3. Execution tests on the spring VM

3.1 Column **"executable from `make`"** (0.5pt)

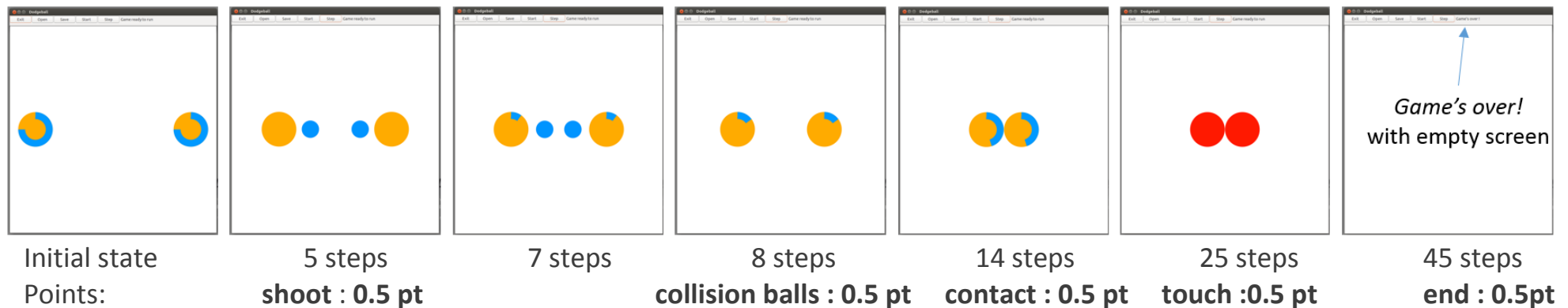*If it does not work at the first trial, remove the 0.5 point.*

Accepted variants throughout the tests:
- When opening a new file with the Open Button, it is **OK** to automatically reset the Start/Stop Button to "**Start**" state
- The specification states *not to move* when a **collision** is detected; it is **OK** to have a slightly bigger **gap** compared to the demo program.
- When two paths of equal length are available ; it is **OK** to make a different choice compared to the demo
- The demo implemented a more sophisticated *spot* choice compared to the specs ; **OK** for moving along a slightly different direction along the shortest path towards the player target.
- When the game ends, it is **OK** to freeze the simulation : what is important is to get the **"Game's over ! "** message

3.2 Column AZ: **[SSO] Step Sans Obstacle(2.50pt)**.

Start the project with F01.txt, either through the command line or the Open button.

The 2 players have the same **count of 15** ; use the **Step** button to run the simulation ; you should obtain this sequence :

Initial state	5 steps	7 steps	8 steps	14 steps	25 steps	45 steps
Points:	shoot : 0.5 pt		collision balls : 0.5 pt	contact : 0.5 pt	touch :0.5 pt	end : 0.5pt

Game's over! with empty screen

Note: it is still OK if the message "Game's over!" appear after one more step

**3.2** Column BA: **[CLSM] Command Line Step Mode (1.5 pt)**

Start the program from the command line as follows:

```
./projet Step F02.txt out.txt
```

Open the file **out.txt** in geany and compare it with the provided file **F02_after_one_step.txt.** The position values should qualitatively match (format can slightly differ) to express that players move towards each other => **0.5 pt.**

The counter should be reset to 0 => **0.5 pt**

it's important to have two balls in out.txt with opposite angles => **0.5 pt**

F02 initial state   **after one step**
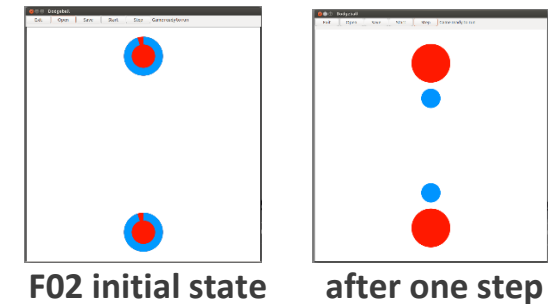
3.3 Column BB : Player chase (2 pts)

- **[S2P] Start 2 Players (1 pt)**

Open **F03.txt** and run the simulation with Start ; it should fire once before getting to contact, then turn to orange and red after the 2 counters reach their max value ; finally both should disappear and we should see the message "Game's over !" => **0.5pt**

