TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Xây dựng ứng dụng

GAME CỜ VUA BẰNG PYTHON

GVHD: Từ Lãng Phiêu
SV: Nguyễn Hoàng Phát - 3121410377
Nguyễn Văn Tân - 3121410444
Nguyễn Minh Tiến - 3121560089
Mai Nguyễn Hoàng Dũng - 3121410109

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

1 PHẦN GIỚI THIỆU

1.1 Giới thiệu về dự án

Dự án làm game cờ vua bằng Python với thư viện Kivy là một nỗ lực sáng tạo nhằm mang lại trải nghiệm chơi cờ vua kỹ thuật số trực quan và hấp dẫn. Mục tiêu chính của dự án là phát triển một ứng dụng cờ vua có giao diện đẹp, dễ sử dụng và chạy trên nền tảng Linux, và. Thư viện Kivy, một công cụ mạnh mẽ và linh hoạt cho việc phát triển ứng dụng GUI đa nền tảng bằng Python, sẽ được sử dụng làm nền tảng chính cho dự án này.

Dự án này sẽ bao gồm các chức năng cơ bản của một trò chơi cờ vua như di chuyển quân cờ, kiểm tra luật chơi hợp lệ, và xác định thắng thua. Bên cạnh đó, nó còn tích hợp các tính năng nâng cao như lưu và tải lại ván cờ, chế độ chơi hai người. Giao diện người dùng sẽ được thiết kế để trực quan và thân thiện, với các hiệu ứng đồ họa mượt mà và phản hồi nhanh chóng.

Một trong những thách thức lớn nhất của dự án là đảm bảo tính chính xác của luật chơi cờ vua và cung cấp trải nghiệm người dùng mượt mà trên nhiều thiết bị. Nhóm phát triển sẽ sử dụng Python cho logic của trò chơi và Kivy để tạo giao diện người dùng, tận dụng các tính năng của Kivy như xử lý cảm ứng và đa nền tảng. Dự án sẽ được quản lý theo phương pháp Agile, với các giai đoạn phát triển, thử nghiệm và cải tiến liên tục để đảm bảo chất lượng sản phẩm cuối cùng.

Ngoài ra, dự án sẽ được mở mã nguồn, cho phép cộng đồng tham gia đóng góp ý kiến và cải thiện. Đây là cơ hội tuyệt vời để những người yêu thích lập trình và cờ vua cùng nhau học hỏi và phát triển kỹ năng. Với sự kết hợp giữa Python và Kivy, dự án này hứa hẹn sẽ mang lại một trò chơi cờ vua số đầy thú vị và thách thức.

Link git hub của dự án: https://github.com/minewaku/chess-game/blob/main/README.md

1.2 Phân công

- *Nguyễn Hoàng Phát - 3121410377*
  Phụ trách phần lập trình các module player, point, side, hintDot, playerInfo.
- *Nguyễn Văn Tân - 3121410444*
  Phụ trách phần lập trình các module promotionPopup, square, surrenderButton, timeCounter.
- *Nguyễn Minh Tiến - 3121560089*
  Phụ trách phần lập trình các module gameScene, logger, move.
- *Mai Nguyễn Hoàng Dũng - 3121410109*
  Phụ trách phần lập trình các module piece, board, main.

2 NỘI DUNG

2.1 Cơ sở lý thuyết

- **Ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình bậc cao, được Guido van Rossum phát triển và ra mắt lần đầu tiên vào năm 1991. Python nổi tiếng với cú pháp rõ ràng, dễ đọc, và dễ học, khiến nó trở thành lựa chọn phổ biến cho cả người mới bắt đầu lẫn các lập trình viên chuyên nghiệp. Một trong những đặc điểm nổi bật của Python là triết lý thiết kế nhấn mạnh vào tính đơn giản và khả năng tái sử dụng mã nguồn.

Python là ngôn ngữ đa năng, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau như phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, phát triển phần mềm, tự động hóa hệ thống, và nhiều lĩnh vực khác. Nó hỗ trợ nhiều phong cách lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm, và lập trình thủ tục.

