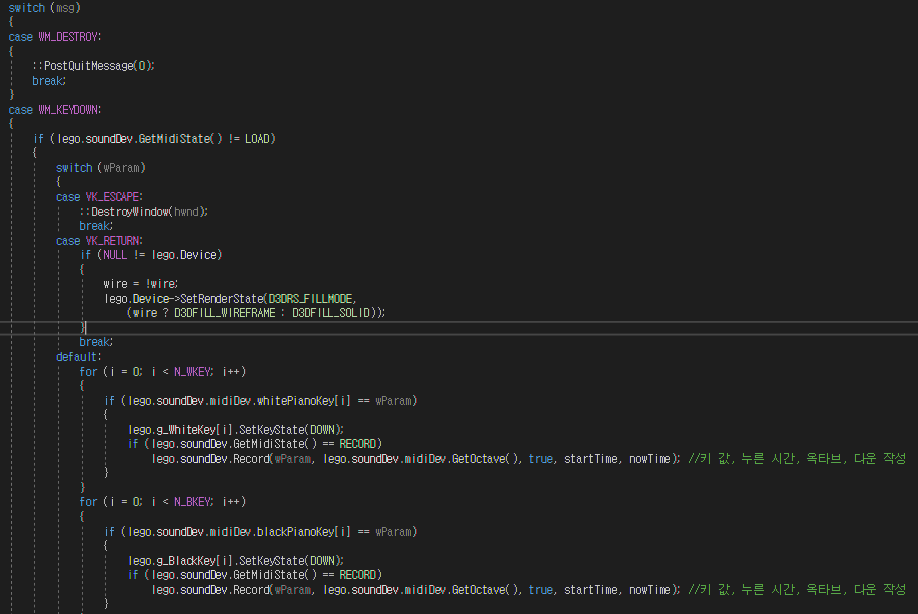
- Use given Midi Class

○ Piano

- Class related to window’s action ( keydown, click )

- Using EnterMsgLoop, WndProc Function, get user’s action



-> KeyDown -> KeyUp

○ VirtualLego

- Create overall interface

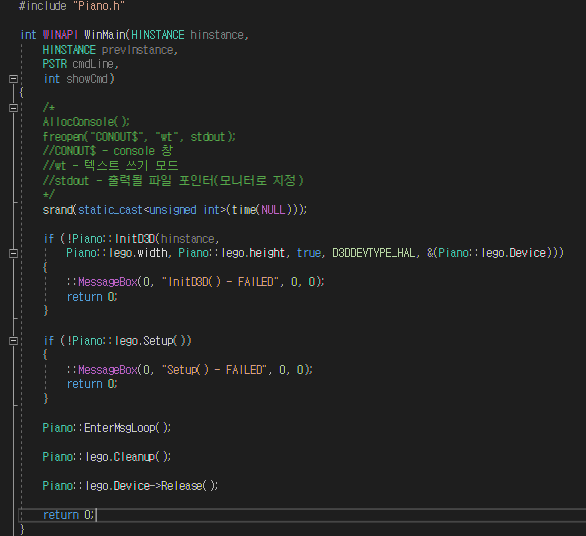
- Draw all sprite, texture, font, image, icon in VirtualLego.

- In every frame, update changes and draw repeatedly.

○ Source

- Have main function

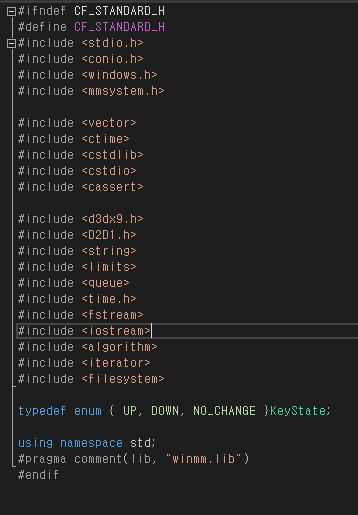
( in windows system, WinMain is main function )



- Including Piano class, create windows and get message ( user’s action ) Continuously

