
RYATTA GROUP
Captain's Mistress Workshop
David Andrews
Ruby Hack Night
July 29, 2015

Captain's Mistress Workshop

During his long sea voyages, Captain Cook would retire to his cabin for extended periods. The crew used to joke that he had a mistress hidden away there. They soon discovered that the Captain had been playing a game with the ship's scientist.

The game came to be known as
'The Captain's Mistress'.

RYATTA GROUP

Rules of Captain's Mistress

1. The game consists of balls and a rack.
2. There are 2 coloured sets of 21 balls each, coloured black and white.
3. The rack has 7 channels (columns) and 6 rows.
4. The rack is oriented vertically so that the balls create 7 stacks.
5. Two players take turns dropping balls into the channels.
6. A ball falls until it lands on top of the existing stack, or the bottom of the rack.
7. Players cannot drop balls into channels that are full.
8. The winner is the first player to create a line of four balls in any direction.

RYATTA GROUP

Do you recognize it?

Masters Traditional Games

RYATTA GROUP

WORLD GAMES

ALL THE FUN IN THE WORLD

Connect Four

VERTICAL CHECKERS GAME

7-Adult

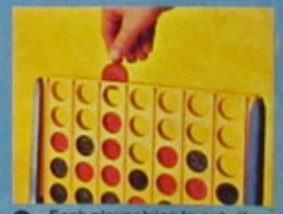
2- Players

A MB Game

Manufactured under licence with Milton Bradley Company U.S.A. by

WORLD GAMES COMPANY LTD. AUCKLAND, N.Z.

1 Players alternate dropping checkers down slots.



2 Each player tries to outwit his opponent.

3 Four in a row wins: across, up-and-down, or diagonally.

Our needs from the very high-level

- The Game and 2 Players:
- The Game entity
  - manages the game board,
  - enforces the rules (including flow of play), and
  - game state (including winning conditions).
- The Player entity
  - answers the question: what is your next move?
- The Player entity will want to ask questions about the game state in order to make decisions about the next move. How do Players ask questions? Through an API!

Divide and conquer

Split up into three teams:

1. Strategy Team 1
2. Strategy Team 2
3. Game Core Team

RYATTA GROUP

Plan for development

Phase 1

- Game Core Team - use TDD to develop the game components
- Strategy Team 1 & 2 - research play strategies

Phase 2

- All - work together to define the API

Phase 3

- Game Core Team - build the API
- Strategy Team 1 & 2 – implement several strategies using the API

Phase 4

- All - pit players against each other and play!

RYATTA GROUP

Useful resources

Game Core Team

- https://www.pivotaltracker.com/n/projects/1396446

Strategy Teams

- http://gizmodo.com/heres-how-to-win-every-time-at-connect-four-1474572099
- https://en.wikipedia.org/wiki/Connect_Four

RYATTA GROUP

Discussion of the solution – the rack

- The team decided the best datatype for storing the rack is an array of channels, each channel storing the ball colour in order from bottom to top
- This datatype has the benefits that Array#push can be used to place new balls, and balls naturally "fall" to their correct locations in the stack. Also, checking the rack for full channels and fullness overall is easy using Array#length
- This datatype has the drawback that it requires manipulation to output. We felt it was preferable to have all of this manipulation in one place rather than spreading checks and tests required by other datatypes across the code.

RYATTA GROUP

Discussion of the solution – printing the rack

- To print the rack we need each row of the rack in order from top to bottom. This requires a bit of manipulation.
- The rack is a compressed (i.e. empty rack spaces are not stored) representation of the channel contents bottom-to-top
- To get rows sorted top-to-bottom we have to do three things:
- 1. expand the rack (i.e. insert the empty rack spaces)
- 2. Array#transpose the contents turning channels into rows
- 3. Reverse the row order so it is top-to-bottom

Discussion of the solution – win detection

- There are four patterns of four balls that need detection: horizontal (in a row), vertical (in a channel), diagonal right (top-left to bottom-right), diagonal left
- The solution uses array manipulation to create four "views" of the rack, each optimal for examining contents in these four orientations
- Balls that cannot participate in a win for a specific orientation are discarded to avoid false positives
- Detection is as simple as finding a continuous string of four balls of the same colour

Discussion of the solution - automatons

- Our computer players, or automatons, were more difficult to program than we had anticipated
- We created one "super easy" automaton that randomly picks an open channel, dubbed "George"
- With a small amount of effort, we should be able to repurpose the win detection code into an API useful for examining the rack for opportunities and threats
- A simple next step would be to build a "three ball" detector and have the automaton chase the opportunity (or block the threat)
- In the meantime, it's fun to play against George, or pit him against himself

RYATTA GROUP

Discussion of the solution – the Game

- The Game entity grew by leaps and bounds near the end of our session
- It has three problems:

1. Lack of focus
2. Many long and complicated methods
3. Insufficient test coverage

RYATTA GROUP