

## Report grading by RB (max 3pts)

1pt for the description of the remaining strategies to put in place

1pt for describing 2 examples documented with images and text

1pt for documenting the working methodology, self-assessment and providing feedback

---

### Execution grading:

The execution is evaluated with the set of provided test files (public and private). In case you want to double check to contest your grade, proceed as follow before sending any message:

- Download the archive file from moodle on the VM
- Make the executable
- Run the test as documented below
- Make screen capture for each step of the test

If you still see a difference regarding the grade, provide all the screen captures with a message documenting *precisely* where you spot the difference.

-----

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

Column : [CL] **Command Line without argument (0.5pt)**

Start the project as follows:

```
./projet
```

It should start by providing the interface with an empty world (an empty grid is fine).

Column : [NC] **NoCrash (1pt)**

Open **c04.txt** and press the Start button ; let it run until the simulation makes some progress => we see the creation of food and of the 3 types of fourmis (3x0.25pt) and at least one fourmilere disappear (0.25pt) without crashing.

**IMPORTANT: *detecting if some execution specific to rendu3 has been done***

This test serves also to detect if the project has done *anything more* compared to the rendu2. In case nothing at all happens in the drawing window for this execution test with c04.txt, RB examines the code to also decide a potential reduction of the code grade.

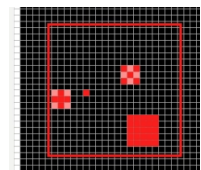
Column : **[FC] Food Creation (1pt)**

Open **c01.txt** and press the Start button ; let it run for 1 min.

As discussed on Discourse, we cannot really count the number of food element created and expect a sharp 1/10 ratio. So here we give the point if there is food created (0.25pt), the food counter is updated (0.25pt) and if the food distribution in space “look” random (0.5pt).

Column : **[DG] Death of Generator / Fourmilere (0.50pt)**

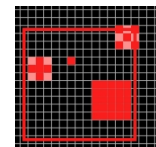
Open **f01.txt** and press the Step button only once. The existing Fourmilere that was visible at file opening disappear and the world is empty.



Init state f01.txt

Column : **[GM] Generator Movement (0.5pt)**

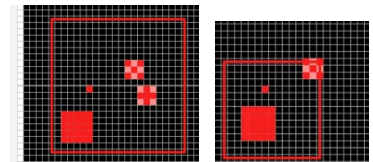
Open **f12.txt** and press the Step button only once . the generator has to make one movement to the left to avoid being on the fourmilere boundary.



Possible final state f12.txt

Column : **[DD] Death of Defensor (0.50pt)**

Open **f02.txt** and press the Step button only once. The existing Defensor that was visible at file opening disappear and the Fourmilere remains with 1 collector and one predator.

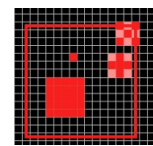


Init & one step of f02.txt

Movements are not graded

Column : **[MD] Movement of Defensor (0.5pt)**

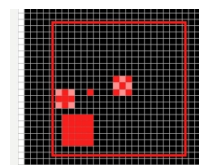
Open **f13.txt** and press the Step button only once. The existing Defensor has to make one movement to the left to avoid staying on the fourmilere boundary



Final Defensor state f13.txt  
Other movements are not graded

Column : **[TF] Total\_Food / Fourmilere (1.00pt)**

Open **f03.txt** ; select the first fourmilere with the next button to see its current state. Then press the Step button 3 times. Check the value of total\_food that should decrease to **0.6** (0.25pt) and **0.2** (0.25pt) and finally the Fourmilere should disappear on the third step (0.5pt).

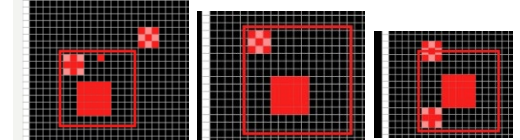


Init state f03.txt,  
Movements are not graded

*NOW edit the source code to set **food\_rate** to zero in the file **constantes.h** before the next tests to prevent the Collector to move too much*

Column : **[CFF] Creation Fourmi FREE (1.50pt)**

Open **f04.txt** ; select the red fourmiliere with the **next** button to see its current state. Then press Step: the initial 3 fourmis disappear (0.25pt) and 1 collector appear (0.25pt). The next Fourmi to appear is a Defensor (0.25pt), then 5 Collectors in 5 steps (0.5pt), then a Predator (0.25pt).

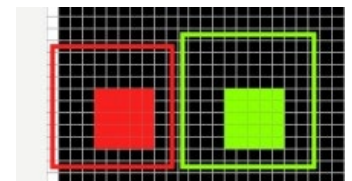


Init & 1 & 2 steps of f04.txt

Movements are not graded but the Generator movement should not prevent the creation

Column : **[CFC] Creation Fourmi CONSTRAINED (1.00pt)**

Open **f05.txt** ; select the red fourmiliere with the next button to see its current state. Then press Step. We should get a Collector (0.25pt), a Defensor(0.25pt), a Collector (0.25) and a Predator(0.25pt). the rest of the sequence is not graded.

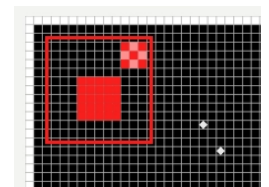


Init state f05.txt,

Movements are not graded but the Generator movement should allow the first 3 creations

Column : **[CF06] Collect Food06 (1.25pt)**

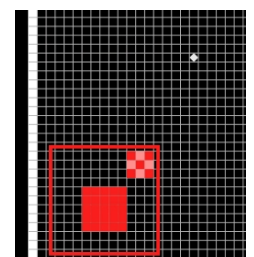
Open **f06.txt** ; select the red fourmiliere with the **next** button to see its current state. The closest food is reachable in straight line in 7 steps (0.25pt), the collector is displayed in LOADED state (0.25pt), it comes back to contact to the fourmiliere (0.25pt) to deliver food (0.25pt) and total\_food increases (0.25pt). The second food is not reachable.



