



MANDARIN SQUARE CAPTURING

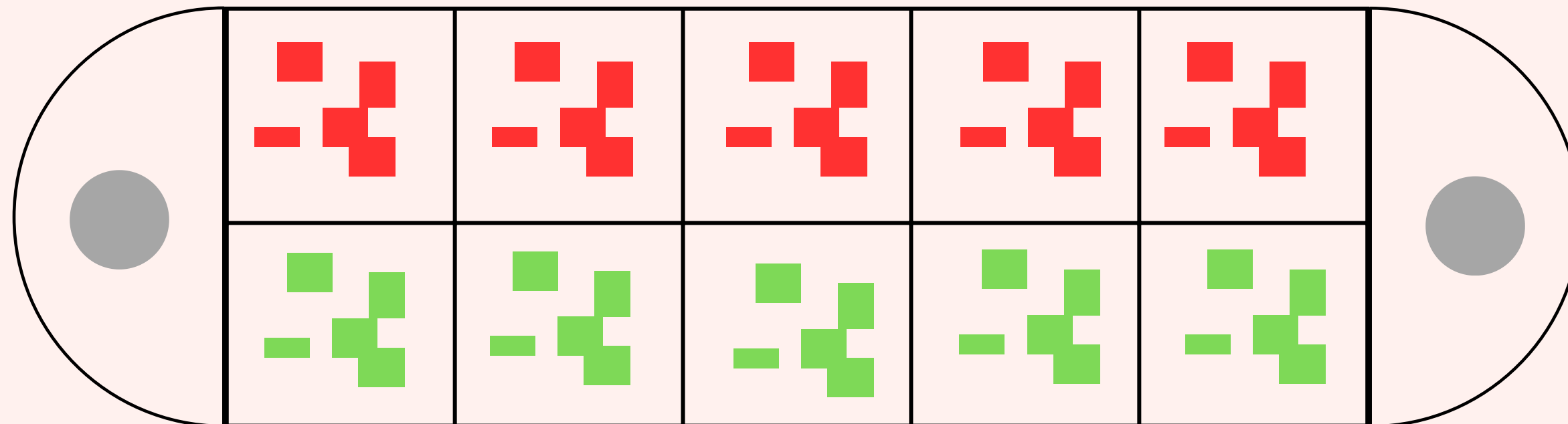
OOP2023.2.Team29



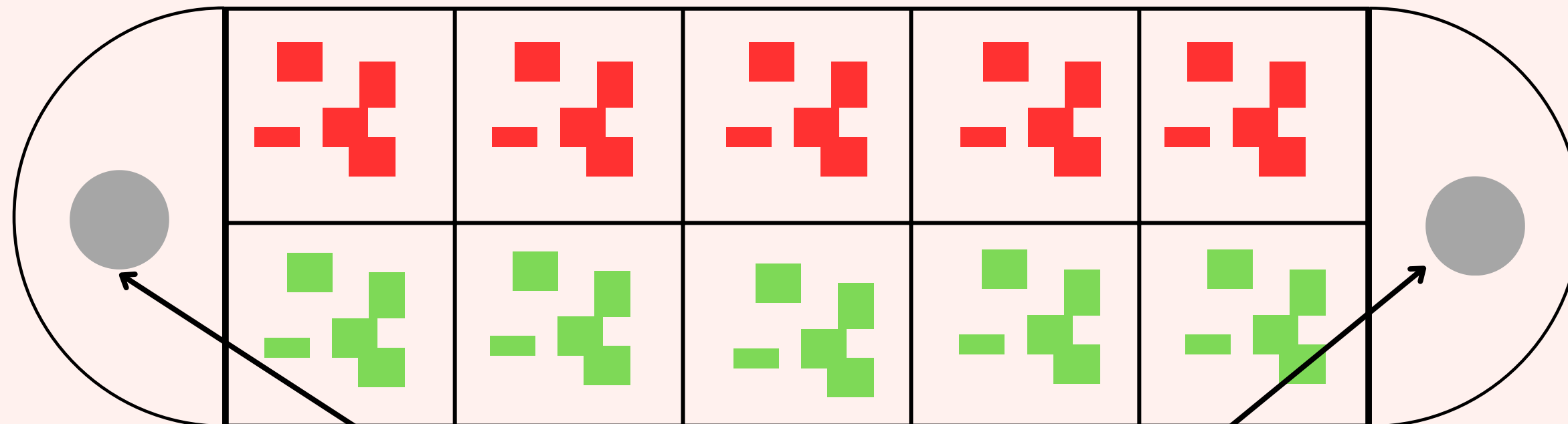
TABLE OF CONTENT

**HOW TO PLAY
UML
OOP TECHNIQUE
DEMO**

HOW TO PLAY

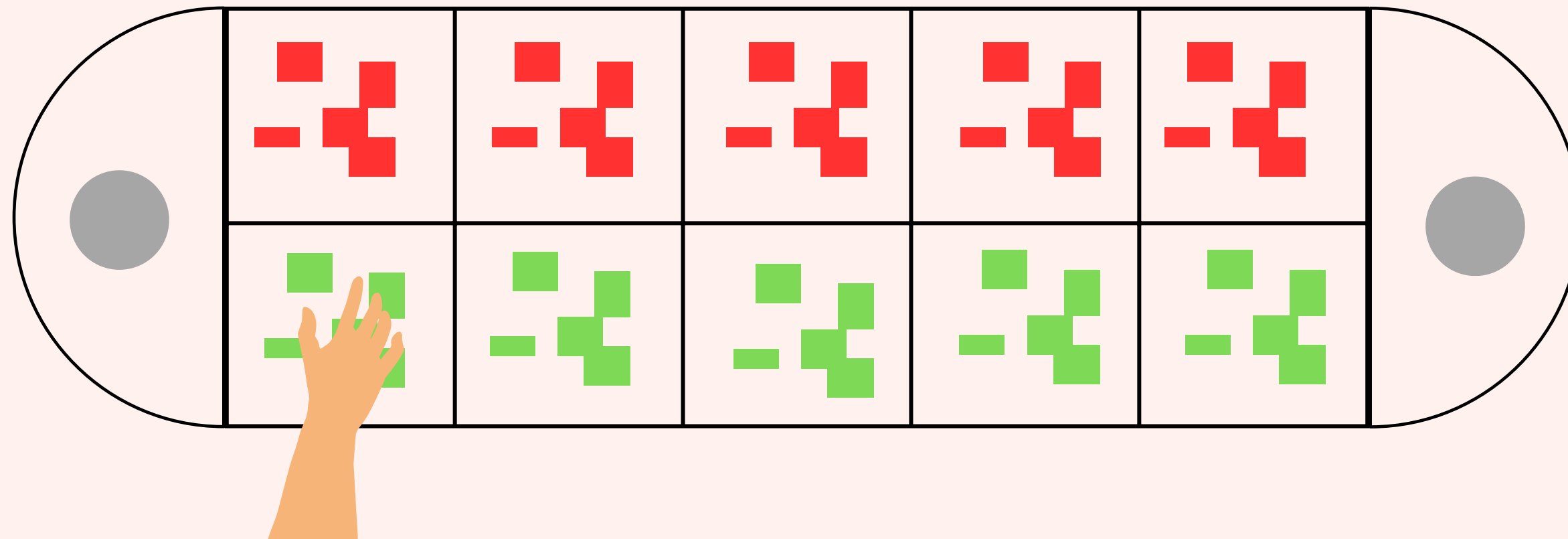


HOW TO PLAY

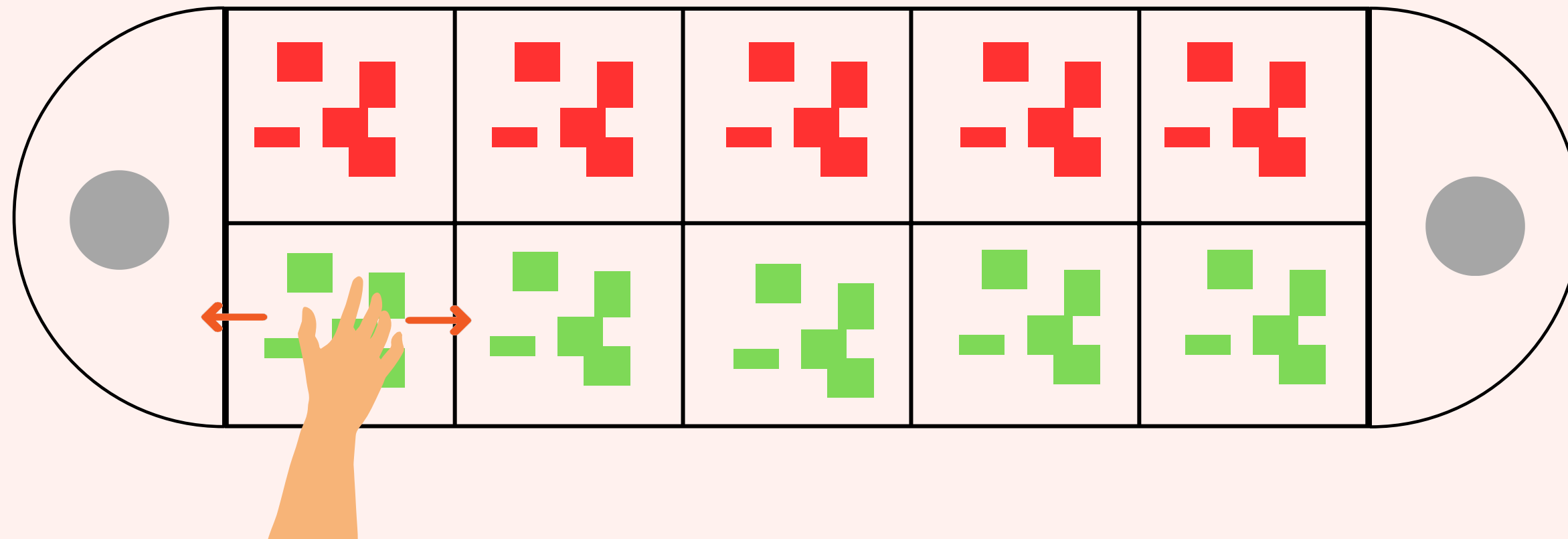


Quan

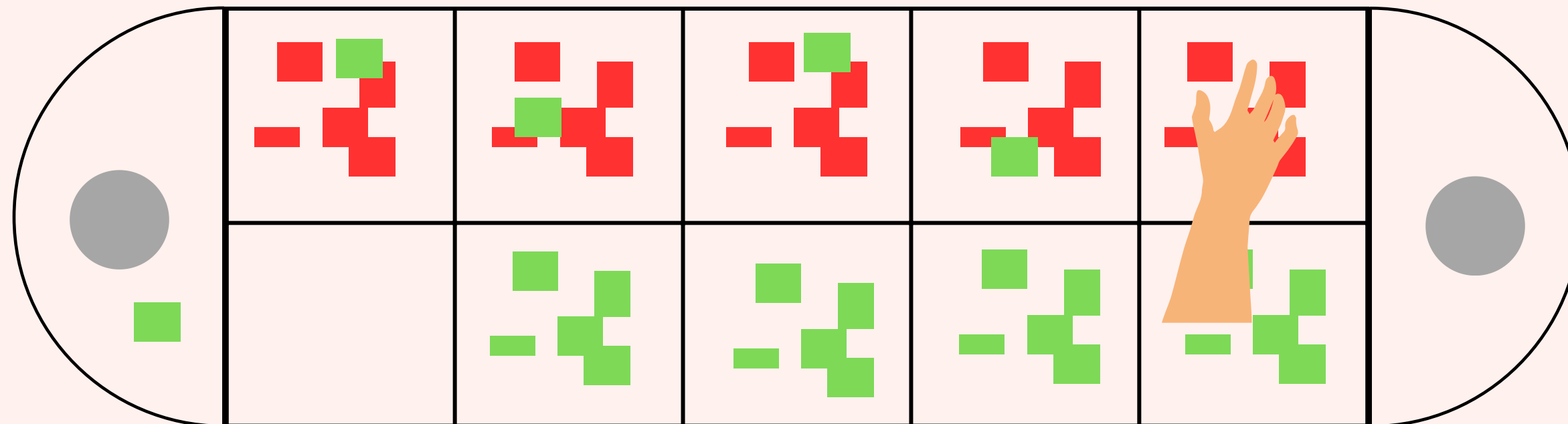
