

## Execution grading:

Column [Arg] 0.5pt => obtaining an executable with `make` gives 0.25pts

**If it does not work at the first trial, remove 0.25 point.** Then try to fix the bug if it is obvious (less than 1 min search in the Makefile or the code). If you feel it requires more work, contact immediately **Ke** at [k.wang@epfl.ch](mailto:k.wang@epfl.ch) so that he asks the group to upload a new version of the archive file (put me in CC). you can use the discord channel for confidential comments.

The other 0.25pt is obtained if **the project can be executed without providing a filename argument on the command line: `./projet`**

The drawing area should be empty.

A penalty of 0.5 pt is applied if no test filename can be provided on the command line.

### Column t28 : rotation sans collision (0.5pt)

Faire tourner avec start ; les coraux tournent dans des sens différents puis meurent après en gros un tour et demi. Le nombre de coraux ne change pas car ils ne sont pas mangés après leur mort.	<table border="1"> <tr><td>mise à jour:</td><td>0</td><td></td><td>357</td><td></td></tr> <tr><td>algues:</td><td>0</td><td></td><td>0</td><td></td></tr> <tr><td>corails:</td><td>2</td><td></td><td>2</td><td></td></tr> <tr><td>charognards</td><td>0</td><td></td><td>0</td><td></td></tr> <tr><td colspan="3"></td><td>fin</td><td></td></tr> </table>	mise à jour:	0		357		algues:	0		0		corails:	2		2		charognards	0		0					fin	
mise à jour:	0		357																							
algues:	0		0																							
corails:	2		2																							
charognards	0		0																							
			fin																							

### Column t29 + t30 : rotation avec auto-collision (0.5pt)

Rotation du second segment avec superpositions et 3 changements de direction puis mort du corail. Ok pour un décalage de 1-2 step.	<table border="1"> <tr><td></td><td>50</td><td></td></tr> <tr><td></td><td>80</td><td></td></tr> <tr><td></td><td>0</td><td></td></tr> <tr><td></td><td>1</td><td></td></tr> </table>		50			80			0			1	
	50												
	80												
	0												
	1												
t29: (trigo, inv_trigo, trigo) 1ière collision à 50, 2ième à 149, 3ième à 248.													
t30: (inv_trigo, trigo, inv_trigo) 1ière collision à 80, 2ième à 178, 3ième à 276.													

### Column t31 : rotation avec collision 2 bords (0.5pt)

Rotation avec collisions avec 2 bords produisant 3 changements de direction aux steps 11, 27 et 43. Ok pour un décalage de 1-2 step. -0.25pt si 4 changements car mauvais test de l'effecteur vis à vis du bord.	
---	--

### Column t32 + 33 : rotation avec collision de 2 coraux (0.5pt)

Rotation avec collisions avec 2 coraux produisant 3 changements de direction. <i>Des décalages importants sont tolérés car cela dépend de l'ordre des tests</i>	<table border="1"> <tr><td></td><td>4</td><td></td><td>61</td><td></td><td>118</td></tr> <tr><td></td><td>0</td><td></td><td>0</td><td></td><td>0</td></tr> </table>		4		61		118		0		0		0
	4		61		118								
	0		0		0								
t32: aux steps 4, 61 et 118													
t33: aux steps 29, 96 et 164													
	<table border="1"> <tr><td></td><td>29</td><td></td><td>96</td><td></td><td>164</td></tr> <tr><td></td><td>0</td><td></td><td>0</td><td></td><td>0</td></tr> </table>		29		96		164		0		0		0
	29		96		164								
	0		0		0								

**Column t34 + 35 : rotation avec collision de 2 coraux (0.5pt)**

<p>Rotation avec collision inter-coraill (step 8) puis intra-coraill (step 26) puis inter-coraill (step 44). Ok pour un decalage de 1-2 step.</p> <p>t34 : aux steps 8, 26 et 44</p> <p>t35 : aux steps 4, 36 et 54</p>	
---	--

