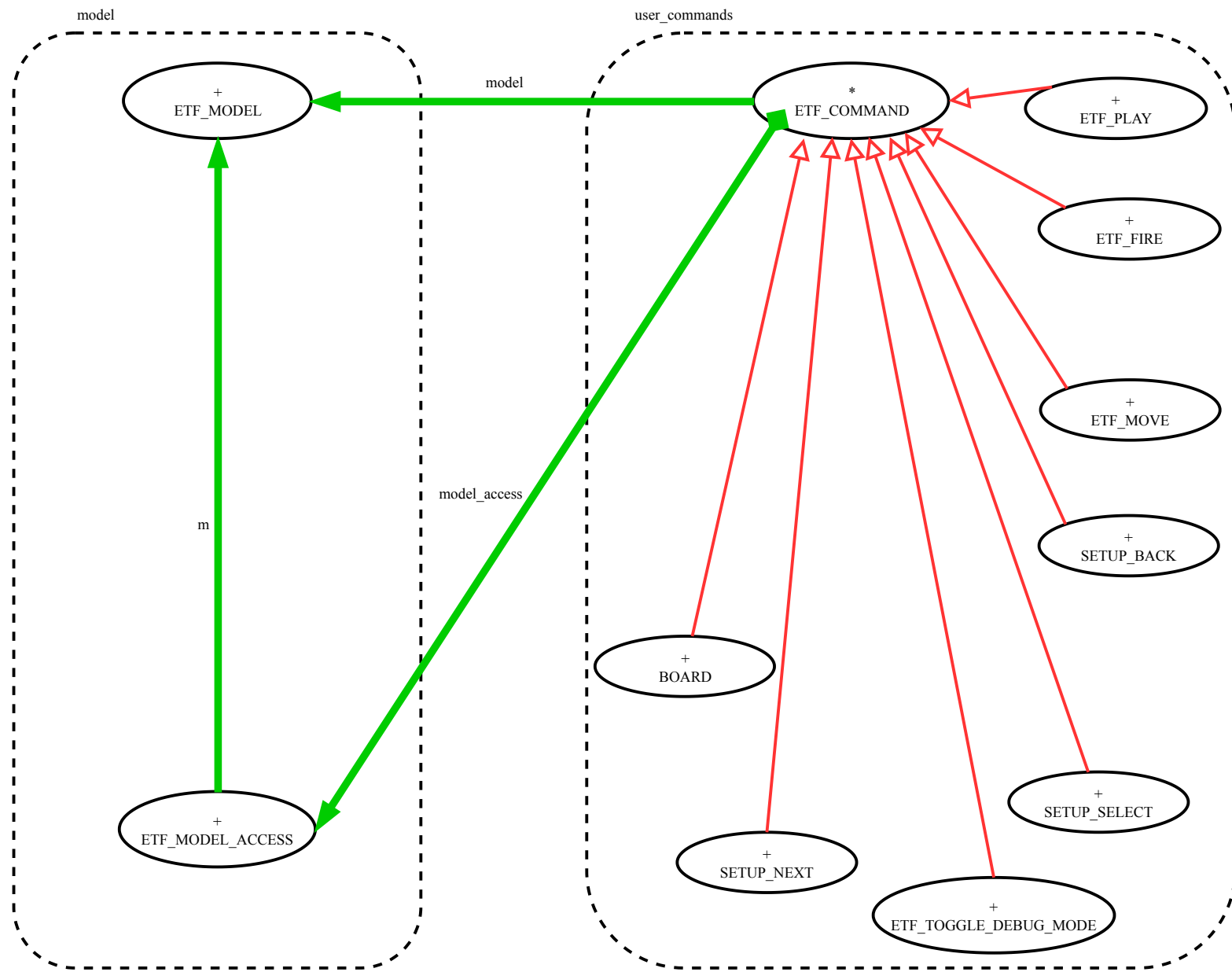# EECS 3311 - Project Bon Diagrams

Varuhn Ruthirakuhan - 215634140

Design diagram for relationship between relevant classes in
space_defender_2

model

user_commands

ETF_MODEL

model

ETF_COMMAND *

+ ETF_PLAY

+ ETF_FIRE

+ ETF_MOVE

+ SETUP_BACK

m

