Board | Cell | Boat | Player | Game

Stored in database:

Game - game history
- connected to player

Player - user login info
- stats

does the Game need to store its Board, Boats and Cells?

not boats because
they would all be shot up anyway;
not cells,
maybe the board? definitely not necessary.
We want stats about the Game,
that will be in the Game object's attributes

we don't need to store anything in the database yet,
except maybe player info and stats.
Just login info to start.
Even that much can be further down the line
— you can still play the game without logging in

M C V

we need to change out the views
so it will display on HTML via erb
instead of on the command line.

We need models for each thing,
we don't need to Save anything yet.
We don't need migrations?
Maybe we do because Rails
wants them to make models.

Game Flow

Game

Player

Board

Cell

Boat

what does the current
codebase look like?

models
Game - Controller?
Player
Board
Cell
Boat

How does this fit into
the structure of a Rails app?
Does it?

Rails

M	C	V
Player	Game	Views
Boat		
Board		
Cell		

Player Score

# games won
shots fired
hits
hit percentage
leaderboard position
~ different leaderboards

Refactor

Game

attributes

- lengths[]
- players[]
- won?
- winner
- score

behaviors

- gen. new player (hum or com)
- setup both players
  - calls new-player
- get score
  - fill out stats
  - store in game's score arry
- setup
  - place fleet for ea. player
- place fleet (random)
  - get a random cell
    - check if boat will fit
      horiz. or vert.