OBJECT-ORIENTED LANGUAGE
MINI-PROJECT

Team 12

Nguyễn Thị Thu Giang 20194750

Tống Thị Thu Anh 20194728

MANDARIN SQUARE CAPTURING

Work assignments: All members working all tasks together via video call on Teams, including design GUI, design diagrams, ideas of how to implement each class and method and coding work.

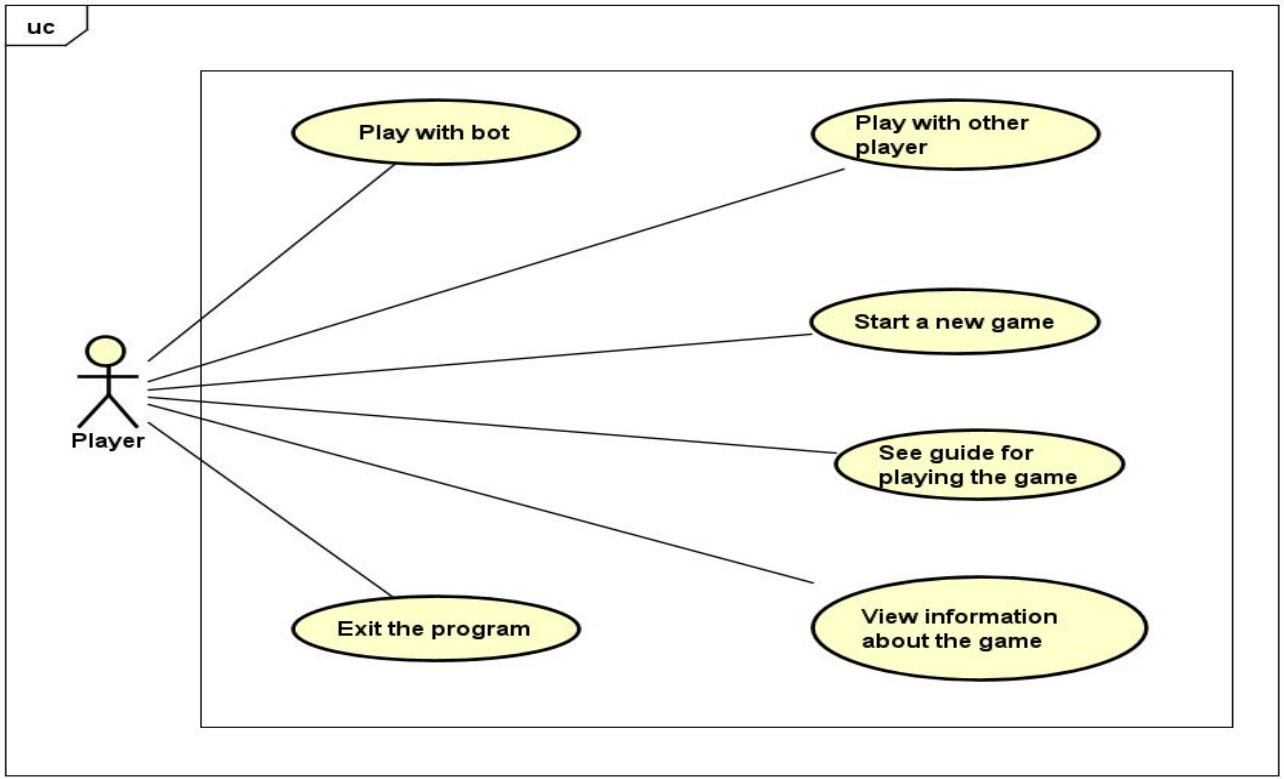
02. PROBLEM STATEMENT

TRADITIONAL GAME: Ô ĂN QUAN

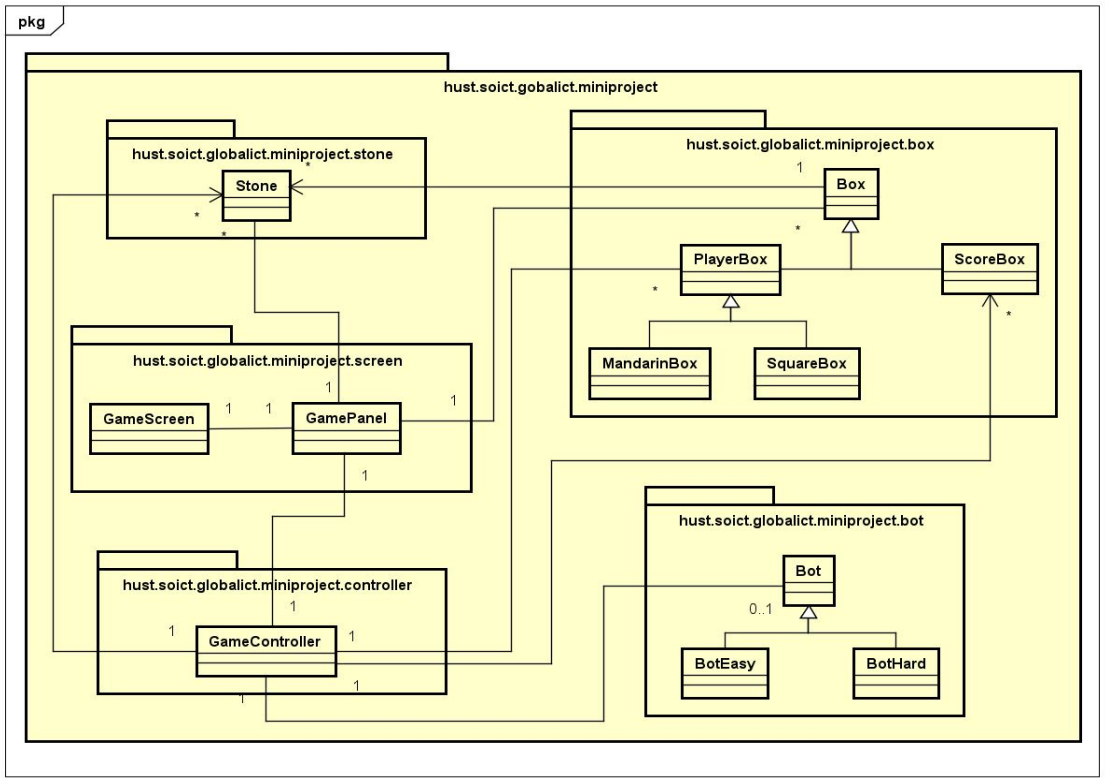
- On the main screen:
  + Start: start the game.
  + Exit: exit the program. Be sure to ask users if they really want to quit the game
  + Help: Show guide for playing the game
- In the game:
  + Game board: The game board consists of 10 squares, divided into 2 rows, and 2 half-circles on the 2 ends of the board. Initially, each square has 5 small gems, and each half- circle has 1 big gem. Each small gem equals 1 point, and each big gem equals 5 points.
  + For each turn, the application must show clearly whose turn it is. A player will select a square and a direction to spread the gems. He got points when after finishing spreading, there is one empty square followed by a square with gems. The score the got for that turn is equal to the number of gems in that followed square (see the gameplay for more details about streaks)
  + The game ends when there is no gem in both half-circles. The application must notify who is the winner and the score of each player.

03.

