Below displays the detailed view of two classes within the model cluster which are related through a client supplier relationship. The classes are shown in detailed view

model

GRID+

feature -- Attributes
  col_size : INTEGER_32
  row_size : INTEGER_32

  grunt_threshold : INTEGER_32
  fighter_threshold : INTEGER_32
  carrier_threshold : INTEGER_32
  interceptor_threshold : INTEGER_32
  pylon_threshold : INTEGER_32

  enemies : LIST [ENEMY]
  friendly_projectiles : LIST [FRIENDLY_PROJECTILE]
  enemy_projectiles : LIST [ENEMY_PROJECTILE]

  enemy_id_counter : INTEGER_32
  projectile_id_counter : INTEGER_32

  grid_char_rows : ARRAY [CHARACTER_8]
  grid_elements : ARRAY [CHARACTER_8]

  game_info : GAME_INFO

feature -- Helper Methods
  can_be_seen (starfighter: STARFIGHTER; enemy_vision: INTEGER_32; enemy_row: INTEGER_32; enemy_column: INTEGER_32) : BOOLEAN
    -- Can an Enemy see a starfighter?
  can_see (starfighter: STARFIGHTER; row: INTEGER_32; column: INTEGER_32): BOOLEAN
    -- Is this spot in the starfighters vision ?
  can_see_enemy (other_enemy: ENEMY; enemy_vision: INTEGER_32; enemy_row: INTEGER_32; enemy_column: INTEGER_32): BOOLEAN
    -- Can an Enemy see another enemy?
  is_in_bounds (row: INTEGER_32; column: INTEGER_32): BOOLEAN
    -- Is this spot in the grid?
    ensure
      in_bounds: (Result ∧ (¬ (row > row_size ∨ row < 1 ∨ column > col_size ∨ column < 1))) ∨ (¬ Result ∧ ((row > row_size ∨ row < 1 ∨ column > col_size ∨ column < 1)))

feature -- Commands
  add_enemy_projectile_fighter (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32; d: INTEGER_32; m: INTEGER_32)
    -- Add a new enemy fighter projectile to the grids enemy projectiles
    ensure
      size_incremented: enemy_projectiles.count = old enemy_projectiles.count + 1
  add_enemy_projectile_grunt (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32; d: INTEGER_32)
    -- Add a new enemy grunt projectile to the grids enemy projectiles
    ensure
      size_incremented: enemy_projectiles.count = old enemy_projectiles.count + 1
  add_enemy_projectile_pylon (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32; d: INTEGER_32)
    -- Add a new enemy pylon projectile to the grids enemy projectiles
    ensure
      size_incremented: enemy_projectiles.count = old enemy_projectiles.count + 1
  add_friendly_projectile_rocket (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32)
    -- Add a new friendly rocket projectile to the grids friendly projectiles
    ensure
      size_incremented: friendly_projectiles.count = old friendly_projectiles.count + 1
  add_friendly_projectile_snipe (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32)
    -- Add a new friendly snipe projectile to the grids friendly projectiles
    ensure
      size_incremented: friendly_projectiles.count = old friendly_projectiles.count + 1
  add_friendly_projectile_splitter (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32)
    -- Add a new friendly splitter projectile to the grids friendly projectiles
    ensure
      size_incremented: friendly_projectiles.count = old friendly_projectiles.count + 1
  add_friendly_projectile_spread (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32)
    -- Add a new friendly spread projectile to the grids friendly projectiles
    ensure
      size_incremented: friendly_projectiles.count = old friendly_projectiles.count + 1
  add_friendly_projectile_standard (row: INTEGER_32; col: INTEGER_32; i: INTEGER_32; t: INTEGER_32)
    -- Add a new friendly standard projectile to the grids friendly projectiles
    ensure
      size_incremented: friendly_projectiles.count = old friendly_projectiles.count + 1
  increment_enemy_id_counter
    -- Increment the id counter for enemies
    ensure
      counter_updated: enemy_id_counter = old enemy_id_counter + 1
  increment_projectile_id_counter
    -- Increment the id counter for projectiles
    ensure
      counter_updated: projectile_id_counter = old projectile_id_counter + 1

