O An Quan

TEAM 26

Hoang Long Vu
20204897

Board, Cell, Player, GUI

Pham Vu Huyen Trang
20207997

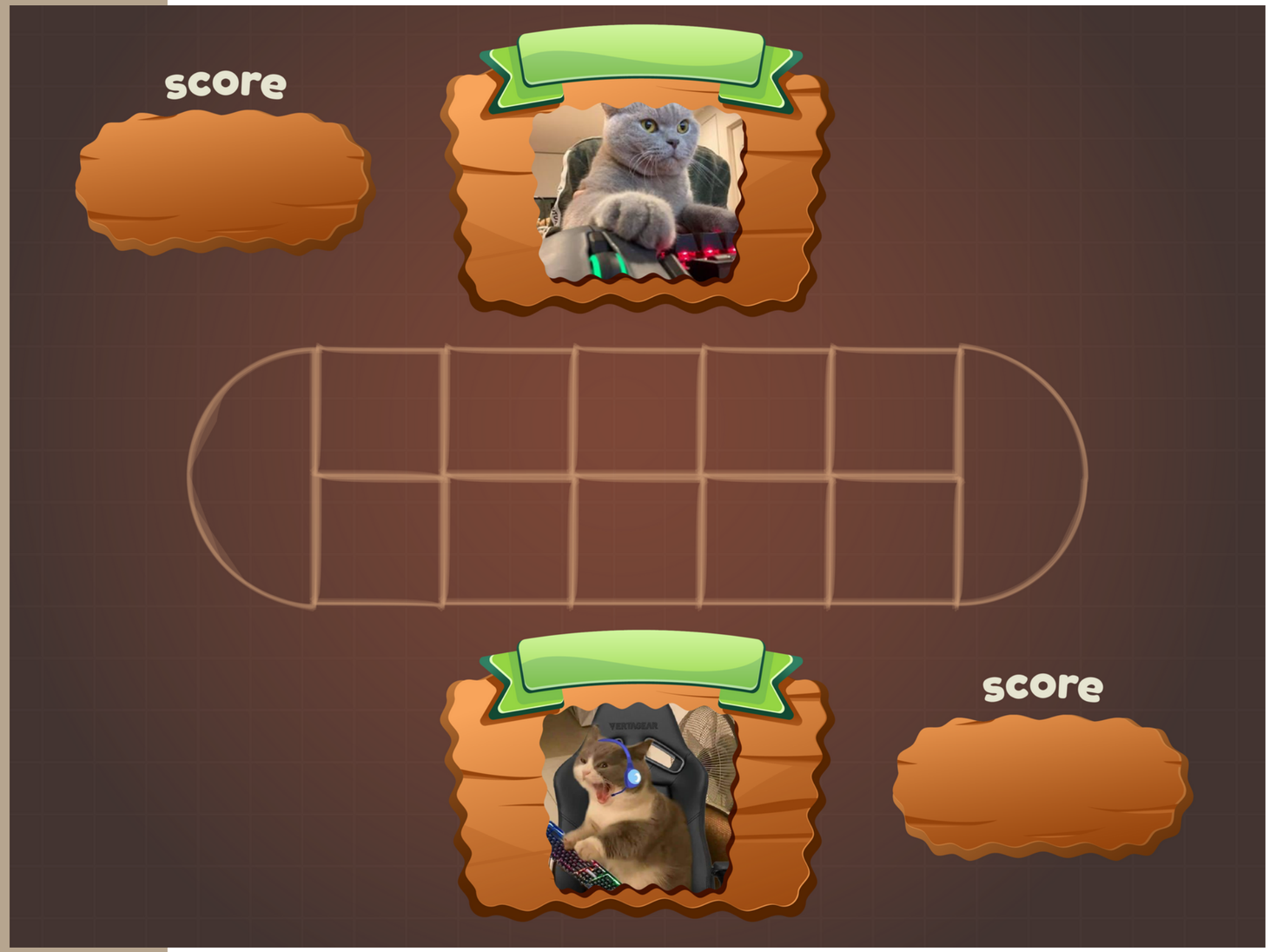
GamePlay, Gem, Player, GUI

Nguyen Huu Vuong
20204898

Canva link: https://www.canva.com/design/DAFFv2prAsw/BztSGsN5imTZRa3MN-mjnA/view?utm_content=DAFFv2prAsw&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

