Simulate plants: a client-server graphic approach

Giuliano Vitali(1), Eugenio Magnanini (2), Enrico Denti (3)

(1) Dept. of Agro-Environmental Science and Technology, Univ. Bologna, Italy, giuliano.vitali@unibo.it
(2) Dept. of Tree Crops, Univ. Bologna, Italy
(3) Dept.of ComputerScience and Electronics, Univ. Bologna, Italy

Introduction

Recently the sector of Functional Structural Plant Modeling (FSPM) has significatively grown (AA.VV., 2004,2007) attracting the attenction of researchers from several branches of biological sciences. Objective of FSP Modelers is using a morphological characterization to improve plant eco-physiological modeling. In some case they move from pure mathematical and geometrical simulators used in animation movies and landscape design, in other cases they are to interpretate external characters of a plant. Several tools has been developed, some freely downloadable, others with a commercial mean. Even when they are available at a collaborative level, the plant simulating engine is often embedded with graphics and few efforts has been made to distinguish the task of making a plant from that of representing it on a screen which also means define a standard to code plants. Communicating a virtual plant between programs with different purposes not only aloow for a better use of computer resourses but could allow for plants interchange between researchers.

Methodology

In this work a virtual plant modeling tool has been developed, which is made of two independent computer programs, a simulator and a graphic interactor (figure 1). The plant simulator is available as a service on a server PC, whereas the client, installed on a networked remote computer, allows the user to interact with the simulator. The user can decide which plant to grow, where and when to do it, then he can observe the plant during its development, and interact with it to have information about its components and also pruning them. To do so XML has been used to code the plant and its features: the language, called VPML, is used to build-up the message exchanged by server and clients. The client has been written in Java, uses the WebStart technology to be installed on a remote machine, and Java3D libraries for graphics and interaction capabilities. The server has a web service active ready to answer to clients requests made of an initial plant, asked to be grown in a certain environment for a period: the answer is made of a set of snapshots of the plant at regular time intervals, so as the user can see the full plant history. Three simulation scales have been considered, the daily one, to simulate seedlings and germination, the daily one for seasonal annual plants and month one for trees. Species dependent growing parameters are maintained on server-side, together with their base shapes, which are only 4 at present: bud, internode, petiole and blade. The server send the client the structure of the plant and location where base graphic shapescan be downloaded from. The structure tells how elements are conneted, their orientation and size. Base elements are coded in VRML, allowing to be freely drawn, and they can also be downloaded locally, together with the plant structure, allowing to observe the plant off-line.

Results and discussions

The development of a fully functional protocol hasben quite time-expensive within the collaboration group because of the levels of protocols to be managed separatly on server and client sides: WebStart, managed within the HTTP, a Web-service (SOAP) used to handle the XML-based protocol, the VPML plant coding/decoding itself and a standard to apply to VRML base graphic elements to correctly mount them (see figure 2).

At present the main problems stay in the complexity of the plants, as VPML decoding on client side is very memory expensive, whereas J3D seems to be able to manage very complex virtual words.

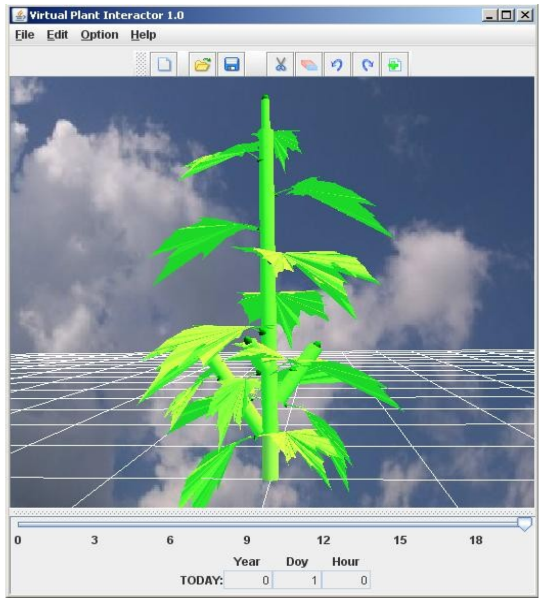

Figure 1 – Client main window with a branched 1 year-old virtual maple.

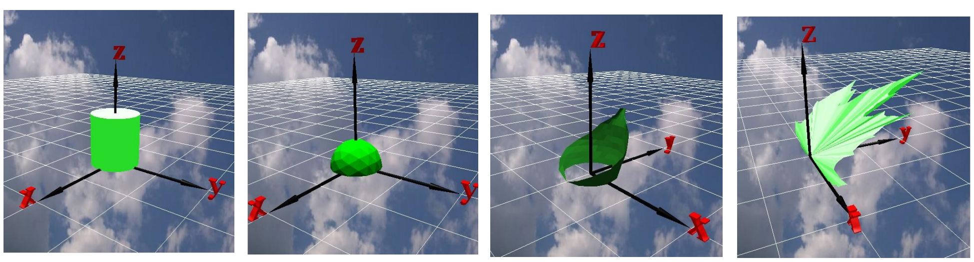

Figure 2 – Base elements used to mount the Virtual Plant from the client applicative.

Conclusions

The development of the project required to define and reach several objectives, as the possibility to work on a virtual-plant project at a team level, with different skillness and competencies, and to define a standard language to represent plant structure separated from the geometrical features. Even if the first objective was fullfilled, it seems to require a powerful server when more clients are run at same time, as in web-games. About the second it is probably a former step toward a proposal for an open format for intrechange plants at structural level. The simulator, is at present very simpe and is still lacking a radiative environment. Java3D allows for self-shading but not for reciprocal shading. Other features which could be important for realistic rendering is physical collisions and gravity, both important to improve space occupation of elements parts in light and other atmospheric interactions. From the other side, the system is per se, already able to simulate a numer of plants.

References

AA.VV., 4th Int.Work. on Functional-Structural Plant Models, Montpellier-FR 7-11 june 2004,423 pp.
AA.VV., 5th Int.Work. on Functional-Structural Plant Models,Napier-NZ 4-9 November 2007,333 pp.