feature -- Turn Commands
  fire
    -- Fire a projectile on the grid from the current weapon selected in the game
    require
      is_alive: game_info.is_alive
  clear_all
    -- Clear all out of bound items that the grid is currently storing
  friendly_projectile_movements
    -- All friendly projectiles perform their specific movements for a turn
    require
      is_alive: game_info.is_alive
  enemy_projectile_movements
    -- All enemy projectiles perform their specific movements for a turn
    require
      is_alive: game_info.is_alive
  update_enemy_vision
    -- All enemies on the grid have their vision updated based on the current items on the grid
    require
      is_alive: game_info.is_alive
  enemy_preemptive_action (type: CHARACTER_8)
    -- All enemies on the grid perform their preempted actions based on the type of command that was performed
    require
      is_alive: game_info.is_alive
  enemy_action
    -- All enemies on the grid perform their actions based on whether they see the starfighter or not
    require
      is_alive: game_info.is_alive
  enemy_spawn
    -- An enemy is spawned using the random generator at a position on the last column of the board
    require
      is_alive: game_info.is_alive
  spawn_interceptor (row: INTEGER_32; column: INTEGER_32)
    -- An interceptor is spawned at the given coordinates on the grid
    require
      is_alive: game_info.is_alive

feature -- Debug Mode Output
  add_enemy_info
    -- Current enemies information on the grid is given to the game info for debug mode
  add_projectiles_info
    -- Current projectiles information on the grid is given to the game info for debug mode

enemies [...]

ENEMY*

feature -- Attributes
  row_pos: INTEGER_32
  col_pos: INTEGER_32

  curr_health: INTEGER_32
  health_regen: INTEGER_32
  health: INTEGER_32

  armour: INTEGER_32
  vision: INTEGER_32
  id: INTEGER_32

  name: STRING_8
  symbol: CHARACTER_8

  seen_by_starfighter: BOOLEAN
  can_see_starfighter: BOOLEAN

  game_info: GAME_INFO

feature {NONE}
  is_turn_over: BOOLEAN

feature -- Commands
  preemptive_action* (type: CHARACTER_8)
    -- Perform the enemy's preemptive action based on the type of command that is performed
  action_when_starfighter_is_not_seen*
    -- Perform the enemy's action for when the starfighter is not seen
    require
      is_not_seen: ¬ can_see_starfighter
  action_when_starfighter_is_seen*
    -- Perform the enemy's action for when the starfighter is seen
    require
      is_seen: can_see_starfighter
  discharge_after_death*
    -- The enemy will discharge its specific scoring item for the starfighter to pick up
    require
      is_in_bounds: game_info.grid.is_in_bounds (row_pos, col_pos)
  regenerate+
    -- Regenerate the enemy's health based on its health regen value
    ensure
      regen_applied: (old curr_health + health_regen > health ∧ curr_health = health) ∨ (old curr_health + health_regen ≤ health ∧ curr_health = old curr_health + health_regen)
  update_can_see_starfighter+
    -- Update the enemy's vision based on whether they can see the starfighter
    ensure
      value_set_correctly: can_see_starfighter = game_info.grid.can_be_seen (game_info.starfighter, vision, row_pos, col_pos)
  update_seen_by_starfighter+
    -- Update the enemy's vision based on whether they are seen by the starfighter
    ensure
      value_set_correctly: seen_by_starfighter = game_info.grid.can_see (game_info.starfighter, row_pos, col_pos)

feature -- Output Helpers
  can_see_starfighter_output+: STRING_8
    -- The enemy's can_see_starfighter attribute is returned as a string output for debugging purposes
    ensure
      correct_output: (can_see_starfighter ∧ Result ~ "T") ∨ (¬ can_see_starfighter ∧ Result ~ "F")
  seen_by_starfighter_output+: STRING_8
    -- The enemy's seen_by_starfighter attribute is returned as a string output for debugging purposes
    ensure
      correct_output: (seen_by_starfighter ∧ Result ~ "T") ∨ (¬ seen_by_starfighter ∧ Result ~ "F")

feature -- Setters
  set_row_pos+(row: INTEGER_32)
    -- Set the row position of the enemy
    ensure
      value_set_correctly: row_pos = row
  set_col_pos+(col: INTEGER_32)
    -- Set the column position of the enemy
    ensure
      value_set_correctly: col_pos = col
  set_curr_health+(i: INTEGER_32)
    -- Set the current health value of the enemy
    ensure
      value_set_correctly: curr_health = i
  set_is_turn_over+(b: BOOLEAN)
    -- Set whether the turn is over for the enemy
    ensure
      value_set_correctly: is_turn_over ≡ b