

The Board Interface

Version 7.0.0.20

Matthias Felleisen

October 10, 2018

Representing the Key Physical Game Piece

The board represents a physical game object, with some minimal pieces of functionality suitable for playing Santorini.

```
(define DIM 5)
(define in-range? (integer-in 0 DIM))
(define init/c (list/c worker? in-range? in-range?))

(init
 ; create the board and place the four worker?s on it
 (->i ((t1 init/c) (t2 init/c) (t3 init/c) (t4 init/c))
  #:pre/name "must be at distinct places"
  (let ([L (locations (list t1 t2 t3 t4))])
    (= (set-count (apply set L)) (length L)))
  (result board?)))

(named-workers
 ; retrieve the workers of the given name on this board
 (-> board? string?
  (list/c worker? worker?)))

(on?
 ; is this the name of a player on this board?
 (-> board?
  (-> string? boolean?)))

(on-board?
 ; does this worker exist on this board?
 (-> board?
  (-> worker? boolean?)))
```

```

(all-directions-to-neighbors
 ; directions to a neighboring field from this worker
 ; GUARANTEE staying put is not a part of the directions
 (-> board? worker?
  (listof (list/c east-west/c north-south/c))))

(stay-on-board?
 ; does this worker remain on board if it moves
 ; ASSUME worker is on board
 (-> board? worker? east-west/c north-south/c
  boolean?))

(height-of
 ; the height of the building where the worker is located
 ; or looking at on this board
 (->i ([b board?] [t worker?])
  ([e-w east-west/c] [n-s north-south/c])
  #:pre/name "on board"
  (or (unsupplied-arg? e-w) (stay-on-board? b t e-w n-s))
  (result natural-number/c)))

(location-free-of-worker?
 ; there is no worker on the specified field on this board
 (->i ((b board?) (t (b) (and/c worker? (on-board? b))))
  (e-w east-west/c) (n-s north-south/c))
  #:pre/name "remains on board" (stay-on-board? b t e-w n-s)
  (result boolean?))

(move
 ; move the worker one step in the given direction
 (->i ((b board?) (t (b) (and/c worker? (on-board? b))))
  (e-w east-west/c) (n-s north-south/c))
  #:pre/name "remains on board" (stay-on-board? b t e-w n-s)
  #:pre/name "a short bldg"
  (< (height-of b t e-w n-s) MAX-HEIGHT)
  (result board?))

(build
 ; add a level to the building that is in the specified direction
 (->i ((b board?) (t (b) (and/c worker? (on-board? b))))
  (e-w east-west/c) (n-s north-south/c))
  #:pre/name "remains on board" (stay-on-board? b t e-w n-s)
  #:pre/name "a short bldg"
  (< (height-of b t e-w n-s) MAX-HEIGHT)
  (result board?))

```

The board interface assumes data representations and certain pieces of functionality from these modules:

- workers, which have a unique identity (first, second) and are associated with a player
`worker? : Any -> Boolean`
- cardinal directions, which consists of `east-west/c` and `north-south/c` directions
- buildings, which have one relevant property here `MAX-HEIGHT`