ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

*

BÀI TẬP LỚN

MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
(Mã học phần: IT3103)

Đề tài 06:

TRÒ CHƠI DÂN GIAN Ô ĂN QUAN

Nhóm: 06
Mã lớp học: 151965
Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang

MSSV	Họ và tên	Lớp	Tỷ lệ %
20215028	Nguyễn Thành Đạt	Việt Nhật 01 - K66	40%
20225804	Lê Thành Đạt	Việt Nhật 03 - K67	20%
20225605	Nguyễn Tuấn Đạt	Việt Nhật 04 - K67	20%
20204947	Nguyễn Thành Đạt	Việt Nhật 03 - K65	20%
20215025	Lê Anh Đạt	Việt Nhật 03 - K66	0%

1. Phân công nhiệm vụ

1.1. Nguyễn Thành Đạt - 20215028

1.1.1. Phân tích và thiết kế
1.1.2. Xây dựng Model
1.1.3. Vẽ biểu đồ Use Case và biểu đồ Class
1.1.4. Triển khai GameBoardController.java
1.1.5. Triển khai EndGameController.java

1.2. Lê Thành Đạt - 20225804

1.2.1. Triển khai MainController.java
1.2.2. Triển khai StartGameController.java
1.2.3. Triển khai HelpController.java
1.2.4. Triển khai ExitController.java
1.2.5. Viết báo cáo

1.3. Nguyễn Tuấn Đạt - 20225605

1.3.1. Xây dựng màn hình Main.fxml
1.3.2. Xây dựng màn hình GameBoard.fxml
1.3.3. Viết báo cáo

1.4. Nguyễn Thành Đạt - 20204947

1.4.1. Xây dựng màn hình Help.fxml
1.4.2. Xây dựng màn hình StartGame.fxml
1.4.3. Xây dựng màn hình EndGame.fxml
1.4.4. Xây dựng màn hình Exit.fxm
1.4.5. Làm Slide

