

game

GAME+

feature -- Constructors

make_from_board(new_board:BOARD)

ensure

board_set: board ~ new_board

feature -- Commands

move_left, move_right (r,c:INTEGER)

require

from_slot_valid_row, from_slot_valid_column,

middle_slot_valid_column,

to_slot_valid_column,

middle_slot_occupied,

to_slot_unoccupied

ensure

slots_properly_set

other_slots_unchanged: board.matches_slots_except(board, r,r,c,c)

move_up, move_down(r,c:INTEGER)

require

from_slot_valid_column,

from_slot_valid_row,

middle_slot_valid_row,

to_slot_valid_row,

from_slot_occupied,

middle_slot_occupied,

to_slot_unoccupied

ensure

slots_properly_set,

other_slots_unchanged:

board.matches_slots_except (board, r, (r-2), c, c)

feature -- Board

bta: BOARD_TEMPLATES_ACCESS

board: BOARD

board

BOARD+

feature -- Auxiliary Commands

set_status(r, c: INTEGER; status: SLOT_STATUS)

require

valid_row: is_valid_row(r)

valid_column: is_valid_column(c)

ensure

slot_set: imp.item(r,c).is_equal(status)

slots_not_in_range_unchanged: matches_slots_except (old current, r, r, c, c)

set_statuses(r1, r2, c1, c2: INTEGER; status: SLOT_STATUS)

require

valid_rows:is_valid_row(r1) and is_valid_row(r2)

valid_columns: is_valid_column(c1) and is_valid_column(c2)

valid_row_range:(r2-r1) ≥ 0

valid_column_range: (c2-c1) ≥ 0

ensure

correct_result: Result = true ⇒

∀i∀j : 1 ≤ i ≤ number_of_rows, 1 ≤ j ≤ number_of_columns:

(i ≥ r1 ∨ i ≤ r2 ∨ j ≥ c1 ∨ j ≤ c2) ⇒ status_of(i,j) ~ other.status_of(i,j)

slots_not_in_range_unchanged:

matches_slots_except(old current, r1, r2, c1, c2)

feature -- Auxiliary Queries

matches_slots_except

(other: BOARD; r1, r2, c1, c2: INTEGER): BOOLEAN

require

consistent_row_numbers: other.number_of_rows = number_of_rows

consistent_column_numbers: other.number_of_columns =

number_of_columns

valid_rows: (r1 ≤ number_of_rows) ∧ (r1 ≥ 1) ∧ (r2 ≥ 1) ∧ (r2 ≤

number_of_rows)

valid_columns: (c1 ≤ number_of_columns) ∧ (c1 ≥ 1) ∧ (c2 ≥ 1) ∧ (c2 ≤

number_of_columns)

valid_row_range: r1 ≤ r2

valid_column_range: c1 ≤ c2

ensure

correct_result: (Result = true) ⇒

∀i∀j : 1 ≤ i ≤ number_of_rows, 1 ≤ j ≤ number_of_columns:

(i < r1 ∨ i > r2 ∨ j < c1 ∨ j > c2) ⇒ status_of(i,j) ~ other.status_of(i,j)

feature -- Implementation

ssa: SLOT_STATUS_ACCESS

bta: BOARD_TEMPLATE_ACCESS

imp: ARRAY2[SLOT_STATUS]

game+

*
PLAYER+
GOOD_PLAYER+
BAD_PLAYER

templates*

access*

status_of+
imp:ARRAY2[...]

bta+

ssa+

slot

*
SLOT_STATUS*
AVAILABLE_SLOT+
UNAVAILABLE_SLOT+
UNOCCUPIED_SLOT+
OCCUPIED_SLOT

SLOT_STATUS_ACCESS

occupied_slot+

unoccupied_slot+

unavailable_slot+

bta

board+

templates+

BOARD_TEMPLATES_ACCESS

+
BOARD_TEMPLATES

tests

+
STUDENT_PLAYER_TESTS+
PRECONDITION_TESTS

bta+

ssa+