Một trong những ưu điểm lớn nhất của Python là hệ sinh thái thư viện phong phú và mạnh mẽ. Các thư viện như NumPy, Pandas, Matplotlib, và TensorFlow giúp các nhà khoa học dữ liệu và kỹ sư trí tuệ nhân tạo xử lý dữ liệu và xây dựng mô hình dễ dàng. Django và Flask là những khung công tác phổ biến để phát triển web, giúp tạo ra các ứng dụng web mạnh mẽ và bảo mật.

Python cũng được biết đến với cộng đồng người dùng lớn và nhiệt tình. Các lập trình viên Python có thể dễ dàng tìm kiếm sự trợ giúp, chia sẻ kiến thức, và đóng góp vào các dự án mã nguồn mở trên các nền tảng như GitHub, Stack Overflow, và Reddit.

Python được cài đặt sẵn trên nhiều hệ điều hành như macOS và Linux, và có thể dễ dàng cài đặt trên Windows. Trình thông dịch Python (Python interpreter) cho phép thực thi mã lệnh ngay lập tức, hỗ trợ việc thử nghiệm và phát triển nhanh chóng.

Nhờ tính linh hoạt, dễ học và mạnh mẽ, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được các công ty công nghệ lớn và các tổ chức giáo dục tin dùng.

Ưu điểm
Cú Pháp Dễ Hiểu và Dễ Học: Python có cú pháp rõ ràng, ngắn gọn và dễ đọc, giúp người mới bắt đầu nhanh chóng làm quen và tiếp cận.
Thư Viện Phong Phú: Python có hệ sinh thái thư viện rộng lớn và mạnh mẽ, từ NumPy và Pandas cho khoa học dữ liệu đến Django và Flask cho phát triển web.
Đa Năng: Python hỗ trợ nhiều phong cách lập trình khác nhau như lập trình hướng đối tượng, lập trình hàm, và lập trình thủ tục.
Cộng Đồng Hỗ Trợ Tích Cực: Python có một cộng đồng người dùng lớn và nhiệt tình, cung cấp rất nhiều tài liệu, hướng dẫn và hỗ trợ.
Khả Năng Mở Rộng: Python có thể tích hợp với các ngôn ngữ khác như C/C++ thông qua các gói mở rộng, cho phép tối ưu hóa hiệu năng khi cần thiết.
Tính Đa Nền Tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux, và các nền tảng di động.

Nhược điểm
Hiệu Năng Thấp Hơn: Python thường chậm hơn so với các ngôn ngữ biên dịch như C++ hoặc Java, đặc biệt là trong các ứng dụng yêu cầu hiệu năng cao.

Quản Lý Bộ Nhớ Kém: Python sử dụng quản lý bộ nhớ tự động (garbage collection), đôi khi dẫn đến việc tiêu thụ bộ nhớ không hiệu quả.

Thiếu Tính Di Động Trên Ứng Dụng Di Động: Mặc dù có thể phát triển ứng dụng di động với Python, nhưng so với các nền tảng phát triển di động chuyên dụng như Swift (iOS) hoặc Kotlin (Android), Python ít phổ biến hơn.

- **Thư viện kivy**

Kivy là một thư viện mã nguồn mở dành cho việc phát triển ứng dụng đa nền tảng với giao diện người dùng phong phú, được viết bằng Python và Cython. Ra mắt lần đầu vào năm 2011, Kivy nhanh chóng thu hút sự chú ý của cộng đồng lập trình nhờ khả năng tạo ra các ứng dụng có giao diện đồ họa đẹp mắt và hiệu năng cao trên nhiều hệ điều hành như Windows, macOS, Linux, Android, và iOS.

Các Đặc Điểm Nổi Bật của Kivy
Đa Nền Tảng: Kivy cho phép viết một lần và chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn. Điều này giúp tiết kiệm thời gian và công sức cho các nhà phát triển khi muốn mở rộng ứng dụng của họ sang nhiều hệ điều hành.