**Column [alg] : t36-t40 : consommation d'algue (2.5pt= 5 x 0.5pt)**

<p><b>t36:</b> 2 steps pour manger une algue ; la longueur du corail augmente</p>	
<p><b>t37:</b> 2 steps pour manger une algue , puis 2 autres pour manger la seconde.</p>	
<p><b>t38:</b> Idem t37 mais creation d'un nouveau segment qu'on voit en faisant 2-3 steps de plus</p>	
<p><b>t39:</b> Idem t38 mais creation d'un nouveau coraill ; ok si au step 4 ou 5</p>	
<p><b>t40:</b> Consommation de l'algue angulairement la plus proche (celle de droite) au step 2 puis 3</p>	

**Column t41 : alternance creation coraill et ajout segment (1pt)**

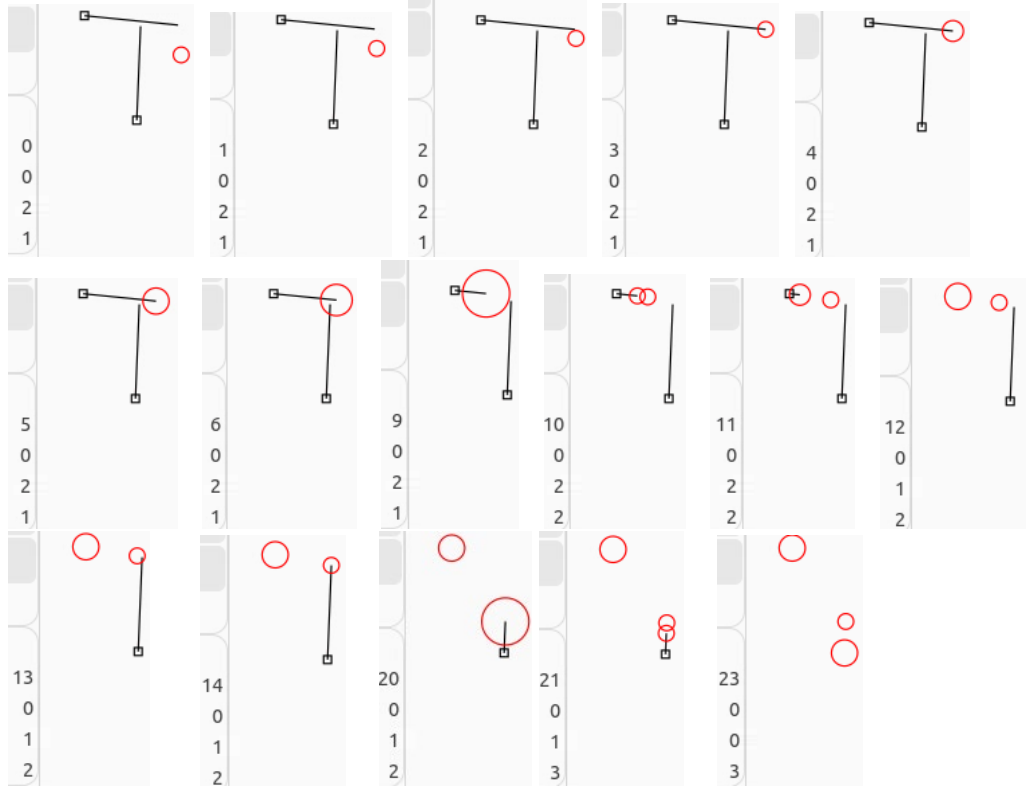
<p>D'abord création d'un nouveau coraill puis creation d'un second segment du coraill parent. Situation init:</p> <p>Info : nombre de... mise à jour: 0 algues: 7 coraills: 1</p> <p>Ce qui importe est l'ordre , on accepte des variantes pour l'indice de mise à jour.</p>	<p>Info : nombre de... mise à jour: 1 2 3 4 algues: 7 6 5 4 coraills: 1 2 2 2</p>
--	---

**Column t42-t43 : extension refusée (1pt)**

<p><b>t42:</b> Extension refusée au step1 à cause collision du bord ; changement de direction (step2)</p>	
<p><b>t43:</b> Extension refusée au step1 à cause collision coraill ; changement de direction (step2)</p>	

