



states

SIMODYSSEY +

feature -- Attributes

galaxy: GRID
explorer: EXPLORER
is_test_game: BOOLEAN

feature -- Commands

abort
require game_in_session
ensure \neg game_in_session

new_game(th: INTEGER ; is_test: BOOLEAN)
require valid threshold: $(1 \leq th \wedge th \leq 101)$
 $\wedge (\neg \text{game_in_session})$
ensure game_in_session \wedge is_test_game = is_test

move_explorer(d: COORDINATE)
require game_in_session $\wedge (\neg \text{sector_in_direction_is_full}(d)) \wedge (\neg \text{is_explorer_landed})$
ensure if_not_lost_the_explorer_is_in_new_position: $(\text{is_explorer_alive}) \Rightarrow$
galaxy.at $((\text{old_explorer_coordinate} + d).\text{wrap_coordinate}((\text{old_explorer_coordinate} + d),$
[1, 1], [shared_info.number_rows, shared_info.number_columns])).has(explorer)

wormhole
require game_in_session $\wedge (\neg \text{is_explorer_landed}) \wedge \text{is_explorer_with_wormhole}$
ensure If_not_lost_the_explorer_is_in_new_position: $\text{explorer.is_alive} \Rightarrow$
galaxy.at(explorer_coordinate).has(explorer)
 \wedge if_explorer_goes_blackhole_he_is_dead: $\text{galaxy.at}(\text{explorer_coordinate}).\text{has_blackhole} \Rightarrow$
explorer.is_dead

land_explorer
require game_in_session $\wedge (\neg \text{is_explorer_landed}) \wedge \text{e_sector_has_yellow_dwarf}$
 $\wedge \text{e_sector_has_planets} \wedge \text{e_sector_has_unvisited_attached_planets}$
ensure
is_explorer_landed

liftoff
require game_in_session $\wedge \text{is_explorer_landed}$
ensure $(\neg \text{is_explorer_landed}) \wedge \text{game_in_session}$

pass
require game_in_session
ensure game_in_session

feature -- Queries

game_in_session: BOOLEAN

sector_in_direction_is_full(d: COORDINATE): BOOLEAN
require d_is_a_direction: d.is_direction

feature -- Explorer Interface

is_explorer_with_wormhole: BOOLEAN
is_explorer_landed: BOOLEAN
explorer_coordinate: COORDINATE
e_sector_has_yellow_dwarf: BOOLEAN
e_sector_has_planets: BOOLEAN
e_sector_has_unvisited_attached_planets: BOOLEAN

STATE *

feature -- Attributes

model: SIMODYSSEY
next_state: STATE

feature -- Controller command / Queries

abort*, land*, liftoff*, move*, pass*, play*, status*, test*, wormhole*

invariant

if_next_state_is_main_menu_state_then_game_is_not_in_session:
attached {MAIN_MENU_STATE} next_state \Rightarrow
 $(\neg \text{model.game_in_session})$
if_next_state_is_play_state_then_game_is_in_session:
attached {PLAY_STATE} next_state $\Rightarrow (\text{model.game_in_session} \wedge$
 $\neg \text{model.is_explorer_landed})$
if_next_state_is_landed_state_then_game_is_in_session:
attached {LANDED_STATE} next_state $\Rightarrow (\text{model.game_in_session} \wedge$
 $\text{model.is_explorer_landed})$

model: ..