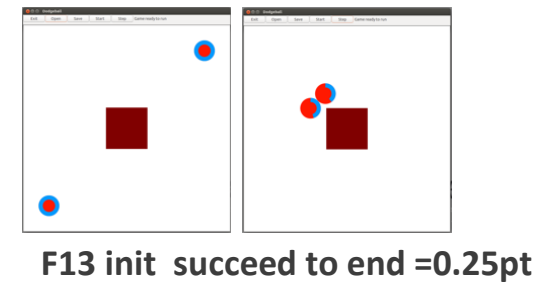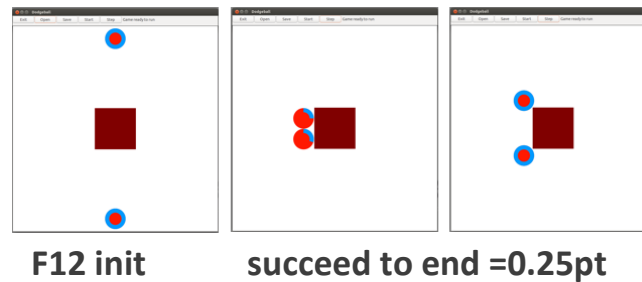The same for **F04.txt** (different init position + start with green color) => **0.5pt**

F03 init state   **F04 init state**

- **[CCP] Chase Closest Player(1pt)**

<u>4 players</u>: open **F05.txt** and run the simulation ; check that each player moves towards its closest neighbor (2 possible variants) => **0.5 pt**
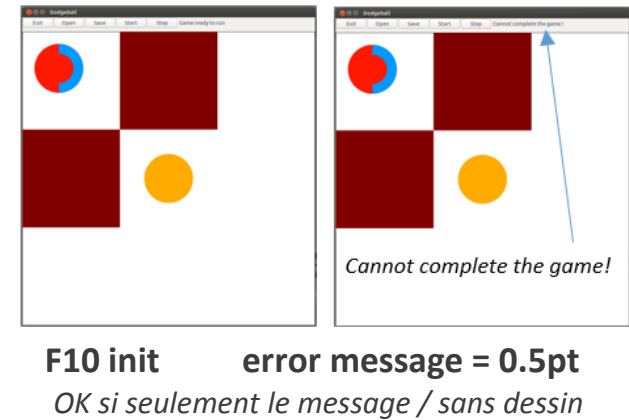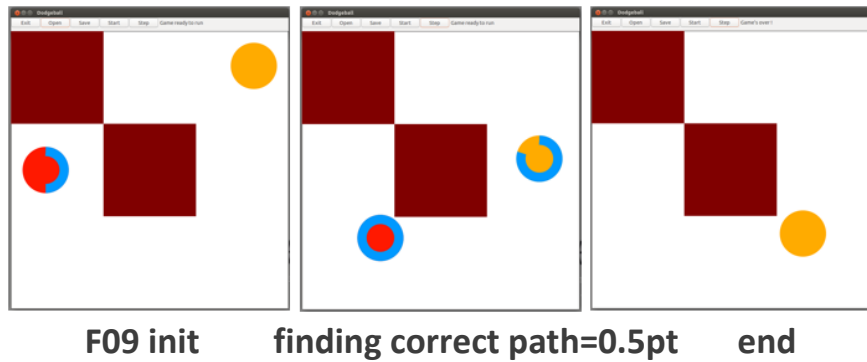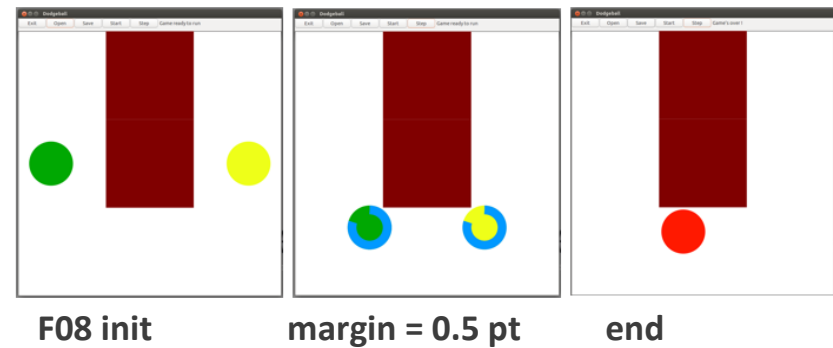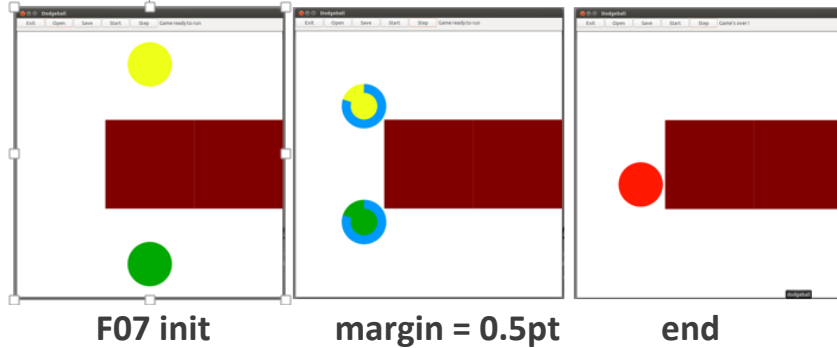
<u>8 players</u>: open **F06.txt** and run the simulation ; check that each player moves towards its closest neighbor => **0.5 pt** (no penalty in case you have to do Start/Stop to see the simulation ; see [P])