Hỗ Trợ Giao Diện Cảm Ứng: Kivy được thiết kế với khả năng hỗ trợ giao diện cảm ứng từ đầu, phù hợp cho việc phát triển ứng dụng di động và các thiết bị có màn hình cảm ứng.

Công Cụ Widget Phong Phú: Thư viện Kivy cung cấp một loạt các widget để xây dựng giao diện người dùng, từ các nút bấm, thanh trượt, đến các thành phần phức tạp như đồ thị và hình ảnh động.

Hiệu Năng Cao: Được viết bằng Python và Cython, Kivy tận dụng tối đa khả năng của phần cứng để cung cấp hiệu năng cao và mượt mà, ngay cả với các ứng dụng đồ họa phức tạp.

Hỗ Trợ Đồ Họa Tăng Tốc (GPU): Kivy sử dụng OpenGL ES 2 để tăng tốc đồ họa, giúp các ứng dụng có giao diện đẹp và hiệu ứng mượt mà.

Tài Liệu và Cộng Đồng Hỗ Trợ: Kivy có tài liệu phong phú và chi tiết, cùng với cộng đồng người dùng tích cực, giúp các lập trình viên dễ dàng tìm kiếm sự trợ giúp và chia sẻ kiến thức.

Ứng Dụng của Kivy Phát Triển Ứng Dụng Di Động: Kivy là lựa chọn tuyệt vời để phát triển ứng dụng di động chạy trên cả Android và iOS.

Ứng Dụng Đa Phương Tiện: Với khả năng xử lý đồ họa và cảm ứng tốt, Kivy thường được sử dụng để phát triển các ứng dụng đa phương tiện và trò chơi.

Giao Diện Người Dùng Tùy Chỉnh: Kivy phù hợp cho các ứng dụng đòi hỏi giao diện người dùng tùy chỉnh và phức tạp, như các hệ thống điều khiển và hiển thị thông tin.

Nhờ các tính năng nổi bật và linh hoạt, Kivy đã trở thành một công cụ quan trọng trong việc phát triển ứng dụng đa nền tảng, mang lại nhiều lợi ích cho các lập trình viên và dự án phát triển phần mềm.

Ưu điểm
Đa Nền Tảng: Kivy cho phép phát triển ứng dụng chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi mã nguồn.

Hỗ Trợ Cảm Ứng: Kivy được thiết kế để hỗ trợ tốt giao diện cảm ứng, lý tưởng cho các ứng dụng di động và các thiết bị cảm ứng.

Đồ Họa Tăng Tốc (GPU): Kivy sử dụng OpenGL ES 2 để tăng tốc đồ họa, giúp ứng dụng chạy mượt mà và hiển thị đẹp mắt.

Widget Phong Phú: Kivy cung cấp nhiều widget có thể tùy chỉnh, giúp dễ dàng xây dựng giao diện người dùng phong phú và phức tạp.

Mã Nguồn Mở: Kivy có giấy phép MIT, cho phép tự do sử dụng, chỉnh sửa và phân phối mã nguồn, khuyến khích sự phát triển và đóng góp từ cộng đồng.

Nhược điểm

Hiệu Năng Chưa Tối Ưu: Dù có hỗ trợ tăng tốc GPU, hiệu năng của Kivy vẫn chưa thể so sánh với các công cụ phát triển ứng dụng di động gốc như Swift cho iOS hoặc Kotlin cho Android.

Độ Phức Tạp: Đối với những người mới bắt đầu, Kivy có thể có chút phức tạp do cấu trúc và cú pháp khác biệt so với các thư viện GUI truyền thống.