model_access

+ BOARD

+ SETUP_SELECT

+ ETF_MODEL_ACCESS

+ SETUP_NEXT

+ ETF_TOGGLE_DEBUG_MODE

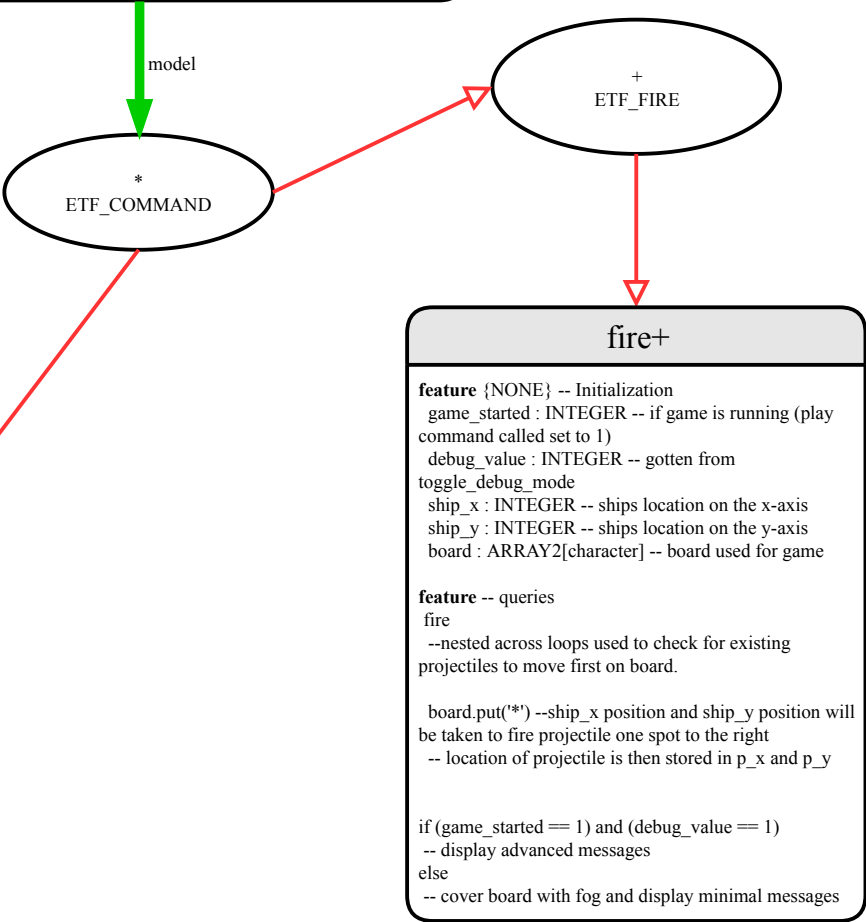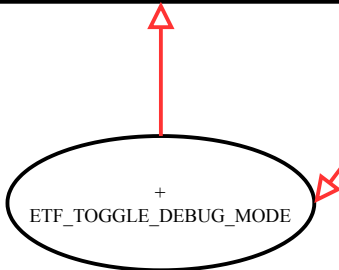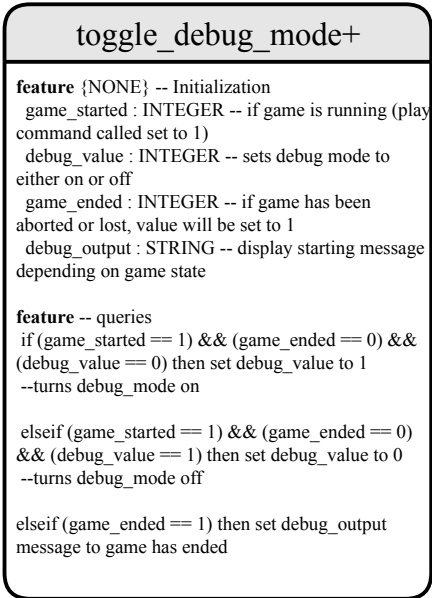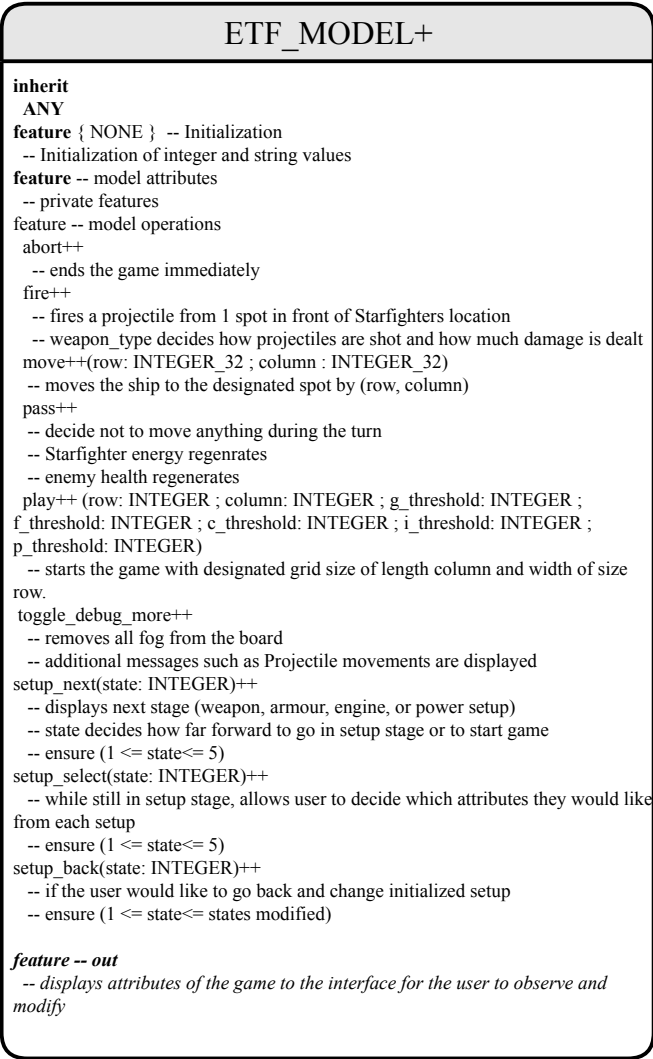# EECS 3311 - Project Bon Diagrams