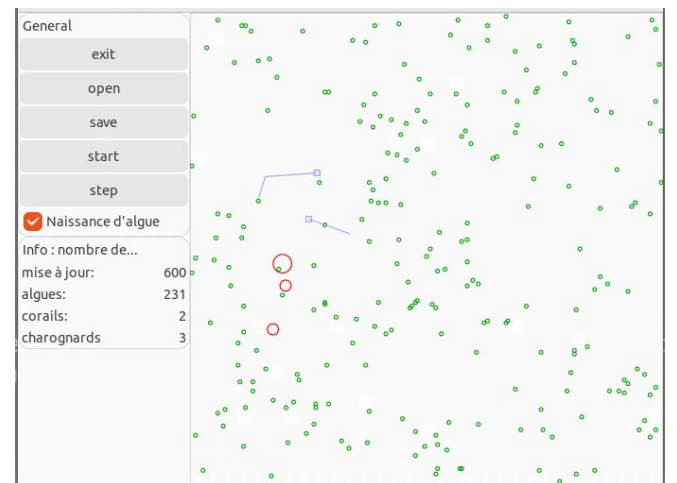
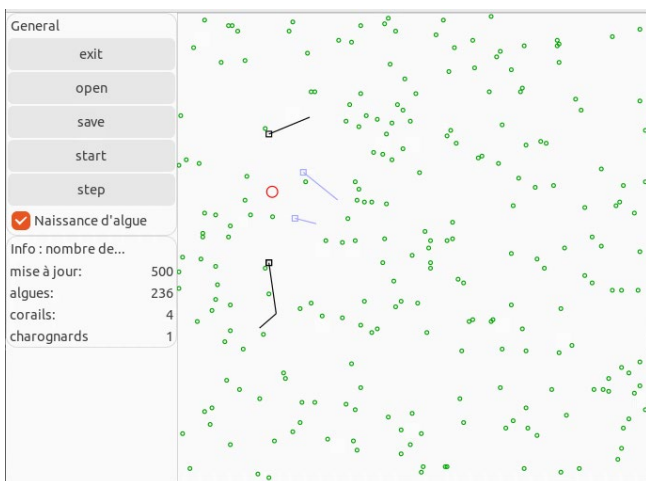
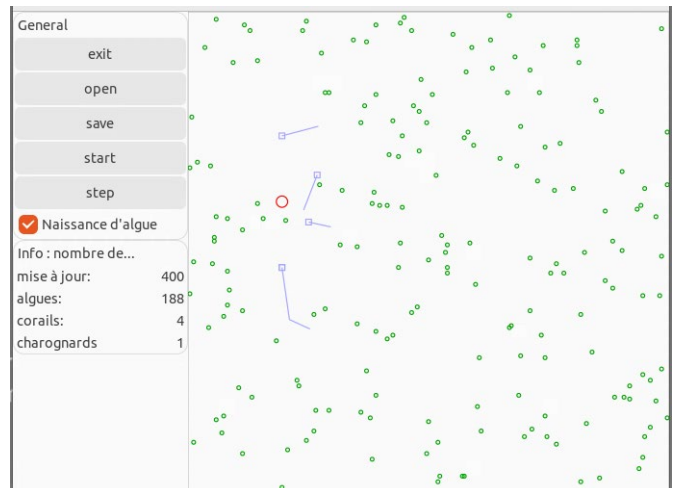
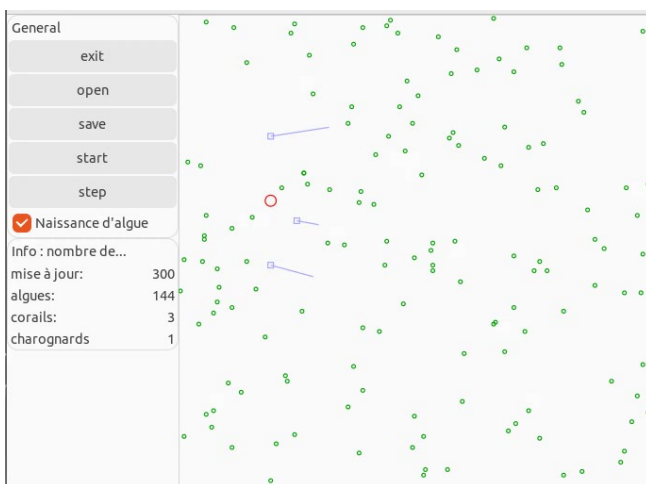
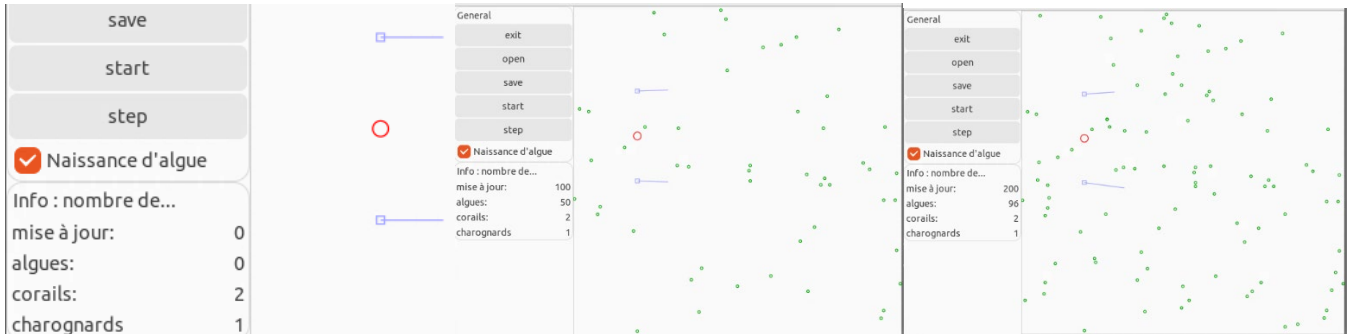
### Column [scav] : déplacement, consommation corail et creation scavenger (1.5pt)

