A Strategy Interface

Version 7.1.0.1

Matthias Felleisen

October 20, 2018

A strategy object knows who plays against whom and can answer questions concerning every interaction during the game.

For the Santorini game, a strategy is an instance of the following class:

```
(class/c

 (init-field
   ; the player that owns the strategy:
   (player string?)
   ; the opponent:
   (other string?))

 (initialization
   ; given a list of occupied places on the board,
   ; this strategy picks the next place for the player's worker
   (->m placements/c place/c))
 (take-turn
   ; given a board, which represents the entire game state,
   ; pick the next action (give up, win, move and build)
   (->m board? action?))
 (dead?
   ; given a board, does this strategy consider the
   ; player's situation a dead end?
   (->m board? boolean?))
 (safe-for
   ; is the given action safe for the specified n rounds
   ; in the current game state?
   ; See Assignment 8 for details.
   (->m action? board? natural-number/c boolean?)))
```

A list of occupied places, `placements/c`, specifies where which player has already placed a worker:

```
(define placements/c
  (listof
   (list/c
    string?   ; who placed a worker
    in-range? ; at x
    in-range? ; at y on the initial board)))
```