Đại học Bách Khoa Hà Nội

Trường Công nghệ thông tin và truyền thông

Trò chơi Ô ăn Quan

Thành viên nhóm 32

Nguyễn Ngoc Anh Thư - 20215143 : thiết kế giao diện

Hoàng Văn Thuấn - 20215140: thiết kế giao diện

Vũ Đức Thịnh - 20210816: Game play

Nguyễn Hoàng Ninh Thuận - 20215141: Game play

1. Yêu câu chi tiết

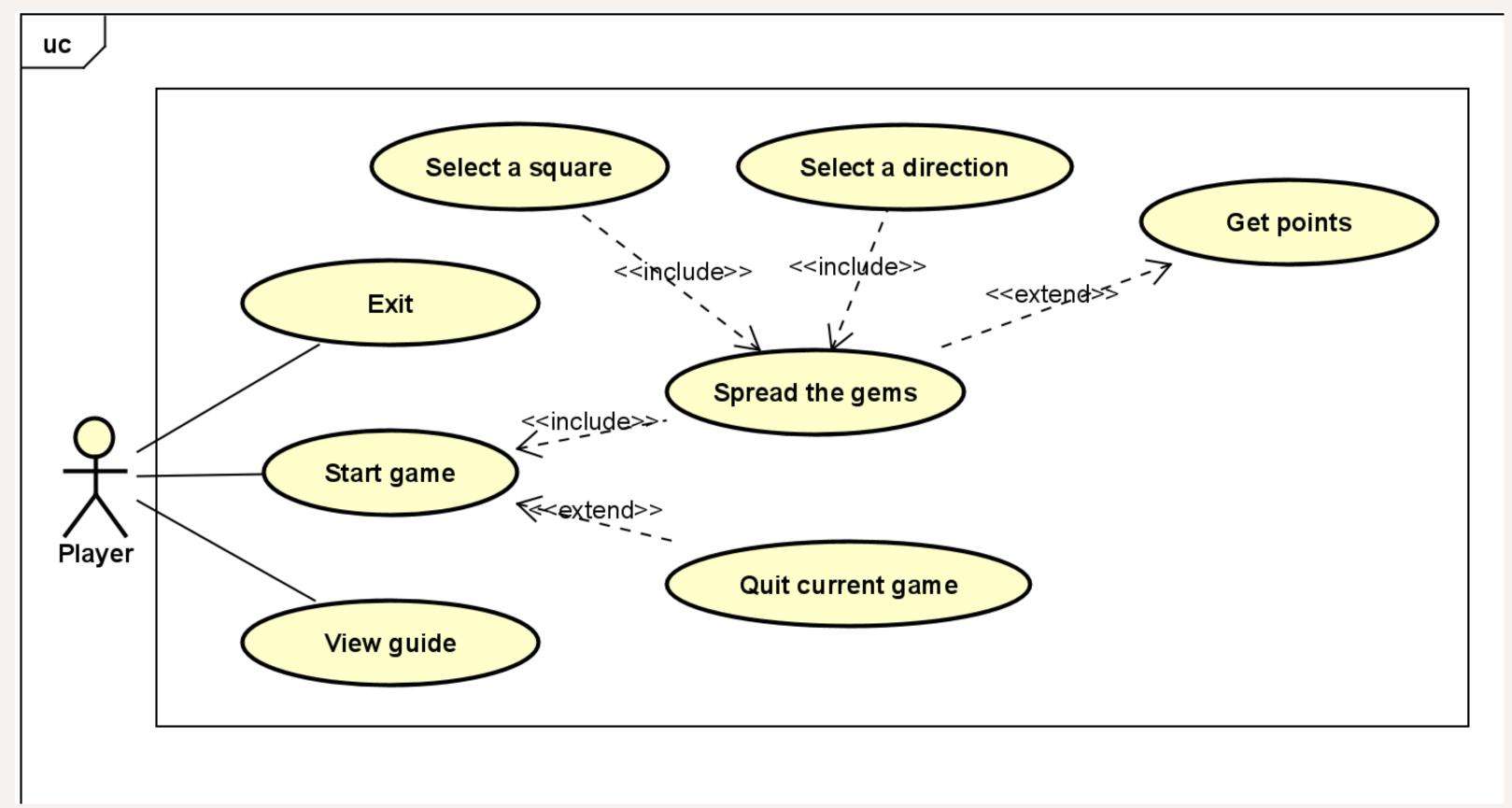
Trên màn hình chính:

- Bắt đầu: bắt đầu trò chơi. Để thuận tiện, không cần phải tạo ra các cấp độ khác nhau
- Thoát: thoát khỏi chương trình. Hãy chắc chắn hỏi người dùng nếu họ muốn thoát khỏi trò chơi
- Trợ giúp: Hiển thị hướng dẫn cho việc chơi game

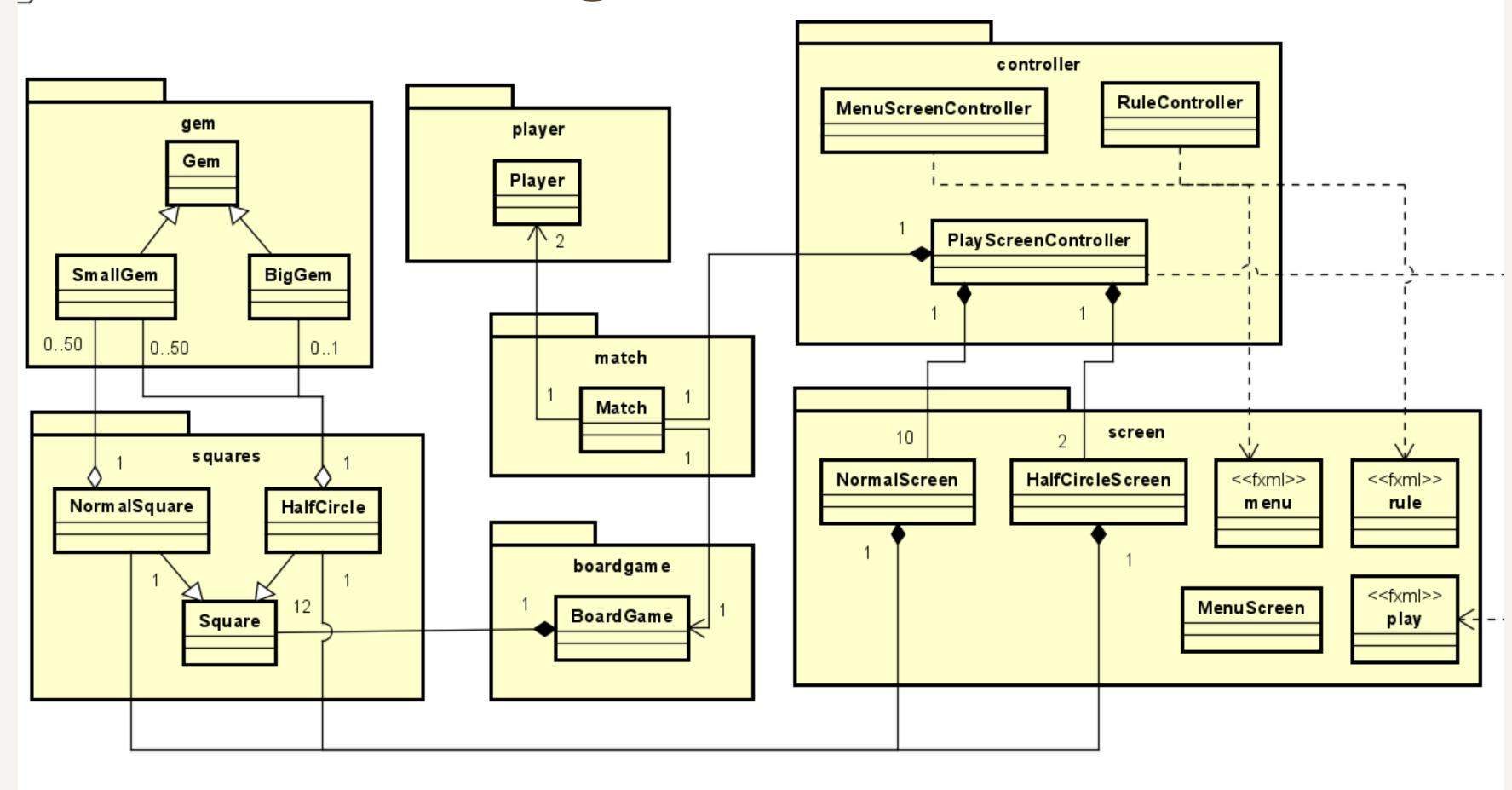
Trong trò chơi:

- Bảng game: Bảng game bao gồm 10 ô, chia thành 2 hàng và 2 nửa vòng tròn ở hai đầu của bảng. Ban đầu, mỗi ô có 5 viên ngọc nhỏ, và mỗi nửa vòng tròn có 1 viên ngọc lớn. Mỗi viên ngọc nhỏ tương đương 1 điểm, và mỗi viên ngọc lớn tương đương 5 điểm.
- Đối với mỗi lượt, ứng dụng phải rõ ràng hiển thị lượt của ai. Người chơi sẽ chọn một ô và hướng để rải các viên ngọc. Anh ta có điểm khi sau khi hoàn thành rải, có một ô trống liền sau một ô có ngọc. Điểm người chơi được cho lượt chơi đó bằng điểm số ngọc có trong ô đó.
 - Trò chơi kết thúc khi không còn ngọc trong cả hai nửa vòng tròn. Ứng dụng phải thông báo người chiến thắng và điểm của mỗi người chơi.
 - Để đơn giản, ta không cần phải xây dựng một bot để chơi với người chơi.

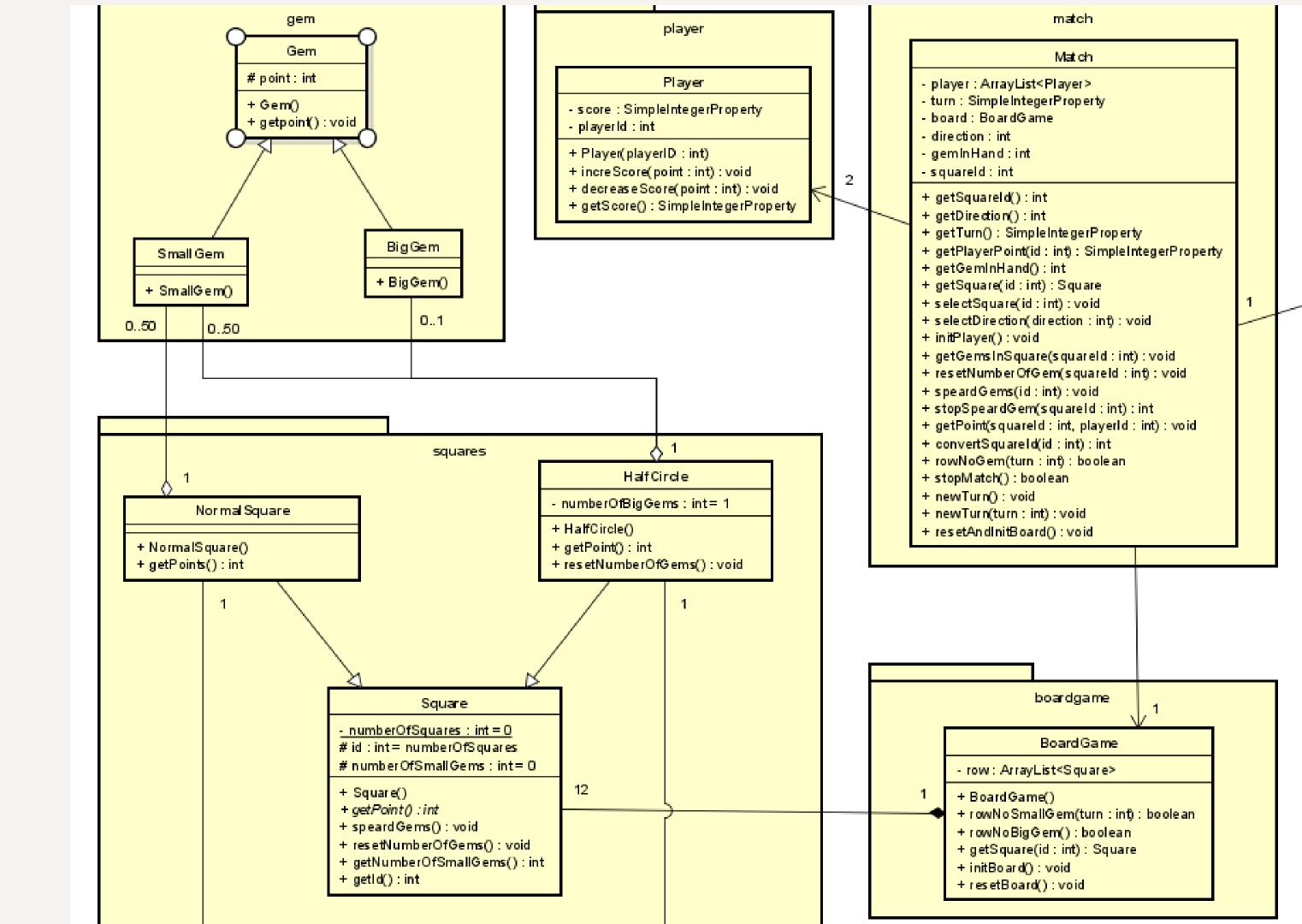
Usecase diagram



General class diagram



Detail class



Package controller

PlayScreenController

- hBox1, hBox2, boardHbox: HBox
- player1TurnLabel, player2TurnLabel, point1 Label, point2Label, turnLabel, winnerLabel, winScore1, winScore2 : Label
- hfCircleAncPane0, hfCircleAncPane6, pauseAncPane0,
 pauseAncPane1, playAncPane, resultAncPane0, resultAncPane1,
