

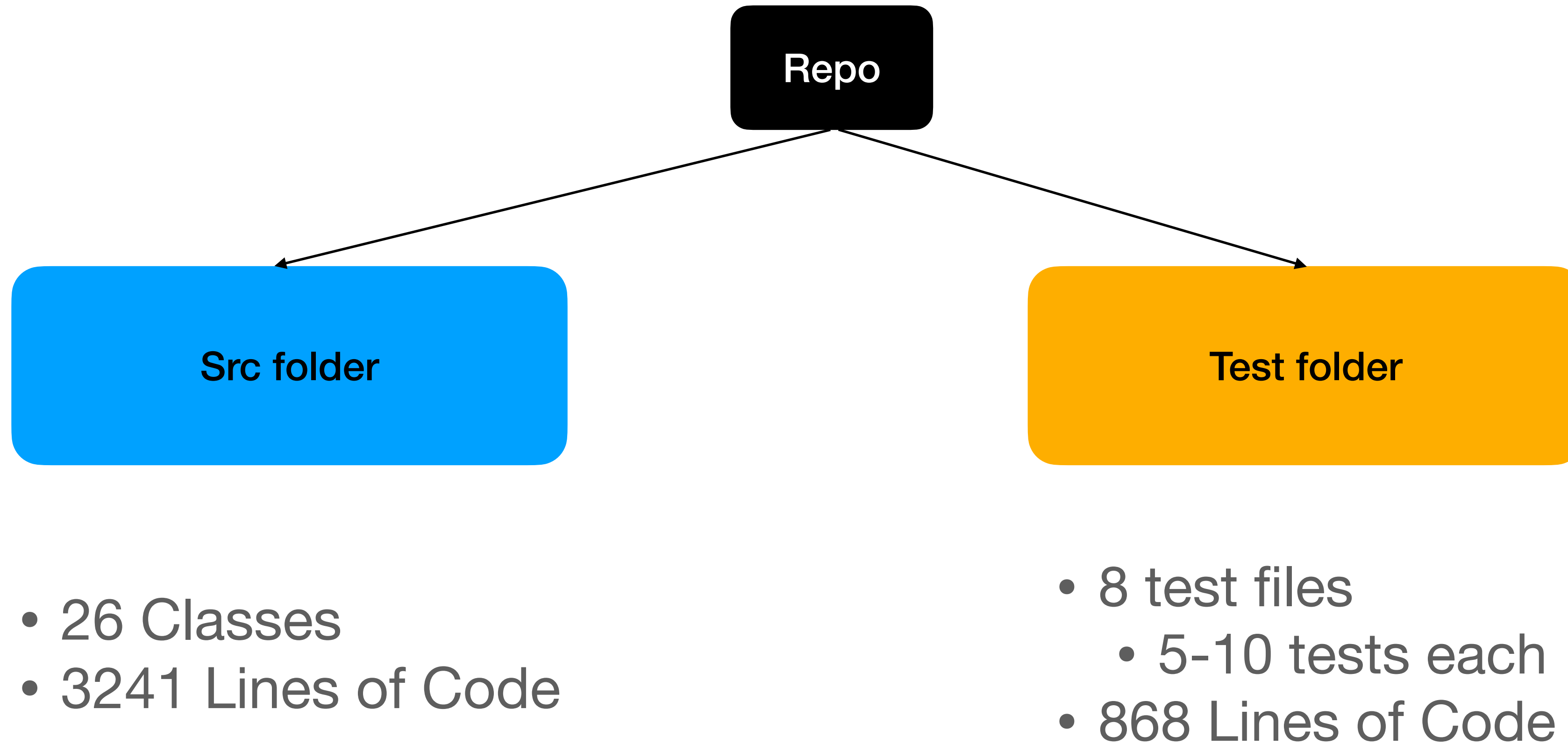
Tic Tac Toe

Software Architecture and Code Structure

Nathan Lauer, October 15, 2020

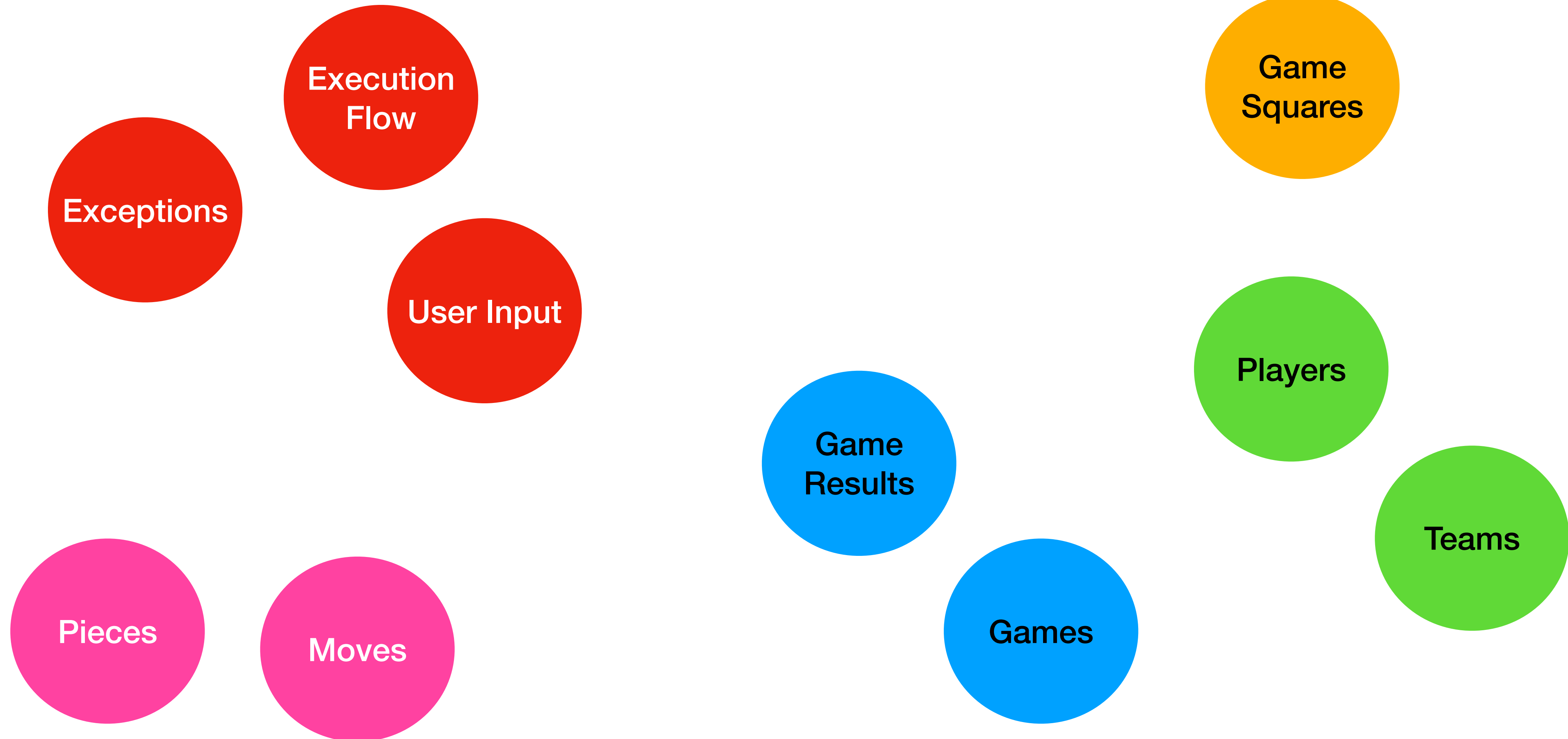
Overview

A quick look at the size of the repo



Src Folder

10 Categories of Classes



Execution Flow

PlayersAndTeamsManager

```
public class PlayerAndTeamManager {  
    private final ArrayList<Player> players;  
    private final ArrayList<Team> teams;  
  
    public List<Team> getTeams() {...}  
    public List<Player> getPlayers() {...}  
    public void configure() {  
        // Prompt user with 5 options:  
        // 0) Finished  
        // 1) Add new Team  
        // 2) Edit Team  
        // 3) Add new player  
        // 4) Edit Player  
    };  
  
    private void addNewPlayer() {...}  
    private void editPlayer(Player player) {...}  
    private void addNewTeam() {...}  
    private void editTeam(Team team) {...}  
}
```

Execution Flow

GameManager

```
public class GameManager {
    private final PlayerAndTeamManager manager;
    private final String[] availableGames = ["Tic Tac Toe", "Custom Tic Tac Toe", "Order and Chaos"];
    private final List<GameResult> results;

    public void run() {
        while(!finished) {
            this.manager.configure();
            int gameToPlay = this.selectGameToPlay();
            switch(gameToPlay) {...}
            // prompt user if finished: no - done, yes: allow reconfiguration of players and teams
        }
        this.printResults();
    }

    private int selectGameToPlay() {...}
    private void playTTT() { this.results.add(new TTT(...).play()) }
    private void playCustomTTT() { this.results.add(new CustomTTT(...).play()) }
    private void playOrderAndChaos() { this.results.add(new OrderAndChaos(...).play()) }
    private void printResults() { ... }
}
```