F05 init / expected behavior   **F06 init / expected behavior**
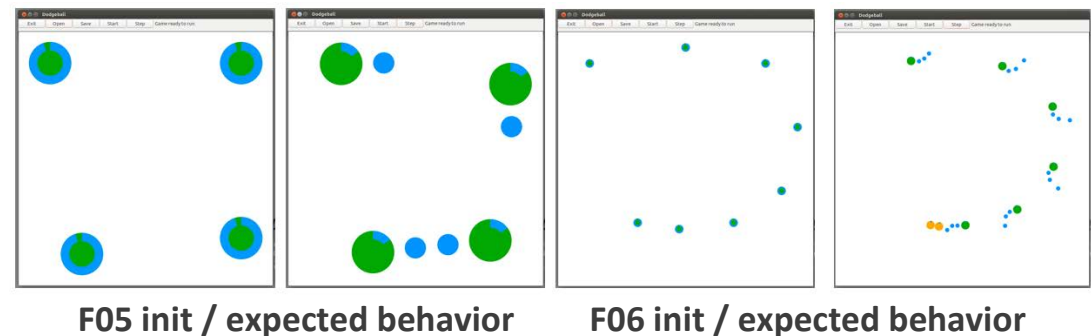
3.4 Column BC: [CO] Constant Obstacle (2.75pt)

This section focuses on the obstacle avoidance with the safety margin with F07(=0.5pt) and F08(=0.5pt), finding a path (F09=0.5pt) or not (F10=0.5pt), and finally allowing to end the game in finite time (0.25pt each for F11, F12 and F13).

F07 init   **margin = 0.5pt**   **end**   **F08 init**   **margin = 0.5 pt**   **end**

F09 init   **finding correct path=0.5pt**   **end**   **F10 init**   **error message = 0.5pt**

OK si seulement le message / sans dessin

Cannot complete the game!

F11 init   **succeed to end =0.25pt**   **F12 init**   **succeed to end =0.25pt**   **F13 init succeed to end =0.25pt**
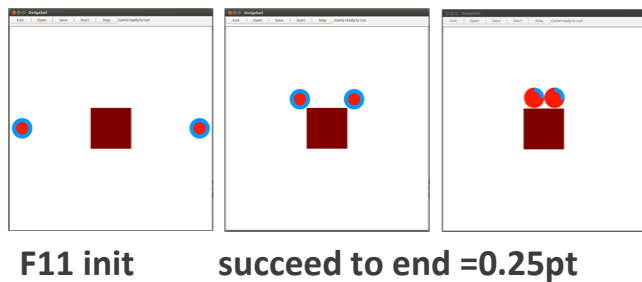
3.5 Column BD: **[VO] Variable Obstacle (1.5pt)**