 turnAncPane: AnchorPane
- + match : Match
- row : ArrayList<AnchorPane>
- squareld : int
- SPREAD TIME : double = 0.5
- TURN APPEAR TIME : double = 1
- iRowNoGem: int
- playerld : int
- backBtn, exitBtn: Button
- + resetToDefaultSquare(): void
- + speardGems(): void
- initialize(): void
- initBoardG(): void
- setTurn(turn : int) : void
- turn Announce(): void
- newTurn():void
- changeColorWhenSpreadG(squareId: int): void
- resetToDefaultColor(squareId:int):void
- getPointLoop(): void
- spreadRowNoGem(turn:int):void
- getScoreInBoard(): void
- resetNumberOfGem(squareId : int) : void
- convertSquareId(squareId:int):int
- backToMenu(): void
- ~ pauseImageClicked(event: MouseEvent): void
- ~ pauseContinueClicked(event: MouseEvent): void
- ~ pauseBackClicked(event: MouseEvent): void
- ~ newGameBtnClicked(event: MouseEvent): void
- ~ exitBtnClicked(event: MouseEvent): void

controller

MenuScreenController

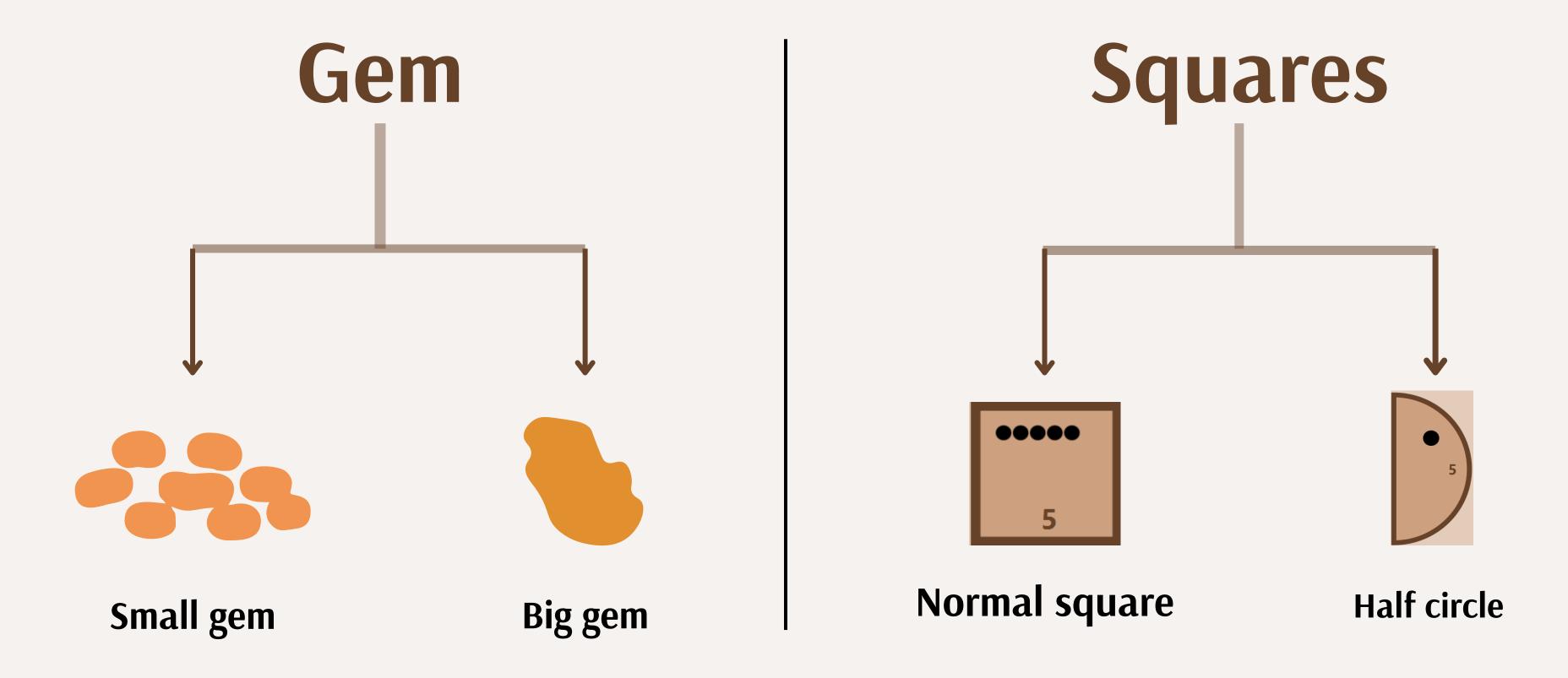
- quitBtn, playBtn, ruleBtn: Button
- quit(stage: Stage): void
- ~ ruleBtnClicked(event: Mouse Event): void
- ~ playBtnClicked(event: MouseEvent): void
- ~ quitBtnClicked(event: MouseEvent): void

