

Casanova: a Language for Computer Games

Review Response

1. Introduction

In the following we try to address and answer the questions, doubts and requests for clarification that the reviewers posed. We have organized the presentation in three sections, one per review, and in tabular form to make certain that all issues are addressed; similar concerns from different reviewers have been grouped together.

2. Reviewer 1

The relevance of this paper is, if it exists at all, rather tangential, I'd say. It happens that I enjoy functional programming, so I found it worth the read---but I'm not sure that a new language for defining games (though clearly to do with interaction of people and programs) will be of interest to EICS as EICS is defined.

From the call for papers: *EICS 2012 is the fourth international conference devoted to engineering usable and effective interactive computing systems. Topics of interest include multi-device interactive systems, new and emerging modalities (e.g., gesture), **entertaining applications** (e.g., mobile and ubiquitous games), safety critical systems (e.g. medical devices), and **design and development methods** (e.g., extreme programming).*

Our main motivation for desiring to contribute to EICS is that, even though our main focus is, indeed, game development, the technologies for game development are the same (or extremely close) techniques that can be used for designing interactive systems.

3. Reviewer 2

The strength of the contribution is weakened by the previous publications about Casanova. I.e., many information provided about Casanova and its design is redundant with what is presented in [10]. The submitted paper lacks a concise statement about where its contribution goes beyond the previous publication about Casanova. (The ACM "Policy on Prior

Publication and Simultaneous Submission" requires at least 25% previously unpublished material.) The authors should address this in their rebuttal.

[ALSO, from **REVIEWER 3**: The biggest problem is overlapping with the authors' previous works such as reference [10], Designing Casanova: a language for games, from 2011.]

[ALSO, from **REVIEWER 4**: There appears to be significant overlap with a previous presentation of the work in [10]. I have reviewed [10], and while it is a much shorter paper and lacks some of the language details presented in this submission, it makes essentially the same argument

The amount of original work seems to be a very important point that we seriously failed to make in the paper, so thank you for helping us bring it to light.

The first remark is that this paper contains a surface similarity with the design paper, exactly because it is an incremental evolution of the original design of the language.

Secondly, a formalization is a very important step (which requires a fairly large effort to produce) in the process of creating Casanova for the following reason:

- It is a universal best practice when designing a language of the ML family to provide a formal framework over which to reason [1,2]; this practice is also very common in other programming languages, such as Haskell [3,4]
- Formalism allows us to implement the language removing much of the guesswork that would otherwise risk the introduction of subtle bugs that would make it very hard for a developer to understand certain undesired behaviors of his application

Finally, a precise description of all the optimizations available in Casanova is a rather relevant addition, together with a preliminary quantitative assessment of their impact on game performance.

Given these considerations, the fully original work in this paper ranges from page 3 (sec 4.1) to page 7 (4.3) and then pages 8 and 9. This amounts to 60% of the paper, or 66% if we exclude the references page.

In conclusion, even though there are strong similarities with the design paper, this is to be expected since the two works are steps taken in quick succession in the same project and thus they share their motivation, general idea and conclusions. The importance of a formalization which gives clear shape to the work and an in-depth discussion of what was only previously hinted at is, in our opinion, of paramount

<p>in the same form.]</p> <p>One broader suggestion: It would be interesting to perform some kind of user study evaluating to what degree using Casanova can make game developers more productive compared to other languages/frameworks.</p> <p>[ALSO, from REVIEWER 3: it would be more interested to show how practitioners find the new programming language in practice, do they prefer it to existing programming toolkits such as the XNA or OpenGL sdk, etc. or with environments such as Alice.]</p> <p>[ALSO, from REVIEWER 4: There is no evaluation presented in the paper, beyond an evaluation of optimized versus non-optimized versions of a single Casanova program. The authors' key claim seems to be that Casanova represents an improvement over state of the art game development practice; however, there are no comparisons of Casanova programs with programs developed by traditional means, using any metric whatever. Are Casanova programs shorter? Are they easier to write? Are they easier to read? Do they perform better? Do programmers prefer Casanova over other alternatives?]</p> <p>* In the case study, projectiles and asteroids manage their own lists of colliders, which are basically inverse relationships.</p> <p>... rendering, and networking. Currently however, the latter two aspects are not yet implemented and both still pose open challenges.</p> <p>[ALSO, from REVIEWER 4: The work appears premature, in that existing Casanova systems do not yet incorporate rendering (and networking, but this is arguably less important). Without the ability to render, in what sense can Casanova be said to support game development? Again, this speaks to the validity of</p>	<p>importance.</p> <p>This is something we absolutely agree upon, to the point that we have started such an evaluation but the results are still preliminary. We are studying the impact of learning game development from scratch through Casanova, both in high-school classrooms and with first year students of the Master in Game Development of the University of Verona; this follows on our previous work [5,6] in the field. It appears that there is compelling data supporting the ease of use of Casanova, especially with non-technical young students, but more data and more evaluation is required.</p> <p>We also have various detailed comparisons of how much shorter a Casanova program is when compared to its F# counterpart. We are also in the process of building different samples in different languages and frameworks, in particular C# and C++ given their widespread use in the game development industry, but the data on this is still preliminary.</p> <p>Indeed, the optimization of predicates also avoids computing the same predicate twice for the same arguments; it should be specified in the paper under <i>Optimizations</i> in the discussion about the optimization of Cartesian products with predicates.</p> <p>Networking is a complex open problem which we are working on. Our work on rendering, on the other hand, has been progressing steadily and is approaching completion; a preview of our work on rendering may be found at http://casanova.codeplex.com/wikipage?title=Papers</p>
--	---

the approach.]	
----------------	--

4. Reviewer 3

<p>- section 4.5 Benchmarks is not at all clear (what was the testing system, how did you measure it, what do you compare against besides your own language)</p>	<p>The gains are compared on a 1.6 GHz Core 2 Duo CPU with an nVidia GeForce 320M GPU. Each optimization is compared against the un-optimized program to show its effectiveness. Further benchmarks that compare our system with existing scene-graphs such as Unity, Ogre and more are underway.</p>
--	---

5. References

- 1 Milner, Robin, Tofte, Mads, Harper, Robert, and MacQueen, David. *The Definition of Standard ML, Revised Edition*. MIT Press, 1997.
- 2 Pierce, Benjamin. *Types and Programming Languages*. MIT Press, Cambridge, Massachusetts , 2002.
- 3 Faxén, Karl-Filip. A static semantics for Haskell. *Journal of Functional Programming* , 12, 5 (2002).
- 4 Jones, Mark P. Typing Haskell in Haskell. In *Haskell Workshop* (1999).
- 5 Maggiore, Giuseppe, Torsello, Andrea, Sartoretto, Flavio, and Cortesi, Agostino. Engaging high school students in computer science via challenging applications. In *Conference on Information technology education (SIGITE)* (New York 2011).
- 6 Costantini, Giulia, Maggiore, Giuseppe, and Cortesi, Agostino. Learning by Fixing and Extending Games. In *Eurographics - Education Papers* (Munich 2009).