Dose 2015

Group 7 - Milan 4

Members of Group 7

Team Milan 4

- Fabrizio Frasca
- Luca Massaron
- Calin Liviu Razvan
- Alberto Fontana

Team Rio Cuarto 7

- Demian Romero
- Romina Miranda

No Contract Design

First Contact

0. Presentation
1. Skill checking
2. Project leader
3. Tasks assignment
4. Communication methods

DR Demian Romero

A: Fabrizio Frasca; Cc: Luca Massaron; Liviu Razvan Calin; Alberto Fontana;

Rispondi a tutti

mer 14/10/2015 19.22

Attività

Hi guys, we are your team mates from Argentina.

We need to choose a project leader to start working. How we do it? Any of you wants to be the project leader?

Sorry for my writing in english, is not very good. We are in touch.

Best regards,

Demián

1. Skill Checking

	Milan	**Argentina**
English	All good	Only one speaks not so fluently
Education	3 from Polimi, 1 from Polito	4th year of university (last)
Eiffel	No knowledge	Lessons with Nazareno
Git	2 Ok, 1 basic knowledge, 1 never used	All good
AI Background	2 following AI course	Followed course of AI

2. Project Leader

- Volunteers
- Lessons Schedule
- Work
- Language Skills
- Technical Skills
- Possibly in Milan

Final Candidates: Luca and Fabrizio

3. Task Assignment

Priority 1	Priority 2	Priority 3
- Bounded depth first search - Bounded breadth first search - Best first search - Hill climbing - Minimax	- Unbounded depth first search with cycle checking - Lowest-cost first search - Heuristic depth first search - Steepest ascent hill climbing - Minimax with alpha beta pruning	- A* Search - Iterative deepening - Principal variation search (NegaScout)

- 13 Algorithms
- 2 algorithms each + 1 TBD (Negascout)
- 10 Single Agent Search + 3 Adversary Search

4. Communication Methods

- Mailing List for official matter (documentation, instructions, meeting hour etc.)
- Telegram group for instant messaging
- Skype for periodic meetings and group working

Eiffel

Pros

- Contract Design
- Basic-like (well readable)
- Object composed only by features
- Better visibility management
- Multiple Inheritance

Cons

- Only one type of cycle
- Basic-like (too much writing)
- No polymorphism
- Bad Exception Handling
- Awful IDE
- No Return in functions
- No “main” method