PROBLEM STATEMENT

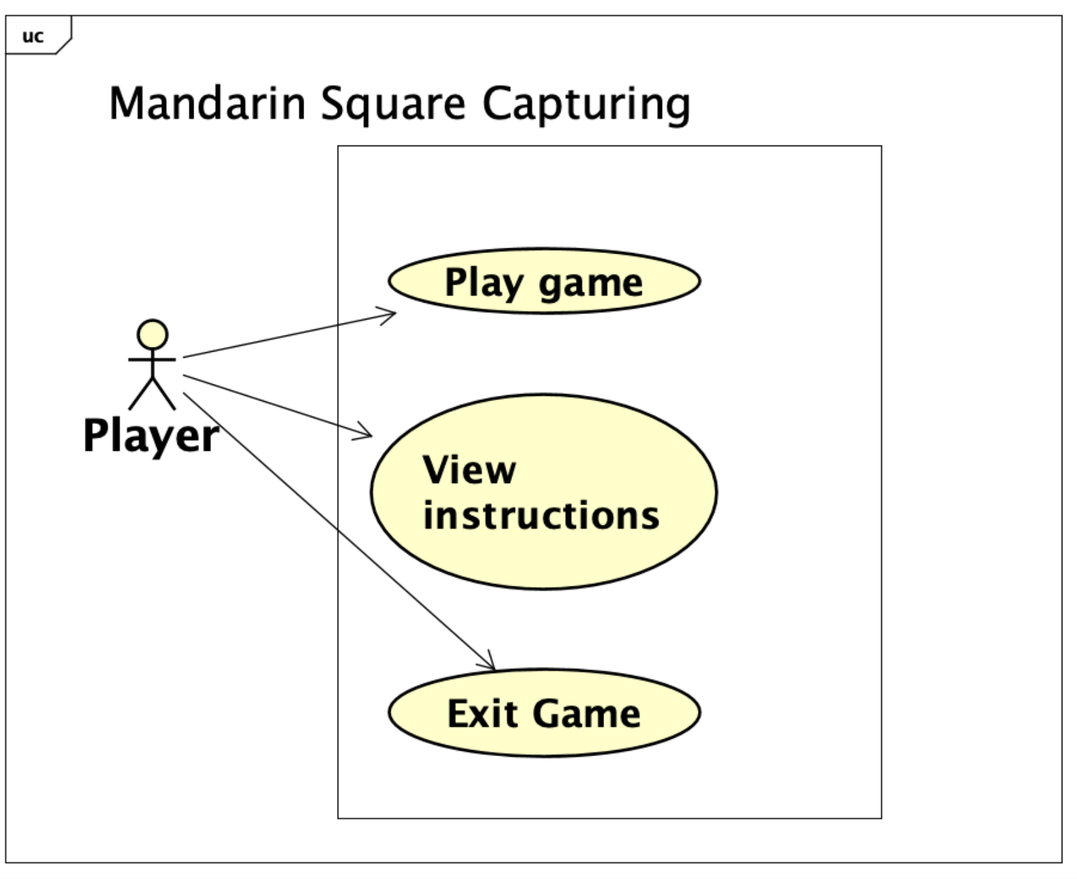
- One board: 10 squares (5 gems each), 2 half circles (1 big gem each)
- Player chooses one cell on their side to start the game.
- Spread the gems until finished.
- The number of gems earned will be calculated as score.
- Game stops when all half circles are empty or the player has no gems on their side to pick up when in turn.
- Player with higher score wins.
- No time constraints.