USE CASE DIAGRAM

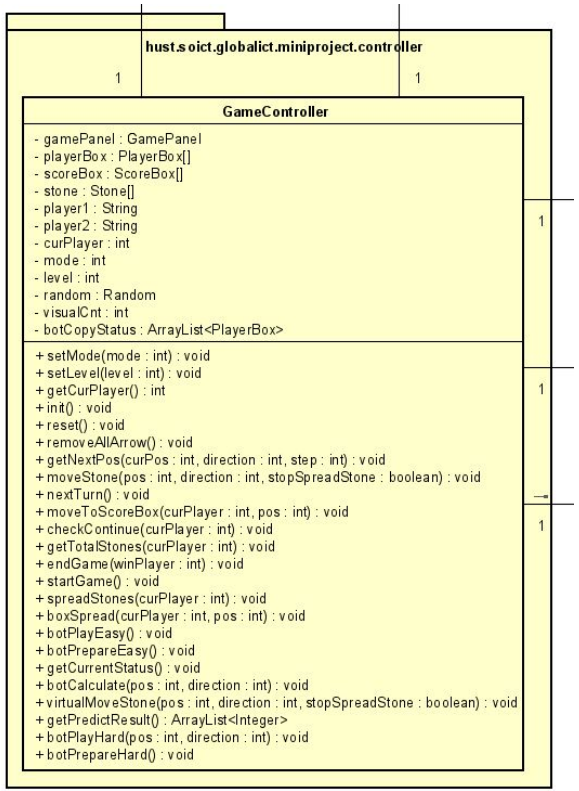


04. GENERAL CLASS DIAGRAM



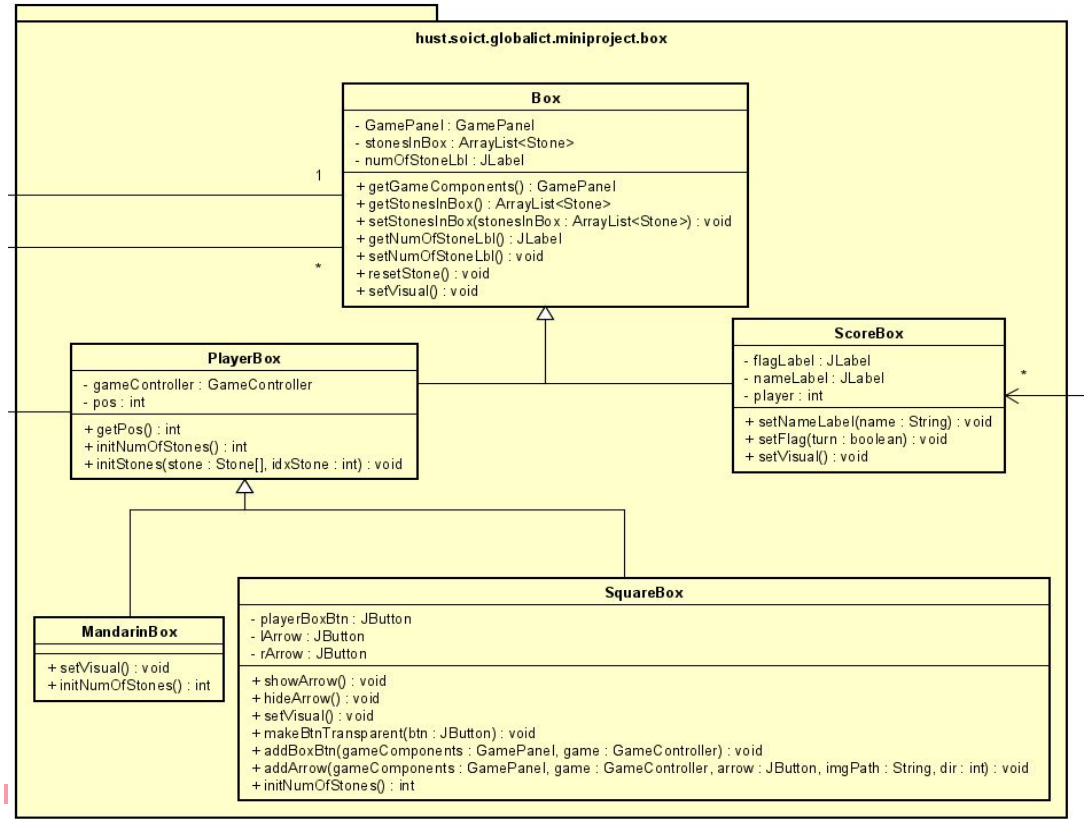
05.