Init state f06.txt

Column : **[CF07] Collect Food07 (0.50pt)**

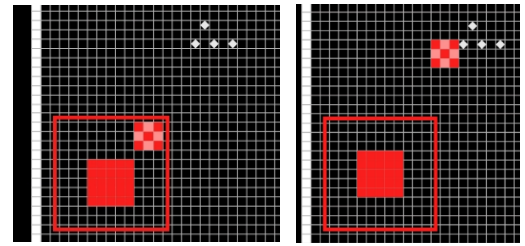
Open **f07.txt** ; select the red fourmiliere with the **next** button to see its current state. The closest food is reachable in 9 steps (0.25pt), the collector comes back to contact to the fourmiliere (0.25pt). We allow some variety of movements, including a zig-zag variant.



Init state f07.txt

Column : **[CF08] Collect Food08 (0.75pt)**

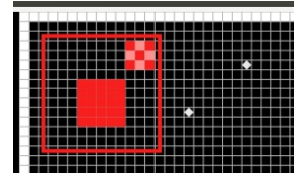
Open **f08.txt** ; The unique reachable food is the top one but both paths are blocked by the other food elements. The least blocked path (with less overlapping) is the one moving **up-left** first (0.5pt). The Collector remains blocked while following this path (0.25pt)



Init & blocked states f08.txt

Column : **[CF09] Collect Food09 (1.00pt)**

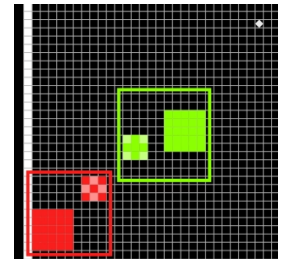
Open **f09.txt** ; The unique reachable food is the top-rightmost one but the bottom path is blocked (at init stage) by the other food element. For the first step, the collector has to start moving rightward towards the world top side (0.5pt). After the first step a variety of movements are accepted to reach the food and bring it back (0.5pt).



Init state f09.txt

Column : **[CD10] Collector-Defensor (0.50pt)**

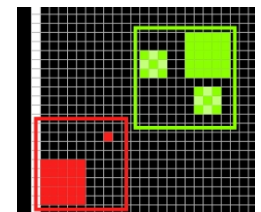
Open **f10.txt** ; The Collector moves in straight line along the diagonal towards the unique closest food, then at the second step its corner is in contact with the corner of the green Defensor that makes it disappear (0.5pt).



Init state f10.txt

Column : **[CPC] Collector-Predator CONSTRAINED(0.50pt)**

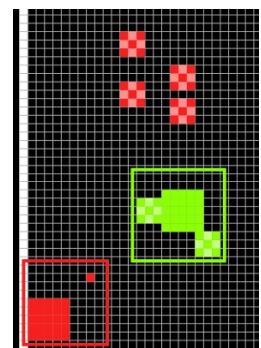
Open **f11.txt** ; The red fourmilierie is in CONSTRAINED mode, so its predator moves towards the closest green Collector (top left)(0.25pt). It takes a few steps to reach the Collector and destroy it (0.25pt).



Init state f11.txt

Column : **[CPM] Collector-Predator Movements (1.00pt)**

Open **f14.txt** ; first the red fourmilierie is in CONSTRAINED mode because it cannot extend (keeps its initial size=0.25pt). The green fourmilierie is in FREE mode: its Collectors should move out (0.25pt). The predator should chases them (0.25pt). after 10 Steps the red Collectors disappear and the red fourmilierie is back to the FREE mode => the predator must come back home (0.25pt).



Init state f14.txt

**ARCHITECTURE EVALUATION: (*only if some execution specific to rendu3 has been done*)****[A1] Architecture features to check for module projet:**

Nb points (max=0.5pts)	Module role / separation of functionalities
[A1] 0.5	Must handle <b>argc</b> and <b>argv</b> ; 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.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 <b>fourmi</b> entities must be managed with a <b>hierarchy of classes</b> ; 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&2)
[A3] 0.50	The module <b>squarecell</b> has to be independent from higher level modules, including gui, and from GTKmm ; only the include of <b>graphic.h</b> 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 <b>Model</b> sub-system with <b>simulation.h only</b> 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&2):

*(only if some execution specific to rendu3 has been done)*

**[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 shows the **default maximum of 2 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 2 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: same as Rendu2 to spare time for execution tests**

***(only if some execution specific to rendu3 has been done)***

**[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 BR shows the default maximum of **4** points for STYLE

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

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

=> **remove 0.25 point per function that is too long [L3]**

In the spreadsheet column BS 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 global comment column BT

Keep the violation\_list alphabetically sorted and separate each entry by a comma.

-----

**In case you want to contest your code grade (architecture, encapsulation, style), proceed as follow before sending any message:**

- Download the archive file from moodle
- Open the code with geany on the VM
- Make screen capture of the feedback associate to the grade
- Make as many screen capture as necessary on geany to document the issue

If you still see a difference regarding the grade, provide all the screen captures with a message documenting *precisely* where you spot the difference.

**We don't accept any message comparing your code grade to another anonymous project**  
**We may re-assess globally your project, hence resulting in a lower grade.**