

PROJECTILE_UPDATE*

feature -- attributes

proj_id: INTEGER
model: GAME
projectile_action_output: STRING

feature

make_proj_update+
-- make feature for the class
make_model
-- creates the client supplier link from model to this class

feature -- Commands

incr_proj_id+
-- increments the current projectile id value by one
decr_proj_id+
-- increments the current projectile id value by one
update_location(p: PROJECTILE)+
-- Updates the projectile `p` location
-- It can either be "out of board" or at a location "[X,Y]"
update_location_list(p_list: LINKED_LIST[PROJECTILE])
-- Updates the location of a list of projectiles
-- Used when multiple projectiles of the same type are spawned
set_projectile_action(output: STRING)
-- Appends `output` to a string called `projectile_action_output`
-- It is used to append output after a projectile moves or is spawned

feature -- Queries

get_projectile(identity: INTEGER; projectile_list: LINKED_LIST[PROJECTILE]): PROJECTILE
-- Returns a projectile based on the identity passed in
-- and the projectile list to look through (enemy or friendly projectile list)

require

identity_exists: $\exists \text{identity} : \text{identity} \in \text{projectile_list}$

ensure

proj_apart_of_list: $\text{projectile} \in \text{projectile_list}$



FRIENDLY_PROJECTILE_UPDATE+

feature -- Attributes

projectiles: LIST[PROJECTILE]
move_output: STRING

feature -- Commands

create_projectile: LINKED_LIST[PROJECTILE]
-- Spawns a new projectile and appends it to a linked list that will be used for output to the game board
-- The spawn also appends the new projectile to the `projectiles` list

move_proj

-- moves all existing projectiles in the list `projectiles`

require

list_not_empty: $\exists x : x \in \text{projectiles}$