

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

PROJECT

Object-Oriented Programming

Electronic Piano

|  |  |
| --- | --- |
| ***Instructor:*** | ***Prof.Nguyen Thi Thu Trang*** |

*Group 13: Le Minh Hieu 20215048*

*Tran Sy Hieu 20215053*

*Nguyen Manh Hieu 20215050*

*Nguyen Viet Hieu 20215051*

I.Assignment of members:

1. Le Minh Hieu 20215048

* Design GUI
* handle button and test the features of app
* Record demo video
* JButton E,F,G,A,B,C\_1,Bb
* Create the frame

1. Tran Sy Hieu 20215053

* Design GUI
* handle button and test the features of app
* JButton C1,D1,E1,F1,D\_3
* Design presentation Slides
* Report editor

1. Nguyen Manh Hieu 20215050

* Design GUI
* Sound of piano
* JButton D\_2,F\_2
* Main writer for the report

1. Nguyen Viet Hieu 20215051

* Design GUI
* JButton btnNewButton
* handle button and test the features of app

II.Mini-project description:

Mini-project requirement:

-Triển khai ứng dụng cung cấp GUI để người dùng chơi đàn piano điện tử ảo.

-Yêu cầu thiết kế:

+ Trên menu chính: tiêu đề ứng dụng, GUI piano, menu trợ giúp, thoát

• Người dùng có thể chơi piano bằng cách tương tác với GUI

• Menu trợ giúp hiển thị cách sử dụng cơ bản và mục đích của chương trình

• Thoát thoát khỏi chương trình. Yêu cầu xác nhận để thoát

+ Trình diễn:

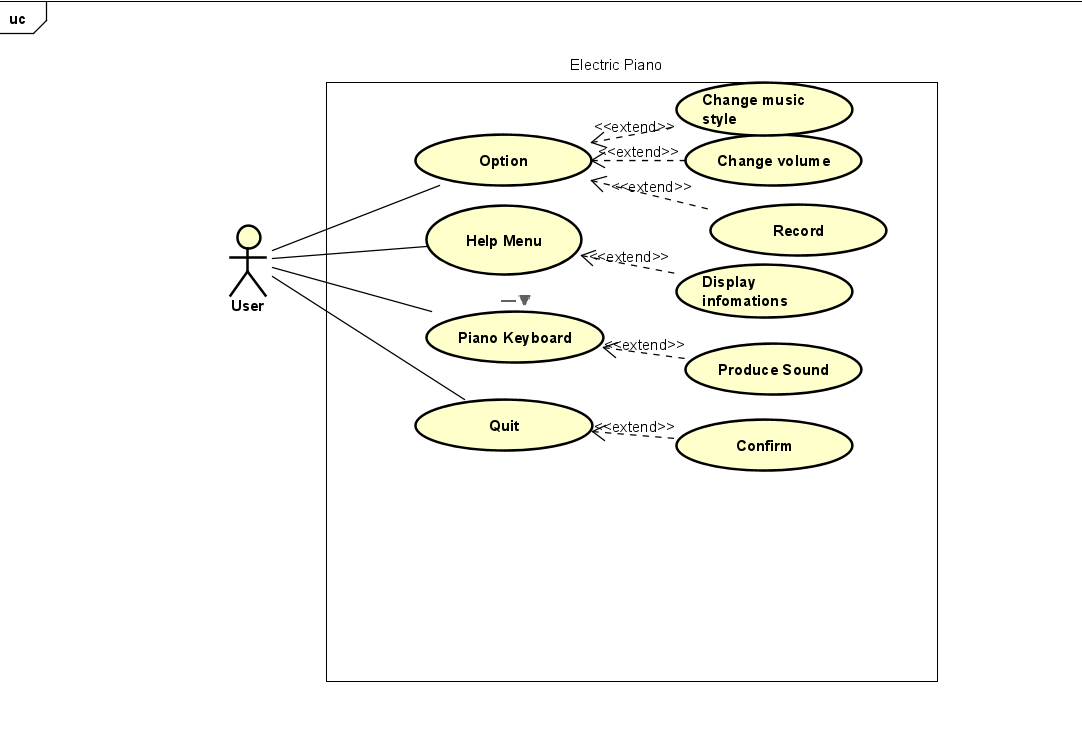
• Bàn phím: C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Xi). Bàn phím được thiết kế càng chi tiết càng tốt.

• Nút tăng/giảm âm lượng (tùy chọn)

• Nút ghi để ghi lại quá trình phát (tùy chọn)

• Nút thay đổi kiểu nhạc (tùy chọn)

2.Use case diagram:



Explanation:

-Chơi đàn piano bằng chuột: Người dùng cũng có thể chơi đàn piano điện tử bằng cách nhấp chuột trái vào phím đàn piano.

-Thoát ứng dụng: bấm vào nút thoát để kết thúc chương trình.