HOW TO PLAY



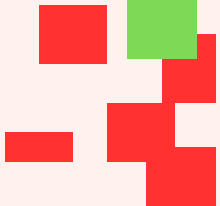
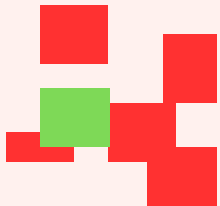
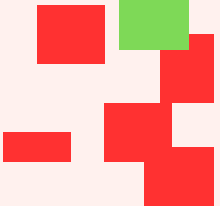
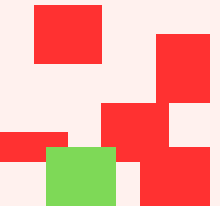
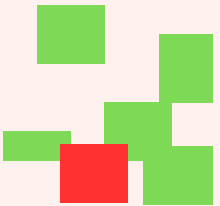
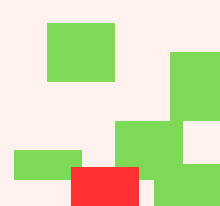
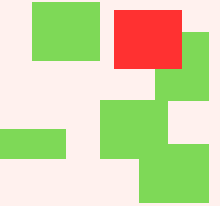
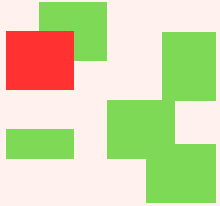
HOW TO PLAY