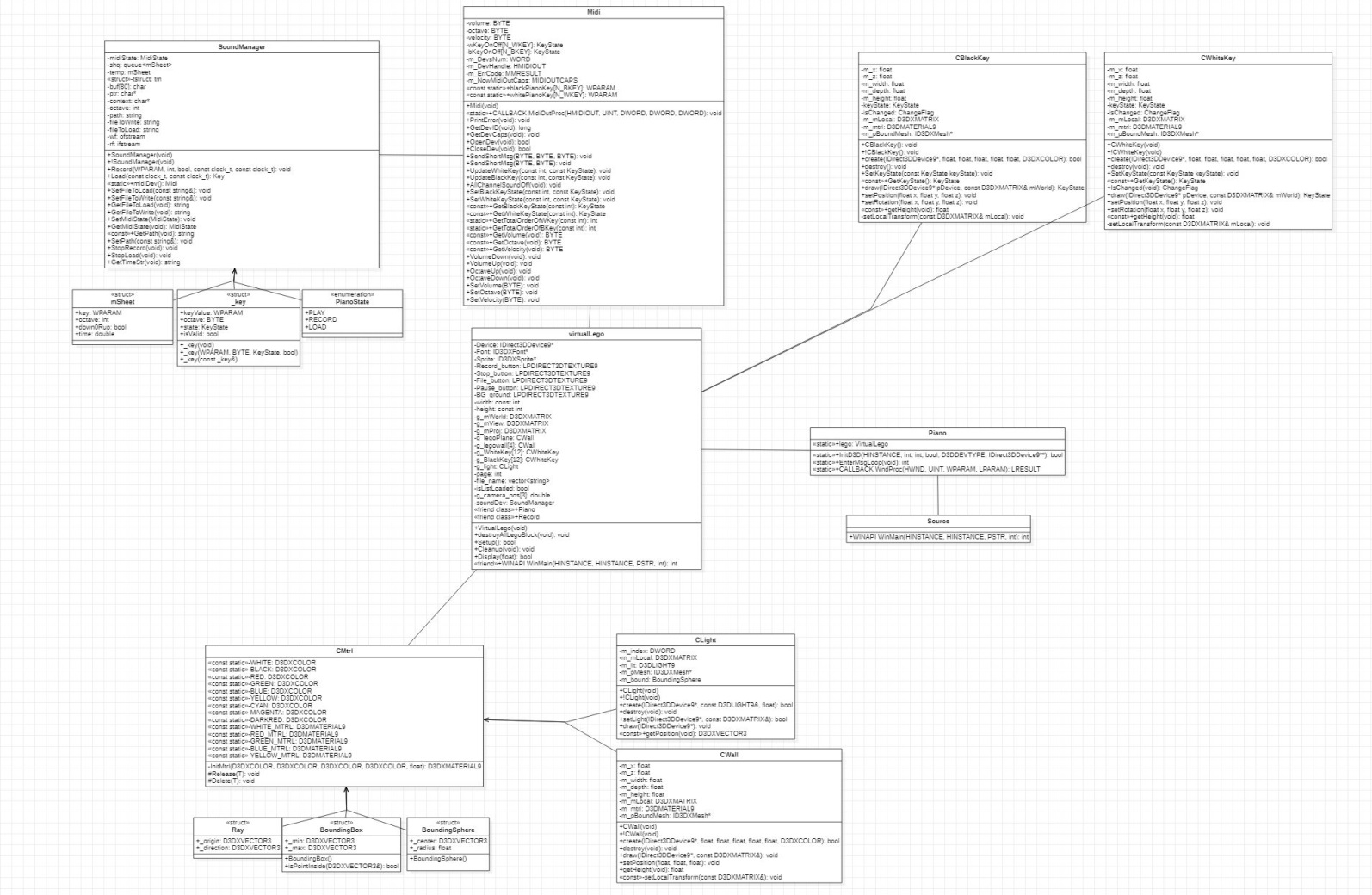
**5) Library Class**

○ Standard



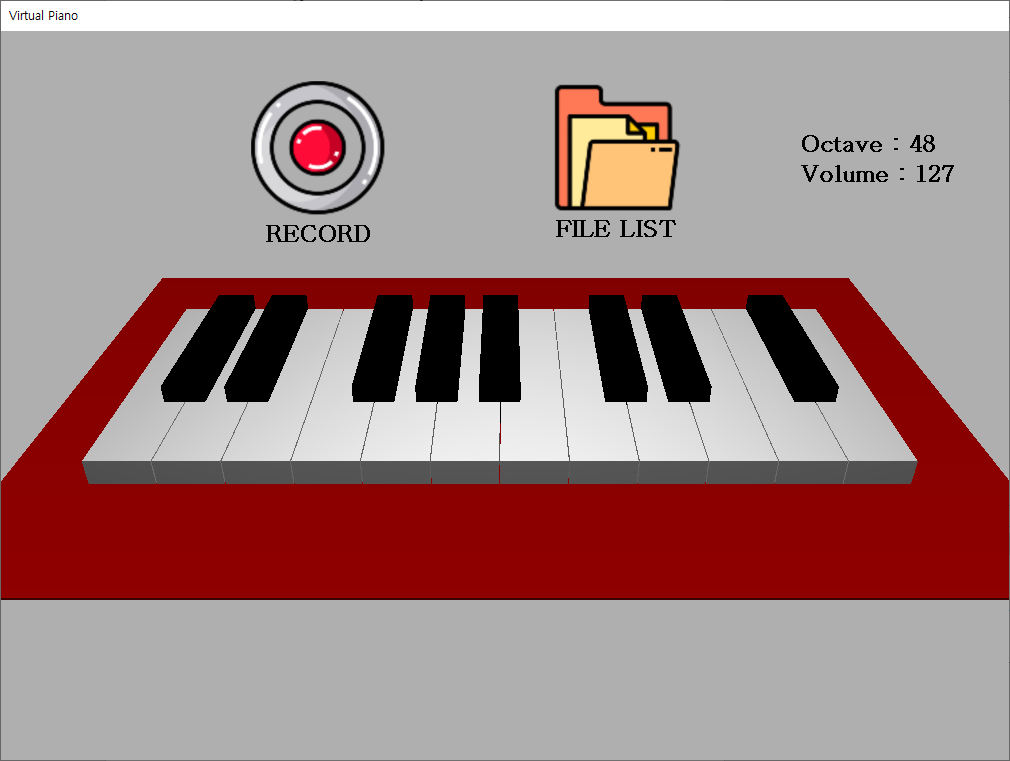
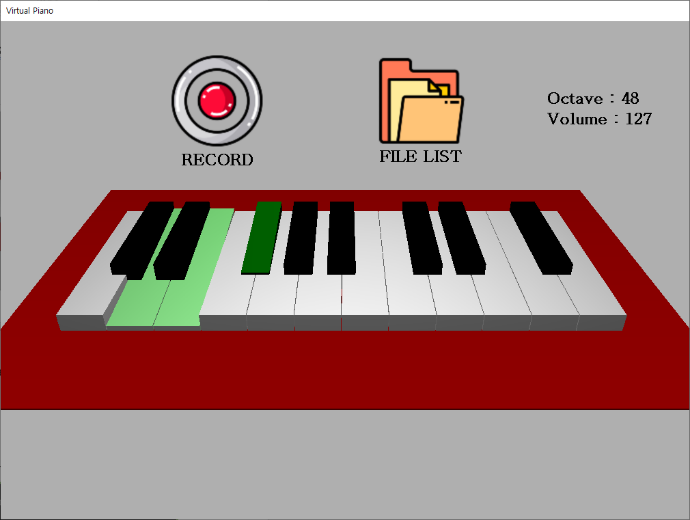
- all other classes include this class