t44: Scavenger se dirige vers l'effector le plus proche d'un corail mort (0.25pt) ; puis le consomme (0.25pt) et se reproduit (0.25pt) ; le nouveau scavenger se dirige vers l'autre corail mort (0.25), le consomme (0.25pt) et se reproduit(0.25pt) aussi.



*Activate the checkbox creating some algae.*

**Column t45 : test perf light avec creation d'algues (1pt)**

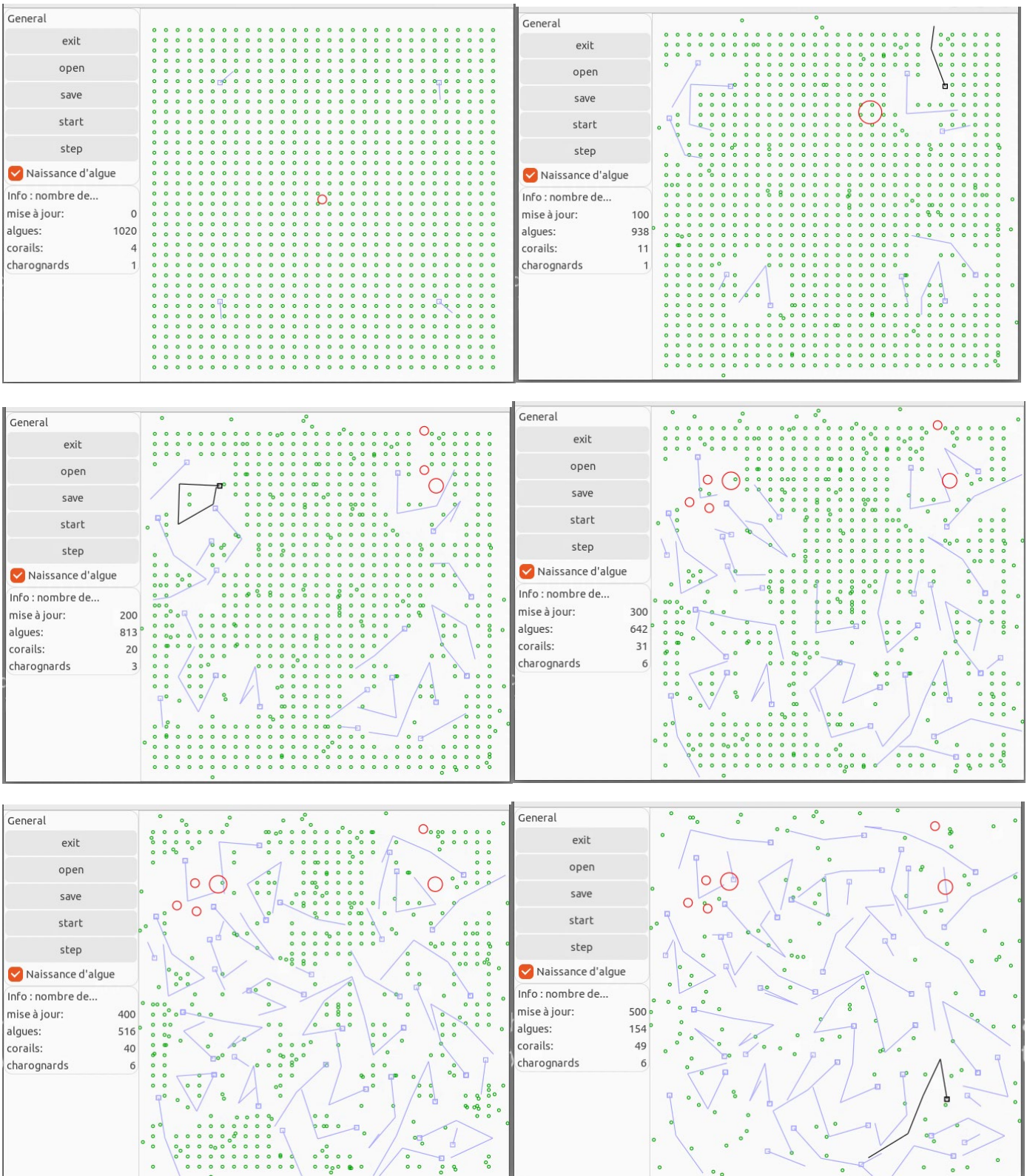


0.5pt if smooth execution (check timer value if too slow ; should be around 25ms)

0.5pt for no crash until first corail eaten by scavenger



## Column t46 : test perf lourde avec creation d'algues (1pt)



0.25pt if smooth execution (check timer value if too slow ; should be around 25ms)

0.5pt if no self-intersection of  $n^{\text{th}}$  segment with one of the  $n-2$  first segments (not tested before)

Ex: top left corner around step 127 with a 3-segments-corail

0.25pt if no crash until second dead corail is eaten

**ARCHITECTURE EVALUATION:**

Nb points (max=0.5pts)	Module role / separation of functionalities
[A1] 0.5	Must use <b>argc</b> and <b>argv</b> , at least to check whether there is an argument.

**[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 <b>lifeform</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 shape:**

Nb points	Module role / separation of functionalities (same as rendu1)
[A3] 0.50	The module <b>shape</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 shape 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</b> only but simulation.h can include other interfaces for its own class needs. OK to include shape.h and gtkmm.h

**[A5] Architecture features to check for module graphic :**

ARCHI pt	If the module <b>graphic</b> 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 BE 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 BF architecture violation comment, note down the corresponding **code(s) : e.g. [A1], [A2.1], [ A2.3], [A3]** etc

#### 4. CLASS ENCAPSULATION / MODULARIZATION: same as rendu1&2

**[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 BG shows the **default maximum of 2 points** for **CLASS**.

=> **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 BH class 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 future assignments.

## 5. CODING STYLE: less criteria as 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).

Note: it is OK that “**case**” is not indented in the **switch** block but controlled instructions have to be indented.

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

**[P2]** Apart from two function/method of max 80 lines, all function/method 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 function/method.

The spreadsheet column BI 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/method that is too long [P2]**

In the spreadsheet column BJ 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 general comment column BK.

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