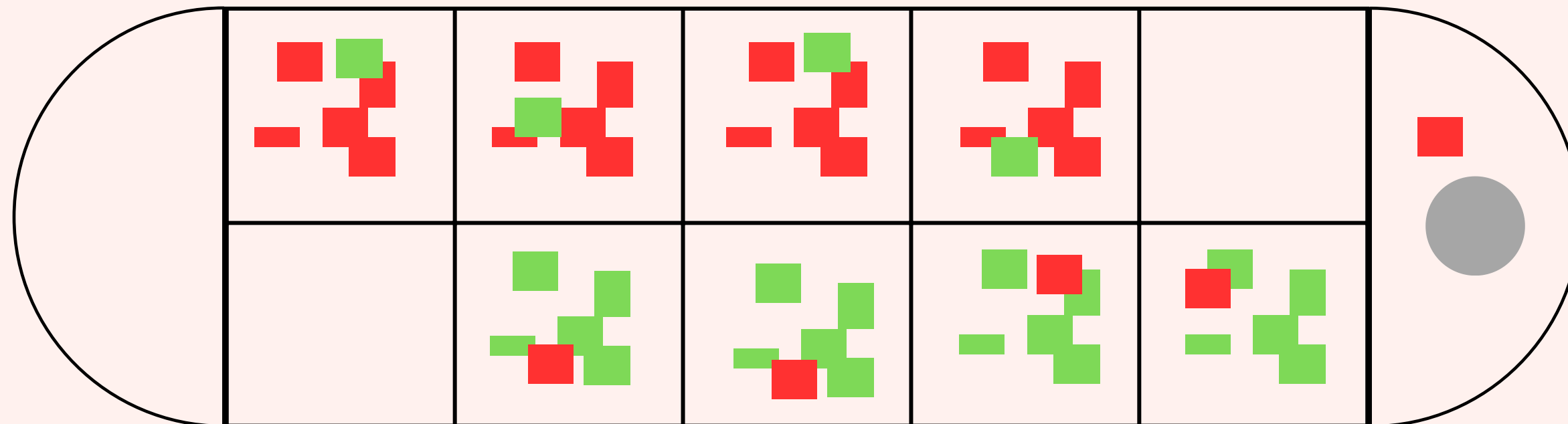
HOW TO PLAY



HOW TO PLAY

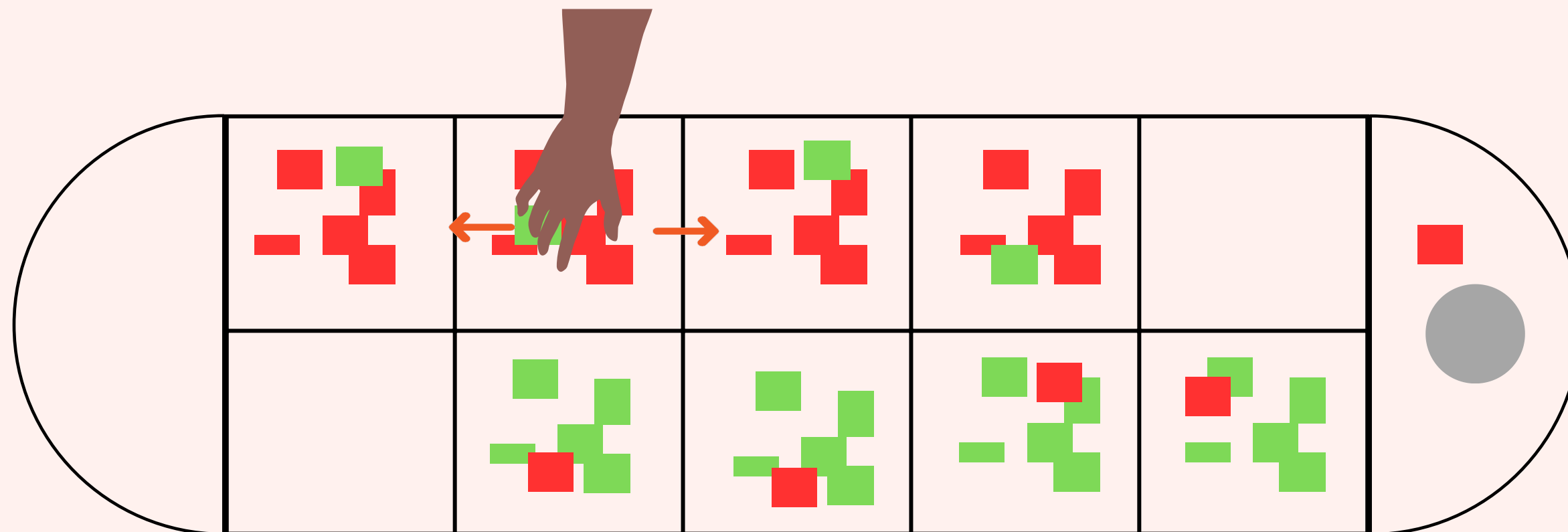
| | | | | | | |
|--|---|---|---|---|---|--|
| |  |  |  |  | | |
| | |  |  |  |  | |