Varuhn Ruthirakuhan - 215634140

Detailed view of toggle_debug_mode and fire class with inheritance relation

ETF_MODEL+

inherit
  ANY
feature { NONE } -- Initialization
  -- Initialization of integer and string values
feature -- model attributes
  -- private features
feature -- model operations
  abort++
   -- ends the game immediately
  fire++
   -- fires a projectile from 1 spot in front of Starfighters location
   -- weapon_type decides how projectiles are shot and how much damage is dealt
  move++(row: INTEGER_32 ; column : INTEGER_32)
  -- moves the ship to the designated spot by (row, column)
  pass++
   -- decide not to move anything during the turn
   -- Starfighter energy regenrates
   -- enemy health regenerates
  play++ (row: INTEGER ; column: INTEGER ; g_threshold: INTEGER ; f_threshold: INTEGER ; c_threshold: INTEGER ; i_threshold: INTEGER ; p_threshold: INTEGER)
   -- starts the game with designated grid size of length column and width of size row.
  toggle_debug_more++
   -- removes all fog from the board
   -- additional messages such as Projectile movements are displayed
setup_next(state: INTEGER)++
   -- displays next stage (weapon, armour, engine, or power setup)
   -- state decides how far forward to go in setup stage or to start game
   -- ensure (1 <= state<= 5)
setup_select(state: INTEGER)++
   -- while still in setup stage, allows user to decide which attributes they would like from each setup
   -- ensure (1 <= state<= 5)
setup_back(state: INTEGER)++
   -- if the user would like to go back and change initialized setup
   -- ensure (1 <= state<= states modified)

feature -- out
  -- displays attributes of the game to the interface for the user to observe and modify

model

ETF_COMMAND
*

+ ETF_FIRE

+ ETF_TOGGLE_DEBUG_MODE

toggle_debug_mode+

feature {NONE} -- Initialization
  game_started : INTEGER -- if game is running (play command called set to 1)
  debug_value : INTEGER -- sets debug mode to either on or off
  game_ended : INTEGER -- if game has been aborted or lost, value will be set to 1
  debug_output : STRING -- display starting message depending on game state

feature -- queries
  if (game_started == 1) && (game_ended == 0) && (debug_value == 0) then set debug_value to 1
  --turns debug_mode on

  elseif (game_started == 1) && (game_ended == 0) && (debug_value == 1) then set debug_value to 0
  --turns debug_mode off

  elseif (game_ended == 1) then set debug_output message to game has ended

fire+

feature {NONE} -- Initialization
  game_started : INTEGER -- if game is running (play command called set to 1)
  debug_value : INTEGER -- gotten from toggle_debug_mode
  ship_x : INTEGER -- ships location on the x-axis
  ship_y : INTEGER -- ships location on the y-axis
  board : ARRAY2[character] -- board used for game

feature -- queries
  fire
   --nested across loops used to check for existing projectiles to move first on board.

   board.put('*') --ship_x position and ship_y position will be taken to fire projectile one spot to the right
   -- location of projectile is then stored in p_x and p_y

if (game_started == 1) and (debug_value == 1)
  -- display advanced messages
else
  -- cover board with fog and display minimal messages