III.Design:

1.General class diagram:

A diagram of a computer

Description automatically generated

2.Detail class diagram:

A diagram of a computer

Description automatically generated with medium confidence

1. **Giới Thiệu**

Đây là một ứng dụng đơn giản nhưng hoạt động, cho phép người dùng tương tác với một đàn piano ảo thông qua giao diện người dùng. Ứng dụng được xây dựng bằng ngôn ngữ lập trình Java và sử dụng thư viện Swing để tạo giao diện người dùng.

1. **Các Chức Năng Chính**
2. **Giao Diện Người Dùng:**

* Cửa sổ chính của ứng dụng có kích thước cố định và chứa các nút đại diện cho các phím piano.
* Các nút được xác định cụ thể về vị trí và kích thước trong giao diện sử dụng ‘**null layout’**.

1. **Tương Tác với Phím Piano:**

* Mỗi nút đại diện cho một phím trên đàn piano.
* Khi người dùng nhấn một nút, âm thanh tương ứng với phím piano đó được phát ra.

1. **Các Thành Phần Chính của Mã**
2. **Lớp ‘Piano’**: Lớp này mở rộng từ **JFrame**, tạo cửa sổ chính cho ứng dụng.

* Khởi tạo cửa sổ chính với tiêu đề và kích thước cố định.
* Tạo một **JPanel** (**contentPane**) để chứa các thành phần giao diện.
* Thiết lập bố cục của **contentPane** là **null** layout, nghĩa là vị trí và kích thước của mỗi thành phần sẽ được xác định cụ thể bằng cách sử dụng **setBounds()**.

1. **Nút đàn Piano (JButton):** Các nút này được tạo để đại diện cho các phím đàn piano.

* Mỗi nút tượng trưng cho một phím và được đặt vị trí cụ thể trong giao diện.
* Mỗi nút có một ActionListener để xử lý sự kiện khi nút được nhấn.

1. **Xử lý Event**

* Mỗi nút piano có một **ActionListener** đi kèm.
* Các phím đen được tạo ra và đặt giữa các phím trắng tương ứng.
* Trong mỗi **ActionListener**, mã xử lý sẽ cố gắng tải file âm thanh tương ứng với phím piano bằng cách sử dụng **AudioSystem** và **Clip** từ gói **javax.sound.sampled**, sau đó phát file âm thanh đó.

1. **Các Phím Đen và Phím Trắng**

* Các phím đen được tạo ra và đặt giữa các phím trắng tương ứng.
* Các phím đen có kích thước nhỏ hơn và được đặt vị trí để phù hợp với bố cục của đàn piano.

1. **Điểm Mạnh**
2. **Đơn Giản và Dễ Sử Dụng:** Ứng dụng cung cấp một giao diện rõ ràng và đơn giản cho người dùng để tương tác với đàn piano ảo.
3. **Âm Thanh Phát Ra Sắc Nét:** Sử dụng **AudioSystem** và **Clip** để phát âm thanh tương ứng với phím piano được nhấn.

Đoạn mã sử dụng và thể hiện một số tính chất của lập trình hướng đối tượng (OOP):

1. Gói (Encapsulation):

* Trong đoạn mã, việc đóng gói thể hiện qua việc nhóm các thành phần giao diện người dùng (như JFrame, JPanel, JButton) và các phương thức xử lý sự kiện vào trong lớp Piano.
* Hành vi của các nút piano, bao gồm việc phát âm thanh, được đóng gói trong các phương thức tương ứng.

1. Trừu Tượng (Abstraction):

* Sử dụng các thành phần giao diện như JButton để trừu tượng hóa việc tạo ra các phím piano trên giao diện người dùng.
* Việc xử lý sự kiện thông qua các phương thức đã được định nghĩa sẵn trừu tượng hóa việc xử lý cụ thể bên trong từng nút.

1. Đa Hình (Polymorphism):

* Mỗi nút piano có một ActionListener và một phương thức actionPerformed. Mặc dù chúng có cùng kiểu (ActionListener), nhưng mỗi phương thức actionPerformed thực hiện hành động riêng biệt khi nút tương ứng được nhấn.
* Mặc dù không sử dụng đa hình một cách rõ ràng, việc triển khai các phương thức actionPerformed cho các nút piano có thể được coi là một dạng cơ bản của tính đa hình.

1. Sự Kiện và Xử Lý Sự Kiện:

* Sử dụng ActionListener để xử lý sự kiện khi người dùng nhấn vào các nút piano, cho phép kích hoạt các hành động cần thiết như phát âm thanh khi cần.

V. Kế Thừa (Inheritance):

* Trong đoạn mã, không có sử dụng trực tiếp kế thừa từ một lớp cha khác. Tuy nhiên, sử dụng các thành phần giao diện có thể được coi là việc sử dụng tính kế thừa gián tiếp.