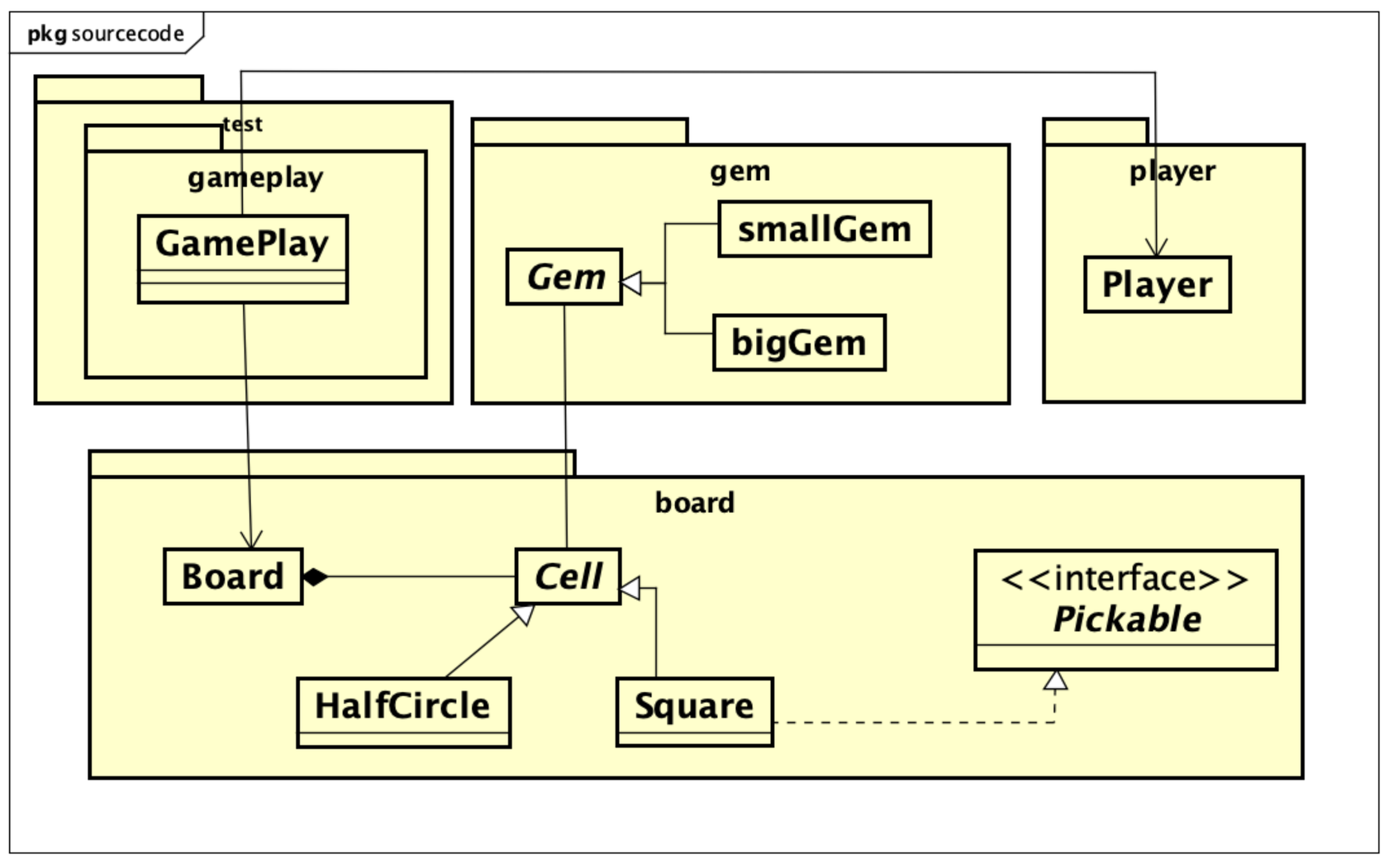
USE CASE DIAGRAM

- *Play game*: When play choose to start the game (by clicking the Start button in the Intro Screen), the program displays a playing board, the player then plays the game following the rules stated in until the game is finished