HOW TO PLAY



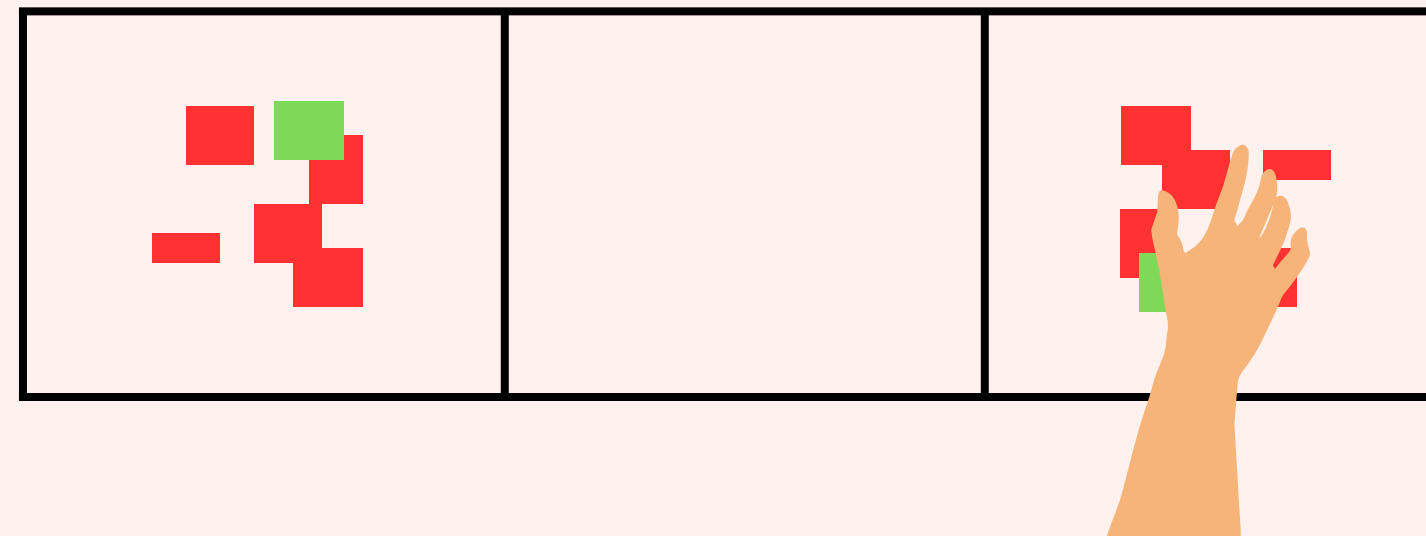
  +6

HOW TO PLAY



HOW TO PLAY

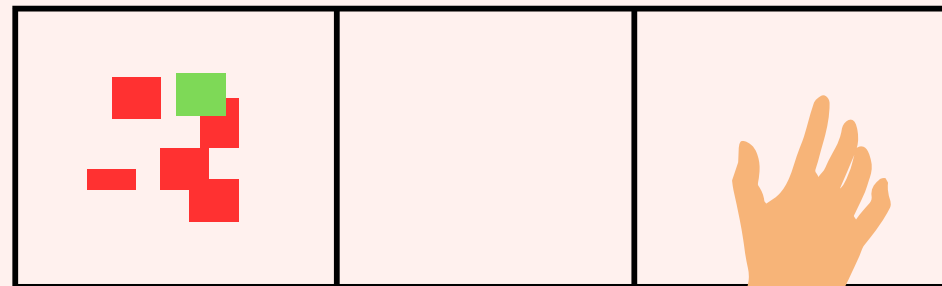
You here



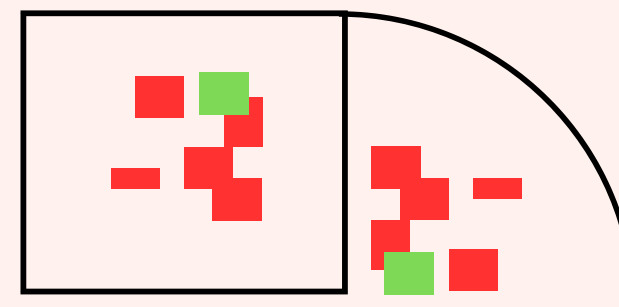
STOP THE TURN AND EARN POINT

HOW TO PLAY

You here

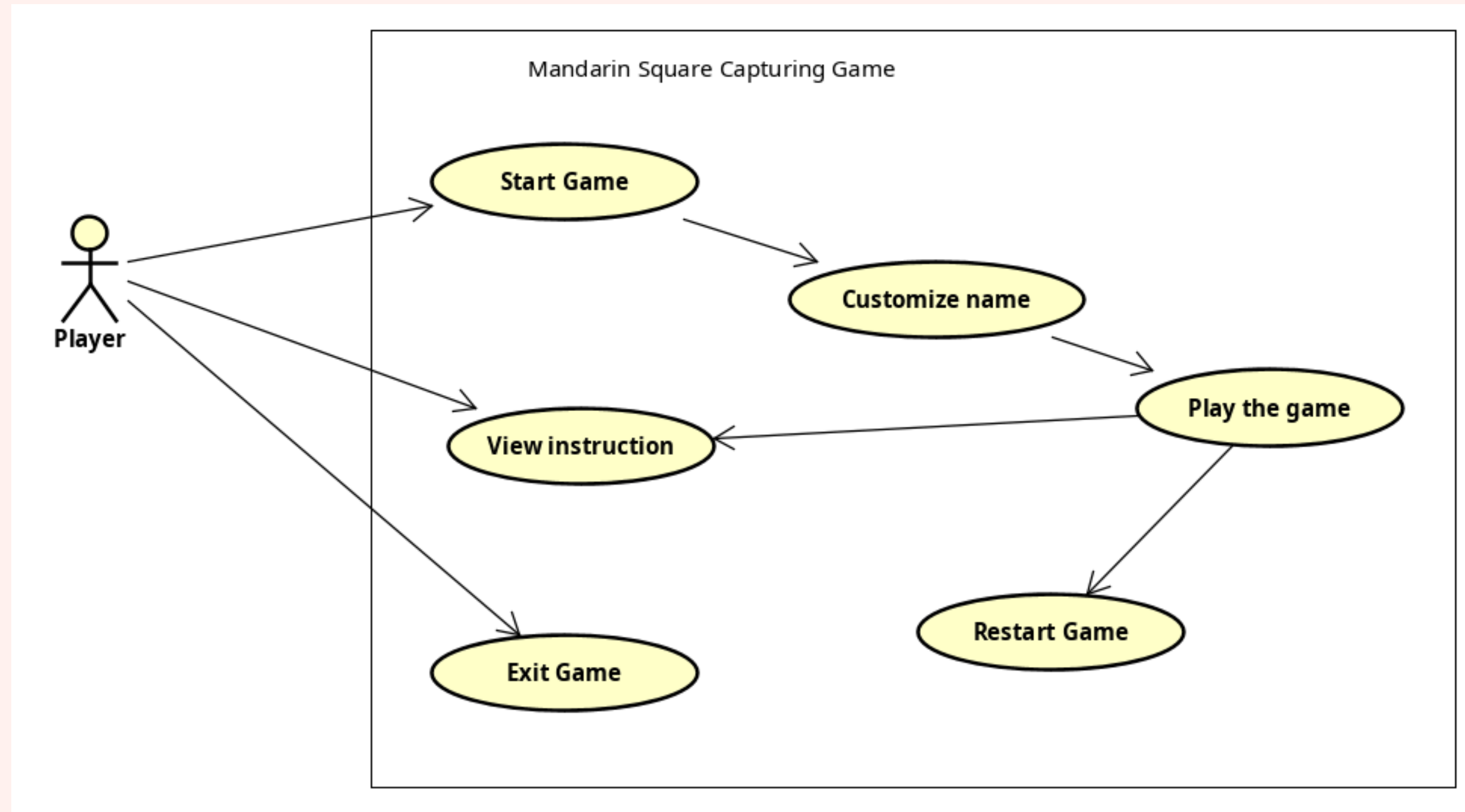


You here

