

Report grading (the pdf has to be provided in the archive file):

Data structures / storage of data sets [2 pts] in Column W and X for comments

[R1] -0.5 pt if the text is a bulky block without a clear structure of paragraphs.

[R2] -1pt if the description of the class hierarchy is missing for “Fourmi” : we expect to have a list of attributes/methods of the superclass and the derived classes

[R3] -0.5pt per missing entity description : **simulation, fourmilere, nourriture, squarecell.**

- Among the provided description, we expect to know where the sets of entities are stored: **nourriture, fourmis, fourmilere.**
- We only ask what types have been designed for **squarecell** (although not recommended it was allowed to design no specific types).

=> **Remove the number of point indicated for each feature that is not achieved**

In the spreadsheets column Report comment, note down the corresponding **code(s)** : e.g.

[R1],[R2],[R3]

Execution grading:

Column : **[Comp]** => **obtaining an executable with make** gives 0.5pts

Column : **[IC] Isolated Commands for correct files (2.00pt)**

Provide c01.txt on the command line as follows:

```
./projet c01.txt
```

Check that the drawing and value of Nb Food in the left column are correct (Fig next page).

Quit the programme with exit after each file ; do the same for c02.txt, c03.txt and c04.txt

- c01.txt : empty simulation ; draw the grid with white border. Nb food value is 0
- c02.txt : Nb Food is 33 ; they draw an EPFL logo in the lower left corner
- c03.txt : Nb food is 0 ; single fourmilier, with 1generator, collector, defensor, predator in RED
- c04.txt : Nb food is 0 ; 25 fourmilieres in a 5 x 5 square. The color order is : red, green, (dark) blue, yellow, magenta (pink), cyan (light blue). Note that yellow is not very visible, that's why in our example we used a kind of grey.

=> 0.5 pt for each successful case

Column : **[DF] Distorsion-free for one correct file (1.00pt)**

Run with one of the correct file that displays correctly and change the window size in the 2 directions to check that we still see the whole simulation space and that there is no distorsion => squares remain squares. See 2 examples next page (bottom).

=> 0.5 pt for each successful direction

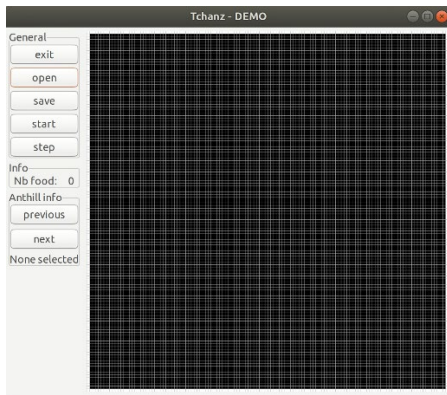
Column : **[NPB] Next and Previous Buttons (1pt)**

Run the program with **c04.txt**. start to check the button “previous” and, after a few clic, check the button “next”. It has to be consistent with the section 2.2.1 of Rendu2: we see the fourmilier index and its current state, and it switches back to “none selected” after going through the max index. In both direction.

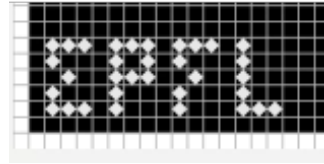
=> 0.5 pt for each button.

Column : **[SSS] Start/Stop/Step Buttons (1pt)**

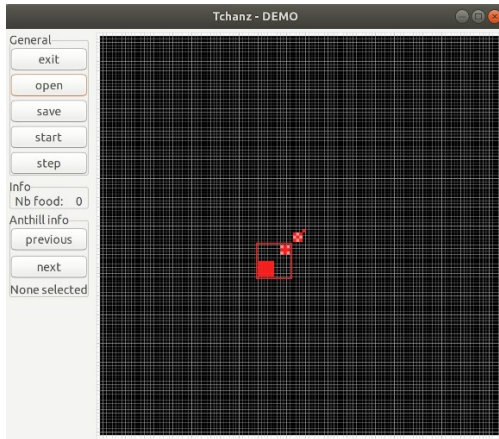
Run the program with **c01.txt**. Check that **start** becomes **stop** and vice-versa (0.25pt). Check that **Start** triggers a timer who display the current value of a counter (0.25pt) and that Stop stops it. Step has to be usable only when the simulation is stopped and to progress by a single unit of the counter (0.5pt).