This section focuses on the update of the shortest path when one (F14= 0.5pt, F016=0.5pt) or more obstacles (F15=0.5pt) disappear.
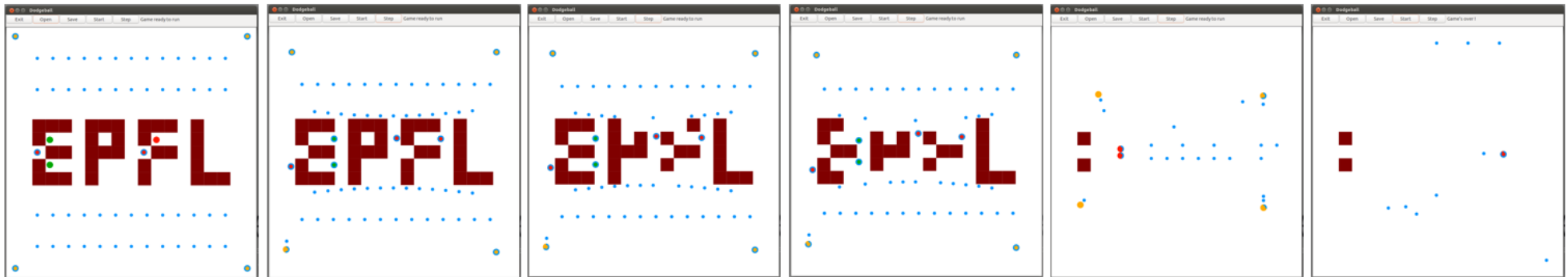

F14 init   before hit   hit=>change path (0.5pt)   near end

F15 init   before hit   1ˢᵗ change (0.25pt)   2ⁿᵈ change (0.25pt)

F16 init   before hit   hit=>change path (0.5pt)   near end
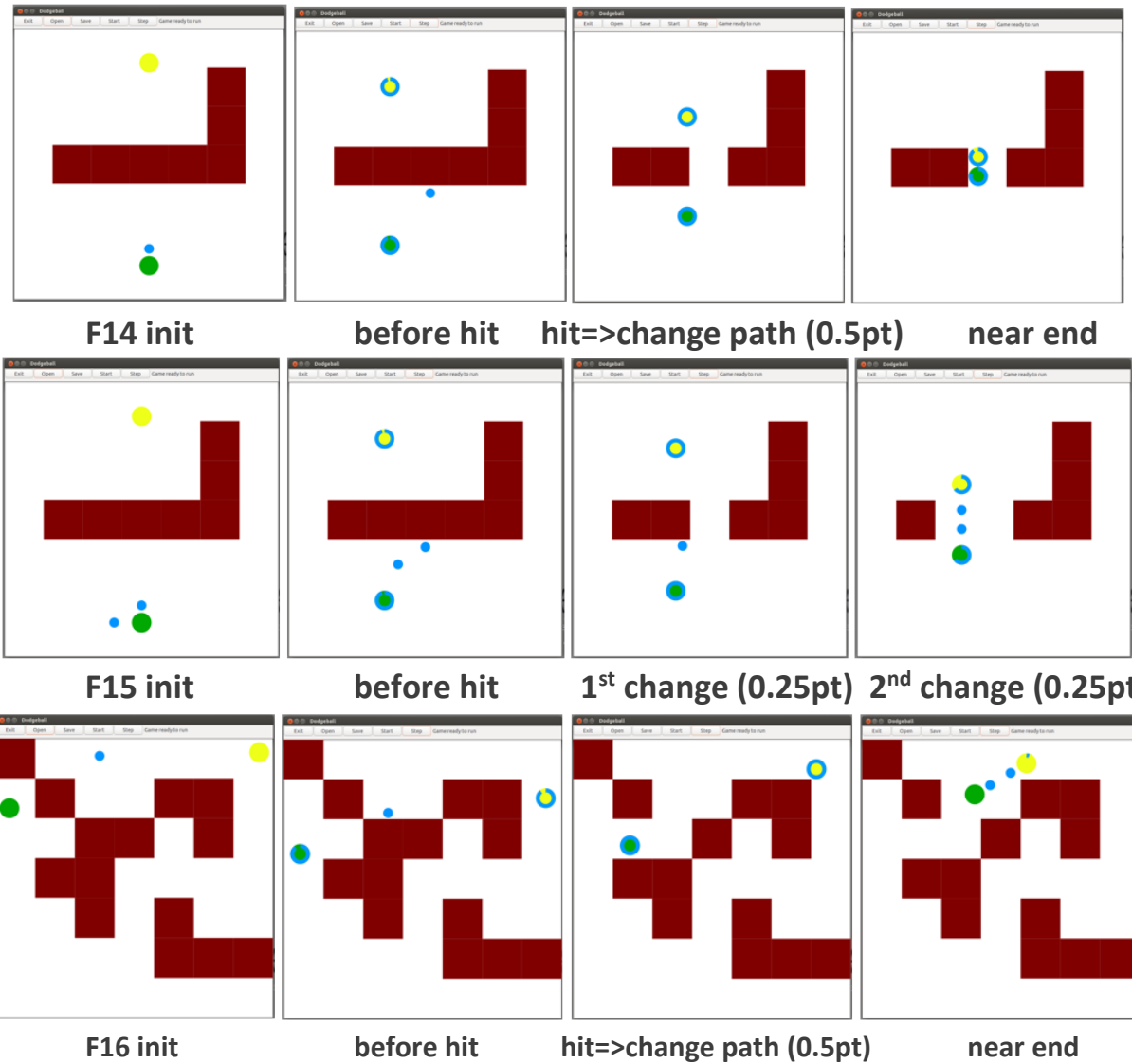
3.6 Column BE: **[P] Performance (1.25pt)** This test assesses the robustness to multiple changes during the simulation