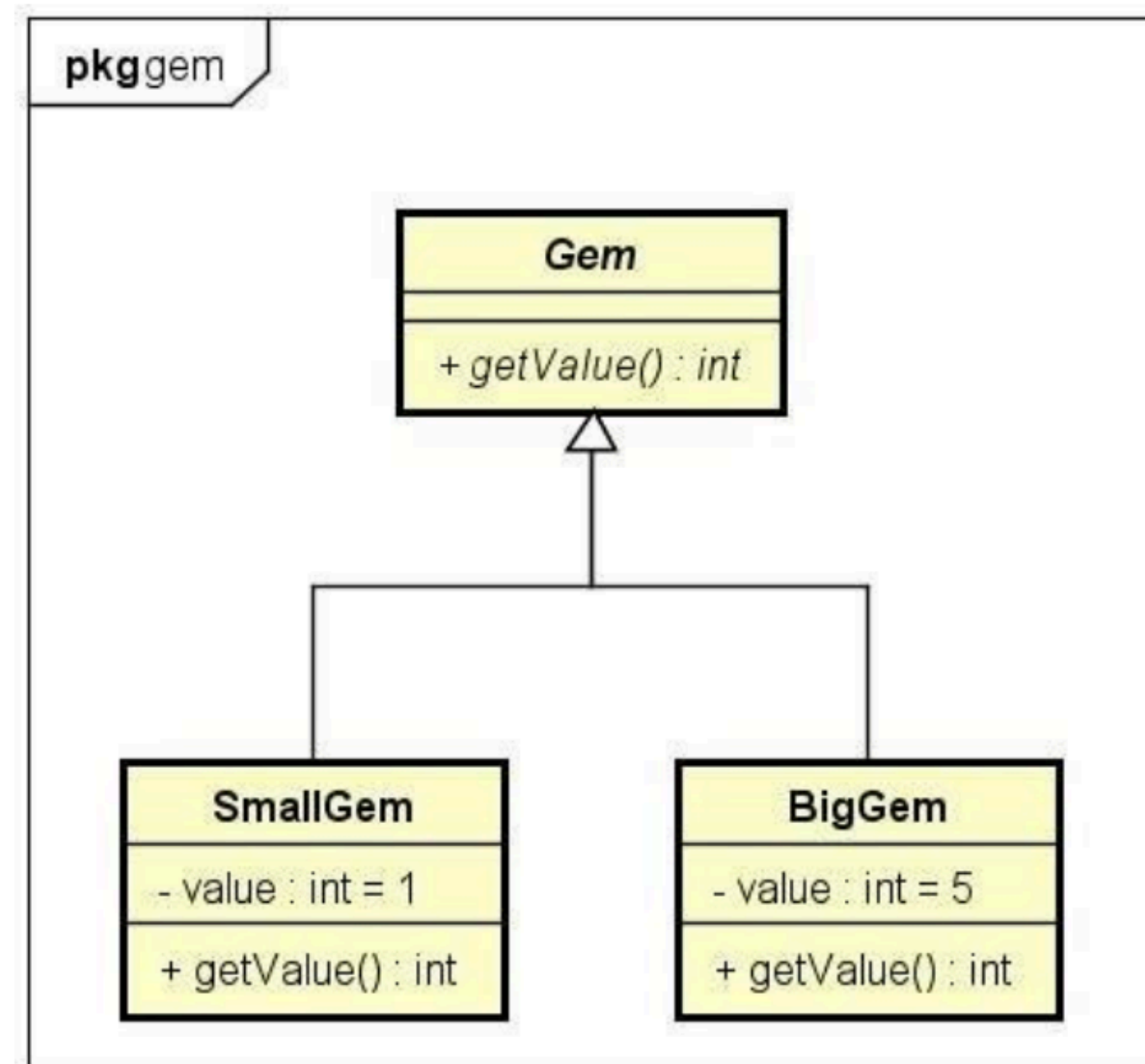


STOP THE TURN WITHOUT EARNING POINT

USE CASE DIAGRAM



CLASS DIAGRAM



CLASS DIAGRAM

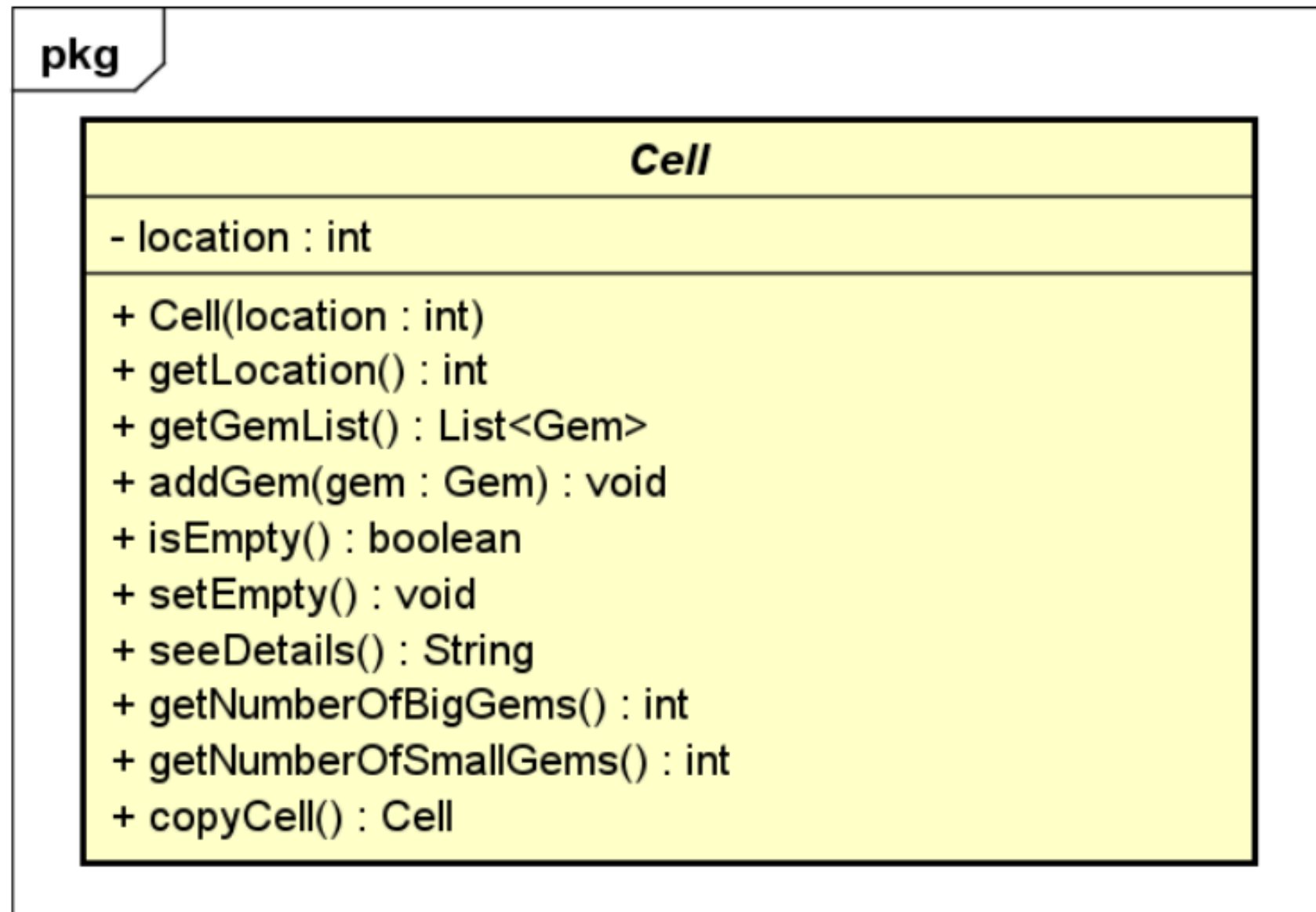
pkg

Board

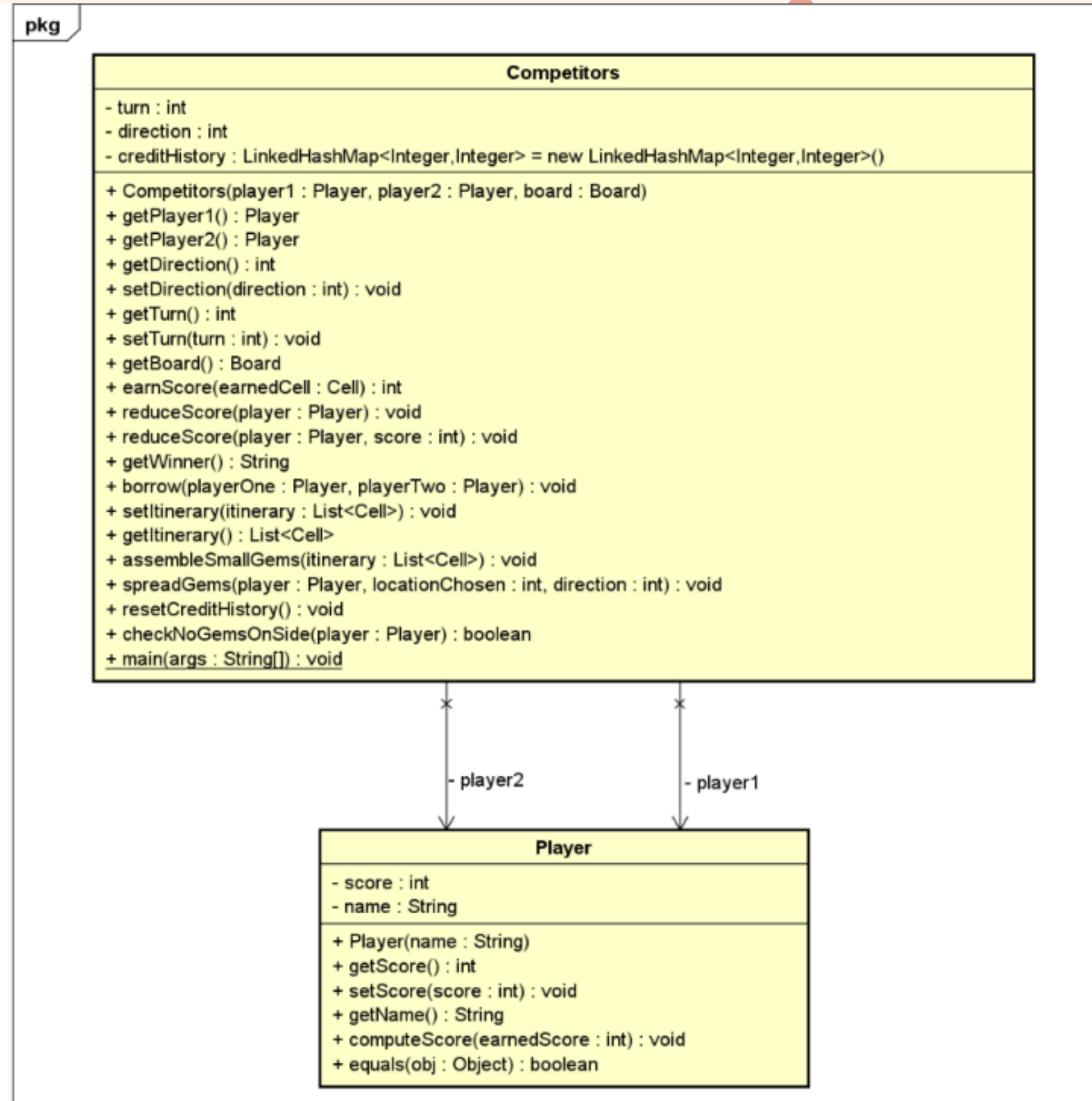
- numSquares : int = 10
- numHalfCircles : int = 2
- numBigGems : int = 2
- numSmallGems : int = 50

+ getNumSquares() : int
+ getNumHalfCircles() : int
+ getNumSmallGems() : int
+ getNumBigGems() : int
+ Board()
- initializeCells() : void
- addGemsToCells() : void
+ getCells() : Cell[]
+ getNextCellCounterClockwise(cell : Cell) : Cell
+ getNextCellClockwise(cell : Cell) : Cell
+ endGame() : boolean