CLASS DIAGRAMS



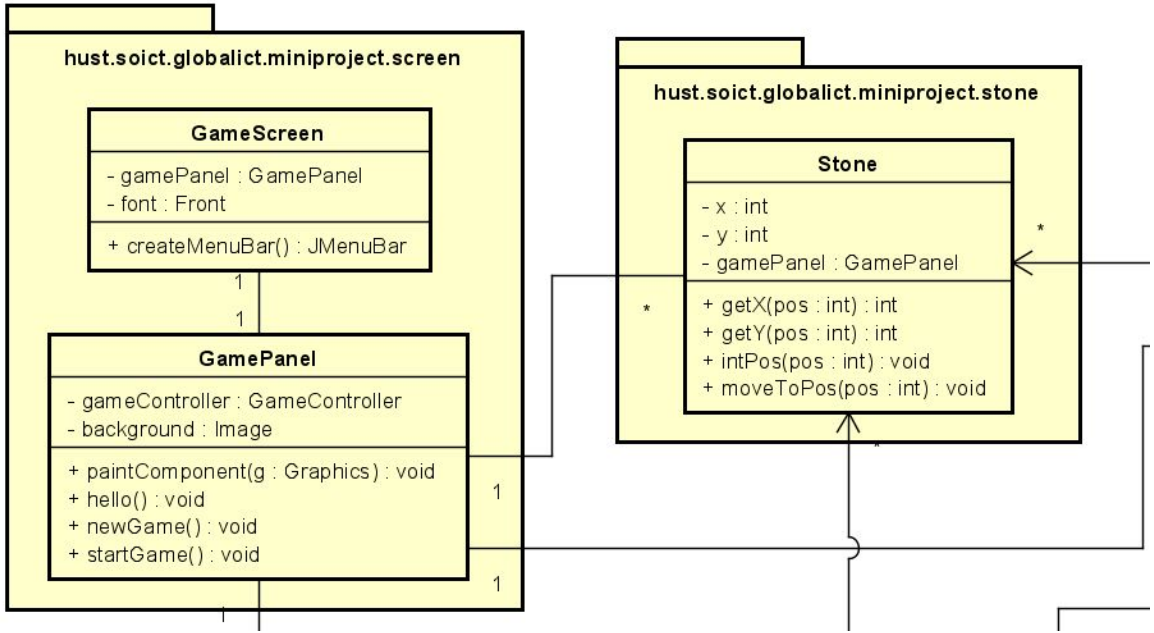
05.