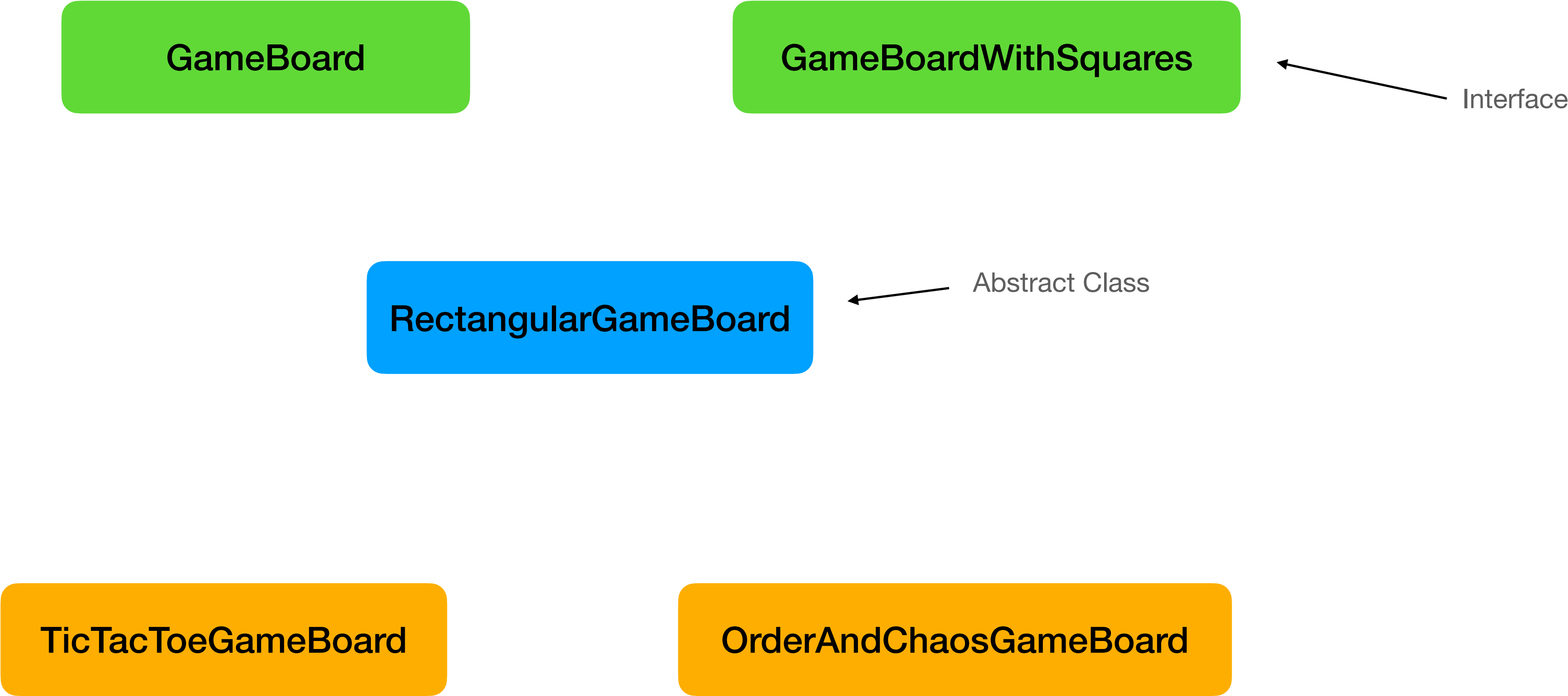
Boards and Game Squares

Game Square

- Game Square class represents a location on some Board
 - Location is given by int (row, col)
- Every game square may have multiple pieces
 - Some games, like chess/checkers have one piece per square.
 - Others, like mahjong, have numerous pieces per square
- public boolean isOccupied(): true if at least one piece on this square
- public boolean playerHasPieceOnSquare(Player): true if a piece on this square is owned by passed in player
- public void placePiece()
- Public GameSquare clone()
 - Allows other classes to control access to a GameSquare
 - For example: if a board class exposes many GameSquares, may not want clients to be able to change GameSquares.
 - Thus, a board's getGameSqaures() method can return a list of clones.

Boards and Game Squares

Game Boards Hierarchy



Boards and Game Squares

Game Boards Hierarchy

- Abstract class RectangularGameBoard
 - Controls access to game squares
 - Checks bounds
 - Executes moves
 - Drawable
- Tic Tac Toe Game Board
 - `super(3,3)` or `super(n,n)` for custom TTT
 - encodes winning conditions for a player
 - checks if game is tied
 - `isMoveLegal()`
- Order and Chaos: the “same” as TicTacToeGameBoard, but winning/tied/legal moves conditions are different

Games and GameResults

Games

```
public interface Game {  
    public void printWelcomeMessage();  
    public GameResult play();  
}
```

Games and GameResults

Games

```
public class TicTacToe implement Game {
    private Player xPlayer;
    private Player oPlayer;
    private TicTacToeGameBoard board;

    public void printWelcomeMessage();
    public GameResultPlay() {
        // Get user move
        // execute move on board
        // prompt board for tie or winning condition
        // return GameResult when finished
    }
}
```

```
public class OrderAndChaos implement Game {
    private Player chaos;
    private Player order;
    private OrderAndChaosGameBoard board;

    public void printWelcomeMessage();
    public GameResultPlay() {
        // Get user move
        // execute move on board
        // prompt board for tie or winning condition
        // return GameResult when finished
    }

    private class OrderAndChaosInput() {...}
}
```