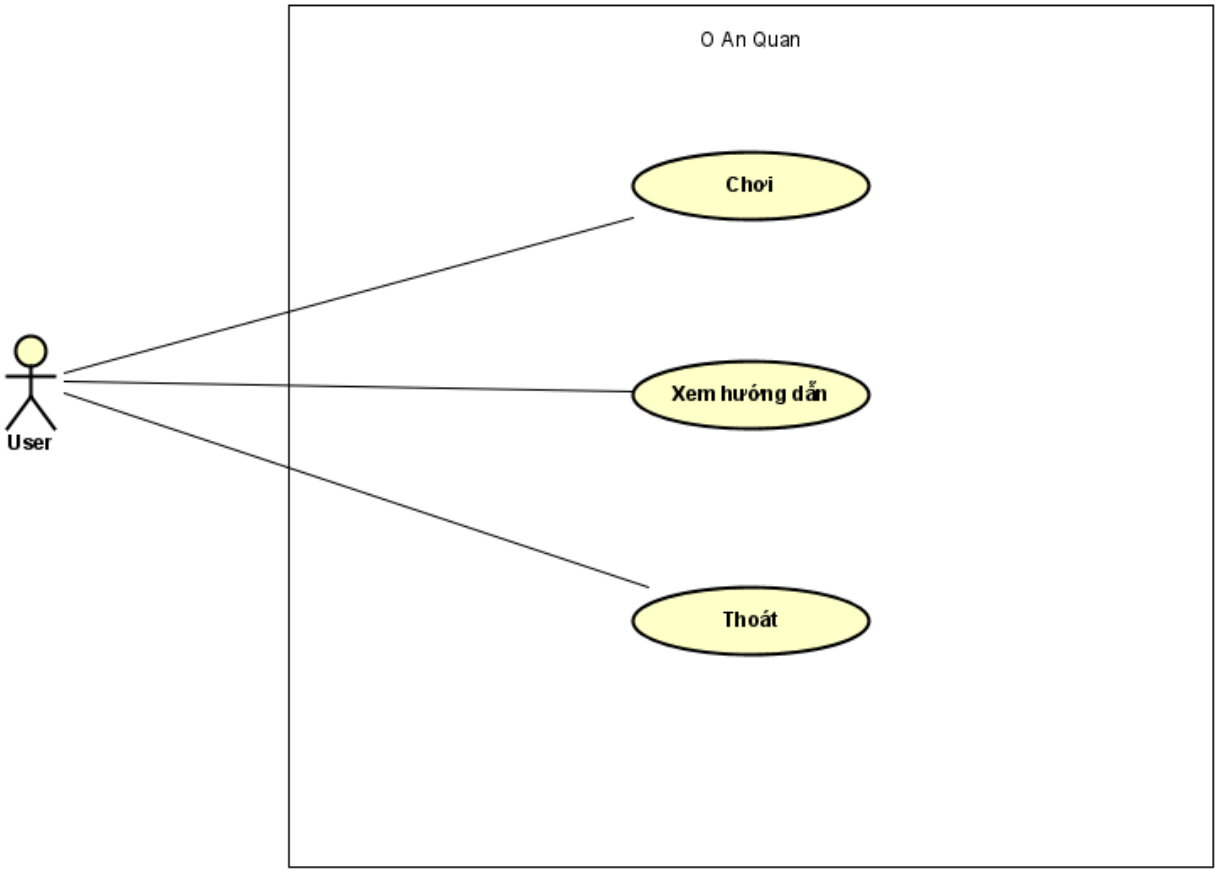
2. Mô tả bài toán

Trò chơi Ô Ăn Quan là một trò chơi dân gian của Việt Nam, thường dành cho 2 người chơi, được chơi trên một bàn chơi bao gồm 10 ô hình vuông (ô dân) và 2 ô hình bán nguyệt (ô quan), mỗi ô đều chứa các viên đá - ô dân chứa 5 viên đá và ô quan chứa 1 viên đá lớn. Trò chơi này không chỉ mang tính giải trí mà còn giúp người chơi rèn luyện tư duy chiến thuật và khả năng tính toán.

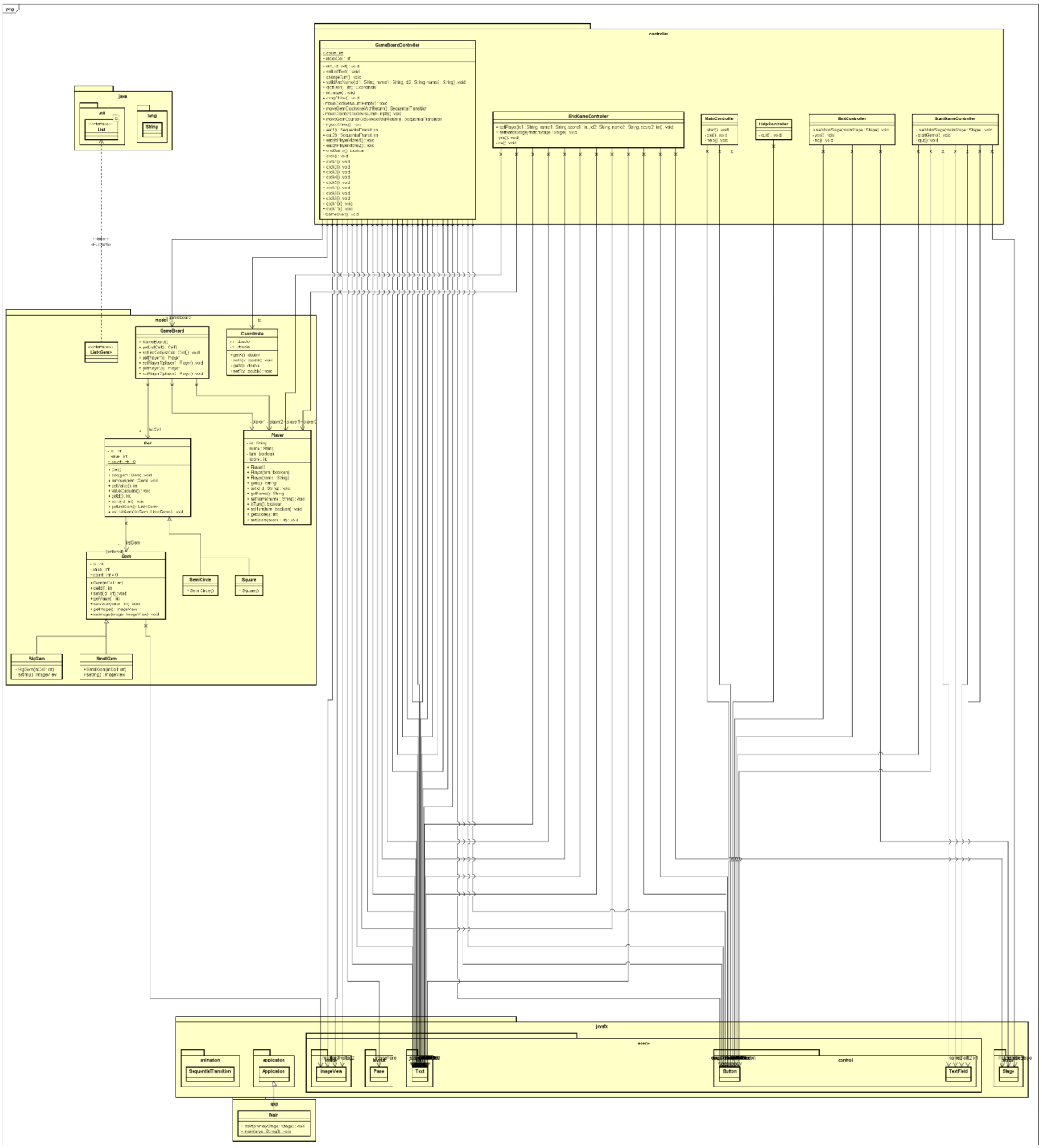
Dự án nhằm phát triển một ứng dụng để triển khai trò chơi ô ăn quan với các chức năng cơ bản như di chuyển đá, thu đá,...