RuleController

- ruleplayBtn, backBtn : Button
- ~ ruleplayBtnClicked(event : MouseEvent) : void
- ~ backBtnClicked(event : MouseEvent) : void

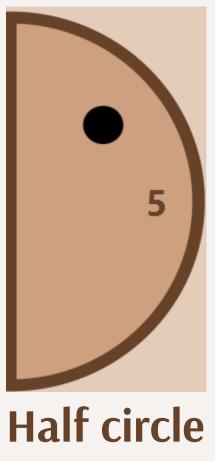
1

Inheritance

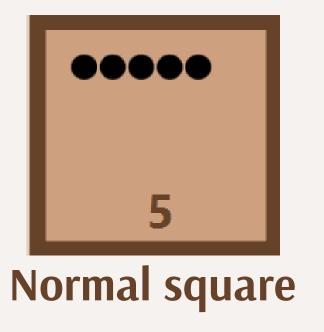


Polymorphism

Method overriding



```
public int getPoint() {
  BigGem bigGem = new BigGem();
  SmallGem smallGem = new SmallGem();
  int point = numberOfBigGems*bigGem.getPoint() +
    numberOfSmallGems*smallGem.getPoint();
  return point;
}
```

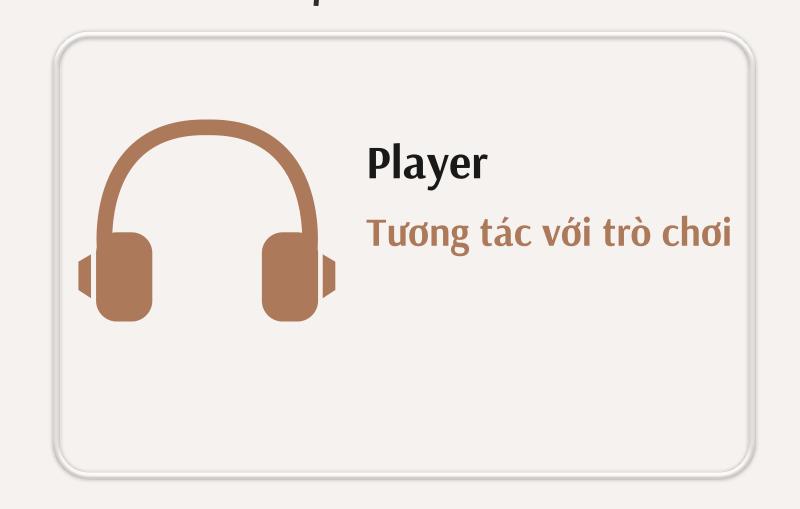


```
public int getPoint() {
   SmallGem smallGem = new SmallGem();
   return smallGem.getPoint()*numberOfSmallGems;
  }
}
```

Association



player : Playerboard : BoardGame





BoardGame

Chứa các ô và đá

Video demo

Demo video