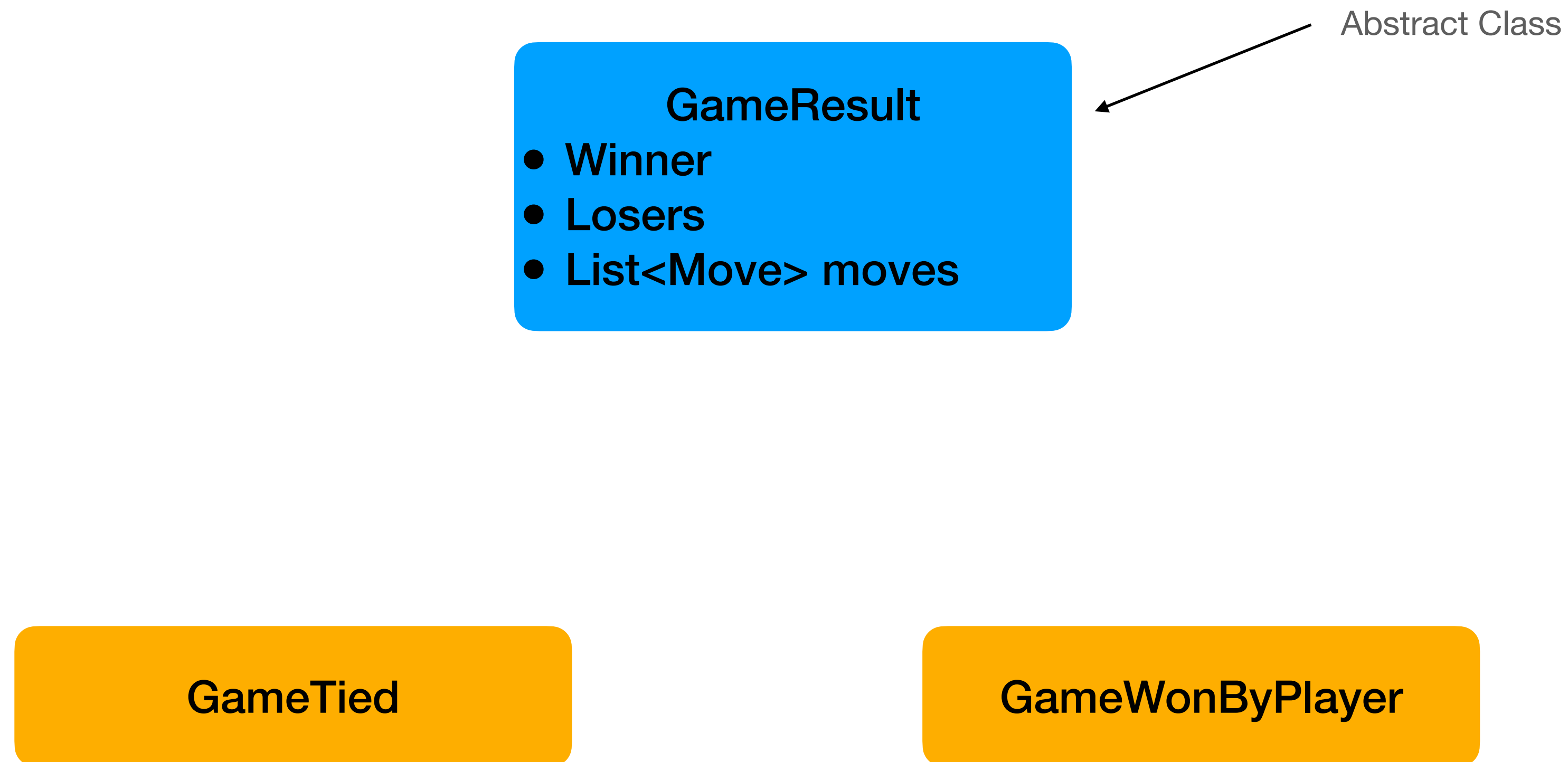
Games and Game Results

Games

- Flow of Tic Tac Toe and Order and Chaos similar
- Separate classes because different semantics in:
 - Messages to user
 - Move types
- In retrospect....
 - Probably could have abstracted these two classes into TurnBasedGame

Games and GameResults

GameResults Hierarchy



Pieces and Moves

Types of pieces and moves

- Class Piece
- Class ColoredPiece: specific type of piece, where identification is color.
 - Applies to games like checkers, connect4, and also Tic Tac Toe!
- Abstract Class Move
- Abstract Class MoveForGameBoardWithSquares
 - placePieceOnSquare
 - movePieceFromSquare
 - removePieceFromSquare, etc.
- Class PlaceNewPieceOnBoardMove