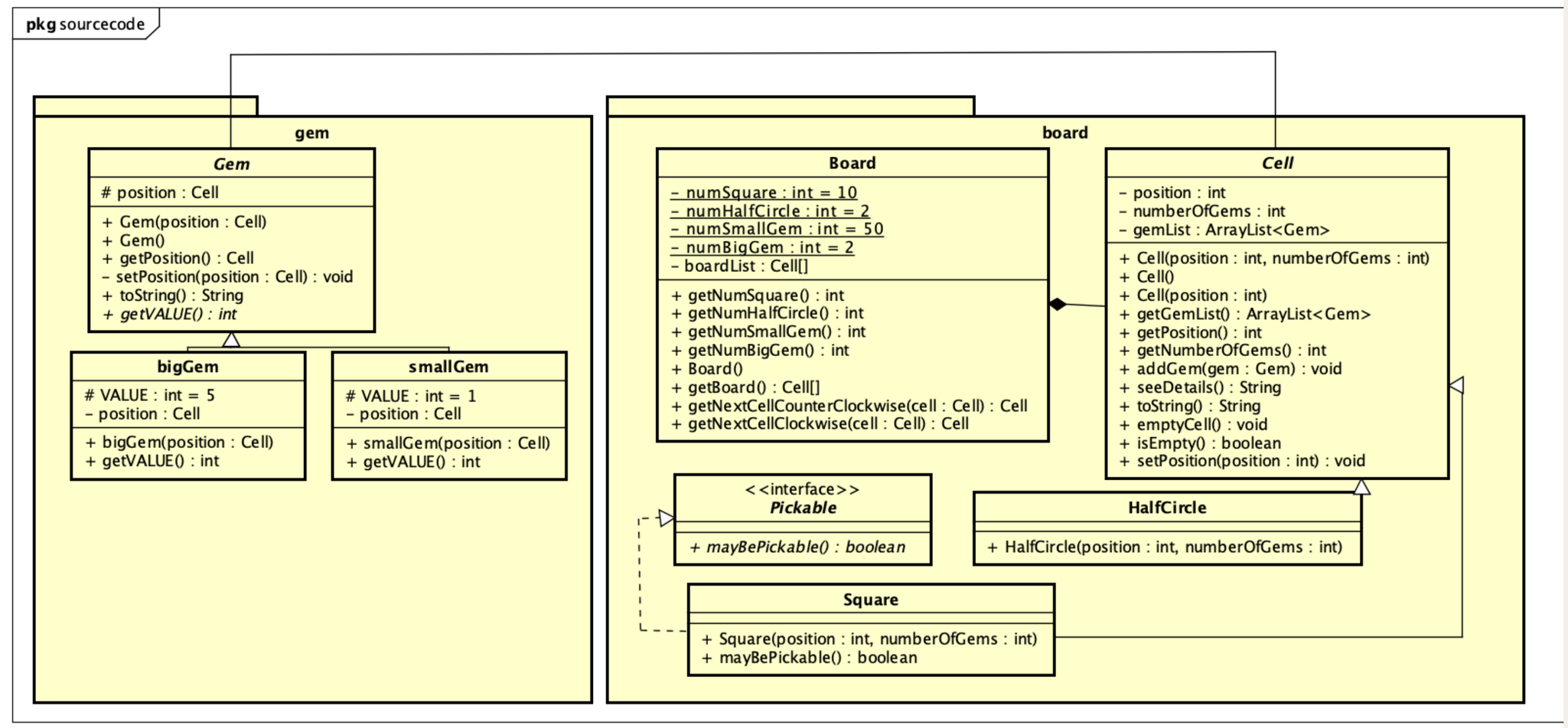
- *Show instructions*: Player can press Help to access the Help Menu from the Intro Screen to read about the instructions, the program should display a board showing rules of the game.