CLASS DIAGRAM



CLASS DIAGRAM



CLASS DIAGRAM

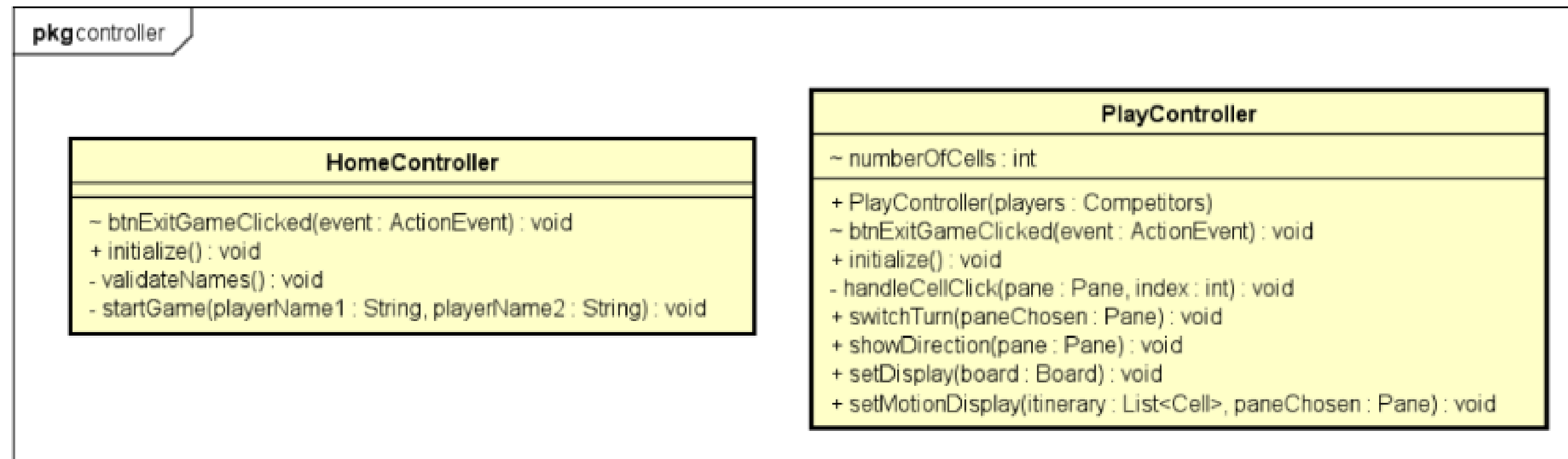
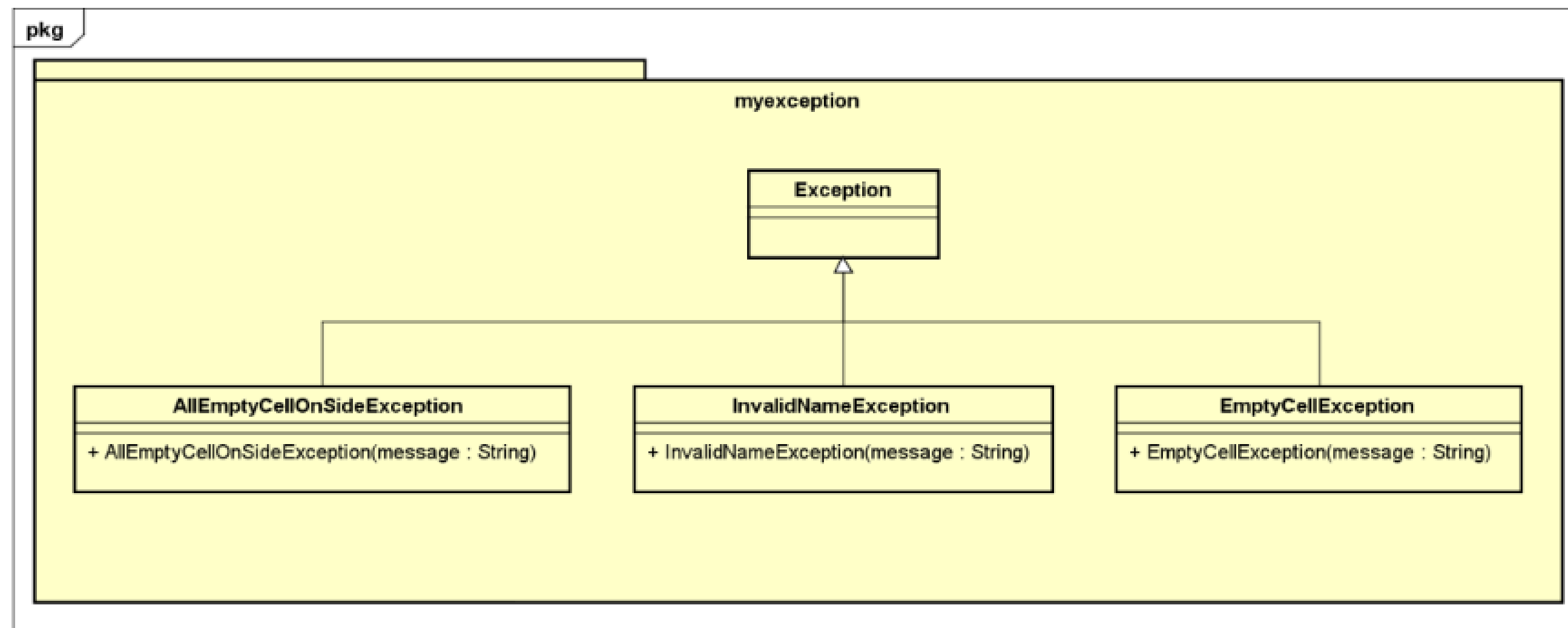
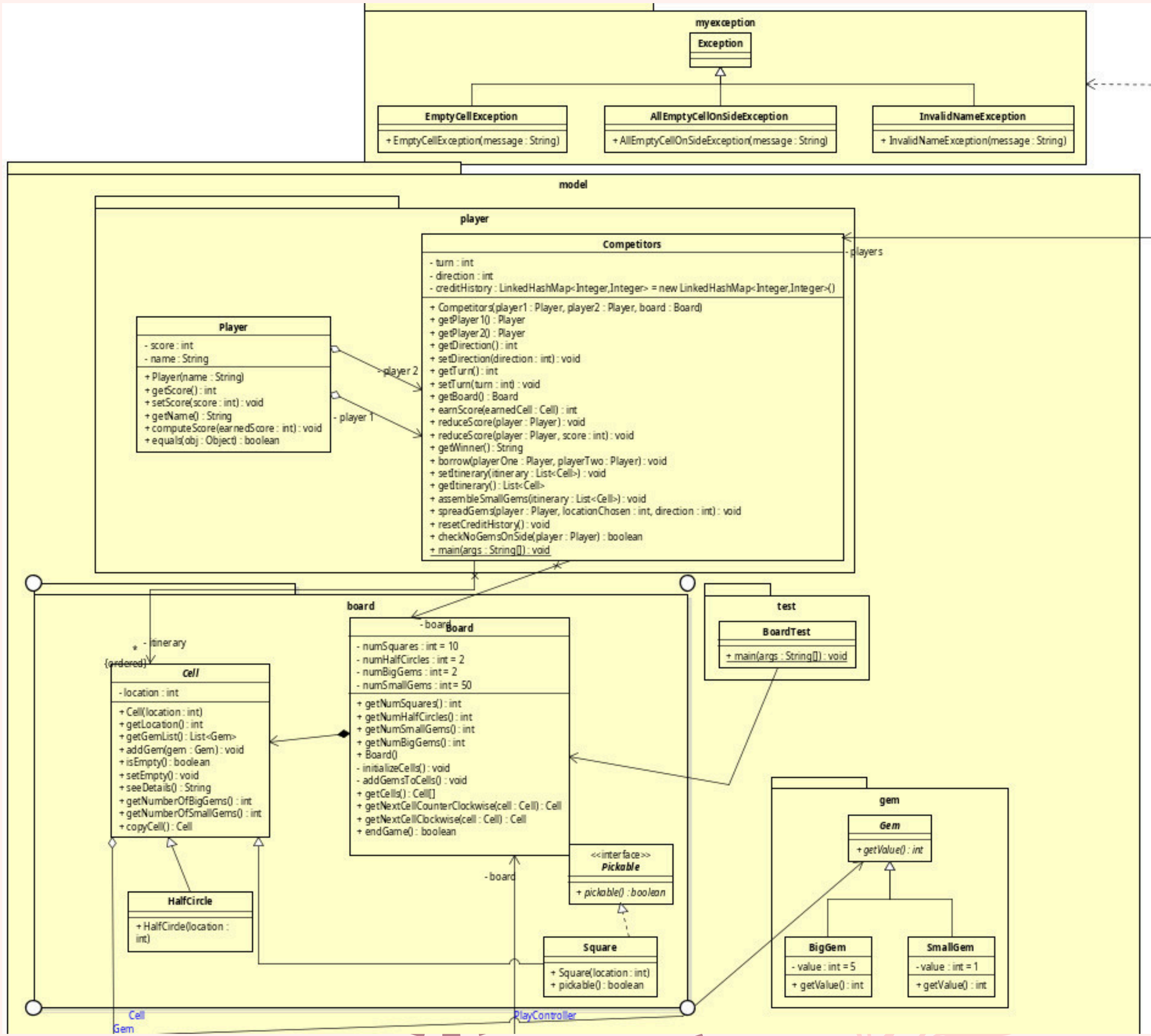