Đồng thời, dự án sẽ cung cấp một ứng dụng có giao diện thân thiện, chức năng dễ sử dụng và có hướng dẫn đầy đủ cho người mới chơi.

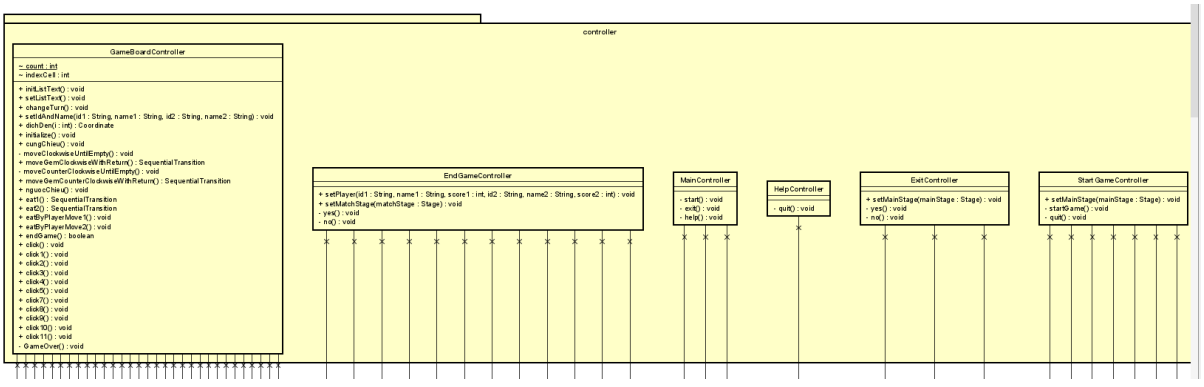
3. Chức năng chính
   3.1. Chơi game
   3.2. Xem hướng dẫn
   3.3. Thoát
4. Thiết kế
   4.1. Sơ đồ Use Case