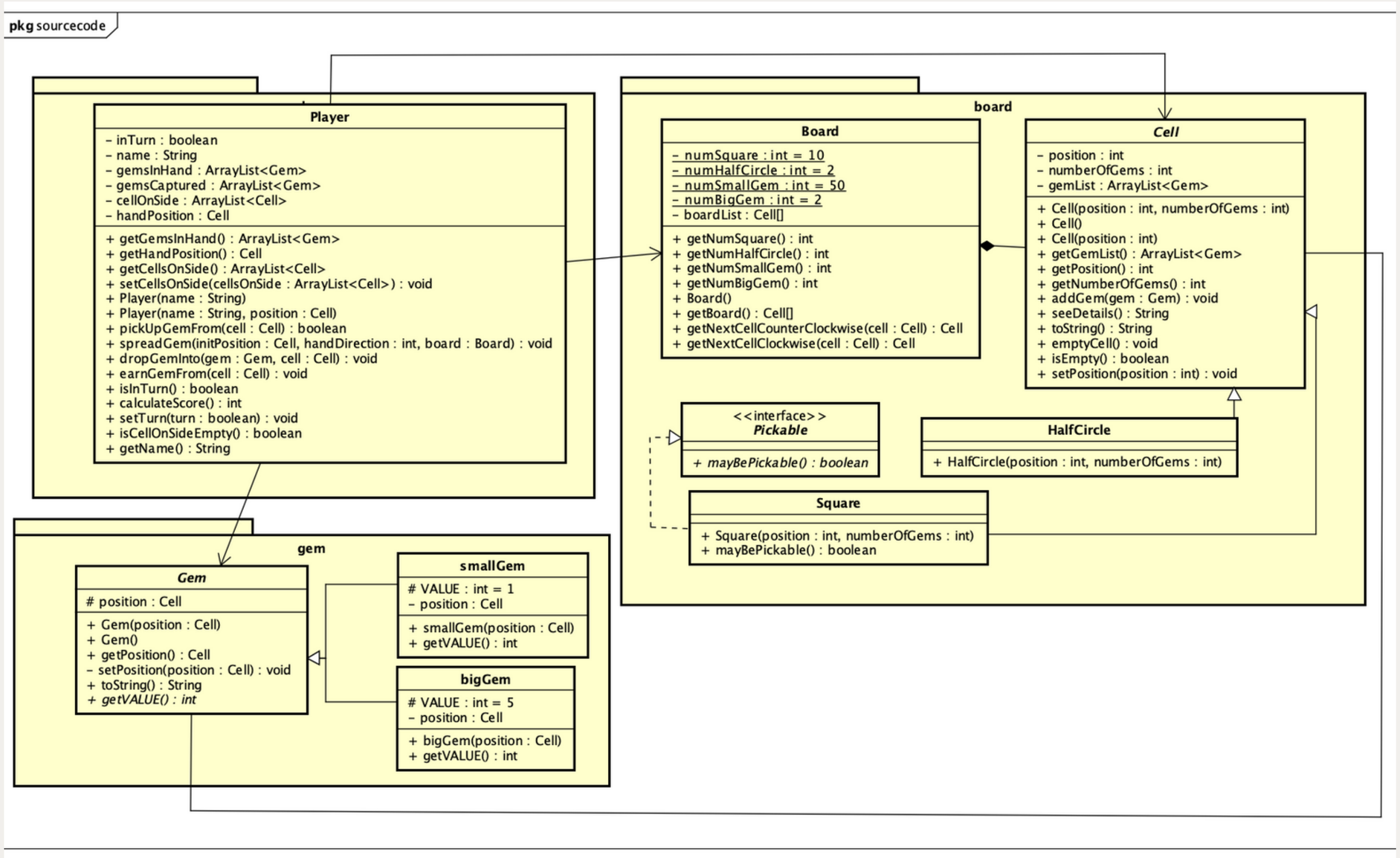
GENERAL CLASS DIAGRAM



DETAILED CLASS DIAGRAM - "PASSIVE"



DETAILED CLASS DIAGRAM - "ACTIVE"



OOP TECHNIQUES - AGGREGATION - ASSOCIATION- COMPOSITION

We have designed ourr model using some OOP techniques

1. Aggregation - Association - Composition

- Gem associates Cell and vice versa
- Gem associates Board
- Board composites Cell
- Board, Gem associates Player
- Board, Gem, Player associates GamePlay

OOP TECHNIQUES - INHERITANCE

2. Inheritance

- An *abstract* class Cell with two child classes Square and Half Circle.

- An *abstract* class Gem and two child classes smallGem and bigGem, each child class is specified by a value: smallGem (value = 1) and bigGem (value = 5)

OOP TECHNIQUES - POLYMORPHISM

3. Polymorphism

- The bigGem and smallGem can be both called as an object of the Gem class, but has different value of the **VALUE** attribute.

- Only the Square is “**pickable**”. When looping through the cells (Cell objects) in a board, depending on the type (Square or HalfCircle), the program will decide if the player can pick up gems from that cell or not.

- In the isPaneEmpty method of the class GamePlayScreenController, the children of the Pane is retrieved via the built-in method getChildren, in which only the Label objects can implement the getText method.

- toString() method is overridden in the bigGem and smallGem."

DEMO VIDEO

Link: https://drive.google.com/file/d/1v8G6gZ2c93rFomoLYvm6NvUkUzqDJN7k/view?usp=sharing