



Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs

University of Vienna
Faculty of Computer Science
Research Group Entertainment Computing
Währinger Straße 29
A- 1090 Vienna, Austria

T +43 (1) 4277-78710
F +43 (1) 4277-878710
helmut.hlavacs@univie.ac.at

To:
Riccardo Focardi
Coordinator
PhD in computer science
Ca' Foscari University, Venice
<http://www.dsi.unive.it/~focardi/>

**Submitted thesis of Giuseppe
Maggiore**

Vienna, 15.1. 2013

This document reviews the thesis submitted by Giuseppe Maggiore with the title

Casanova: a language for game development.

The thesis is structured into 11 chapters and 7 appendices. The basic topic is the derivation and evaluation of a new programming language called Casanova. This language is specifically aimed for developing computer games. Developing computer games is an extremely tedious, lengthy, and error prone task, sometimes involving thousands of people. The usual programming languages - C++, C#, JavaScript and others - are in fact general purpose languages not specifically tailored to program games, resulting in the above development overhead.

The hypothesis behind Casanova is that a language specifically tailored to develop computer games should improve the general development process, increase the productivity, and decrease costs and errors.

The proposed language is a functional language inspired by predecessors such as Haskell or F#. The reason for this choice is the possibly large expressiveness of such languages, requiring less code, but helping the programmer to avoid unnecessary bugs.

Synopsis

In chapter 1, Mr. Maggiore provides an in-depth motivation and introduction into the problem field, and describes the research challenges he wants to tackle.

Chapter 2 describes the basic structure as seen in many games. This consists mainly of a continuous game loop, integration of movement differential equations, game state and state machines, and drawing the game state to the screen.

In chapter 3 Mr. Maggiore reviews other comparable approaches, their properties and how they work, and selects three state-of-the art tools/environments, which without doubt on the one hand work in a complimentary way, and on the other hand are very popular and widespread. Out of this review, he motivates why it makes sense to design a new language specifically designed to write computer games.

Chapter 4 then describes the basic design choices made for Casanova, the language that he designed for this purpose. This description is prolonged in chapter 5, where Mr. Maggiore describes the syntax elements of Casanova, by using a context-free grammar. This description proves to be surprisingly simple, at first glance the number of language constructs seems to be quite low. It turns out that this is in fact one of the design goals, for keeping the language simple and learnable.

Chapter 6 then describes the language semantics, its performance and correctness. He also provides an implementation based on existing F# environments and compilers. Through this clever trick he is able to provide a mature development environment, extensive libraries and compilers for his new language. This implementation is more detailed in chapter 7. Here Mr. Maggiore also describes implementation issues like rule containers, scripting, coroutines, and how to use caching to improve the performance significantly.

Chapter 8 then describes case studies of games programmed with Casanova, in this case the classic Game of Life, a shooter, a strategy game and more. By providing code snippets, Mr. Maggiore describes how to implement game parts.

The evaluation continues in chapter 9, where Mr. Maggiore provides more evaluation results, including a comparison of the complexity of Casanova compared to a set of other approaches, input management, implementation of state machines, and drawing. The chapter continues with a quantitative performance evaluation, followed by an experiment involving high-school and master students and being focused on the learnability of Casanova.

The last but one chapter 10 extensively discusses how Casanova answers the research questions asked at the start of the thesis. The treated topics involve issues like rendering, libraries, IDEs, networking, AI, and general shortcomings of the approach. Finally, chapter 11 concludes the thesis, summarizing again the main contributions of the thesis.

Scientific Contribution

The thesis fulfills all criteria for a computer science PhD thesis. It is focused, with a clear goal, and the author fulfills all research goals as stated at the thesis start. The main contribution, a novel programming language for computer games, must be called a major contribution to the field. The author achieves to establish an innovative concept, with the prospect of having not only contributed to science, but also having created a new language that is actually used by many people for this very purpose of programming games.

Mr. Maggiore furthermore shows an impressive and deep knowledge and understanding of the major corner stones of his work, including how to program computer games, and concepts of programming languages in general, and functional programming in particular. Mr. Maggiore also knows to how formulate research questions, how to find appropriate answers, and how to scientifically evaluate his answers.

As a side matter, the author's language is excellent, and the thesis is a delight to read.

Points of Critique

The main part of the thesis seems rather short. However, this effect is ameliorated when also taking into account the extensive appendix, which partially could also be used in the main body of the thesis.

Conclusion

I hereby certify that the reviewed thesis should be accepted as thesis without modification.

Dr. Helmut Hlavacs

Dr. Helmut Hlavacs