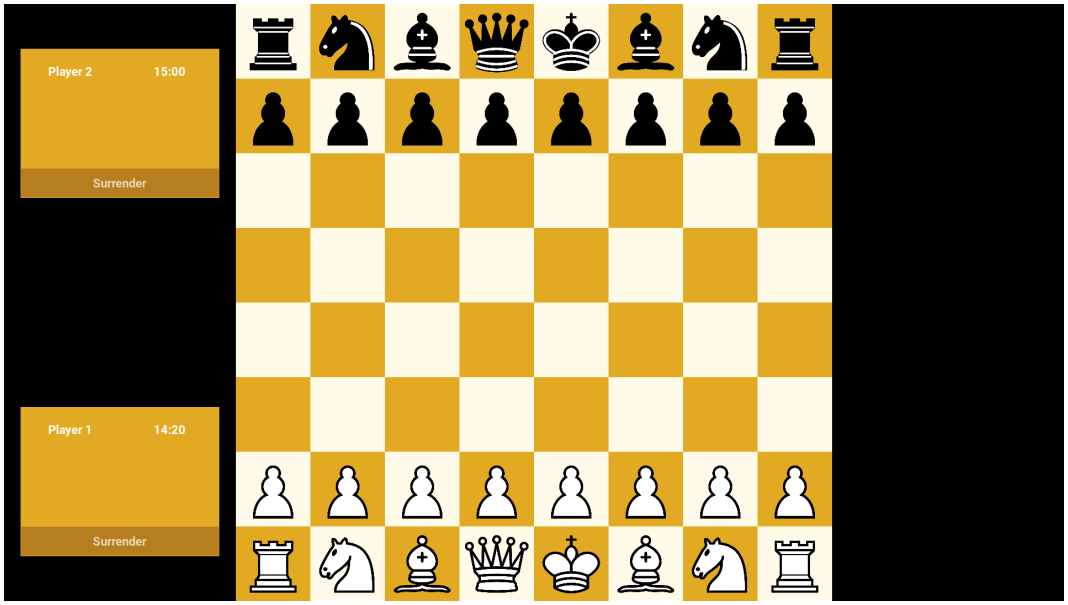
Thiếu Tài Liệu và Hỗ Trợ: Mặc dù có cộng đồng hỗ trợ, tài liệu và các tài nguyên học tập cho Kivy vẫn chưa phong phú như một số thư viện và công cụ phát triển khác.

Kích Thước Ứng Dụng Lớn: Ứng dụng phát triển bằng Kivy có thể có kích thước lớn hơn so với các ứng dụng gốc do sự phụ thuộc vào nhiều thư viện và lớp trung gian.