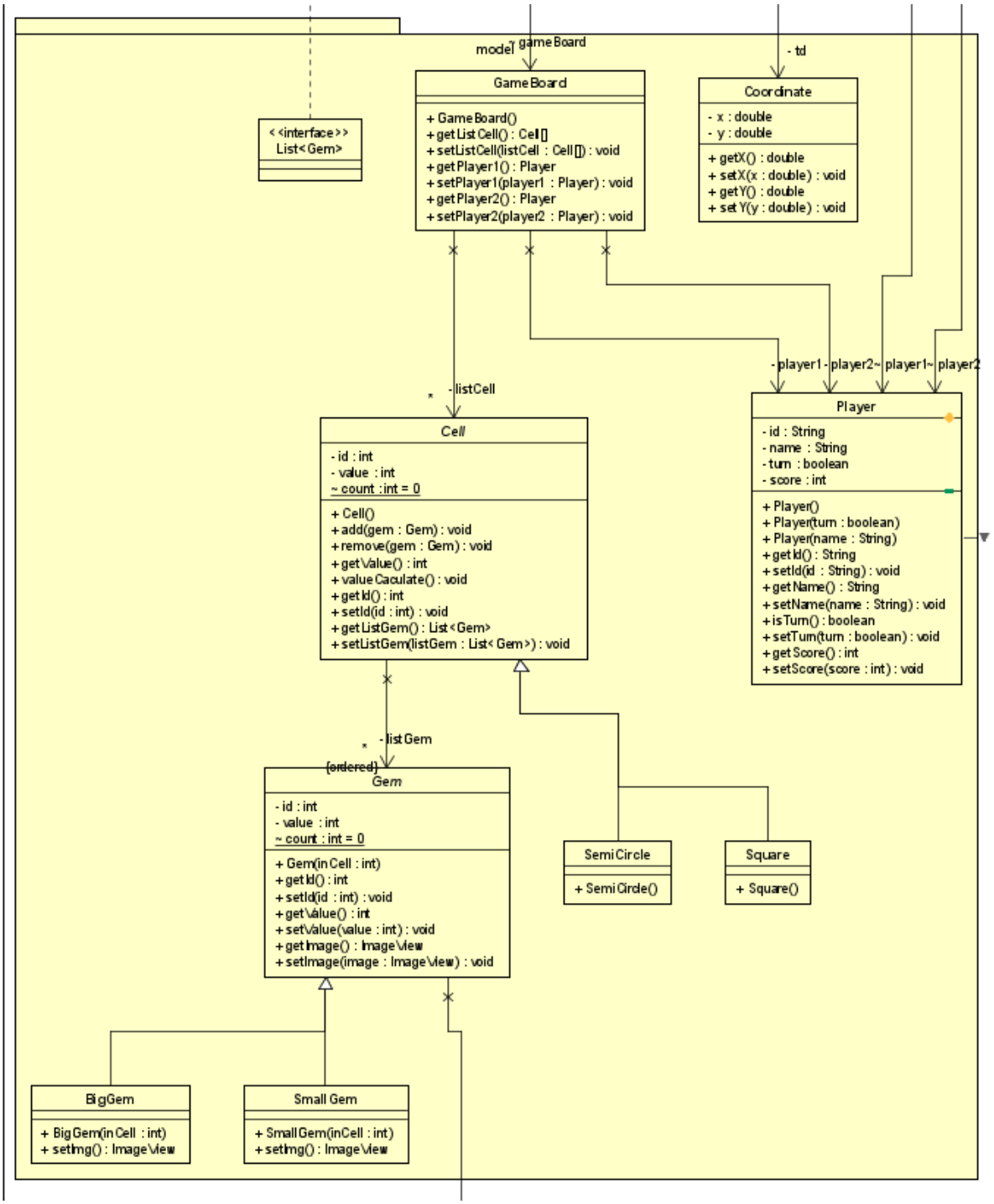
   4.2. Sơ đồ Class
      4.2.1. Sơ đồ chung:



4.2.2. Package controller



4.2.3. Package model



5. Kỹ thuật lập trình OOP

5.1. Đóng gói (Encapsulation)

5.1.1. Tóm tắt lý thuyết:

- Là cơ chế che dấu thông tin và chi tiết cài đặt của đối tượng.
- Chỉ cung cấp một số phương thức public để truy cập và thay đổi dữ liệu bên trong đối tượng.

