Contents

1 Introduction **3**
1.1 Games and video games . . . 3
1.2 Building games . . . 4
1.3 Problem statement . . . 4
1.3.1 Research question and process . . . 5
1.3.2 Positive consequences . . . 5
1.4 Outline . . . 5

2 Related work **7**
2.1 What is a game? . . . 7
2.2 Game development . . . 7
2.2.1 Hardware . . . 8
2.2.2 Multimedia API . . . 8
2.2.3 Engines . . . 8
2.2.4 Editors . . . 8
2.2.5 Visual + GPL . . . 8
2.2.6 DSL . . . 8
2.3 Research questions? . . . 8

3 Requirements for general game development languages **9**

4 Our proposal: the Casanova language **11**
4.1 System description . . . 11
4.2 Syntactic choices description . . . 11

5 Language implementation and evaluation **13**
5.1 Compiler structure . . . 13
5.2 Performance optimization . . . 13
5.2.1 State machine optimization . . . 13
5.2.2 Event optimization . . . 13
5.2.3 Event optimization and queries . . . 13

6 Language usability and evaluation **15**
6.1 Usability first impact . . . 15
6.2 Usability in depth . . . 15

7 Applications **17**
7.1 Portability . . . 17

8 Discussion **19**

9 Conclusions **21**

Chapter 1

Introduction

1.1 Games and video games

- Games and video games. What is a game (definition)?
- Games are grouped by genre. How many genres?
- Commercial games. Mainly used for entertainment. Commercial games are *big*. Commercial games society impact.
- Games are not only used for entertainment.
- Games are *useful* for applications so-called *serious*.
- What is a serious game?
- Examples of serious games.
- What are their impact in our society?
- Who build serious games (developers)?
-
  - Serious developers -¿ description
  - Researchers -¿ description
  - Indie developers (special case) -¿ description
- Despite their cultural impact and usefulness the above developers do not enjoy the same budget as for commercial games developers.

- These budget limitations decrease the freedom for serious games developers to choose new features to implement, to investigate new interaction paradigms, to try new kind of game experience, etc.

- What are these limitations?

- Among the possible limitations (obstacles) we identified *tools* as a big limitation.

- Tools for making games are limited and not suitable.

- Games are inherently complex. If we also stack to this not suitable tools then the results is dramatic in terms of complexity and expenses. Indeed many serious games projects never see the light of day.

- Hence, we introduce Casanova.

- Casanova is a language for making games. Casanova is aimed to reduce the complexity of the tools layer, so eventually to give serious games developers the opportunity to make games with less effort hence to increase serious games likelihood of success.

1.2 Building games

- discusses difficulties in building games, difficulties between AAA and research/indie games

- tools in use

- languages for game development

  - Why a language? Why not a general purpose language? Inspiration from paper DS L's as the ultimate abstraction [Paul Hudak]

1.3 Problem statement

Orchestration in games with Casanova 2 REQUIREMENTS TABLE

Code	Language requirement	Run-time requirement	Name	Description
R1	Yes	No	Syntax and semantics	the language should use specific constructs designed around typical aspects present in games
R2	No	Yes	Performance	the resulted game should be fast enough to guarantee a smooth run-time experience
R3	Yes	No	Learning curve	the language should be easy to handle for novice developers
R4	Yes	No	Usability	the language should speed up development processes of those developers who master Casanova in the long-term
R5	No	Yes	Applications	the game should be run-able on the most used devices and software platforms
R6	Yes	No	Reliability	the language back-end should help developers to reduce the amount of mistakes while building games

1.3.1 Research question and process

1.3.2 Positive consequences

...

1.4 Outline

...

Chapter 2

Related work

2.1 What is a game?

- Low level, a program that indefinitely interacts with hardware components
- High level (computer science), the derivative of the game state with respect of the time applied to the game state itself

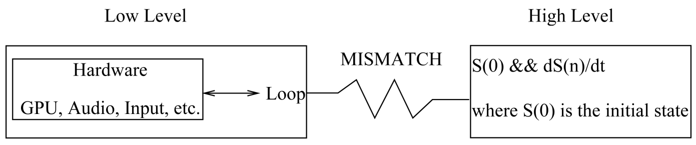

Figure 2.1: Game formalization (overview)

2.2 Game development

- Research in game development tries to solve the gap/mismatch between the low level constraints and the high level description of a game.
- **How?** Historically a hierarchy has come to life, which is described by the picture below:

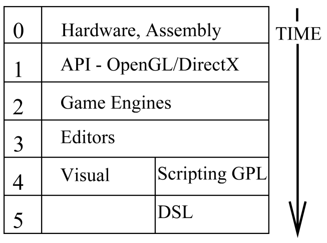

Figure 2.2: Game development tools evolution

2.2.1 Hardware

introduction, examples (assembly games), pros, and issues

2.2.2 Multimedia API

introduction, examples (DirectX, OpenGL), pros, and issues

2.2.3 Engines

introduction, examples (Ogre, XNA, UnityEngine), pros, and issues

2.2.4 Editors

introduction, examples (UnityEditor, UnrealEngine), pros, and issues

2.2.5 Visual + GPL

introduction, examples(GameMaker,RPGmaker), pros, and issues

2.2.6 DSL

introduction, examples(Casanova), pros, and issues

2.3 Research questions?

For example:

- To what extent a game should be designed around the domain of games and what are the requirements
- To what extent game development would benefit from the integration of DSLs into the development processes and to what level of abstraction
- ...

Chapter 3

Requirements for general game development languages

...

Chapter 4

Our proposal: the Casanova language

FEATURES TABLE

Requirement	Casanova Feature	Description
R1	Casanova syntax and semantics	...
R2	Events optimization	...
R3	Students test	...
R4	Usability test	...
R5	Making games with Casanova	...
R6	Layered compiler	...

4.1 System description

...

4.2 Syntactic choices description

Orchestration in games with Casanova 2 [paper]

Chapter 5

Language implementation and evaluation

5.1 Compiler structure

5.2 Performance optimization

...

5.2.1 State machine optimization

Orchestration in games with Casanova 2 [paper]

5.2.2 Event optimization

High performance encapsulation in Casanova [paper]

5.2.3 Event optimization and queries

High performance encapsulation in Casanova [paper]

Chapter 6

Language usability and evaluation

6.1 Usability first impact

Rotterdam usability study - [paper]

6.2 Usability in depth

Rotterdam usability study - [paper]

Chapter 7

Applications

- Dyslexia
- RTS
- Contact
- Lego

7.1 Portability

The compiler output C# code. The Mono framework allow to run C# code on several platform such as Windows, Linux, and Mac.

Chapter 8

Discussion

...

Chapter 9

Conclusions

...