Eiffel Configuration File .ecf

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<system xmlns="http://www.eiffel.com/developers/xml/configuration-1-14-0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.eiffel.com/developers/xml/configuration-1-14-0 http://www.eiffel.com/developers/xml/configuration-1-14-0.xsd" name="
eiffel-ai-search" uuid="C7C425FB-CC6A-4F1D-A3D3-7765C54B8A0C" library_target="eiffel-ai-search">
        <target name="eiffel-ai-search">
                <root all_classes="true"/>
                <option warning="true" is_attached_by_default="false" void_safety="none">
                        <assertions precondition="true" postcondition="true" check="true" invariant="true" loop="true" supplier_precondition="true"/>
                </option>
                <setting name="console_application" value="true"/>
                <library name="base" location="$ISE_LIBRARY\library\base\base.ecf"/>
                <library name="testing" location="$ISE_LIBRARY\library\testing\testing.ecf"/>
                <cluster name="eiffel-ai-search" location=".\">
                        <file_rule>
                                <exclude>/EIFGENs$</exclude>
                                <exclude>/CVS$</exclude>
                                <exclude>/.svn$</exclude>
                        </file_rule>
                        <cluster name="single-agent-search" location=".\single_agent_search\"/>
                        <cluster name="single-agent-search-engines" location=".\single_agent_search\single_agent_search_engines\"/>
                        <cluster name="single-agent-search-examples" location=".\single_agent_search_examples\"/>
                        <cluster name="adversary-search" location=".\adversary_search\"/>
                        <cluster name="adversary-search-engines" location=".\adversary_search\adversary_search_engines\"/>
                        <cluster name="adversary-search-examples" location=".\adversary_search_examples\"/>
                </cluster>
        </target>
</system>
```

AI Library

Closed List

What is it?
Why?

Example of implementation in BDFS

Example with water jar and simpler problem

Importance of remove from visited states the ones with greater depth when decreasing the depth

Closed List

What is it?

- A control which allow to **not generate any state that was visited** before.

Why?

- **no wasting time** by expanding states that have already been encountered before on some other path.
- moreover it **makes an algorithm like BDFS complete**, it means that if the solution is in the defined bound it will be found sooner or later

Price

- Requires every state that was visited to be kept in memory, resulting in a possibly **high consumption of memory**.

Closed List - implementation

For instance, in the Bounded Depth First Search algorithm this was implemented using a list obj to store the already visited states and then:

- before adding any child of a state to the frontier check if it was already visited or if it isn't already in the frontier
- If the max depth was reached remove the states with greater depth from the visited once because could be a shorter path to that state

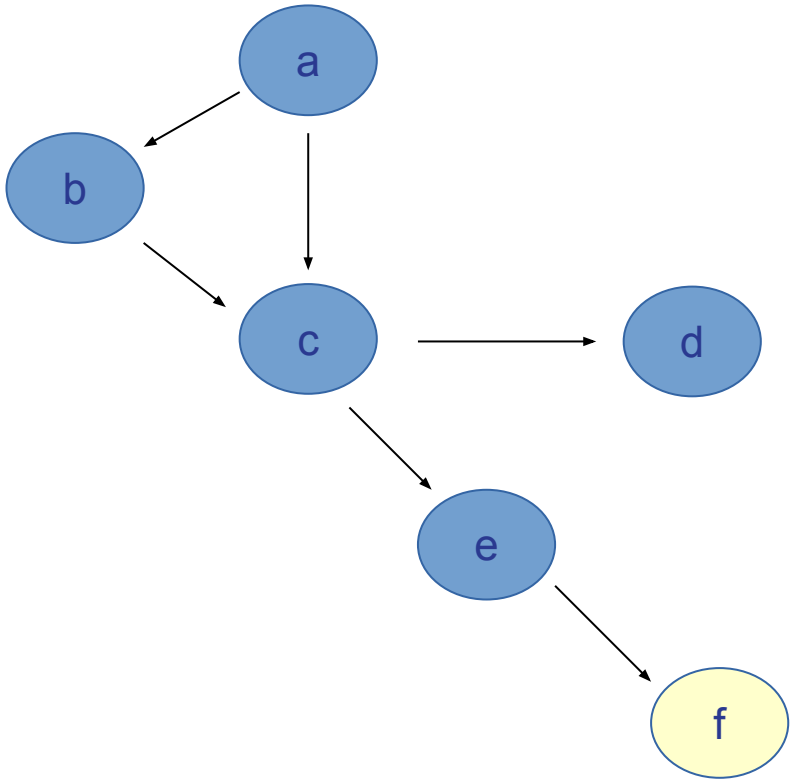
The feature may be enabled or disabled

Closed List – decreasing depth

Max depth: 3

Initial state: a

Goal state: f

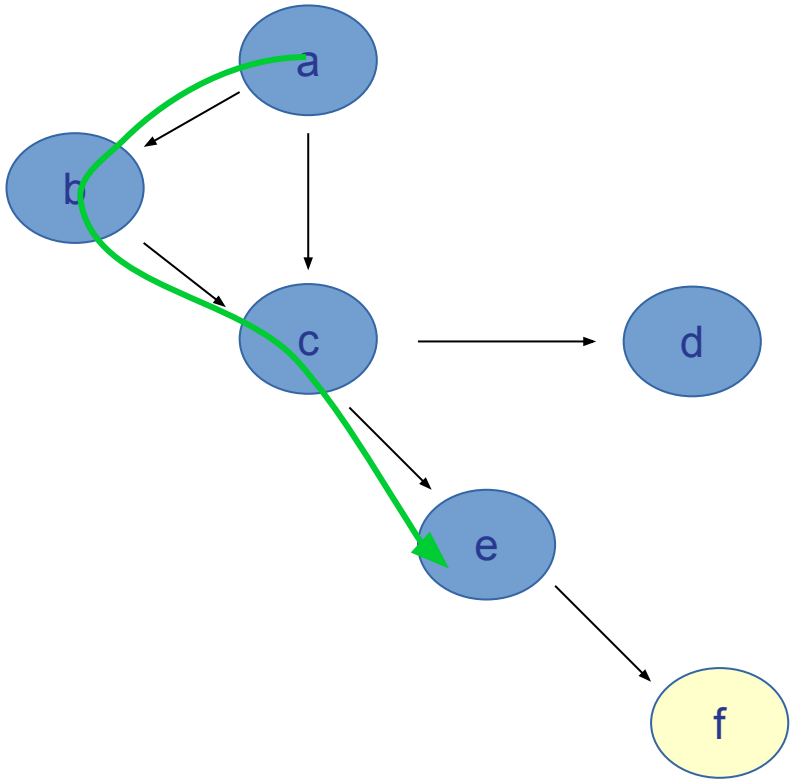


Closed List – decreasing depth

Max depth: 3

Initial state: a

Goal state: f

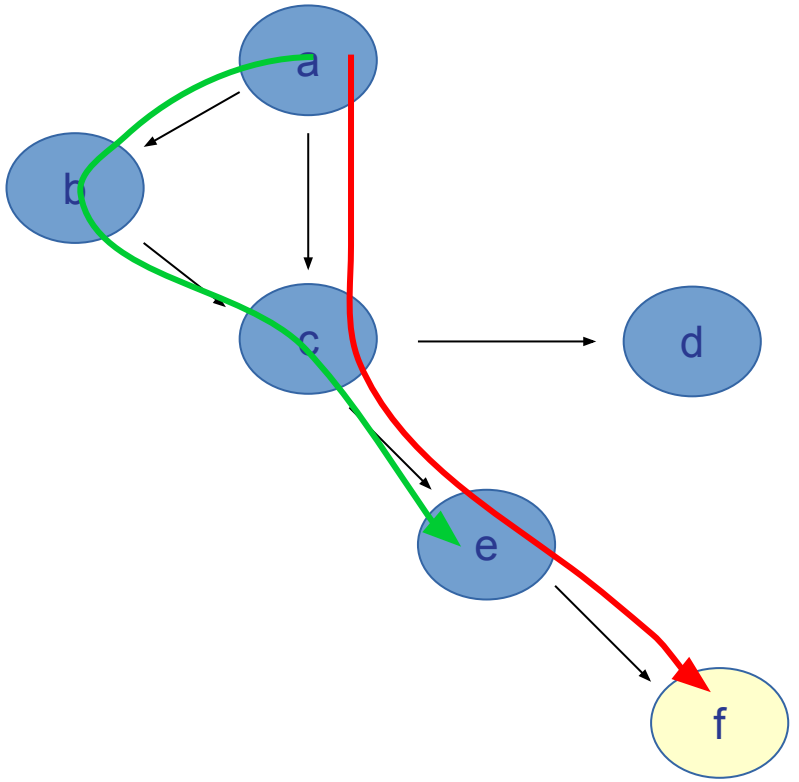


Closed List – decreasing depth

Max depth: 3

Initial state: a

Goal state: f



Closed List – example with BDFS

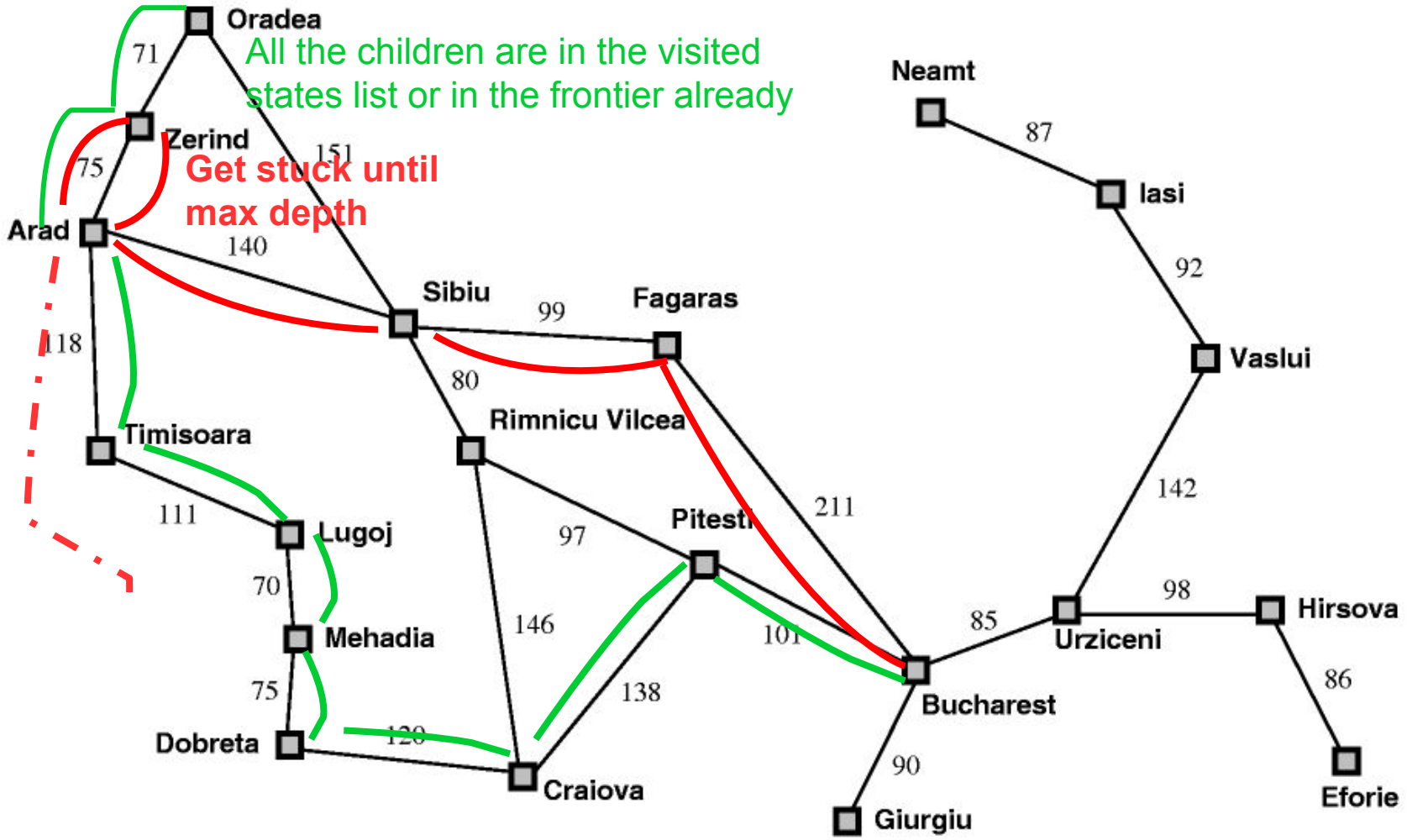
Water Jar Puzzle:

- Closed list disabled: unsolvable, the algorithm get stuck between one or more states
- Closed list enabled: find the solution with max depth 16 in a few milliseconds, visiting 34 states

Road to Bucharest (max depth set to 20):

- Closed list disabled: path length 19, visited states 88. The algorithm get stuck between two cities until the max depth is reached
- Closed list enabled: path length 7, visited states 10

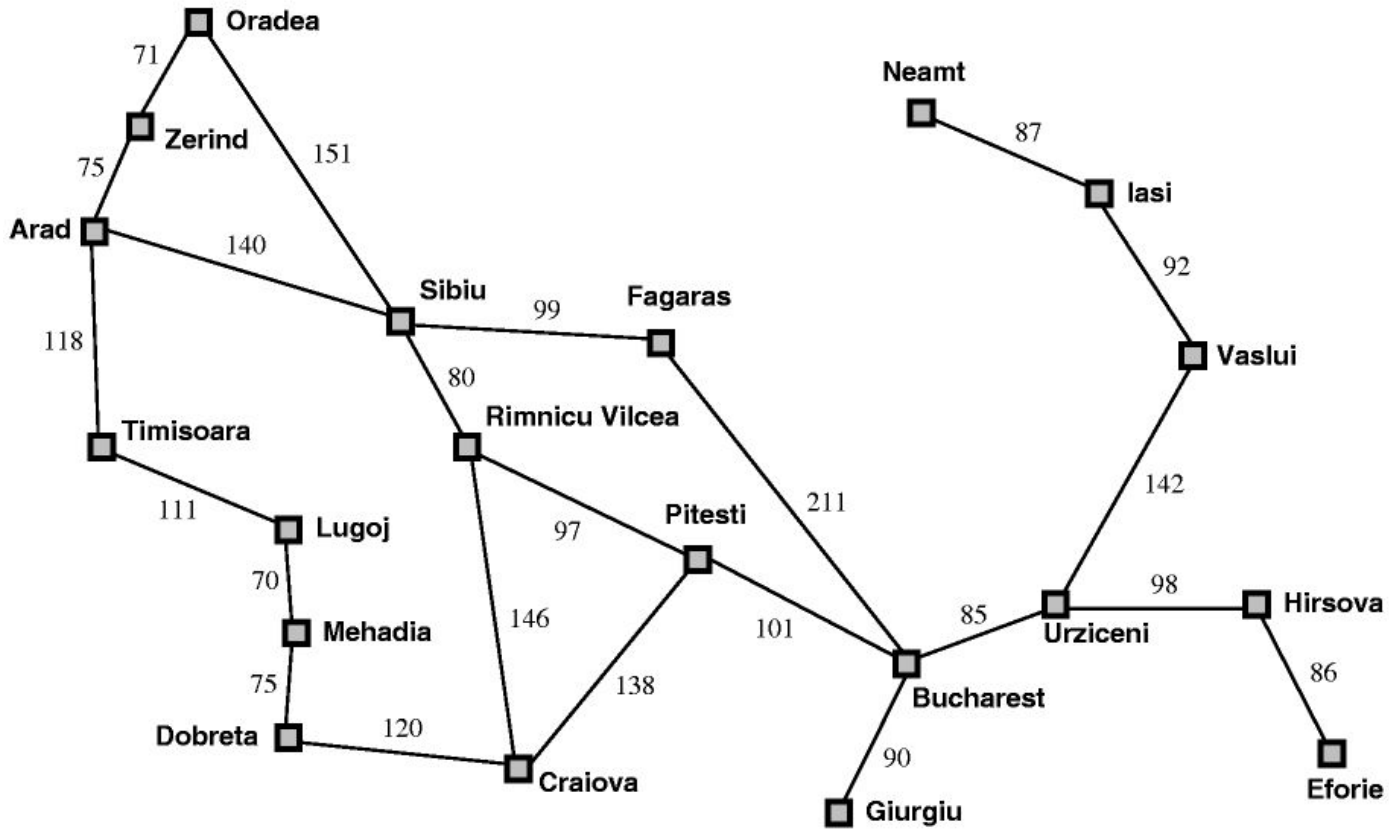
Closed List – road to Bucharest



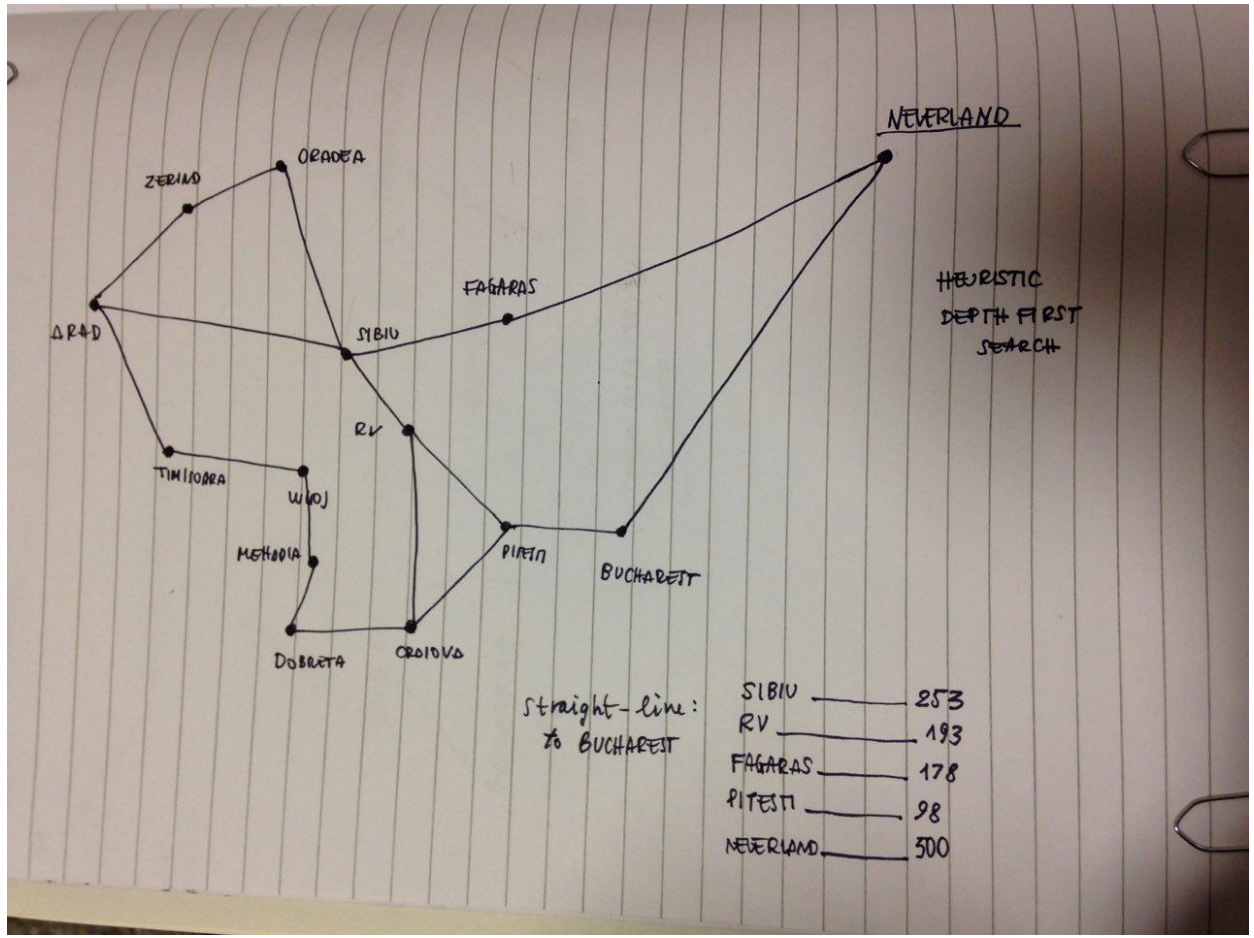
Test

1. The Water Jar Puzzle and its drawbacks
2. The need for another test case
3. The Road to Bucharest problem
4. Extending the test case to stress heuristic engines
5. Adversary search tests missing