Figure 8. 'PlayController' and 'HomeController' Classes

CLASS DIAGRAM

Class Diagram: Exception





EXPLANATION OF OOP TECHNIQUE INHERITANCE

1.

The class SmallGem and BigGem inherit from the abstract class Gem

2.

The class HalfCircle and Square inherit from the class Cell

EXPLANATION OF OOP TECHNIQUE

ENCAPSULATION

1.

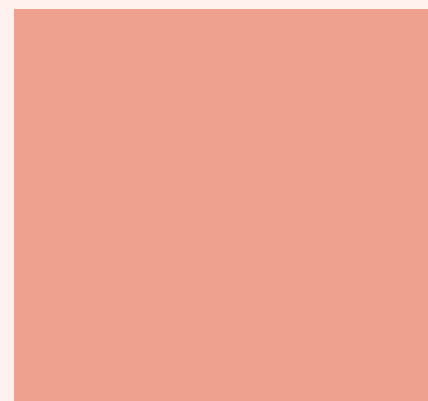
Classes in package models (like player, board and gem) use encapsulation to protect their private attributes

2.

Each class has its fields (attributes) and methods (functions) bundled together

EXPLANATION OF OOP TECHNIQUE ABSTRACTION

Pickable interface to distinguish the pickable cell and non-pickable one (for example, square and the half circle).



EXPLANATION OF OOP TECHNIQUE ASSOCIATION

1. The class Competitors is associated with the Players in a one-to-many relationship
2. Board are also associated with Cell in one-to-many relationship

EXPLANATION OF OOP TECHNIQUE

AGGREGATION

1.

Cell aggregates Gem

2.

Competitors class is
aggregation of player 1 and
player 2

COMPOSITION

The board is composed
of Cell objects

EXPLANATION OF OOP TECHNIQUE

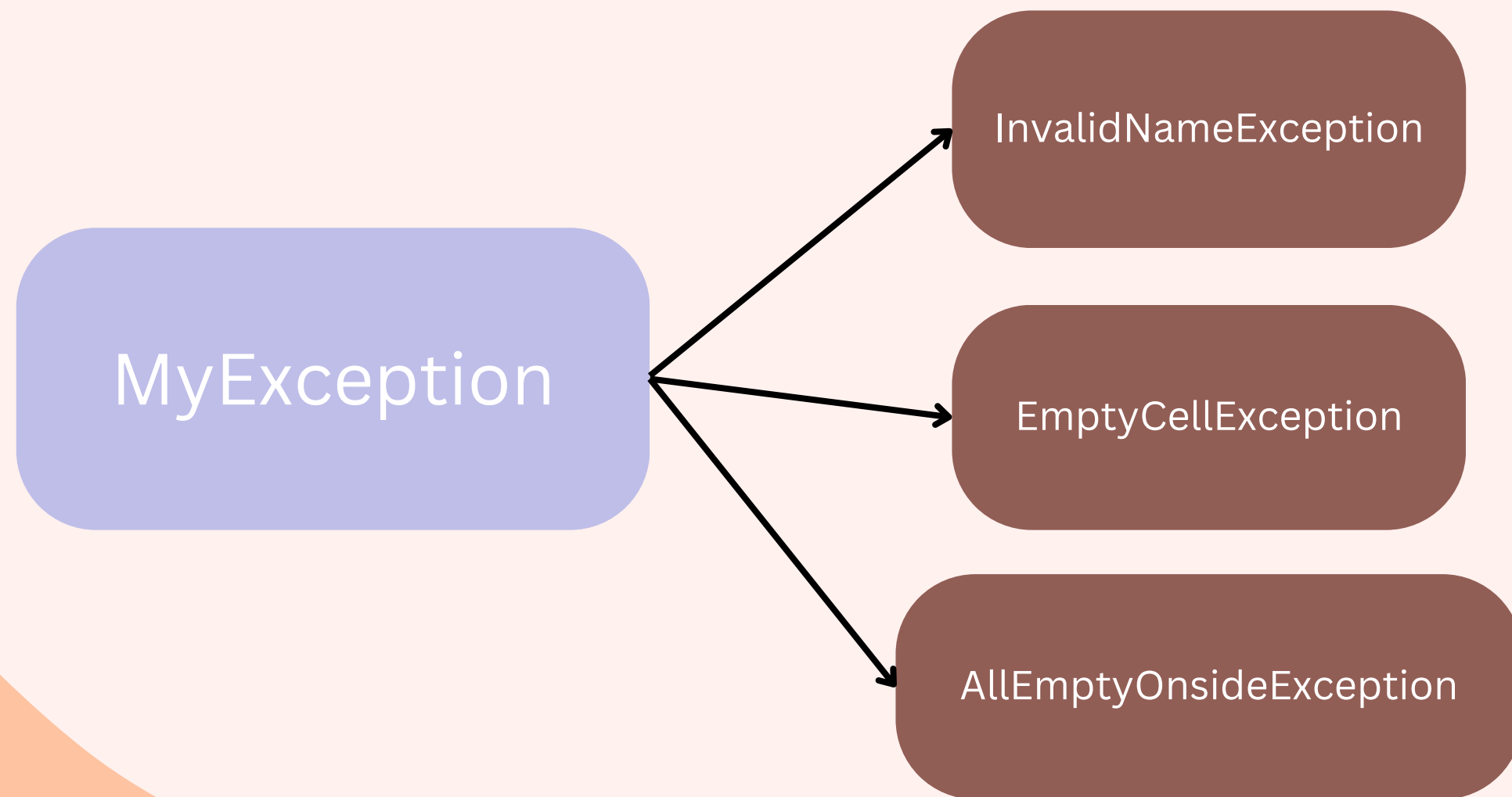
REALISATION

All squares implement
Pickable Interface

POLYMORPHISM

It can be utilized in both BigGem and SmallGem as object
of Gem class, but with distinct attributes and method

EXPLANATION OF OOP TECHNIQUE EXCEPTION



DEMO





**THANK
YOU**