5.1.2. Áp dụng:

- Lớp Coordinate:

```
public class Coordinate {
    private double x;
    private double y;

    public double getX() {
        return x;
    }
    public void setX(double x) {
        this.x = x;
    }
    public double getY() {
        return y;
    }
    public void setY(double y) {
        this.y = y;
    }
}
```

- Trong lớp GameBoardController:

```
private Coordinate td = new Coordinate();

public Coordinate dichDen(int i) {

    switch (i) {
        case 0:
            td.setX(40+Math.random()*40);
            td.setY(30+Math.random()*120);
            break;
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:
            td.setX(10+i*100+Math.random()*60);
            td.setY(10+Math.random()*60);
            break;
        case 6:
            td.setX(600+Math.random()*40);
            td.setY(30+Math.random()*120);
            break;
        case 7:
        case 8:
        case 9:
        case 10:
        case 11:
            td.setX(10+(12-i)*100+Math.random()*60);
            td.setY(110+Math.random()*60);
            break;
        case 13:
            td.setX(100+Math.random()*60);
            td.setY(300+Math.random()*60);
            break;
        case 12:
            td.setX(100+Math.random()*60);
            td.setY(-200+Math.random()*60);
            break;
    }
    return td;
}
```

- Tương tự với một số lớp khác cũng áp dụng nguyên

tắc Đóng gói.

5.2. Trừu tượng hóa (Abstraction)

5.2.1. Tóm tắt lý thuyết:

- Là quá trình che giấu chi tiết cài đặt và chỉ j những chức năng cần thiết cho người dùng.
- Thực hiện thông qua: Lớp trừu tượng (abstract class) và Giao diện (interface).

5.2.2. Áp dụng:

- Lớp Gem:

```
public abstract class Gem {
    private int id;
    private int value;
    private ImageView image;
    static int count = 0;

    public Gem(int inCell) {
        count++;
        this.id = count;
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getValue() {
        return value;
    }
    public void setValue(int value) {
        this.value = value;
    }

    public ImageView getImage() {
        return image;
    }
    public void setImage(ImageView image) {
        this.image = image;
    }
}
```

- Tương tự với lớp Cell.

5.3. Kế thừa (Inheritance)

5.3.1. Tóm tắt lý thuyết:

- Là cơ chế cho phép lớp con kế thừa thuộc tính và phương thức của lớp cha.
- Giúp tái sử dụng mã nguồn và mở rộng tính năng của hệ thống

5.3.2. Áp dụng:

- Lớp SmallGem và BigGem kế thừa lớp Gem:

```java
public class BigGem extends Gem{

    public BigGem(int inCell) {
        super(inCell);
        this.setValue(10);
        this.setImage(setImg());
    }

    public ImageView setImg() {
        File file = new File("src/testUI/big-gem.png");
        ImageView img = new ImageView(file.toURI().toString());
        img.setFitWidth(50);
        img.setFitHeight(50);
        return img;
    }
}
```

```java
public class SmallGem extends Gem{

    public SmallGem(int inCell) {
        super(inCell);
        this.setValue(1);
        this.setImage(setImg());
    }

    public ImageView setImg() {
        File file = new File("src/testUI/small-gem.png");
        ImageView img = new ImageView(file.toURI().toString());
        img.setFitWidth(20);
        img.setFitHeight(20);
        return img;
    }
}
```

- Tương tự: Lớp Square và lớp SemiCircle kế thừa từ lớp Cell

5.4. Đa hình (Polymorphism)

5.4.1. Tóm tắt lý thuyết:

- Là khả năng một đối tượng có thể hiện hành vi khác nhau h khi được gọi thông qua các phương thức giống nhau.
- Gồm: Method overloading và method overriding.