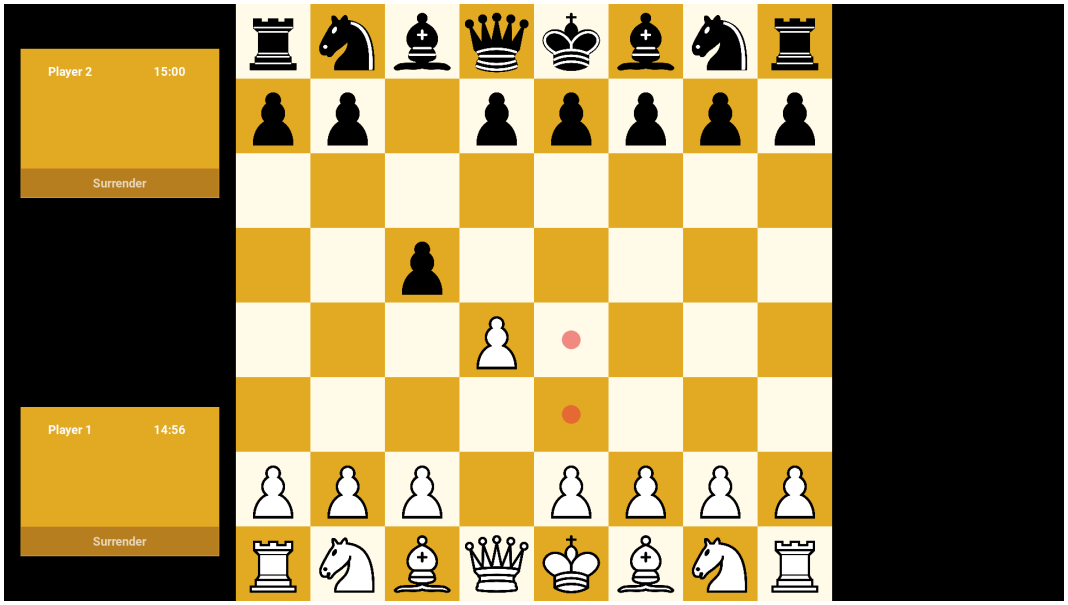
2.2 Thiết kế ứng dụng

2.2.1 Thiết kế giao diện

- Giao diện bàn cờ



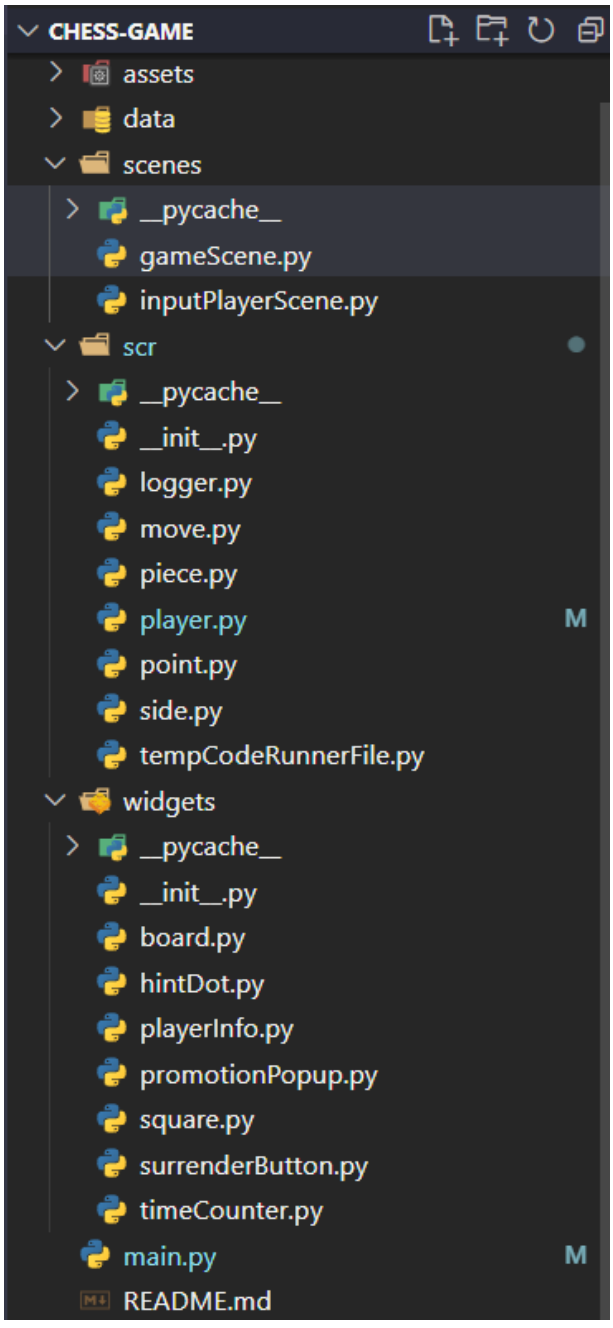
- Giao diện trận đấu đang diễn ra



- Giao diện chiến thắng

2.2.2 Thiết kế xử lý

Cấu trúc của dự án:



gameScene.py

```
from scr.side import Side
from scr.logger import Logger

from kivy.uix.floatlayout import FloatLayout
from kivy.uix.label import Label
from kivy.uix.popup import Popup

from widgets.board import Board
from widgets.playerInfo import PlayerInfo

class gameScene(FloatLayout):
    def __init__(self, fullscreen_width, fullscreen_height, player_1,
        player_2,**kwargs):
        super(gameScene, self).__init__(**kwargs)
        self.counter = 0
        self.fullscreen_width = fullscreen_width
        self.fullscreen_height = fullscreen_height

        self.player_1 = player_1
        self.player_2 = player_2

        self.player_panel_black = PlayerInfo(player=self.player_2,
            background_color='#e2aa23', size_hint=(None, None),
            size=(((fullscreen_width - fullscreen_height) / 2) - (fullscreen_width
            / 32), fullscreen_height / 4), pos_hint={'center_x':
            (((fullscreen_width - fullscreen_height) / 2) / 2) / fullscreen_width,
            'center_y': 0.8})
        self.player_panel_white = PlayerInfo(player=self.player_1,
            background_color='#e2aa23', size_hint=(None, None),
            size=(((fullscreen_width - fullscreen_height) / 2) - (fullscreen_width
            / 32), fullscreen_height / 4), pos_hint={'center_x':
            (((fullscreen_width - fullscreen_height) / 2) / 2) / fullscreen_width,
            'center_y': 0.2})

        self.board_panel = Board(player_panel_black=self.player_panel_black,
                                 player_panel_white=self.player_panel_white,
                                 player_1=self.player_1,
                                 player_2=self.player_2,
                                 square_size=fullscreen_height / 8,
                                 rows=8, cols=8, padding=0, spacing=0,
                                 size_hint=(None, None), size=(fullscreen_height,
                                     fullscreen_height),
                                 pos_hint={'center_x': 0.5, 'center_y': 0.5})

        self.add_widget(self.board_panel)
        self.add_widget(self.player_panel_black)
        self.add_widget(self.player_panel_white)


    def end_game(self, winner):
        Popup(title='Alert', content=Label(text=f"{winner.username} win!"),
```

```
            size_hint=(None, None), size=(300, 200)).open()
        Logger(winner=winner, player_1=self.player_1, player_2=self.player_2,
            log=self.board_panel.log).writeLog()
        self.player_panel_black.timeCounter.stop_counter()
        self.player_panel_white.timeCounter.stop_counter()
        self.restart_game()


    def restart_game(self):
        self.player_1.capturedList = []
        self.player_2.capturedList = []

        self.counter = self.counter + 1

        self.clear_widgets()
        self.switchSide()


    def switchSide(self):
        if self.counter % 2 == 0:
            self.player_panel_black = PlayerInfo(player=self.player_2,
                background_color='#e2aa23', size_hint=(None, None),
                size=(((self.fullscreen_width - self.fullscreen_height) / 2) -
                (self.fullscreen_width / 32), self.fullscreen_height / 4),
                pos_hint={'center_x': (((self.fullscreen_width -
                self.fullscreen_height) / 2) / 2) / self.fullscreen_width,
                'center_y': 0.8})
            self.player_panel_white = PlayerInfo(player=self.player_1,
                background_color='#e2aa23', size_hint=(None, None),
                size=(((self.fullscreen_width - self.fullscreen_height) / 2) -
                (self.fullscreen_width / 32), self.fullscreen_height / 4),
                pos_hint={'center_x': (((self.fullscreen_width -
                self.fullscreen_height) / 2) / 2) / self.fullscreen_width,
                'center_y': 0.2})
            self.player_1.side = Side.WHITE
            self.player_2.side = Side.BLACK
        else:
            self.player_panel_black = PlayerInfo(player=self.player_1,
                background_color='#e2aa23', size_hint=(None, None),
                size=(((self.fullscreen_width - self.fullscreen_height) / 2) -
                (self.fullscreen_width / 32), self.fullscreen_height / 4),
                pos_hint={'center_x': (((self.fullscreen_width -
                self.fullscreen_height) / 2) / 2) / self.fullscreen_width,
                'center_y': 0.8})
            self.player_panel_white = PlayerInfo(player=self.player_2,
                background_color='#e2aa23', size_hint=(None, None),
                size=(((self.fullscreen_width - self.fullscreen_height) / 2) -
                (self.fullscreen_width / 32), self.fullscreen_height / 4),
                pos_hint={'center_x': (((self.fullscreen_width -
                self.fullscreen_height) / 2) / 2) / self.fullscreen_width,
                'center_y': 0.2})
            self.player_1.side = Side.BLACK
```

```
66              self.player_2.side = Side.WHITE
67
68          self.board_panel = Board(player_panel_black=self.player_panel_black,
69                                   player_panel_white=self.player_panel_white,
70                                   player_1=self.player_1,
71                                   player_2=self.player_2,
72                                   square_size=self.fullscreen_height / 8,
73                                   rows=8, cols=8, padding=0, spacing=0,
74                                   size_hint=(None, None), size=(self.fullscreen_height,
                                         self.fullscreen_height),
75                                   pos_hint={'center_x': 0.5, 'center_y': 0.5})
76
77          self.add_widget(self.board_panel)
78          self.add_widget(self.player_panel_black)
79          self.add_widget(self.player_panel_white)
```