CLASS DIAGRAMS

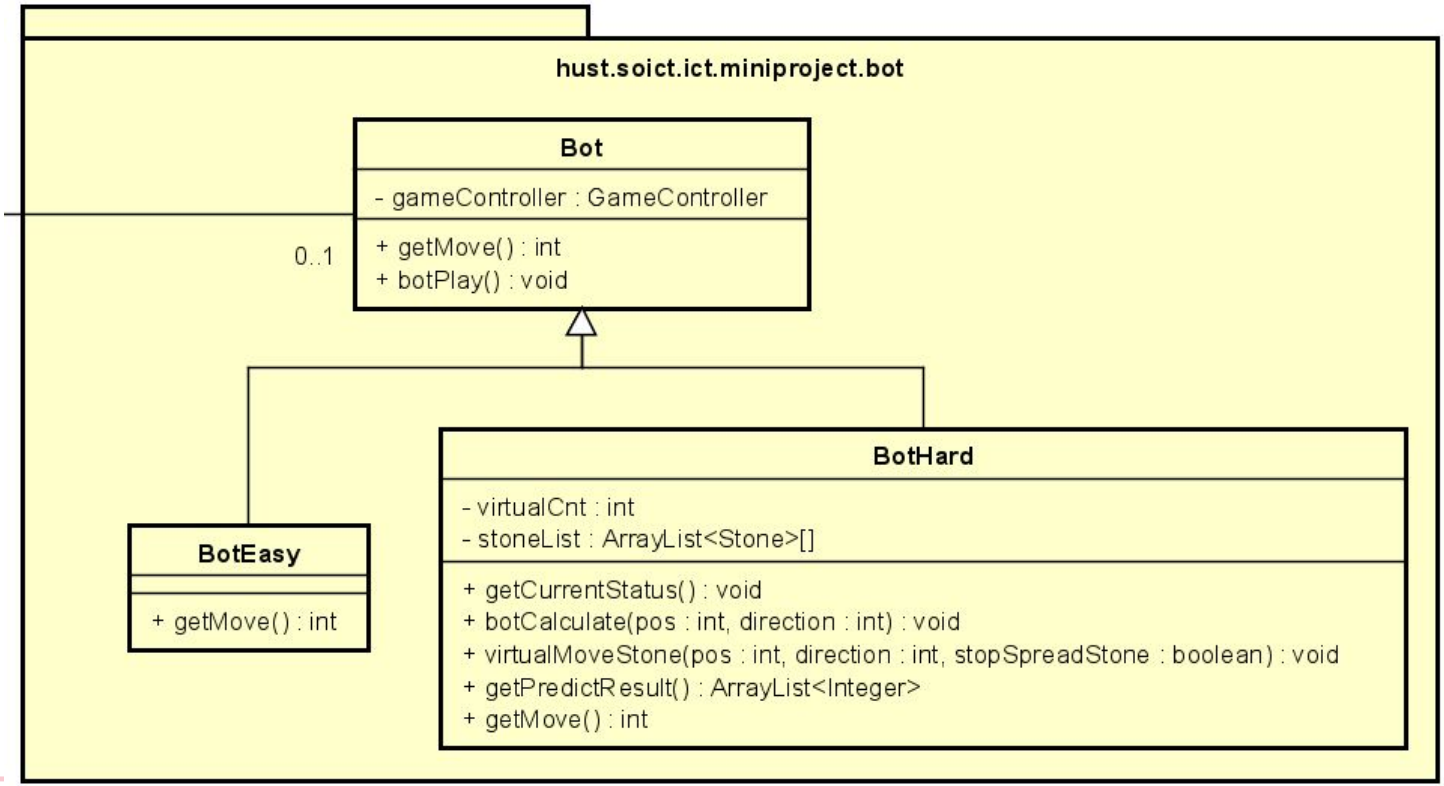


05.

CLASS DIAGRAMS



05. CLASS DIAGRAMS



06. OOP TECHNIQUES

INHERITANCE

- Class ScoreBox and abstract class PlayerBox inherit Box
- Multi-level inheritance: Class MandarinBox and SquareBox inherit PlayerBox
- BotEasy and BotHard inherit Bot

ASSOCIATION

- Classes Box and Stone have one-to-many Association relationship:
- GamePanel is a component of GameScreen (JFrame) (one-to-one association)
- Classes Box, Stone have some components (JButton, JLabel, Image) of GamePanel (many-to-one association)
- Class GameController associate with PlayerBox, ScoreBox, Stone (one-to-many association) and Bot (one-to- zero or one)

06. OOP TECHNIQUES

POLYMORPHISM

Medthod init() in GameController:

```
public void init() {
    for(int i=0; i<5; i++) playerBox[i] = new SquareBox(gameComponents, this,i);
    for(int i=6;i<11;i++) playerBox[i]=new SquareBox(gameComponents, this,i);
    playerBox[5] = new MandarinBox(gameComponents, this, 5);
    playerBox[11] = new MandarinBox(gameComponents, this, 11);
    for(int i=0; i<12; i++) playerBox[i].setVisual();

    scoreBox[0] = new ScoreBox(gameComponents, 0);
    scoreBox[0].setVisual();
    scoreBox[1] = new ScoreBox(gameComponents, 1);
    scoreBox[1].setVisual();

    for(int i=0; i<70; i++) {
        stone[i] = new Stone(gameComponents);
        stone[i].setVisible(false);
    }
}
```

06. OOP TECHNIQUES

POLYMORPHISM

Class PlayerBox:

```
public abstract int initNumOfStones();

public void initStones(Stone[] stone, int idxStone) {
    int num = initNumOfStones();
    for(int i=0; i<num; i++) {
        stone[idxStone].initPos(pos);
        getStonesInBox().add(stone[idxStone++]);
    }
}
}
```