```
feature -- Status Setting

        enable_visited_state_control
                        -- Sets the status so that the search performed is actually a
                        -- graph-search, i.e. the states that have been visited yet
                        -- are not considered in the search anymore.
                        -- By enabling the control you'll speed up the search. Do like
                        -- this if the problem is particularly reticular, i.e. it
                        -- contains many cycles in its state-space.
                        --
                        -- PAY ATTENTION: use this feature ONLY when the number of the
                        -- problem states is relatively small. Don't enable the control
                        -- if you've got little memory available on your machine or if
                        -- the state-space is quite big.
                do
                        visited_state_control_is_enabled := true
                end
```

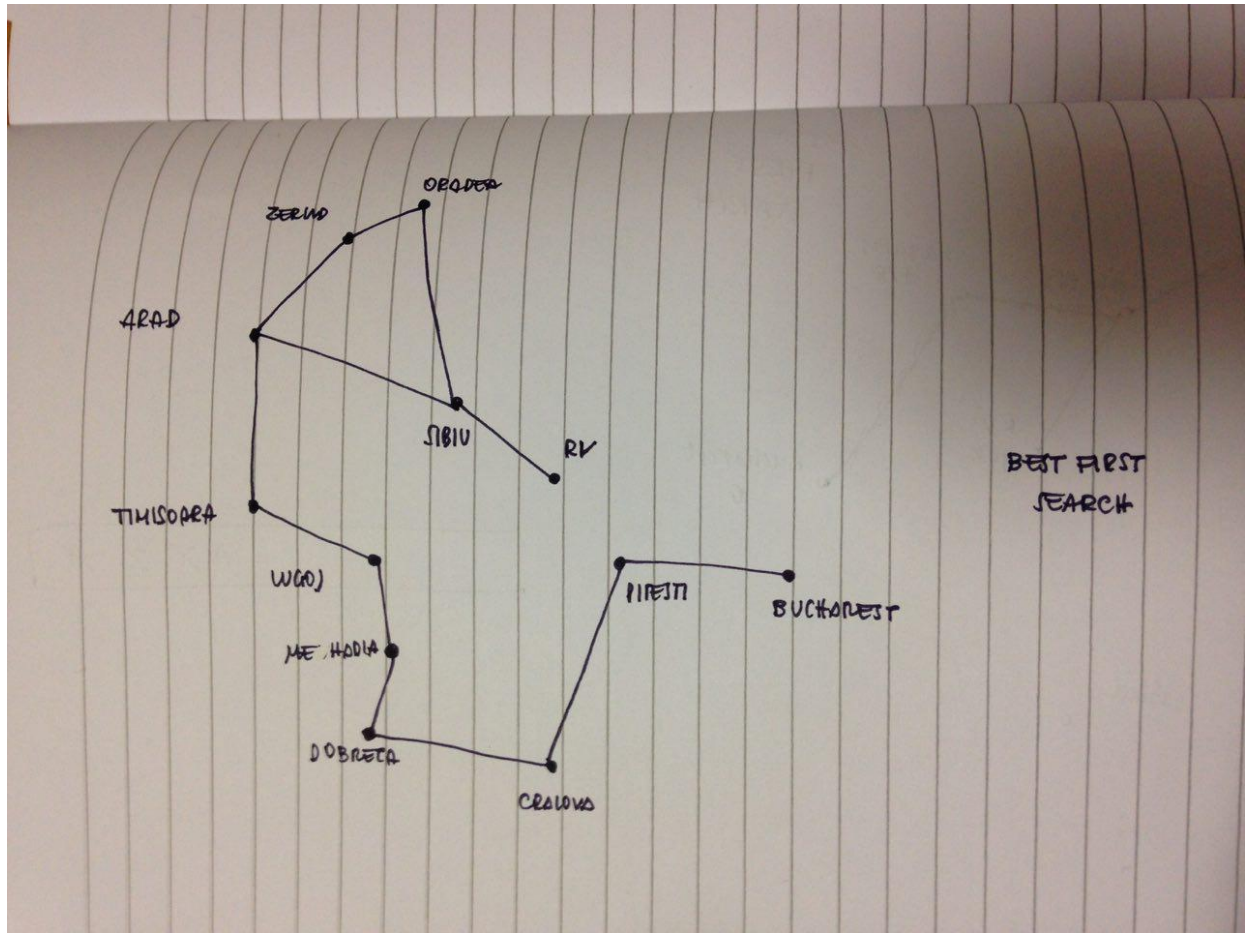


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



EiffelStudio Console (Do not Close)

```
 -> Arad  -> Sibiu  -> Fagaras  -> Neverland  -> Bucharest
---> nr of visited states: 6

---> PAY ATTENTION: with the visited state control disabled, the search will get stuck by wandering from SIBIU to FAGARAS and viceversa
```



EiffelStudio Console (Do not Close)

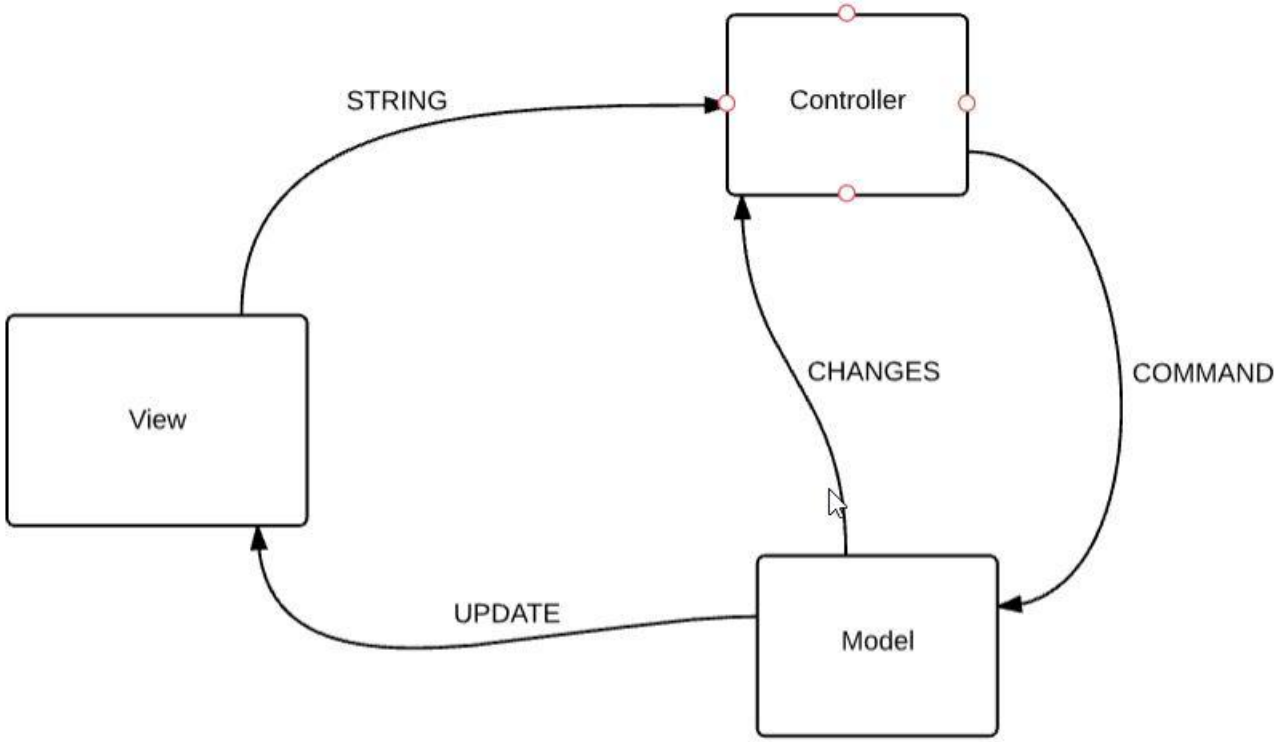
```
---> Solved :D here's the path to the solution:

  -> Arad  -> Timisoara  -> Lugoj  -> Mehadia  -> Dobreta  -> Craiova  -> Pitesti  -> Bucharest

---> nr of visited states: 11

---> PAY ATTENTION: with the visited state control disabled, the search will get stuck by wandering from sibiu to rm and viceversa.
```

Application

Mancala Game

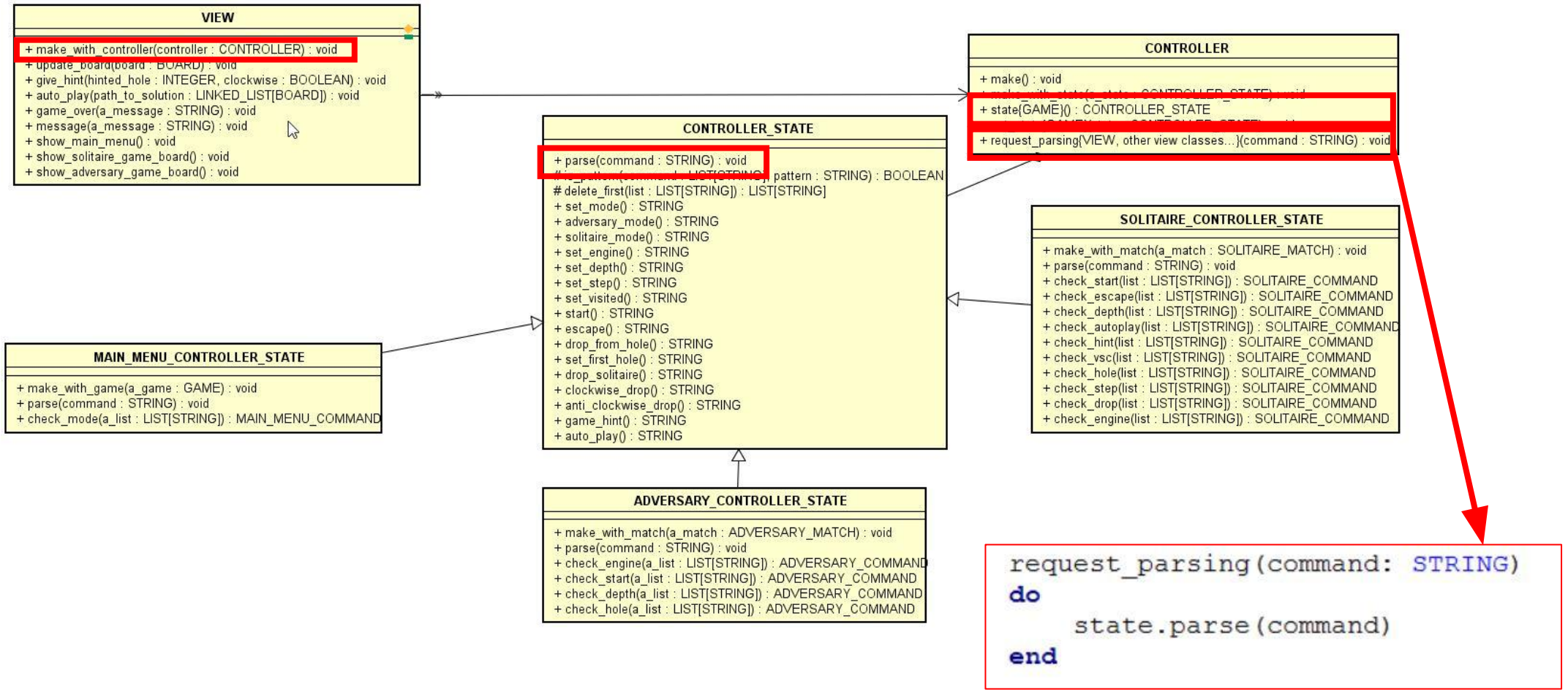
MVC Pattern

1. Simple Application
2. Easy job division
3. Already used
4. Common approach



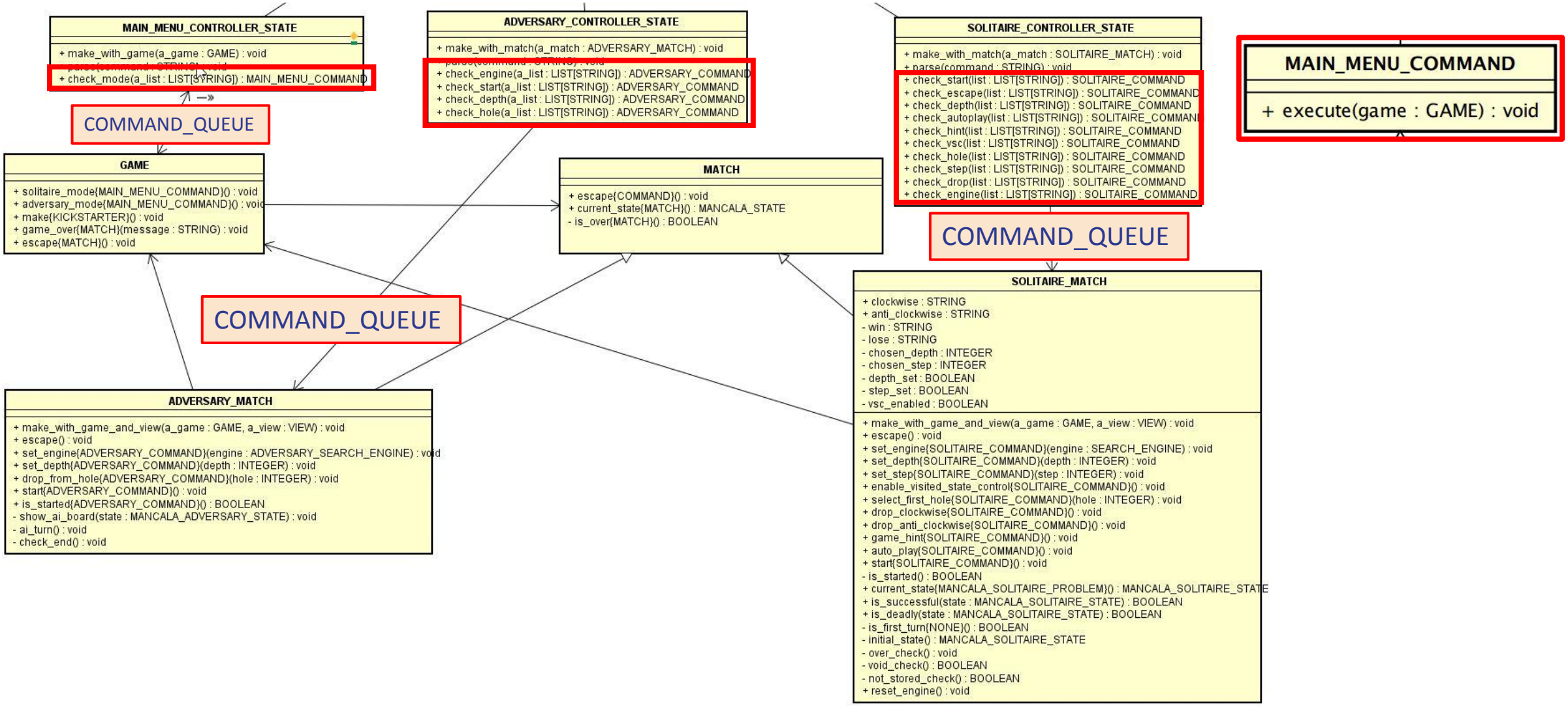
View - Controller

The State Pattern is used to filter different set of commands: Main Menu, Solitaire and Adversary



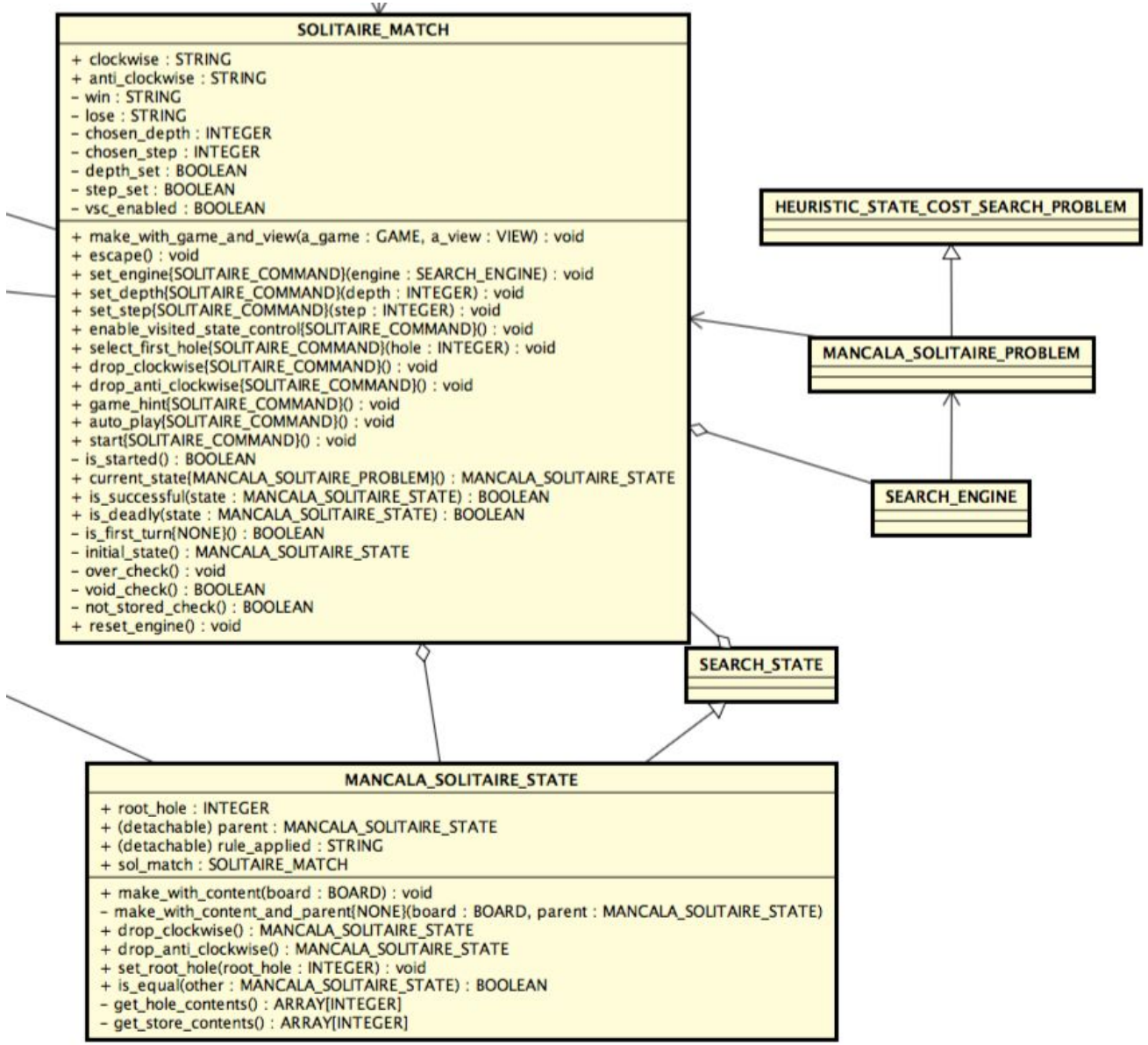
Controller - Model

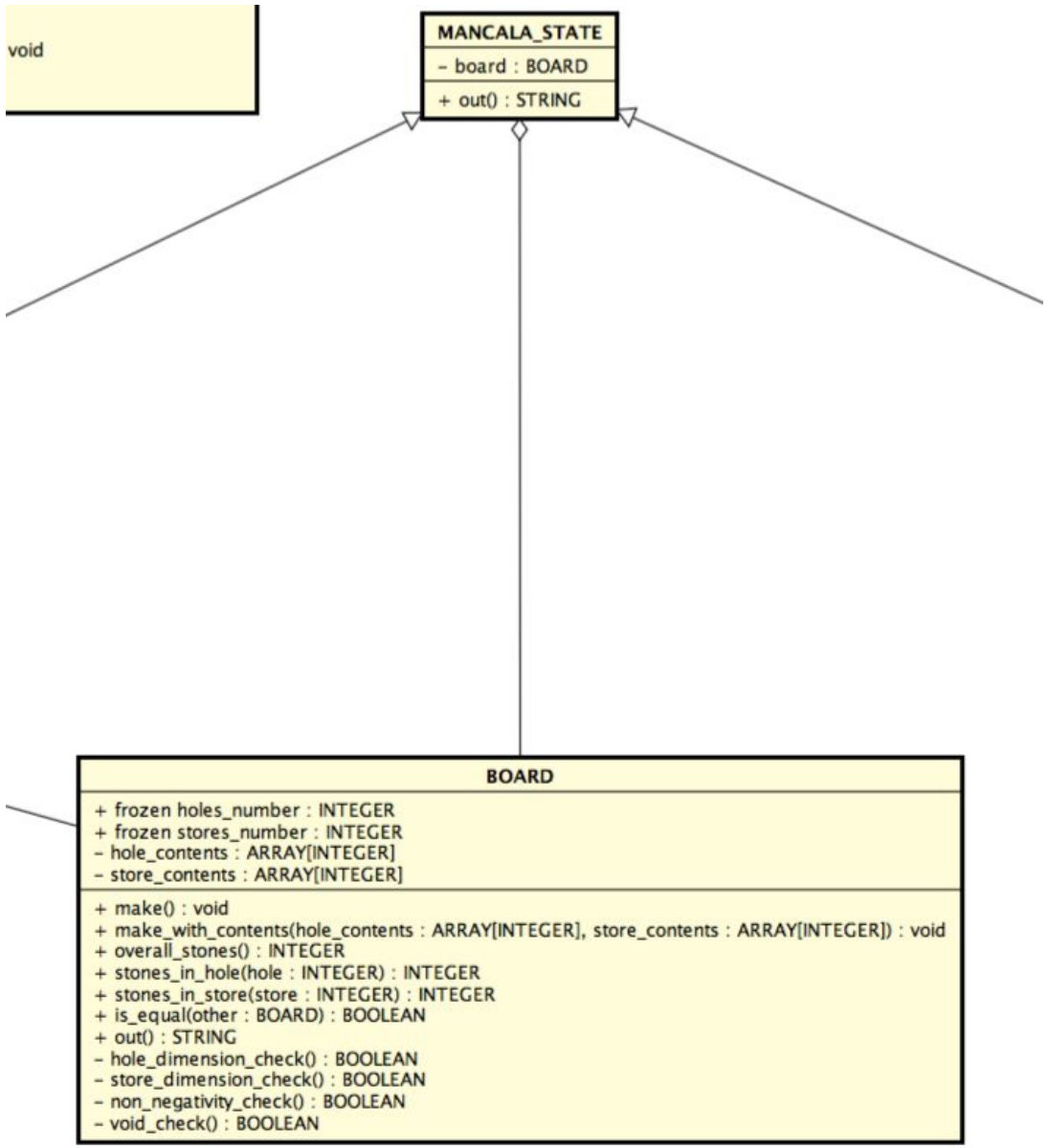
Command Pattern is used to translate View messages into Model features

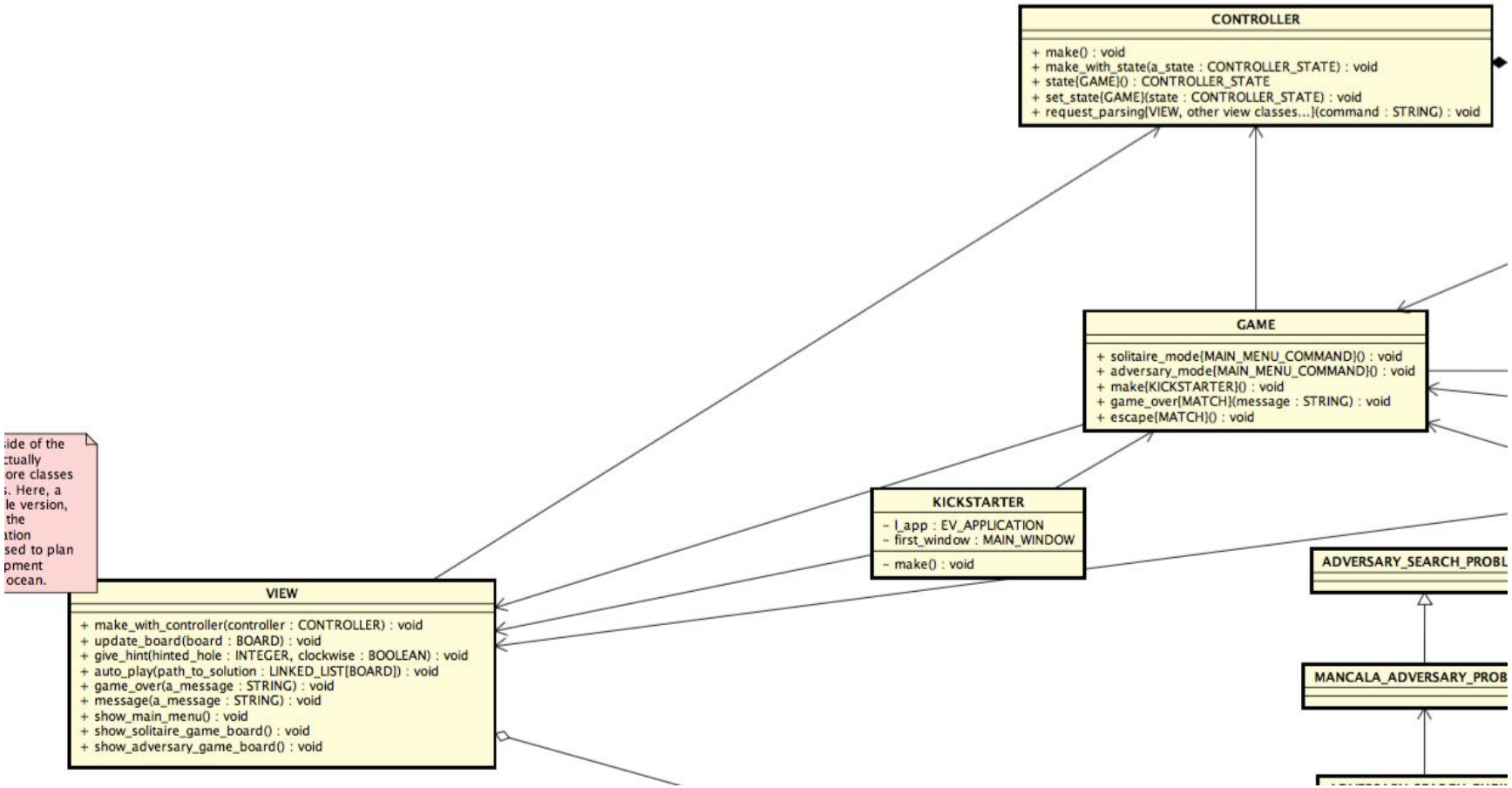


Designing the Model

1. Two models into one
2. The board shared class
3. Integrating the AI library
4. Keeping the game and the AI engine separated
5. Interfacing with the view


SOLITAIRE_MATCH
+ clockwise : STRING
+ anti_clockwise : STRING
- win : STRING
- lose : STRING
- chosen_depth : INTEGER
- chosen_step : INTEGER
- depth_set : BOOLEAN
- step_set : BOOLEAN
- vsc_enabled : BOOLEAN
+ make_with_game_and_view(a_game : GAME, a_view : VIEW) : void
+ escape() : void
+ set_engine{SOLITAIRE_COMMAND}(engine : SEARCH_ENGINE) : void
+ set_depth{SOLITAIRE_COMMAND}(depth : INTEGER) : void
+ set_step{SOLITAIRE_COMMAND}(step : INTEGER) : void
+ enable_visited_state_control{SOLITAIRE_COMMAND}() : void
+ select_first_hole{SOLITAIRE_COMMAND}(hole : INTEGER) : void
+ drop_clockwise{SOLITAIRE_COMMAND}() : void
+ drop_anti_clockwise{SOLITAIRE_COMMAND}() : void
+ game_hint{SOLITAIRE_COMMAND}() : void
+ auto_play{SOLITAIRE_COMMAND}() : void
+ start{SOLITAIRE_COMMAND}() : void
- is_started() : BOOLEAN
+ current_state{MANCALA_SOLITAIRE_PROBLEM}() : MANCALA_SOLITAIRE_STATE
+ is_successful(state : MANCALA_SOLITAIRE_STATE) : BOOLEAN
+ is_deadly(state : MANCALA_SOLITAIRE_STATE) : BOOLEAN
- is_first_turn{NONE}() : BOOLEAN
- initial_state() : MANCALA_SOLITAIRE_STATE
- over_check() : void
- void_check() : BOOLEAN
- not_stored_check() : BOOLEAN
+ reset_engine() : void
HEURISTIC_STATE_COST_SEARCH_PROBLEM
MANCALA_SOLITAIRE_PROBLEM
SEARCH_ENGINE
SEARCH_STATE
MANCALA_SOLITAIRE_STATE
+ root_hole : INTEGER
+ (detachable) parent : MANCALA_SOLITAIRE_STATE
+ (detachable) rule_applied : STRING
+ sol_match : SOLITAIRE_MATCH
+ make_with_content(board : BOARD) : void
- make_with_content_and_parent{NONE}(board : BOARD, parent : MANCALA_SOLITAIRE_STATE)
+ drop_clockwise() : MANCALA_SOLITAIRE_STATE
+ drop_anti_clockwise() : MANCALA_SOLITAIRE_STATE
+ set_root_hole(root_hole : INTEGER) : void
+ is_equal(other : MANCALA_SOLITAIRE_STATE) : BOOLEAN
- get_hole_contents() : ARRAY[INTEGER]
- get_store_contents() : ARRAY[INTEGER]


void
MANCALA_STATE
- board : BOARD
+ out() : STRING
BOARD
+ frozen holes_number : INTEGER
+ frozen stores_number : INTEGER
- hole_contents : ARRAY[INTEGER]
- store_contents : ARRAY[INTEGER]
+ make() : void
+ make_with_contents(hole_contents : ARRAY[INTEGER], store_contents : ARRAY[INTEGER]) : void
