

# ELECTRONIC PIANO

## GROUP 12

---

- Ngo Thi Thu Huyen 20200289: Design GUI
- Vu Tan Khai 20200311 : Sound of piano
- Nguyen Duy Khanh 20204914: Music recording
- Nguyen Ngoc Khanh 20204915: Design GUI



# Problem Statement

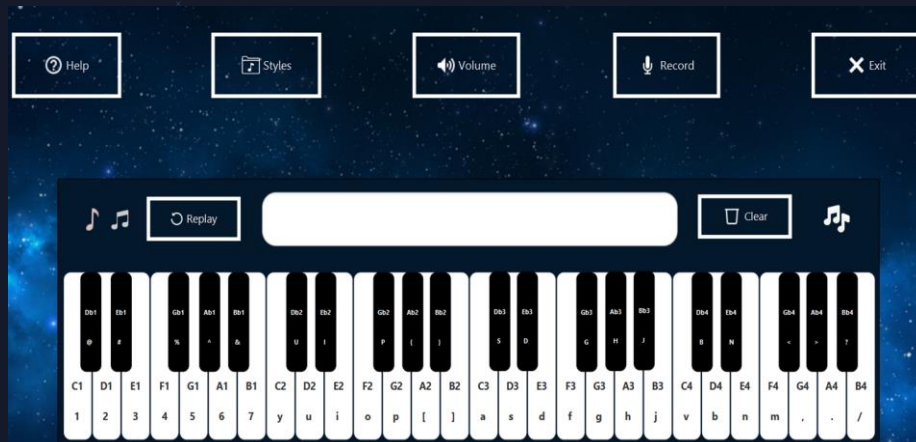
Piano is a popular musical instrument. In this project, we implemented an application that provides GUI for the user to virtually play an electronic piano

On the main menu: title of the application, piano GUI , help menu, quit

- User can play the piano by interacting with GUI
- Help menu shows the basic usage and aim of the program
- Quit exits the program. Remember to ask for confirmation

Keyboard: C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Xi), ... and other functional button:

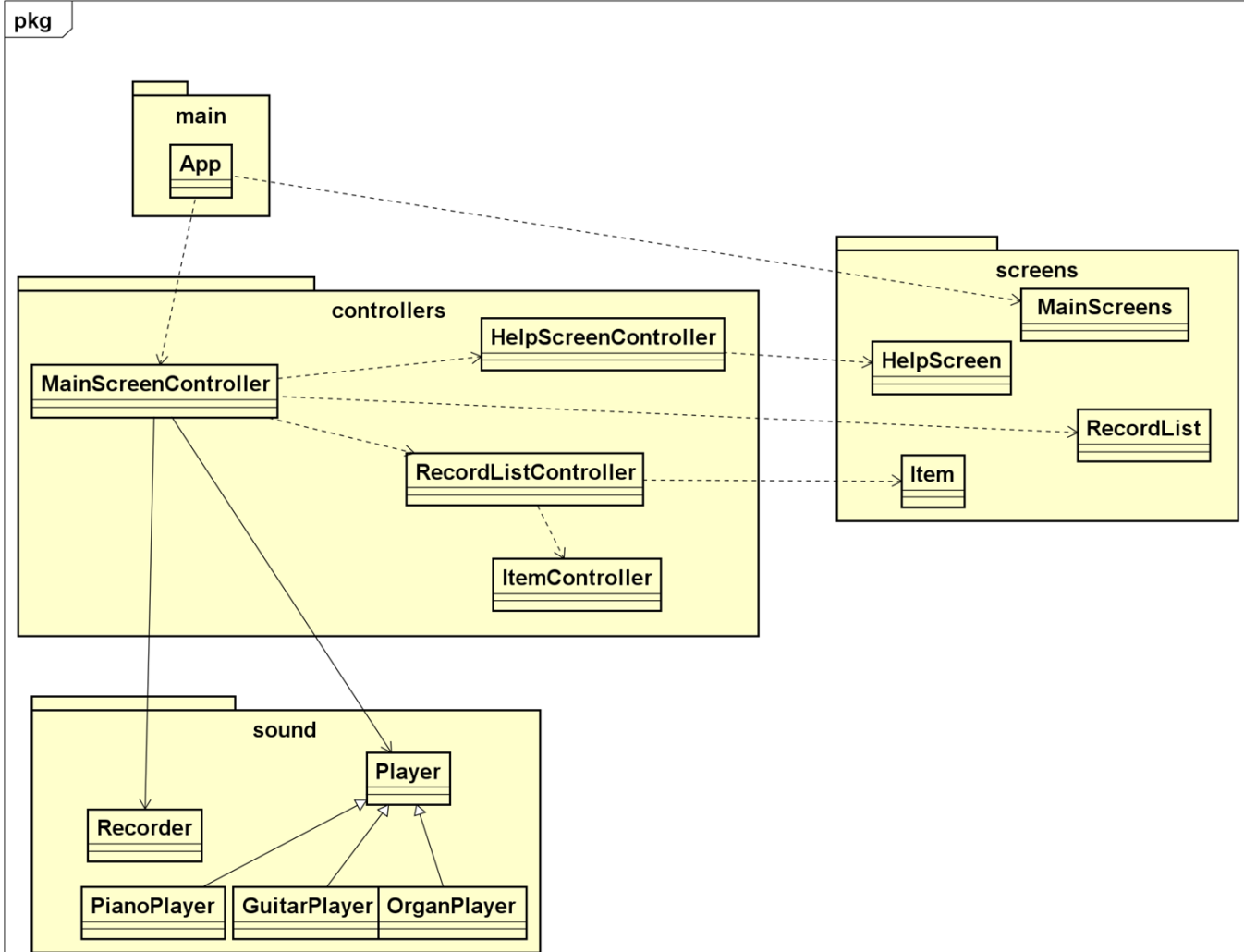
- A button for increasing/decreasing volume
- A record button to record the play and play the record
- A button for changing music style
- Music sheet to write down notes played and replay it