Method reset() in GameController:

```
int cnt = 0;
for(int i=0; i<12; i++) {
    playerBox[i].resetStone();
    playerBox[i].initStones(stone, cnt);
    cnt += playerBox[i].initNumOfStones();
    playerBox[i].setNumOfStoneLbl();
}
```

06. OOP TECHNIQUES

POLYMORPHISM

Class Bot:

```
public abstract int getMove();

public void botPlay() {
    int timePrepare = new Random().nextInt(2)+1;
    Timer timer = new Timer(0, new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            try {
                Thread.sleep(1000*timePrepare);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            int move = getMove();
            if(move < 5) gameController.moveStone(move, 1, false);
            else gameController.moveStone(move-5, -1, false);
        }
    });
    timer.setRepeats(false);
    timer.start();
}
```

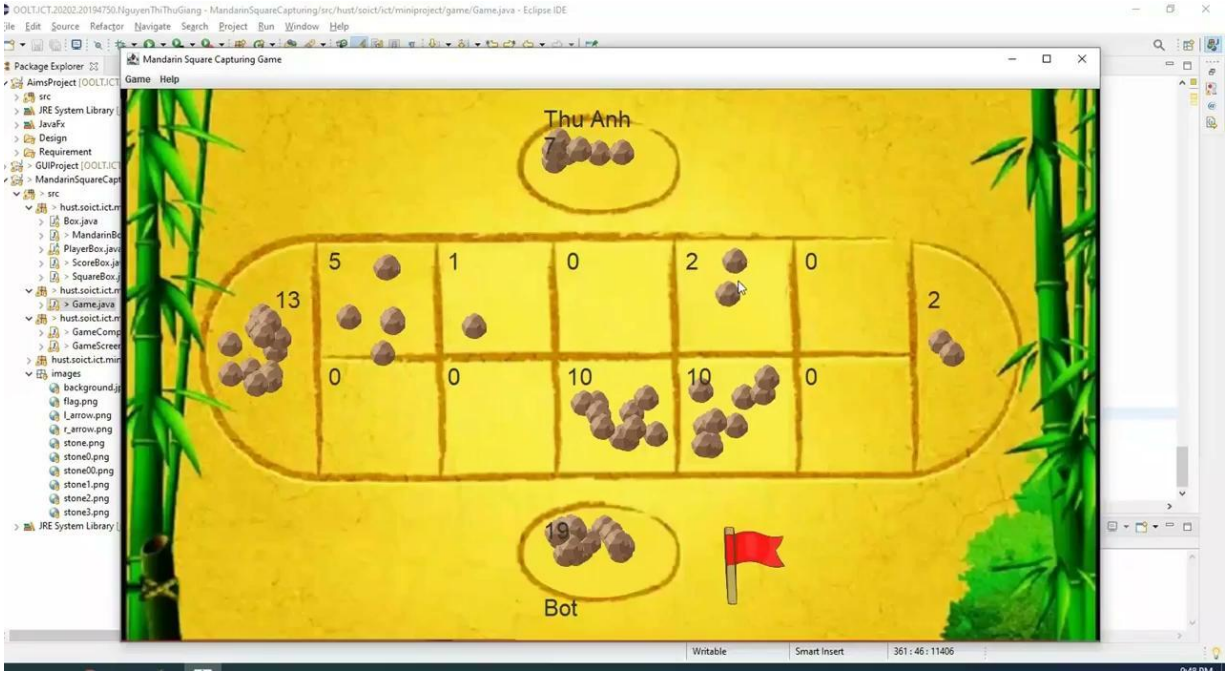
Method reset() in GameController:

```
if(mode == 1) {
    if(level == 1) bot = new BotHard(this);
    else bot = new BotEasy(this);
    if(curPlayer == 1) bot.botPlay();
    else firstPlay = false;

}
```

06.

DEMO SCENARIO



Link of demo video for the application: [LINK]