# **Tournament**

The Tournament object manages multiple games, potentially involving multiple players, where the games might be running sequentially or in parallel

## Knows:

Players

Number of games

Game outcomes

Total scores

Type of tournament e.g. knock-out or pool

## Knows how to:

Create and execute a sequence of games, managing black/white assignment to players

Run games in sequence or in parallel

# **Game**

Think of the Game object as a kind of refereree/game manager, who asks the players for their decisions and then implements them, checking that rules are being followed.

## Knows:

Players (and their colours)

Board

## Knows how to:

Set up the board

Set up players, including automated players

Manage turns, including inability to move  
 - request instruction from a player  
 - execute the move on the board  
 - re-draw the board/squares/pieces as needed, with suitable delays

Keep score of game

Identify when game is over

Cancel/restart the game

Record a list of moves

# **Board**

## Knows:

Its origin coordinates on the canvas

Dimensions (number of squares on a side, and square size)

The list of Squares

## Knows how to:

Draw itself, delegating to Squares

Count squares that are empty, black or white

Retrieve a square by square number

Identify the square covered by given canvas location

Identify list of valid placements for a colour

Implement a move (by flipping necessary pieces) returning list of changed squares

Create a new copy of board, with new pieces from a hypothetical move.

# **Square**

## Knows:

The Piece it contains (if any)

## Knows how to:

Draw itself, incl. delegating to Piece if occupied

Accept a piece

Find all its neighbours, allowing for edges

Find any neighbours of a specified colour

# **Piece**

## Knows:

Which side is up (i.e. its current colour)

Square it is on

## Knows how to:

Draw itself relative to the square

Flip

# **Player**

## Knows:

Board

## Knows how to:

Advise if cannot move

Decide on move

# **Human player (controller)**

Allow movement around the board, highlighting current location

Accept location input

# **Automated player**

Evaluate a potential move