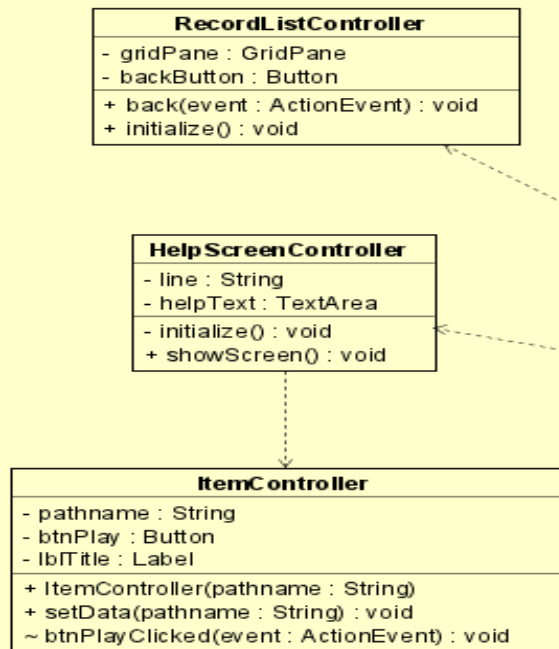
## Use case diagram



# General class diagram



## package controllers



## controllers

### MainScreenController

```

- inUse : Set<Character> = new HashSet<>()
~ pressedWhiteNStyle : String = "-fx-backg...
~ releasedWhiteNStyle : String = "-fx-backg...
~ pressedBlackNStyle : String = "-fx-backg...
~ releasedBlackNStyle : String = "-fx-backg...
- keyString : String = "1!2@3#4$5%6^7&8*9(0)-_={[]};:,<.>/?"
- btnVolume : Button
- volumePane : AnchorPane
- volumeSlider : Slider
- OptionStyles : HBox
- btnStyles : Button
- buttonMap : Map<Character,Button> = new HashMap<>()
- player : Player = new PianoPlayer()
- startRecordButton : Button
- finishRecordButton : Button
- playRecordButton : Button
- btnClear : Button
- notePlayed : TextField
- recorder : Recorder = new Recorder()

~ usePiano() : void
~ useGuitar() : void
~ useOrgan() : void
- initialize() : void
- showRecordButton(event : ActionEvent) : void
+ hideVolumeController() : void
+ closePlayer() : void
- reduceLyric(song : String) : String
- getKeyString(c : char) : String
+ pressNote(c : char, shiftOn : boolean) : void
- resetStyle(button : Button) : void
+ releaseNote(c : char, shiftDown : boolean) : void
- setKeyMap() : void
- setMouseEvent() : void
~ btnHelpPressed(event : ActionEvent) : void
~ btnExitPressed(event : ActionEvent) : void
~ btnStylesPressed(event : ActionEvent) : void
~ startRecord(event : ActionEvent) : void
~ finishRecord(event : ActionEvent) : void
~ playRecord(event : ActionEvent) : void
~ startReplay(event : ActionEvent) : void
~ clearText(event : ActionEvent) : void
~ showList(event : ActionEvent) : void
  
```

# sound package and main package

## sound

### Recorder

```
- count : int = 0
~ file_path : String
~ fileType : AudioFormat.Type = AudioFormat.Type.WAVE
~ line : TargetDataLine

~ getAudioFormat() : AudioFormat
~ start() : void
+ finish() : void
+ begin() : void
+ main(args : String[]) : void
+ play() : void
```

### Player

```
- volume : byte
- text : String = ""
# jfPlayer : RealtimePlayer

+ getText() : String
+ setText(str : String) : void
+ Player()
+ stop() : void
- getNote(str : String, octave : int) : Note
# playNote(str : String, octave : int) : void
+ setVolume(volume : float) : void
+ getVolume() : byte
+ playNote(note : String) : void
+ stopNote(note : String) : void
```

### GuitarPlayer

```
+ GuitarPlayer()
+ playNote(note : String) : void
+ stopNote(note : String) : void
```

### OrganPlayer

```
+ OrganPlayer()
+ playNote(note : String) : void
+ stopNote(note : String) : void
```

### PianoPlayer

```
+ PianoPlayer()
+ playNote(note : String) : void
+ stopNote(note : String) : void
```

## main

### App

```
+ start(stage : Stage) : void
+ main(args : String[]) : void
```

## Inheritance

Player

```
void playNote(String);  
void stop();
```

PianoPlayer

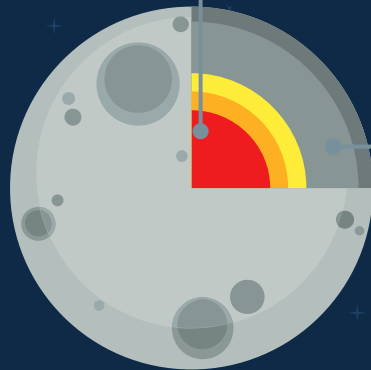
GuitarPlayer

OrganPlayer

## Encapsulation

Fields

```
private byte volume;  
private Player jfPlayer;
```



Methods

```
public byte getVolume();  
public void setVolume();  
public void playNote();
```

## Method polymorphism

```
void playNote(str: String, octave : int)
```

```
void playNote(str: String)
```

## Object polymorphism

```
public class PianoPlayer extends Player {  
    @Override  
    public void playNote(String note) {  
        playNote(note, 5);  
    }  
    // ...  
}
```

```
public class OrganPlayer extends Player {  
    @Override  
    public void playNote(String note) {  
        playNote(note, 4);  
    }  
    // ...  
}
```

---

```
C1.setOnMousePressed(e -> {  
    player.playNote("C1");  
});
```



# Association

## MainScreenController



- player : Player
- recorder : Recorder

## Player



Manage application sound  
Log music sheet

## Recorder



Record audio  
Replay recorded audio

# Record Video Demo

<https://drive.google.com/file/d/15ZqrS5VQMVxqEJqMDgXy39Y1hQuKwGFr/view?usp=sharing>