

ETF_MODEL+

```
feature -- supplier attributes

    starfighter: STARFIGHTER
    total_score: SCORE
    game_state: GAME_STATE
    print_state: PRINT_STATE

feature -- collections
    enemy_collection: HASH_TABLE[ENEMY, INTEGER]
    enemy_projectile_collection: HASH_TABLE[ENEMY_PROJECTILE, INTEGER]
    friendly_projectile_collection: HASH_TABLE[FRIENDLY_PROJECTILE, INTEGER]
    states: ITERABLE[STATE]

feature-- ids
    projectile_id, enemy_id: INTEGER
    score: INTEGER

feature -- booleans
    in_game, in_setup_mode, in_debug_mode: BOOLEAN

feature -- game commands
    play(row: INTEGER_32 ; column: INTEGER_32 ; g_threshold: INTEGER_32 ; f_threshold: INTEGER_32 ;
        c_threshold: INTEGER_32 ; i_threshold: INTEGER_32 ; p_threshold:
INTEGER_32)
        -- Initially used to enter setup_mode and to cache the threshold value

    play_game
        -- used when in_game state

pass
    -- SF passes

fire
    do
        states[weapon_choice].fire
        -- fires based on weapon choice
    end

move(row: INTEGER; column: INTEGER)
    -- SF moves

special
    do
        states[power_choice].special
        -- use special based on power selection
    end

abort
    -- game aborts

feature -- enemy releated features
report_enemies
    -- reports all enemies that are still `on_board`

spawn_enemy(row: INTEGER; column: INTEGER)
    -- natural enemy spawns at location [row, column]

preemptive_action(str: STRING)
    -- phase 5
    -- preemptive action of all enemies `on_board`
    -- oldest to newest

action
    -- phase 5
    -- Enemy action of all enemies `on board`

feature -- queries
enemy_presence [row: INTEGER; column: INTEGER]: INTEGER
    -- returns an `id` of an on_board enemy at location [row, column]

enemy_projectile_presence [row: INTEGER; column: INTEGER]: INTEGER
    -- returns an `id` of an on_board enemy projectile at location [row, column]

friendly_projectile_presence [row: INTEGER; column: INTEGER]: INTEGER
    -- returns an `id` of an on_board friendly projectile at location [row, column]

feature -- projectile related
report_projectiles
    -- reports all projectiles that are `on board`

invariant
    contradiction:
        (in_game and in_setup_mode) = false
```

starfighter+

STARFIGHTER+

```
feature -- sf_attributes
    total_health, total_energy, total_move, total_move_cost, total_vision: INTEGER
    current_health, current_energy, total_armour, total_projectile_cost, total_projectile_damage: INTEGER

feature -- SF identity attributes
    id: INTEGER
    initial_pos, old_pos, pos : TUPLE[row: INTEGER; column: INTEGER]

feature -- model access
    model_access: ETF_MODEL_ACCESS

feature -- queries
    seen_by_sf (row: INTEGER; column: INTEGER)
        -- can starfighter the position [row, column]

feature -- commands

    setup
        -- loops for the current equipment selection and steup starfighter initially

    add_health (h: INTEGER)
        -- adds h to current health

    add_energy (e: INTEGER)
        -- adds e to current energy

    subtract_health (h: INTEGER)
        -- subtracts h from current health

    subtract_energy (e: INTEGER)
        -- subtract e from current energy

    apply_health_regen
        -- health regeneration of starfighter

    apply_energy_regen      name_of_attribute
        -- energy regeeration of starfighter

    set_pos[row: INTEGER; column: INTEGER ]
        -- updates the starfighter pos to [row, column]

    set_old_pos[row: INTEGER; column: INTEGER ]
        -- updates the starfighter old_pos to [row, column]

invariant
    destroyed:
        current_health <=0 implies (model.in_game = false)
```