The simulation should terminate in finite time ; variants are OK due to potential local path direction difference with the demo.

No point if it is necessary to wait more than 1 min for the first step (start evaluating the source code during that time)
Penalty of 0.5 pt if it is necessary to alternate Start/Stop or use Step to see the progress of the simulation.

4. Rendu3 ARCHITECTURE EVALUATION: (much less than rendu2 to spare time for execution tests)

The spreadsheet column AR shows the default maximum of **1** point for ARCHITECTURE.

[A2] <u>**Architecture features to check for the Model sub-system**</u>:

Nb points	Module role / separation of functionalities
[A2.2] 1	There must be NO dependency to GTKmm in any Model module

=> **Remove the point if you find a dependency expressed by including one header file from GTKmm**

In the spreadsheets column AS architecture violation_comment, note down **[A2.2]** and the header file included in the Model

5. CLASS ENCAPSULATION / MODULARIZATION: (same as rendu2 but smaller total penalty)

The spreadsheet column AT shows the **default maximum of 1 point** for **Class encapsulation**.

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

It is allowed to have static variables in the implementation of a module (indicate a warning if there are too many of them).

No problem for struct <u>model</u> made visible in tools.h by the tools module.

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

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

The *2 accepted exceptions* are:
- **getters** methods that fits onto the same line as the function prototype.
- **constructors** limited to an initialization list.

=> **Remove 0.5 point per interface that is not correctly externalized** (max 1 pt).

The total of removed points from C1 and C2 is maximum 1 pt.

In the spreadsheet column AU class violation_comment, note down the corresponding **code [C1], [C2]** together with the **interface name** and the **public attribute name**.

6. CODING STYLE (less than rendu2 to spare time for execution tests)

The spreadsheet column AV shows the default maximum of 2 points for STYLE

[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)

     ABOUT constructor initialization list: we have no convention at the moment ; avoid evaluating indentation of this aspect.

     ABOUT switch: we have no convention for indenting a **block associated to a case** compared to the simpler layout without block. Such a block is necessary when we need to declare local variables ; avoid evaluating this configuration.

 [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.

=> remove 1 point per coding style criteria that is violated (max 2 pt)

In the spreadsheet column AW violation_list, note down the **code** representing the violated criteria followed by the file name and the **line numbers** it occurs. For instance **[L2] gui.cc 57,65,80-84** means that this set of lines are violating the wrapping criteria in gui.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 AX

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