- Mô tả

  Lớp gameScene quản lý giao diện trò chơi, xử lý bố trí của bảng và các bảng thông tin người chơi. Nó cung cấp chức năng để kết thúc trò chơi, ghi lại kết quả, đặt lại trạng thái trò chơi, và chuyển đổi bên của người chơi.

  from kivy.uix.floatlayout import FloatLayout

  from kivy.uix.label import Label

  from kivy.uix.popup import Popup

  FloatLayout: Một lớp bố trí từ Kivy cho phép định vị các widget con tương đối so với kích thước của bố trí.

  Label: Một widget từ Kivy để hiển thị văn bản.

  Popup: Một widget từ Kivy để hiển thị các hộp thoại bật lên.

  from widgets.board import Board

  from widgets.playerInfo import PlayerInfo

  Board: Một widget tùy chỉnh cho bảng trò chơi, xử lý khu vực tương tác chính của trò chơi.

  PlayerInfo: Một widget tùy chỉnh hiển thị thông tin về người chơi, chẳng hạn như tên, điểm số, hoặc các mảnh đã bị bắt.

  from scr.side import Side

from scr.logger import Logger

Side: Là một lớp mô-đun chứa các hằng số hoặc logic liên quan đến các bên (ví dụ, trắng hoặc đen) của một trò chơi, có thể là cờ vua hoặc một trò chơi bảng tương tự.

Logger: Lớp này xử lý việc ghi lại các sự kiện trò chơi, như ghi nhận kết quả của một ván đấu.

main.py

```
from kivy.app import App
from kivy.core.window import Window
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.image import Image

from scr.player import Player
from scr.side import Side

from scenes.gameScene import gameScene


class Main(App):
    def build(self):
        fullscreen_height = 700
        fullscreen_width = 700

        player_1=Player(Side.WHITE, "Player 1")
        player_2=Player(Side.BLACK, "Player 2")

        game_scene = gameScene(fullscreen_width=fullscreen_width,
            fullscreen_height=fullscreen_height, player_1=player_1,
            player_2=player_2)

        root = FloatLayout()
        root.add_widget(game_scene)
        return root

if __name__ == '__main__':
    Main().run()
```

- Mô tả
  Đoạn mã này thiết lập một ứng dụng Kivy cơ bản:
  Tạo một cửa sổ với kích thước cố định.
  Khởi tạo hai đối tượng người chơi với các màu khác nhau.
  Tạo một cảnh trò chơi (gameScene) với các người chơi đã tạo.

Thêm cảnh trò chơi vào bố trí chính (FloatLayout).
Khởi động ứng dụng.

```
from kivy.app import App
from kivy.core.window import Window
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.image import Image
from scr.player import Player
from scr.side import Side
from scenes.gameScene import gameScene
```

App từ kivy.app: Lớp cơ bản để tạo ứng dụng Kivy.
Window từ kivy.core.window: Quản lý cửa sổ ứng dụng.
FloatLayout từ kivy.uix.floatlayout: Bố trí các widget con một cách tự do theo tỷ lệ phần trăm.
Image từ kivy.uix.image: Widget hiển thị hình ảnh.
Player từ scr.player: Lớp đại diện cho người chơi.
Side từ scr.side: Lớp hoặc mô-đun chứa các hằng số hoặc logic liên quan đến các bên (trắng hoặc đen).
gameScene từ scenes.gameScene: Lớp tùy chỉnh quản lý cảnh trò chơi.