+ overall_stones() : INTEGER
+ stones_in_hole(hole : INTEGER) : INTEGER
+ stones_in_store(store : INTEGER) : INTEGER
+ is_equal(other : BOARD) : BOOLEAN
+ out() : STRING
- hole_dimension_check() : BOOLEAN
- store_dimension_check() : BOOLEAN
- non_negativity_check() : BOOLEAN
- void_check() : BOOLEAN


CONTROLLER
+ make() : void
+ make_with_state(a_state : CONTROLLER_STATE) : void
+ state{GAME}() : CONTROLLER_STATE
+ set_state{GAME}(state : CONTROLLER_STATE) : void
+ request_parsing{VIEW, other view classes...}(command : STRING) : void
GAME
+ solitaire_mode{MAIN_MENU_COMMAND}() : void
+ adversary_mode{MAIN_MENU_COMMAND}() : void
+ make{KICKSTARTER}() : void
+ game_over{MATCH}(message : STRING) : void
+ escape{MATCH}() : void
KICKSTARTER
- l_app : EV_APPLICATION
- first_window : MAIN_WINDOW
- make() : void
VIEW
+ make_with_controller(controller : CONTROLLER) : void
+ update_board(board : BOARD) : void
+ give_hint(hinted_hole : INTEGER, clockwise : BOOLEAN) : void
+ auto_play(path_to_solution : LINKED_LIST[BOARD]) : void
+ game_over(a_message : STRING) : void
+ message(a_message : STRING) : void
+ show_main_menu() : void
+ show_solitaire_game_board() : void
+ show_adversary_game_board() : void
ADVERSARY_SEARCH_PROBL
MANCALA_ADVERSARY_PROB

Thread

We thought it would be important to develop a multithread application with the aim of let the control of the app to the user, even after he ask for an automatic solution method, like 'hint' or 'solve'.

Especially in the solitaire mode, the execution of the ai algorithm could take a lot of time.

It might not even terminate cause the insolubility of the problem.

Thread - problem

We weren't able to synchronize the execution of the threads in a good way for the execution of the program and in the same way between different O.S..

We had different behaviors changing the O.S., for instance in Windows the application starts with a blank window while on Unix based O.S. it starts normally, but pretty often it get stuck and doesn't responds to commands any more.

Probably the differences are due to the different way in which the systems manage the scheduling of threads.

Thread – eiffel “tools”

The most important classes are:

- **THREAD**: use this class as an ancestor for each class that allows starting new threads.
- **PROXY**: use this class to refer to other threads' objects.

In class THREAD, there is a procedure launch. This procedure calls another, execute, deferred in THREAD. To describe the behavior of a thread on execution, simply redefine procedure execute in the appropriate descendant of THREAD.

Thread – proxy mechanism

To maintain the safety and consistency of a multithreaded application, within the use of other threads objects, there is a **Basic Thread Rule:**
“A thread must never keep references to another thread's objects.”

When a thread needs to access another thread's object, it will use a proxy for that object -- an instance of the EiffelThreads class PROXY[G].

Thread – proxy mechanism

To record the object foreign_object of type FOREIGN_TYPE in a proxy, it will execute