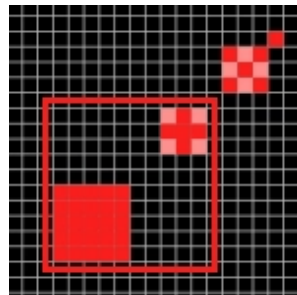
c01.txt



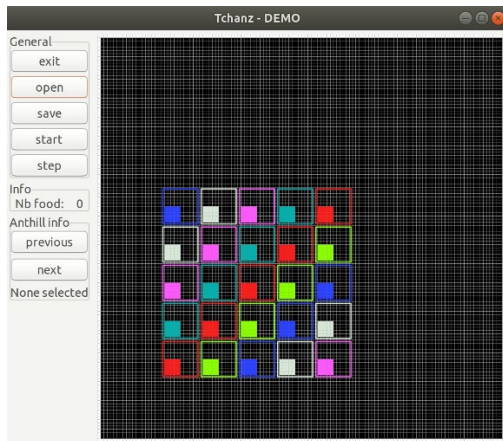
c02.txt (detail)



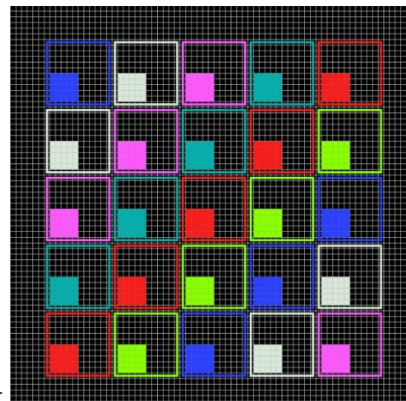
c03.txt



c03.txt (detail)



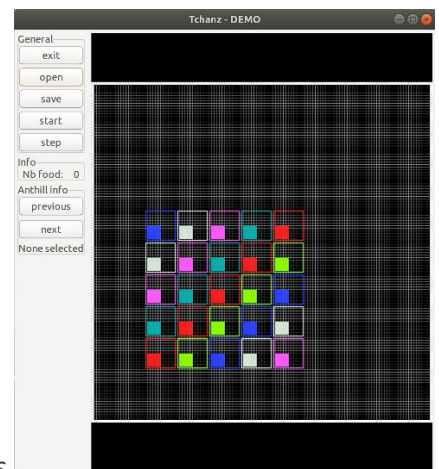
c04.txt



c04.txt (detail)



2 distortion-free cases



Column : [Kb] keyboard (0.5pt)

Run the program with **c01.txt**. Check that 's' does the same as start/stop (0.1), '1' the same as Step (0.1), 'n' the same as Next (0.1pt), 'p' the same as previous (0.1pt) and 'q' the same as Exit (0.1pt).

column : [OSO] Open1-Save1-Open2-OpenSaved (2pts)

Case 1 (1pt):

- run the program with file **c03.txt**.
- use the Save button to save the simulation with the name **ddd.txt** (0.5pt)
- use the Open button on file **c02.txt**. the drawing must be updated (0.25pt)
- use the Open button on file **ddd.txt**. the drawing must be updated (0.25pt)

Case 2 (1pt): same as case 1 but use the start-stop-step button before loading c02.txt

Column : [ED] Error Detection and resulting behavior (1.pt)

- Run the program with file **t01.txt** => The error message should appear in the terminal (pop-up window is ok too) and the program should NOT quit and the drawing area should be empty (0.5pt).
- Use the Open button on file **c02.txt**. the drawing must be updated (0.25pt)
- Use the Open button on file **t23.txt**. the error message should appear etc... (0.25pt)

ARCHITECTURE EVALUATION:**[A1] Architecture features to check for module projet:**

Nb points (max=0.5pts)	Module role / separation of functionalities
[A1] 0.5	Must handle argc and argv ; OK to have one instance of Simulation but no definition of class. It's also normal to have one instance of the interface class.