5.4.2. Áp dụng:

- Method overloading:
  - Trong lớp Player, overloading các phương thức khởi tạo Player:

```
public Player() {
    this.score = 0;
}

public Player(boolean turn) {
    this.score = 0;
    this.turn = turn;
}

public Player(String name) {
    this.name = name;
    this.score = 0;
}
```

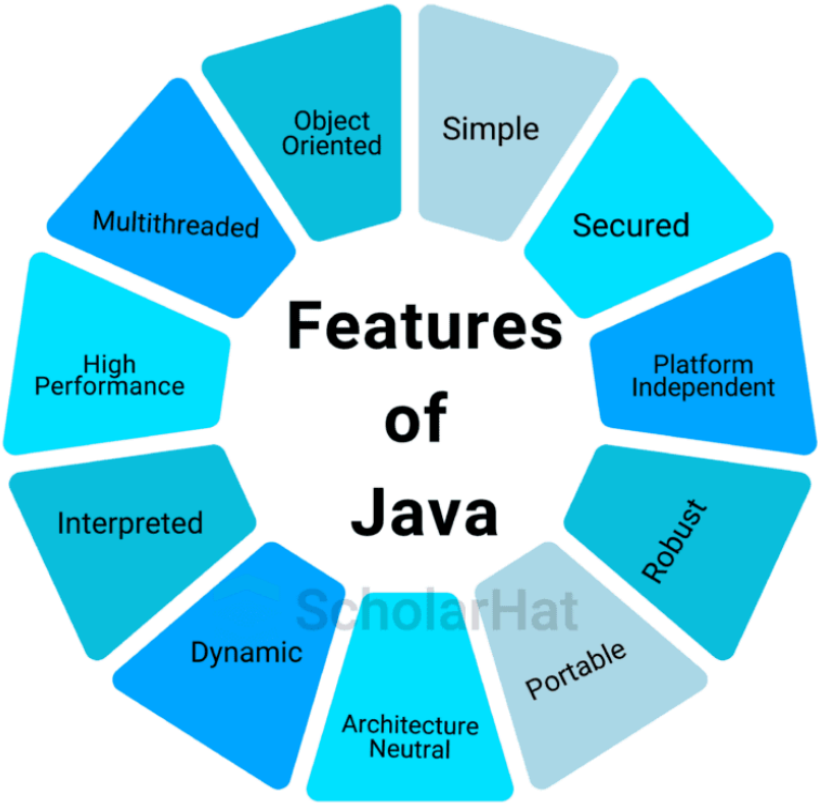
- Method overriding:
    - Trong lớp Main, overriding phương thức start của lớp java.application.Application.

```
@Override
public void start(Stage primaryStage) throws Exception {
    Parent root = FXMLLoader.load(getClass().getResource("Main.fxml")); // Đảm bảo tệp FXML có tên đúng
    Scene scene = new Scene(root);

    primaryStage.setResizable(false);

    primaryStage.setOnCloseRequest(event -> {
        // Hủy sự kiện đóng cửa sổ
        event.consume();
        System.out.println("Nút Close đã bị chặn!");
    });

    primaryStage.setTitle("Main");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```

6. Công nghệ sử dụng

6.1. Ngôn ngữ lập trình Java

Java là một trong những ngôn ngữ lập trình hướng đối tượng, giúp dễ dàng tái sử dụng và mở rộng code. Nhờ sử dụng bytecode và JVM, Java bảo bảo tốc độ và hiệu suất ổn định. Java cung cấp thư viện đa dạng hỗ trợ nhiều lĩnh vực từ xử lý dữ liệu, mạng đến giao diện đồ họa. Ngôn ngữ Java được sử dụng phổ biến trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.



6.2. Giao diện với JavaFX

JavaFX là một nền tảng giao diện người dùng đồ họa (GUI) được phát triển bởi Oracle, được thiết kế để tạo ra các ứng dụng GUI giàu chức năng và đa nền tảng. Nó cung cấp một bộ toàn diện các thành phần giao diện người dùng, hiệu ứng đồ họa và khả năng tích hợp chặt chẽ với các công cụ phát triển phổ biến như:

- **FXML:** Là một định dạng XML cho phép mô tả giao diện người dùng. Thay vì viết mã giao diện bằng Java thuần, FXML cho phép nhà phát triển tách riêng giao diện và logic xử lý. Việc này giúp code trở nên sáng sủa và dễ bảo trì hơn.
- **Scene Graph**: Đây là công cụ đồ họa tích hợp trong JavaFX, cho phép tạo và thao tác các hiệu ứng đồ họa phức tạp như xoay, thu phóng và hoạt ảnh