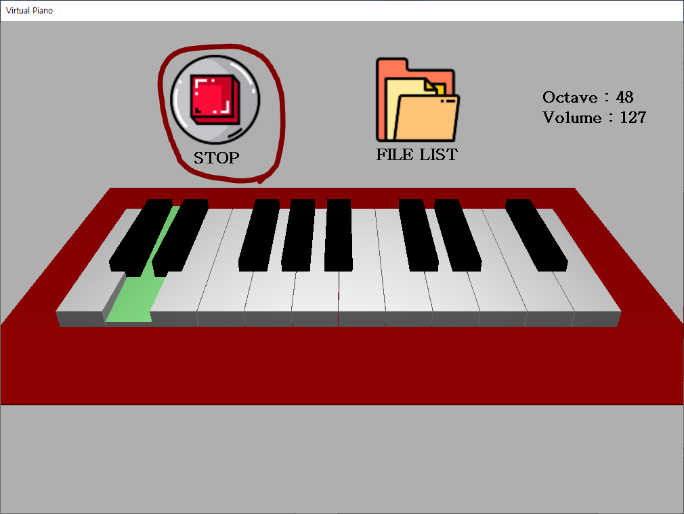
**5. UML Design for our system**

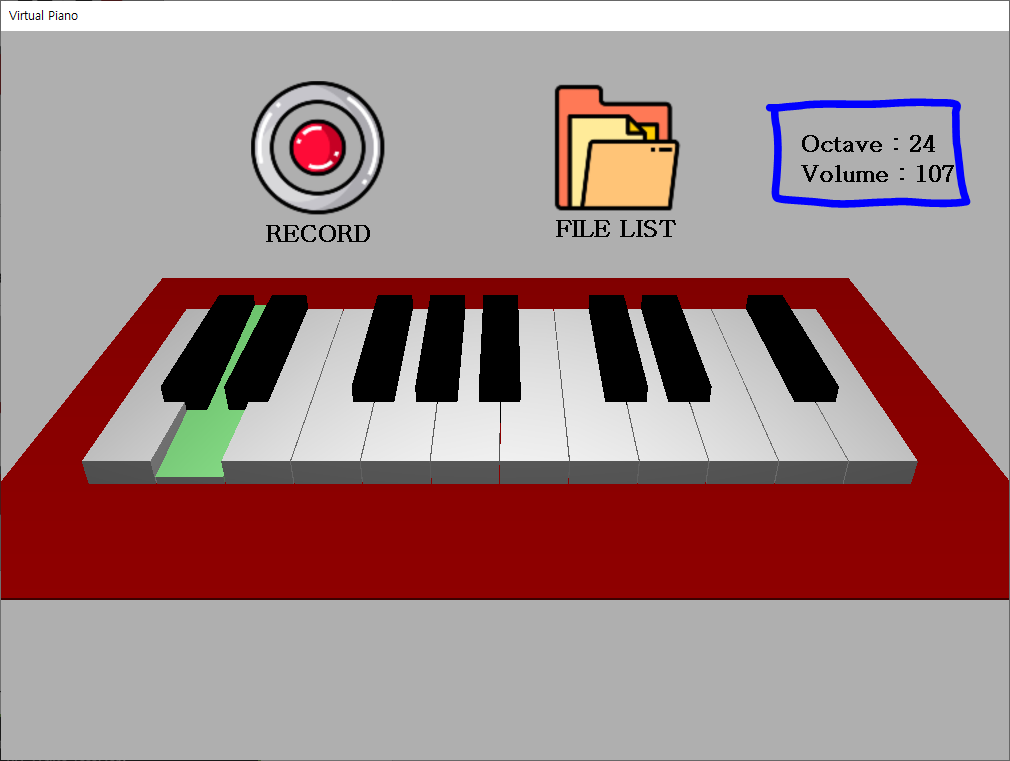


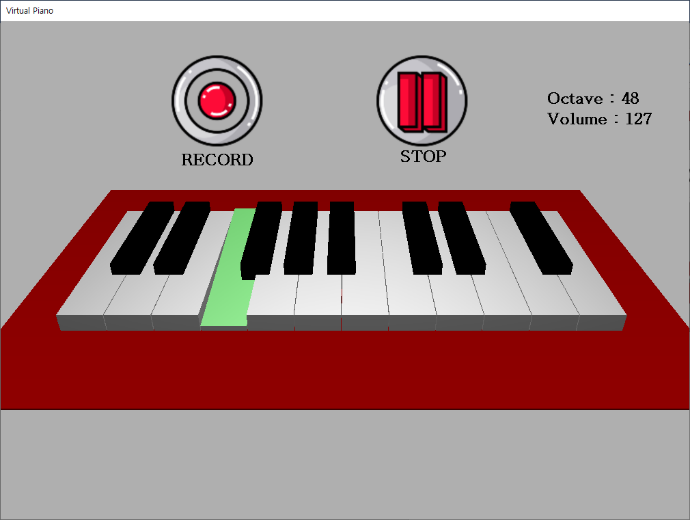
**6. Execution Result**

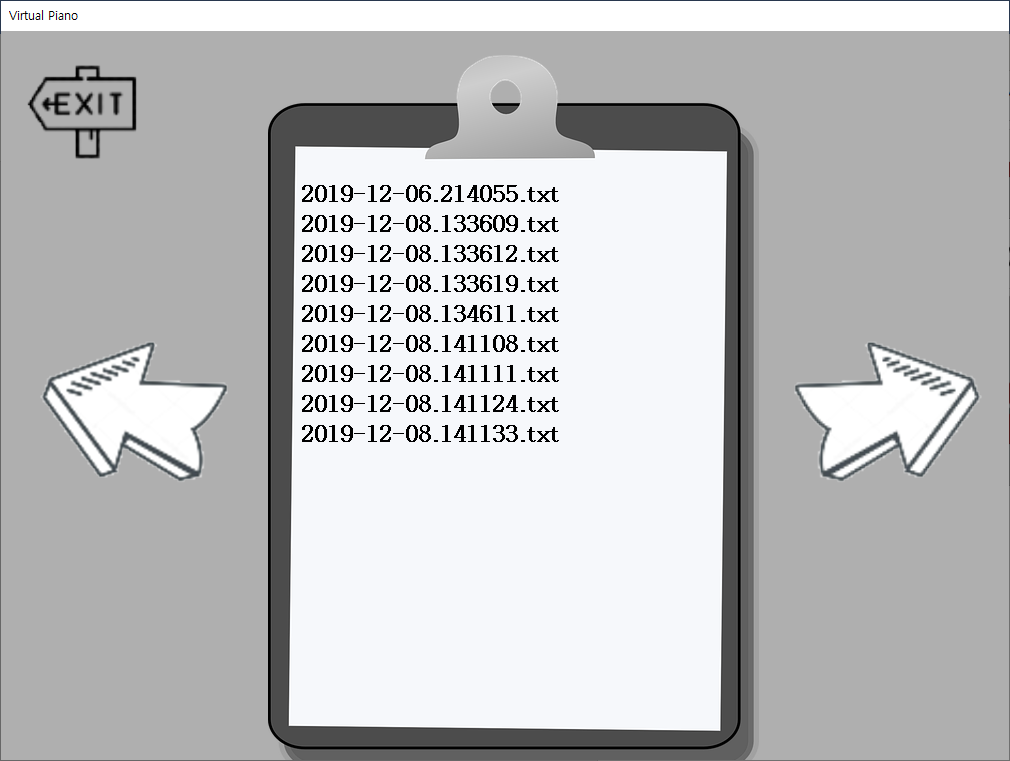
1) main interface 2) when pushed key



3) change Octave, Volume 4) push Record Button



5) Open File List 6) Load File



**7. How we applied Object Oriented Concepts**

1) Encapsulation : We separate many functions and variables into some classes and object

2) Inheritance : We use inheritance system

( CMtrl <-> other interface classes )

**8. Conclusion**

-We make Virtual Piano that can Record, Load file as well as Play piano.

-We delete almost all bugs we found.

-Users can play freely.