[A2] Architecture features to check for the Model sub-system:

Nb points	Module role / separation of functionalities
[A2.1] 0.50	Simulation must declare a class ; simulation.h must NOT be included in the lower-level modules ;
[A2.2] 0.50	There must be NO dependency to GTKmm in any Model module
[A2.3] 0.50	The fourmi entities must be managed with a hierarchy of classes ; they can be defined in the same module or different modules

[A3] Architecture features to check for module squarecell:

Nb points	Module role / separation of functionalities (same as rendu1)
[A3] 0.50	The module squarecell has to be independent from higher level modules, including gui, and from GTKmm ; only the include of graphic.h is allowed Ex : including the appendix A = « constantes.h » in the squarecell module is a clear violation of the architecture specification.

[A4] Architecture features to check for module gui:

Nb points	Module role / separation of functionalities
[A4] 0.25	connection with the Model sub-system with simulation.h only but simulation.h can include other interfaces for its own class needs. OK to include squarecell.h

[A5] Architecture features to check for module graphic :

A few groups have been allowed to adopt a variant of the proposed architecture that does not have the **graphic** module. Check the report if this module is not present ; in such a case the gui module gather all the relevant information from the Model to manage the display with GTKmm.

ARCHI pt	If the module graphic is present: Module role / separation of functionalities
[A5] 0.5pt	Same rule as for [A3]: no dependency to higher level of the Model or to gui

The spreadsheet column shows the **default maximum of 2 point** for ARCHITECTURE.
=> **Remove the number of point indicated for each feature that is not achieved, but not more than 2 pts.**

In the spreadsheets column architecture violation comment, note down the corresponding **code(s) : e.g. [A1], [A2.1], [A2.3], [A3]** etc

Class ENCAPSULATION / MODULARIZATION (same as for Rendu1):

[C0] Incomplete implementation: the max number of points is reduced in case of partial implementation. Do not waste time to figure out this in detail ; it should be obvious that a large fraction of the code is missing : ***Report the case to RB who will have a look and calibrate the reduced max.***

[C1] Encapsulation violation : using any **global variable** or making any attribute public is strictly forbidden in any modules, including **public** static attributes (no problem for methods and static methods).

It is allowed to have static variables in the implementation (.cc) of a module or variables declared in the unnamed namespace, or **private** static attribute (indicate a warning if there are too many of them). Indicate a BIG warning in case some static variables appear in the interface of a module.

[C2] Externalization of methods' definition : whenever a module interface shows a class interface, it should contain only method prototypes. The method definition must be externalized in the module implementation.

The only *accepted exception* of method definition in the class interface are the **constructors** or **getters** methods that fits onto the same line as the function prototype.

The spread sheet column AJ shows the **default maximum of 3 points**.

=> **Remove 1 point per public attribute or global variable** (max 2pt).

=> **Remove 1 point per interface that is not correctly externalized** (max 2 pt).

The total of removed points from C1 and C2 is maximum 3 pts.

In the spreadsheet column Encapsulation violation_comment, note down the corresponding **code [C1],[C2]** together with the **interface name** and the **public attribute name**. Indicate that it must be corrected in the next assignment.

CODING STYLE: less criteria for Rendu2 to spare time for execution tests

[L1] Indentation rules have been ignored **more than 4 times** ; read carefully [the conventions](#) before considering this penalty because we accept some variants. Please note that we don't indent the public/private keywords in class declaration. Indicate only a **warning** if the whole code is consistent in the use of multiple brace styles (e.g. two styles are used but always in the same way, for the same control instructions)

[L2] There are **more than 4 wrapping line** in the code (more than 87 char); Indicate only a warning if 4 wrapping lines or less.

[L3] Apart from two functions of max 80 lines, all function size must not exceed 40 lines (+tolerance of 2 lines) with geany (with the default font size). Recommend to apply the principle of abstraction in case of too long functions.

The spreadsheet column shows the default maximum of **4** points for STYLE

=> **remove 1 point max for [L1]**

=> **remove 1 point max for [L2]**

=> **remove 1 point per function that is too long [L3]**

In the spreadsheet column AM violation_list, note down the **code** representing the violated criteria followed by the **filename** and the **line number** it occurs. For instance **[L2]simulation.cc57,65,80-84** means that this set of lines are violating the wrapping criteria in the file simulation.cc. If the same type of violation occurs more than 5 times, you mention briefly how much larger the problem is in the violation comment column AH

Keep the violation_list alphabetically sorted and separate each entry by a comma.