```
my_proxy.put (foreign_object)
```

where my_proxy is of type PROXY [FOREIGN_TYPE]. To access the object, it will then use the expression

```
my_proxy.item
```

This expression should only be used as a target of feature applications, as in

```
my_proxy.item.do_something
```

and **never assigned to an attribute (as in my_attrib := my_proxy.item)**, as this would cause a violation of the Basic Thread Rule, and unpredictable results.

Thread – once routines

Eiffel introduced the powerful mechanism of once routines. A once routine has a body that will be executed only for the first call; This provides a simple way of sharing objects in an object-oriented context.

For multithreaded applications, the appropriate semantics is that once routines must be called once per thread (rather than once per process).

Solitaire Heuristic Function

1. Solitaire Mancala: an unsolvable problem
2. Fail to apply the most common approach
3. A customized approach: Life Expectancy

```
feature {ANY} -- Heuristic search related routines

    heuristic_value (state: MANCALA_SOLITAIRE_STATE): REAL
        local
            weight: REAL_32
        do
            -- The evaluation is basically made by considering
            -- the number of stones in the stores weighted
            -- upon the life expectancy of state, that is
            -- number of descendants at a certain parametric
            -- depth.

            if is_successful(state) or else has_successful_descendant(forecast_depth, state) then

                Result := 0

            elseif state.root_hole = 0 then -- first state

                Result := match.initial_stones_in_holes

            else

                weight := (descendants_at_depth(forecast_depth, state).to_real / (branching_factor.power(forecast_depth)).floor)

                Result := overall_stones - weight * stored_stones(state)
            end

        end
```

```
feature{NONE} -- Heuristic Utilities

    forecast_depth: INTEGER = 4

    branching_factor: INTEGER = 2

    overall_stones: INTEGER
        do
            Result := match.initial_stones_in_holes
        end

    stored_stones(s: like initial_state): INTEGER
        local
            i: INTEGER
            stored: INTEGER
        do
            stored := 0

            from
                i:=1
            until
                i>s.board.stores_number
            loop

                stored := stored + s.board.stones_in_store(i)

                i := i + 1
            end

            Result := stored
        end
```

```
descendants_at_depth(d: INTEGER; s: like initial_state): INTEGER
    local
        ancestors: LINKED_LIST[MANCALA_SOLITAIRE_STATE]
        descendants: LINKED_LIST[MANCALA_SOLITAIRE_STATE]
        i: INTEGER
        lv: INTEGER
    do
        from
            create descendants.make
            create ancestors.make
            lv := d
            descendants.extend(s)
        until
            lv=0
        loop
            ancestors.copy(descendants)

            descendants.wipe_out

            from
                i:=1
            until
                i>ancestors.count
            loop
                descendants.append(get_successors(ancestors.i_th(i)))

                i := i + 1
            end

            lv := lv - 1

        end

        Result := descendants.count

    end
```

```
has_successful_descendant(d:INTEGER; s: like initial_state):BOOLEAN
    local
        ancestors: LINKED_LIST[MANCALA_SOLITAIRE_STATE]
        descendants: LINKED_LIST[MANCALA_SOLITAIRE_STATE]
        i: INTEGER
        lv: INTEGER
        goal_found: BOOLEAN
    do
        from
            create descendants.make
            create ancestors.make
            lv := d
            descendants.extend(s)
            goal_found := false

        until
            lv=0 or else goal_found
        loop
            ancestors.copy(descendants)

            descendants.wipe_out

            from
                i:=1
            until
                i>ancestors.count
            loop
                descendants.append(get_successors(ancestors.i_th(i)))

                i := i + 1
            end

            from
                i:=1
            until
                i>descendants.count or else goal_found
            loop

                if is_successful(descendants.i_th(i)) then
                    goal_found := true
                end