- **CSS (Cascading Style Sheets)**: Tương tự như HTML, JavaFX cũng hỗ trợ CSS để tạo ra các giao diện người dùng đẹp mắt và dễ dàng tùy chỉnh.
- **Thành phần giao diện người dùng**: JavaFX cung cấp một bộ phong phú các thành phần giao diện người dùng, chẳng hạn như nút, trường văn bản, danh sách và bảng. Các thành phần này có thể được tùy chỉnh dễ dàng để đáp ứng các nhu cầu cụ thể của ứng dụng.

6.3. Mô hình MVC (Model - View - Controller)

MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính, chia thành *ba phần* được *kết nối với nhau* và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Các thành phần trong MVC:

- **Model:** Có nhiệm vụ thao tác với database, chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu. Controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View
- **View:** Là giao diện người dùng (User Interface), Chứa các thành phần tương tác với người dùng như menu, button, image, text,... Nơi nhận dữ liệu từ Controller và hiển thị
- **Controller:** Điều khiển và kết nối giữa Model và View. Đảm nhận vai trò tiếp nhận yêu cầu từ người dùng, thông qua Model để lấy dữ liệu sau đó thông qua View để hiển thị cho người dùng.

7. Hình ảnh sản phẩm

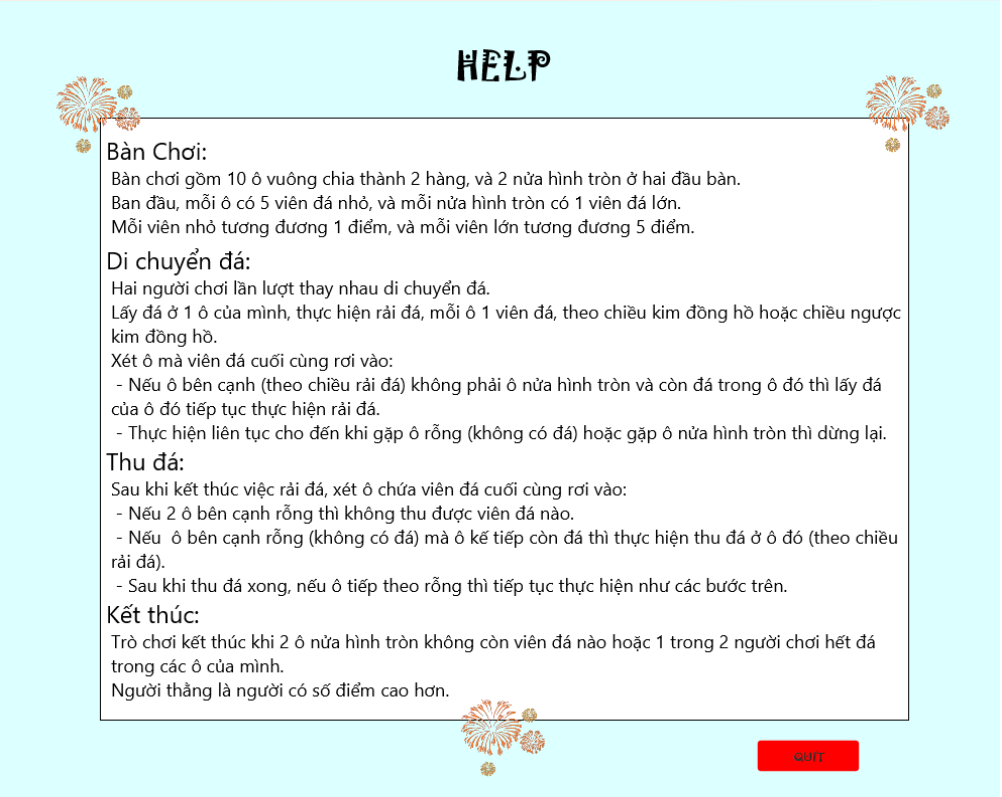
7.1. Màn hình Chính

- Màn hình Chính cung cấp các chức năng chính của Game:
  - Chơi game:
    - Nhấn vào nút “Start” để chơi game.
  - Hướng dẫn:
    - Nhấn vào nút “HelHelp” để đọc hướng dẫn.
  - Thoát:
    - Nhấn vào nút “Exit” để thoát chương trình.

7.2. Màn hình Start Game

- Màn hình Start Game cung cấp cho người dùng thiết lập thông tin cho người chơi:
    - Text Field ID: Cho phép người dùng nhập ID của 2 người chơi.
    - Text Field Name: Cho phép người dùng nhập Tên của 2 người chơi.
    - Nhấn vào nút “Start” để bắt đầu chơi game.
    - Nhấn vào nút “Quit” để quay lại màn hình Chính

7.3. Màn hình Help



- Màn hình Help cung cấp thông tin về luật chơi.
    - Nhấn vào nút “Quit” để quay lại màn hình Chính.

7.4. Màn hình GameBoard

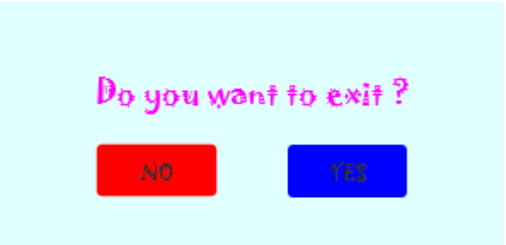