2.3 Cài đặt

Mở terminal và chạy các lệnh sau:

- *Tải các thư viện cần thiết*
  pip install python
  pip install kivy

- *Tải source code*
  git clone https://github.com/minewaku/chess-game.git

- *Thực thi ứng dụng*
  python main.py

Cờ vua là một trò chơi chiến lược hai người chơi được chơi trên lưới 8x8 được gọi là bàn cờ. Mỗi người chơi bắt đầu với 16 quân cờ: một vị vua, một nữ hoàng, hai tân binh, hai hiệp sĩ, hai giám mục và tám con tốt. Mục tiêu của trò chơi là kiểm tra vua của đối thủ của bạn, có nghĩa là đặt nhà vua vào vị trí kiểm soát (bị đe dọa bắt giữ) và không có động thái hợp pháp để trốn thoát.

Chiếu tướng không hoạt động, người chơi phải đầu hàng thông qua nút đầu hàng để kết thúc trận đấu!

3 TỔNG KẾT

Dự án làm game cờ vua bằng Python và thư viện Kivy đã mang đến một trải nghiệm lập trình thú vị và đầy thách thức, đồng thời tạo ra một sản phẩm có giá trị thực tiễn cao. Chúng tôi đã thành công trong việc phát triển một ứng dụng cờ vua đa nền tảng, có thể chạy mượt mà trên hệ điều hành Linux. Sử dụng Kivy, chúng tôi xây dựng được giao diện người dùng thân thiện, hỗ trợ cảm ứng và có đồ họa đẹp mắt, tạo ra một trải nghiệm chơi cờ vua trực quan và hấp dẫn.

Ứng dụng không chỉ cung cấp các chức năng cơ bản của cờ vua như di chuyển quân cờ và kiểm tra luật chơi hợp lệ mà còn bao gồm các tính năng nâng cao như lưu và tải lại ván cờ, chế độ chơi hai người.

Quá trình phát triển dự án đã giúp chúng tôi học hỏi và cải thiện nhiều kỹ năng lập trình, từ việc quản lý logic trò chơi, xử lý giao diện người dùng đến tối ưu hóa hiệu năng. Chúng tôi đã áp dụng phương pháp Agile trong quản lý dự án, đảm bảo liên tục cải tiến và phản hồi nhanh chóng đối với các vấn đề phát sinh.

Dự án này cũng đã khẳng định sức mạnh và tính linh hoạt của Python cùng thư viện Kivy trong việc phát triển ứng dụng đa nền tảng. Đồng thời, chúng tôi hy vọng sản phẩm này sẽ là nguồn cảm hứng và tài liệu học tập hữu ích cho cộng đồng lập trình viên, đặc biệt là những người quan tâm đến việc phát triển ứng dụng trò chơi.

Nhìn lại toàn bộ quá trình, chúng tôi cảm thấy hài lòng với những gì đã đạt được. Từ ý tưởng ban đầu đến sản phẩm hoàn thiện, chúng tôi đã vượt qua nhiều thử thách kỹ thuật và mang lại một ứng dụng chất lượng. Chúng tôi tin rằng, với sự phát triển và hoàn thiện không ngừng, ứng dụng cờ vua này sẽ còn được nâng cấp và cải tiến hơn nữa trong tương lai.

Tài liệu

[1] *Bài giảng Phát triển phần mềm mã nguồn mở*, Từ Lãng Phiêu, Trường Đại học Sài Gòn.

[2] *Bài giảng Ngôn ngữ lập trình Python*, Trịnh Tấn Đạt, Trường Đại học Sài Gòn.

[3] *Introduction to Python Programming*, UDAYAN DAS, AUBREY LAWSON, WILEY CHRIS MAYFIELD, NARGES NOROUZI, Rice University, 2024