                i := i+1

          end

          lv := lv - 1

      end

      Result := goal_found

 end
```

Adversary Heuristic function

- H0: First valid move (furthest valid bin from my home)
- H1: How far ahead of my opponent I am (My Mancala – Opponent's Mancala)
- H2: How close I am to winning (> half)
- H3: How close opponent is to winning (> half)
- H4: Number of stones close to my home
- H5: Number of stones far away from my home
- H6: Number of stones in middle of board (neither close nor far from home)

Source: www.ittc.ku.edu/publications/documents/Gifford_ITTC-FY2009-TR-03050-03.pdf

Result: H1, H3 and H2 were the best Heuristics in order

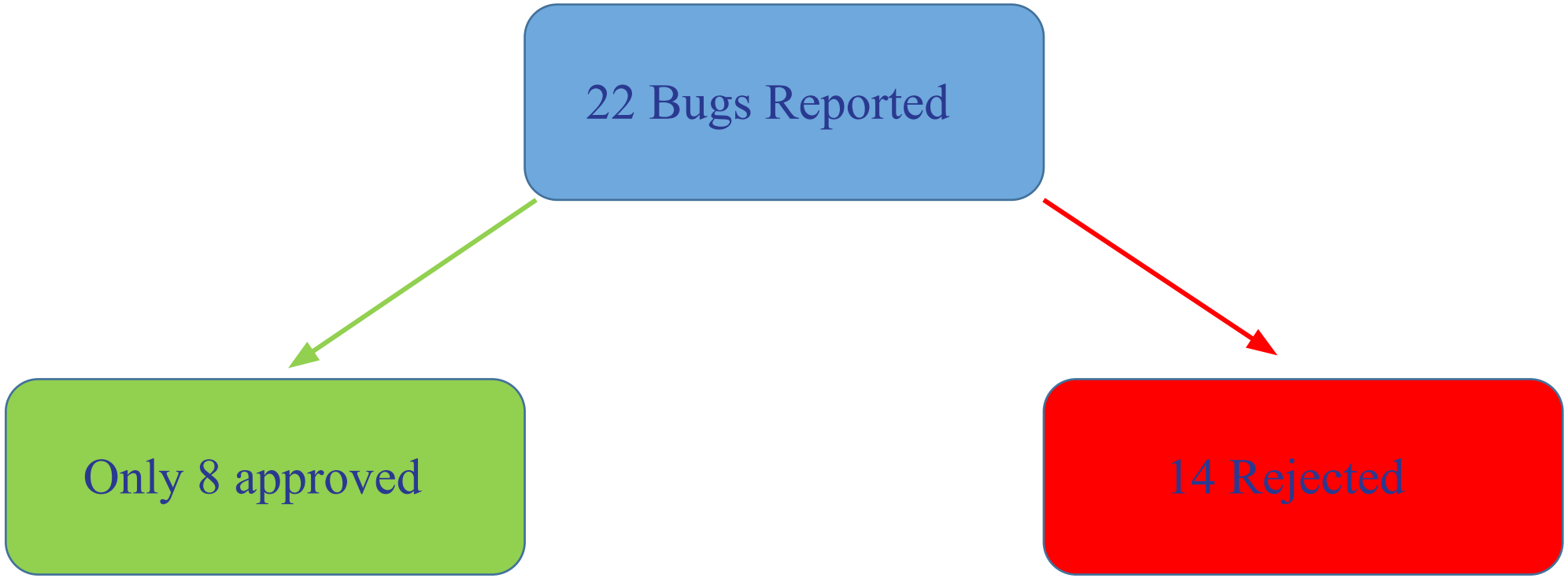
Adversary Heuristic function

```
value (state: MANCALA_ADVERSARY_STATE): INTEGER
    local
        score:INTEGER
    do
        if max_wins (state) then
            score:= max_value
        else
            score:= score + state.max_advantage
            score:= score + state.max_to_win
            score:= score + (state.max_can_eat * 2)
            if state.max_has_additional_turn then
                score:= score + 5
            end
        end
        if min_wins (state) then
            score:= min_value
        else
            score:= score - state.min_to_win
            score:= score - (state.min_can_eat * 2)
            if state.min_has_additional_turn then
                score:= score - 5
            end
        end
        Result := score
    end
```

- H1: How far ahead of my opponent I am (My Mancala – Opponent's Mancala)
- H2: How close I am to winning (> half)
- M1: How much can I eat (Doubled)
- M2: I have an additional turn
- H3: How close opponent is to winning (> half)
- M3: How much can my opponent eat
- M4: My opponent has an additional turn

Bug Competition

Group 7 Bugs



Approved Bugs

#	Bug	Cause
1	Exception while starting Adversary game with negative depth	Due to misunderstanding message feature in VIEW was not working in main menu
2	Exception while starting Adversary game with 0 depth	As point 1
3	Can start Solitaire game with negative depth	Was working in Multi Thread but after switching to Single thread there was an error adapting the code because was not performed by the class developer
4	Can start Solitaire game with negative step	As point 3
5	Final values not shown in Adversary game	Final Value check did not update the view
6	Label always shows “YOUR TURN” in Adversary Game	Should work in Multi Thread but using Single labels are updated only at the end of the function call (using buttons should solve it)
7	Holes buttons disabled after clicking on an empty one	Blocking buttons should not be implemented in Single thread, was actually removed and solved but not merged
8	Negascout Exception	Still unknown

Rejected Bugs

Main Reasons:

1. Duplicates
2. Bad Documentation
3. Not a Bug
4. Cannot replicate

Title
low cost first search autoplay probably infinite loop

Class
AUTO_PLAY_COMMAND

Routine
execute

Description
low cost first search click button autoplay, window froze, no response

"Probably" infinite loop?
COMMAND.execute?