- Màn hình GameBoard cung cấp màn hình chơi game cho người chơi:
    - Thông tin hai người chơi:
        - Ô thu đá về: Ô vuông cạnh mỗi ảnh đại diện của mỗi người chơi, cho biết số đá mà người chơi đó đã thu được.
        - Ảnh đại diện: Hình ảnh chính giữa thông tin của mỗi người chơi.
        - ID: Cho biết ID của người chơi - đã được nhập ở màn hình Start Game.
        - Tên: Cho biết Tên của người chơi - đã được nhập ở màn hình Start Game.
        - Điểm: Cho biết điểm hiện tại của người chơi có được trong trận đấu.
        - Ô đang chọn: Cho biết ô đang chọn để di chuyển đá của người chơi (nếu là rỗng thì là người chơi đang không chọn hoặc là chưa tới lượt người chơi đó).
        - Cờ: Cho biết lượt di chuyển đá là của người chơi nào.
    - Bàn chơi:
        - Gồm 12 ô: 10 ô hình vuông và 2 ô hình bán nguyệt. Cạnh mỗi ô có các số để thể hiện số điểm mà ô đó đang chứa.

- Nếu tới lượt chơi, người chơi có thể chọn 1 trong các ô đá của mình để di chuyển đá (người chơi 1 chỉ có thể chọn các ô từ 1 - 5; người chơi 2 chỉ có thể chọn các ô từ 7 - 11).
- Khi chọn 1 ô, màn chơi sẽ hiện 2 nút "<" và ">" để người chơi có thể chọn hướng di chuyển đá.

7.5. Màn hình End Game

- Màn hình End Game cung cấp các thông tin sau khi kết thúc trận đấu:
  - Bao gồm thông tin 2 người chơi (ID, Tên, Điểm) và người chơi đã giành chiến thắng.
  - Người chơi có thể tiếp tục chơi nếu nhấn vào nút "Yes" hoặc quay lại màn hình Chính nếu bấm vào nút "No".

7.6. Màn hình Exit



- Màn hình Exit giúp người chơi xác nhận "có muốn thoát chương trình không?":
  - Nhấn vào nút "Yes" để thoát.
  - Nhấn vào nút "No" để quay lại màn hình Chính.

8. Demo

Link demo: link tại đây.

9. Định hướng phát triển:

- Kết nối với database để lưu thông tin người chơi, lịch sử trận đấu…
- Cho phép chơi trên các máy khác nhau
- Thêm một số chức năng khác như:
    - Cài đặt: thêm cài đặt như cài đặt tốc độ di chuyển đá, giá trị của từng viên đá…
    - Ghi lại lịch sử: lưu lịch sử trận đấu, người dùng có thể tìm kiếm được trận đã chơi, mở rộng ra là xếp hạng người chơi.