

A generic renderer engine for evo-devo models

Sylvain Tournois, Sylvain Cussat-Blanc, Jonathan Pascalie, Yves Duthen

Université de Toulouse - IRIT - CNRS - UMR 5505

sylvain.tournois@gmail.com, sylvain.cussat-blanc@irit.fr, jonathan.pascalie@irit.fr, yves.duthen@irit.fr

Abstract - Introduction

Whereas cell-based models have very different simulation mechanisms, the graphical representation tends to be always the same. In order to save the work of graphical interface development, this paper presents a prototype of a generic multi-platform renderer engine for evo-devo models. It proposes a list of configurable cell states and animations put together in a simulation file that describes the simulation story.

Description of the functioning

Split of the simulation into 3 files

This renderer is based on a 3 human-readable files that describe the simulation. The *state file* describes all possible states of a cells. It gives the list of the possible cell states and their properties such as the name, the color of the cell nucleus, of the cytoplasm and of the membrane. The *action file* contains of all possible cell actions in the simulation. Each action is linked to an animation that will provide a graphical representation. The current version proposes 15 of the most important animations (NOOP, mitosis, move, absorb, reject, apoptosis, etc.). More animations can be easily added. Once the states and the actions are described, the *data file* presents the sequence of animations and states changes that happen during the simulation.

The data file

The sequence of actions, contained in the data file, describes the whole simulation. Each line of the file describes a simulation step. Each line begins with an integer value that represents the simulation step number. Following the number, all cells that change their state are given. A cell is represented by a unique identifier. If the cell position in the environment has

been modified without any action (add of a new cell for example), the coordinate can be given. A new identifier automatically create the cell in the given position in the environment. Then, the new state of a cell is given. Finally, the action and its progress (optional, usefull to pause or to control the speed of the action) can be applied to the cell. This action will trigger the corresponding animation, as described in the action file.

Functionalities

The renderer engine allows an *a-posteriori* visualization of an evo-devo simulation. This is an advantage because it is possible to navigate backward and forward on the simulation timeline. A direct access to every simulation step is then possible. Step by step progress in the timeline, simulation pause and stop are other possible controls. Zoom in and zoom out are also available to visualize more details on a subpart of the simulation. Chapters can be added in the data file in order to go directly to a particular event.

Conclusion and Future work

This prototype of simulation player has been tested on three very different evo-devo models develop in our own research team. The integration was very easy and required less than one day of work. Figure 1 illustrates two examples of utilization of the renderer engine with two of these models [1,2]. This software will be available in the next months on the website <http://www.blabla.fr>.

Many functionalities can be added to this rendering software. The ergonomics of the renderer can be improved in order to get it closer to a conventional video player (simulation loading procedures, possible exports, etc.). A multi-layer visualization could also be interesting to implement in order to visualized different aspects of the simulation (physical, chemical, decomposition of parallel tasks, etc.). Each layer has to be enough generical in order to represent all kind of simulations. Finally, an interface could be imagined in order to connect directly the renderer engine to the simulator. The engine will keep all the features previously presented. The model could send each time step to the renderer, which build on the fly the timeline. The step forward feature will not be accessible anymore but it will be possible to navigate in the past of the simulation.

References

- [1] S. Cussat-Blanc, H. Luga, Y. Duthen. From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134–141. MIT Press, 2008.
- [2] J. Pascalie, V. Lobjois, H. Luga, B. Ducommun, Y. Duthen. A Checkpoint-Orientated Model to Simulate Unconstrained Proliferation of Cells. In *ECAL'11*, MIT Press, 2011.

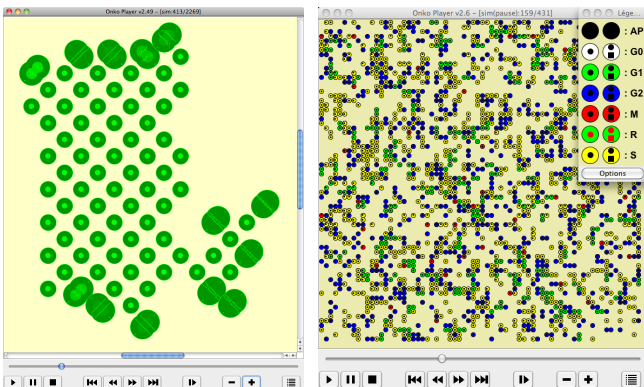


Figure 1: Example of obtained visualization with the renderer engine and two different evo-devo models.