```
execute(match: ADVERSARY_MATCH)
do
    match.drop_from_hole (hole)
end
```

Title
depth exploration set 0 still could begin game, hint button first raise warning t

Class
GAME_HINT_COMMAND

Routine
execute

Description
depth exploration set 0 still could begin game, hint button first raise warning then window froze.

Depth in which engine?
Depth = 0 already reported first day

Bug Competition:
Inspecting other teams' code

1. Where to look
2. Cohesion and coherence
3. (Missed) Separation between Game and AI

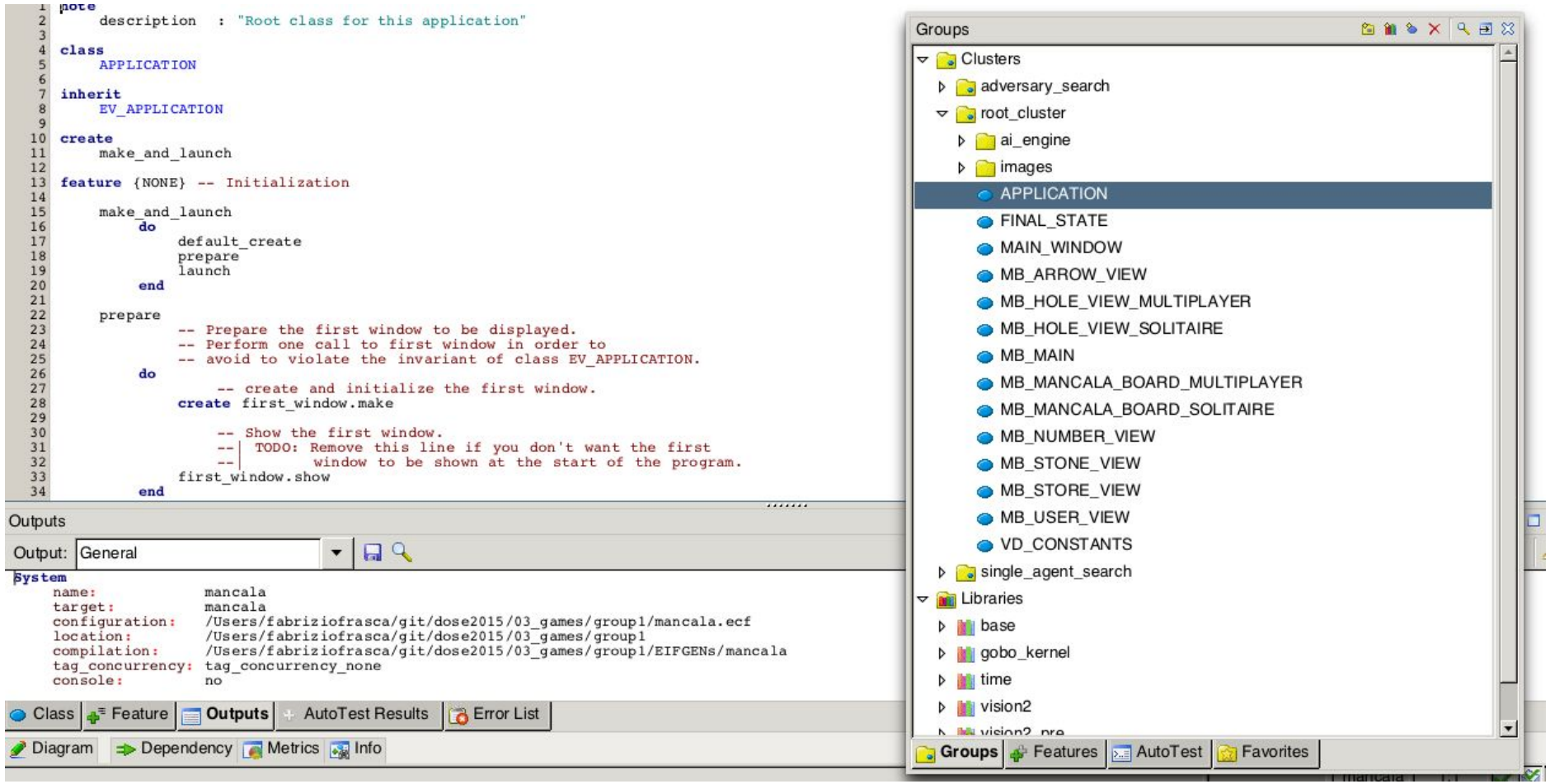
Noired						
Noired	GUI is not allowing to set the Iterative deepening step	Group 10	Dec 13, 2015	MAIN_WINDOW	NONE	APPROVED
Noired	GUI is not allowing to set the Iterative deepening step	Group 9	Dec 13, 2015	MAIN_WINDOW	NONE	REJECTED
Noired	GUI is not allowing to set the Iterative deepening step	Group 2	Dec 13, 2015	V_GAME_VIEW	NONE	APPROVED
Noired	Wrong use of iterative deepening step	Group 10	Dec 13, 2015	ITERATIVE_DEEPENING_SEARCH_ENGINE	perform_search	APPROVED
Noired	Wrong use of iterative deepening step	Group 9	Dec 13, 2015	ITERATIVE_DEEPENING_SEARCH_ENGINE	bounded_depth_first_search	APPROVED
Noired	GUI is not allowing to set the Iterative deepening step	Group 1	Dec 13, 2015	MAIN_WINDOW	NONE	APPROVED
Noired	Wrong use of iterative deepening step	Group 1	Dec 13, 2015	ITERATIVE_DEEPENING_SEARCH_ENGINE	depth_first_search	APPROVED
Noired	Automatic play's path to solution does not start from the current state	Group 6	Dec 13, 2015	LOGIC_SOLITAIRE_MANCALA	solution	APPROVED
Noired	Game hint is always dropping from the first state of the game, regardless of the	Group 6	Dec 13, 2015	LOGIC_SOLITAIRE_MANCALA	help	APPROVED

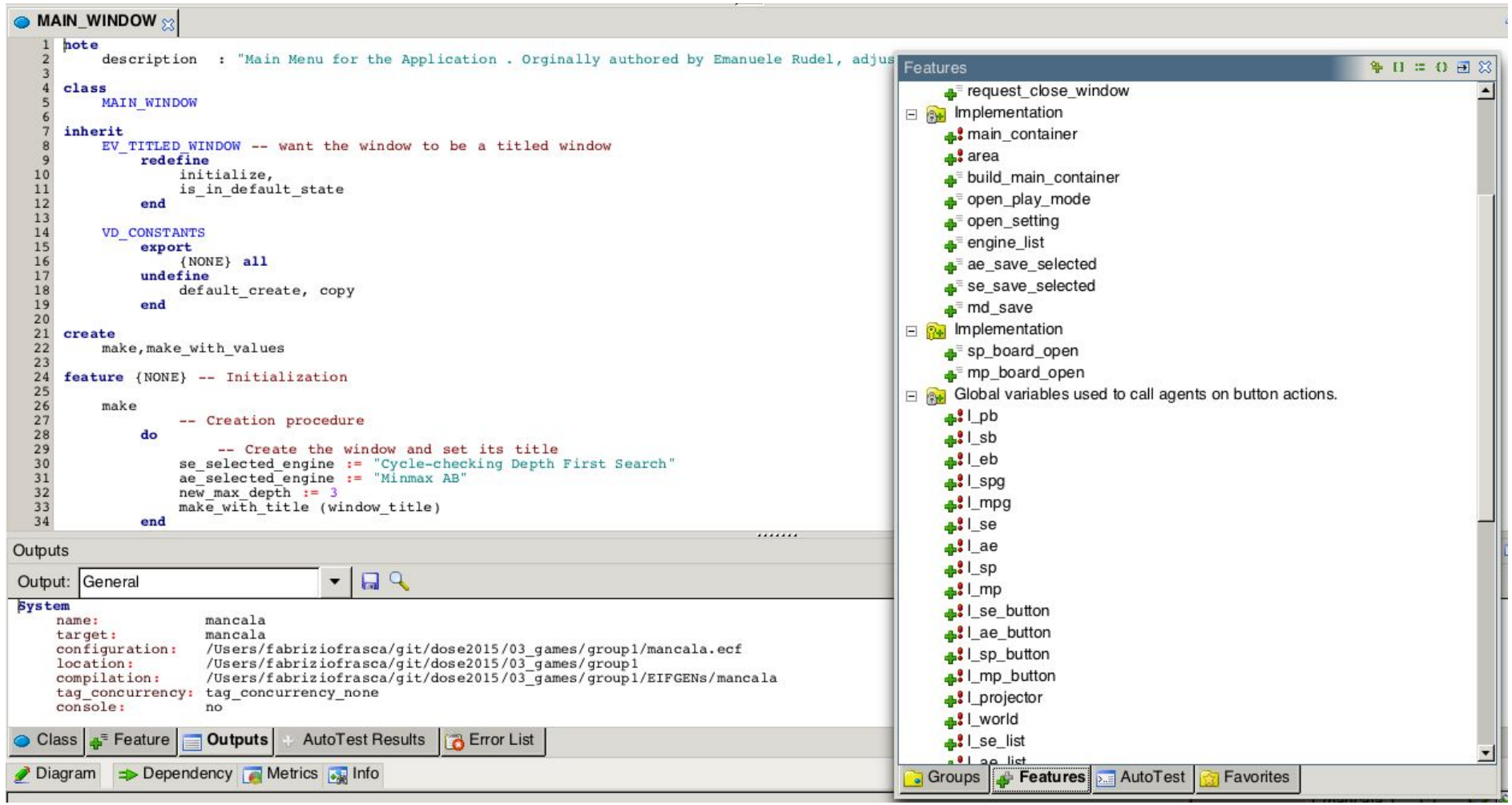
'Routine name'

Noired	Game hint is always dropping from the first state of the game, regardless of the current state	Group 6	Dec 13, 2015	LOGIC_SOLITAIRE_MANCALA	help	APPROVED
Noired	Invalid index precondition violation	Group 10	Dec 13, 2015	MANCALA	get_successors	APPROVED

Noired	get_successors feature is not working	Group 1	Dec 12, 2015	MANCALA_SOLITAIRE_PROBLEM	get_successors	APPROVED
Noired	Autoplay is not showing the solution found.	Group 1	Dec 12, 2015	MB_MAIN	state_received	APPROVED
Noired	Game hint is not changing the current state, even if the search is successful	Group 1	Dec 12, 2015	MB_MAIN	state_received	APPROVED
Noired	Autoplay is not synchronized with the obtained solution	Group 2	Dec 10, 2015	C_SINGLE_GAME_CONTROL	solve_game	OPEN
Noired	Successors are computed for dead states too	Group 2	Dec 10, 2015	M_MANCALA_SINGLE_GAME	get_successors	OPEN
Noired	Noired Autoplay is calling drop moves on a finished games, too restrictive precondition for them	Group 2	Dec 10, 2015	M_MANCALA_SINGLE_GAME	counterclockwise_move	OPEN
Noired	Autoplay is calling drop moves on a finished games, too restrictive precondition for them	Group 2	Dec 10, 2015	M_MANCALA_SINGLE_GAME	clockwise_move	OPEN
Noired	Initial state of the search has a parent	Group 2	Dec 10, 2015	M_MANCALA_SINGLE_GAME	calculate_hint_for_current_initial_state	REJECTED

Noired	Initial state of the search has a parent	Group 2	Dec 10, 2015	M_MANCALA_SINGLE_GAME	solve_game	APPROVED
Noired	Visited states is not working	Group 2	Dec 10, 2015	LOWEST_COST_FIRST_SEARCH_ENGINE	perform_search	OPEN
Noired	No path cost computed	Group 2	Dec 10, 2015	WRAPPER_STATE_COST_SEARCH_PROBLEM	get_cost	OPEN
Noired	Iterative Deepening step cannot be set from the initial window	Group 3	Dec 2, 2015	NONE	NONE	APPROVED
Noired	Game Over message not showing in some cases	Group 3	Dec 2, 2015	NONE	NONE	APPROVED
Noired	Autoplay has strange behavior in showing the path to solution	Group 3	Dec 2, 2015	NONE	NONE	REJECTED


note
description : "Root class for this application"
class
APPLICATION
inherit
EV_APPLICATION
create
make_and_launch
feature {NONE} -- Initialization
make_and_launch
do
default_create
prepare
launch
end
prepare
-- Prepare the first window to be displayed.
-- Perform one call to first window in order to
-- avoid to violate the invariant of class EV_APPLICATION.
do
-- create and initialize the first window.
create first_window.make
-- Show the first window.
-- TODO: Remove this line if you don't want the first
-- window to be shown at the start of the program.
first_window.show
end
Outputs
Output: General
System
name: mancala
target: mancala
configuration: /Users/fabriziofrasca/git/dose2015/03_games/group1/mancala.ecf
location: /Users/fabriziofrasca/git/dose2015/03_games/group1
compilation: /Users/fabriziofrasca/git/dose2015/03_games/group1/EIFGENs/mancala
tag_concurrency: tag_concurrency_none
console: no
Class
Feature
Outputs
AutoTest Results
Error List
Diagram
Dependency
Metrics
Info
Groups
Clusters
adversary_search
root_cluster
ai_engine
images
APPLICATION
FINAL_STATE
MAIN_WINDOW
MB_ARROW_VIEW
MB_HOLE_VIEW_MULTIPLAYER
MB_HOLE_VIEW_SOLITAIRE
MB_MAIN
MB_MANCALA_BOARD_MULTIPLAYER
MB_MANCALA_BOARD_SOLITAIRE
MB_NUMBER_VIEW
MB_STONE_VIEW
MB_STORE_VIEW
MB_USER_VIEW
VD_CONSTANTS
single_agent_search
Libraries
base
gobo_kernel
time
vision2
Groups
Features
AutoTest
Favorites


MAIN_WINDOW
note
description : "Main Menu for the Application . Orginally authored by Emanuele Rudel, adjus
class
MAIN_WINDOW
inherit
EV_TITLED_WINDOW -- want the window to be a titled window
redefine
initialize,
is_in_default_state
end
VD_CONSTANTS
export
{NONE} all
undefine
default_create, copy
end
create
make,make_with_values
feature {NONE} -- Initialization
make
-- Creation procedure
do
-- Create the window and set its title
se_selected_engine := "Cycle-checking Depth First Search"
ae_selected_engine := "Minmax AB"
new_max_depth := 3
make_with_title (window_title)
end
Features
request_close_window
Implementation
main_container
area
build_main_container
open_play_mode
open_setting
engine_list
ae_save_selected
se_save_selected
md_save
Implementation
sp_board_open
mp_board_open
Global variables used to call agents on button actions.
l_pb
l_sb
l_eb
l_spg
l_mpg
l_se
l_ae
l_sp
l_mp
l_se_button
l_ae_button
l_sp_button
l_mp_button
l_projector
l_world
l_se_list
Groups
Features
AutoTest
Favorites
Outputs
Output: General
System
name: mancala
target: mancala
configuration: /Users/fabriziofrasca/git/dose2015/03_games/group1/mancala.ecf
location: /Users/fabriziofrasca/git/dose2015/03_games/group1
compilation: /Users/fabriziofrasca/git/dose2015/03_games/group1/EIFGENs/mancala
tag_concurrency: tag_concurrency_none
console: no
Class
Feature
Outputs
AutoTest Results
Error List
Diagram
Dependency
Metrics
Info

Foreign mates: Recap

Library	Their code seemed well written but it didn't even compile, lucky were just little errors (not creating an object, not resetting a variable). We spent a lot of time during the testing phase because they did not understand well how to set up test and still often their code did not compile, like they did not even try to execute it before pushing. We ended up correcting some part ourselves.
App	They were not so happy to work on the GUI but accepted and made a very nice Interface. They collaborate a lot with us during the final phase, the last day they woke up very early to stay with us all time till the deadline.
Communication	No problem at all. They answered always in little time (considering the time zone) and participated to conversations on Skype and Telegram. At the end they asked for our pictures and congratulated for our nice work and leadership. They got the best mark in this project (considered like a final thesis) with professor Nazareno.

Dose- Romina Miranda
Avrei voluto dire loro che erano molto buoni leader Lucas e Fabrizio
12:58