Casanova 2.0

A basic introduction

Dr. Giuseppe Maggiore

NHTV University of Applied Sciences
Breda, Netherlands

Game development

Interests

- A huge field
- Increasing artistic and narrative value [**?**]
- Largely unexplored potential fields of application

Games sales

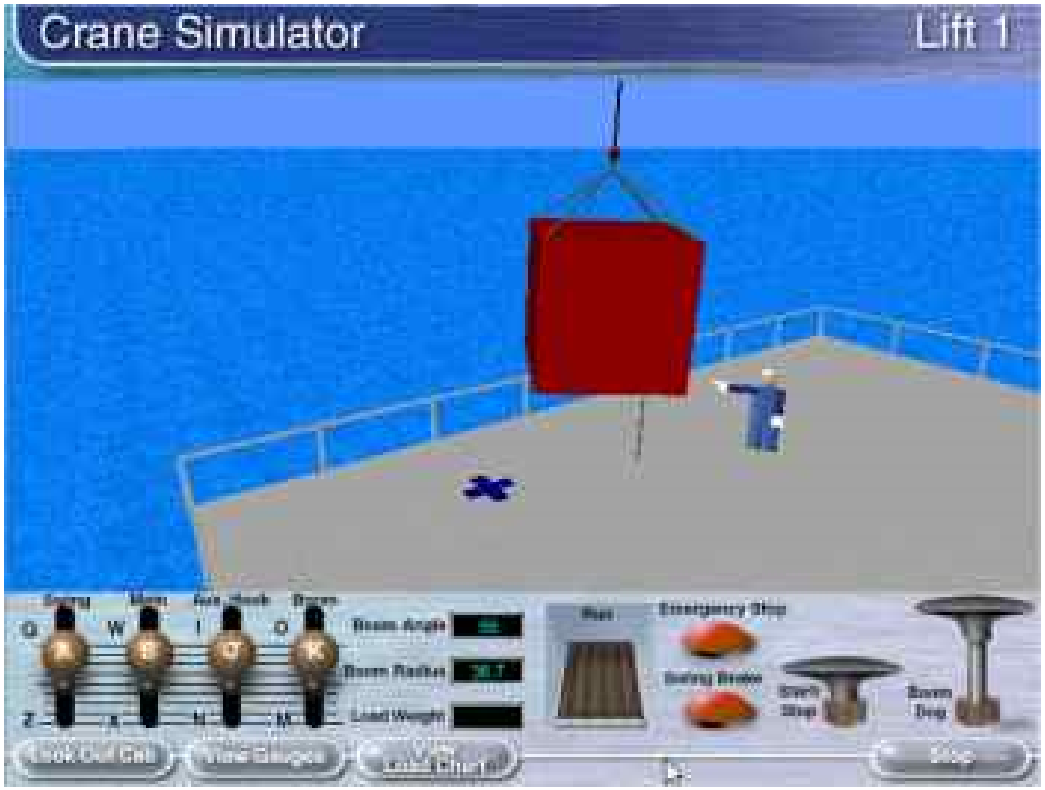

United States video game industry (1995-2007)
Revenues (in billions USD)
20
18
16
14
12
10
8
6
4
2
0
1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
NPD sales (billions) 3.2 3.3 5.1 6.2 6.9 6.6 9.4 10.3 10 9.9 10.5 12.5 17.94

Fields of application

- As a supporting tool for training [**?**]
  - *military*
  - *crane operation*
  - *medicine*
  - ...

Games for training

Games for training



Games for training

Game development

Fields of application

- As a supporting tool for education [**?**]
  - Math
  - Chemistry
  - History
  - ...

Games for education

Game development

Fields of application

- As a supporting tool for research
  - Psychology [**?**]
  - Philosophy
  - AI [**?**]
  - ...

Games for research

Games for research

Games for research

Unexplored?

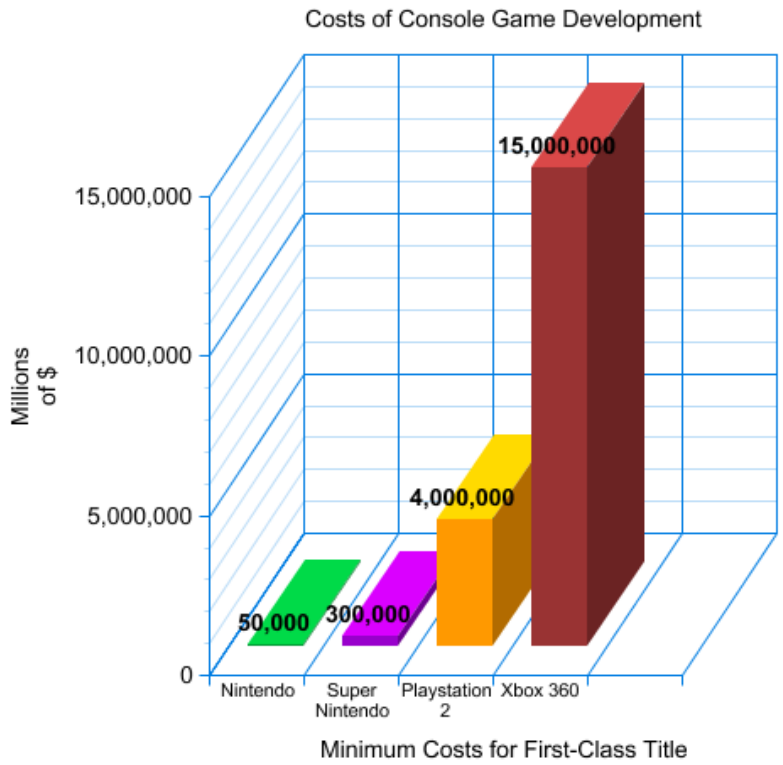
- So many opportunities: **why is it unexplored?**

Game development

Unexplored?

- So many opportunities: **why is it unexplored?**
- Because it is incredibly expensive to make an even decent game
- Low-quality game means distraction because of poor interaction

Games for research



Game development

Why is it expensive?

- Art is expensive
  - Procedural generation [**?**] moves some cost from art to programming
- Programming is expensive
  - Most programming technology is *inadequate* for games
  - It is designed for other fields of application
  - Lots of time for trivial tasks
  - Lots of time for debugging

Game development

Taming the costs

- Reuse code and art
- Can yield games similar to each other (surprise there!)

Game development

Reusing code and art

- Does not help us at all!
- Research and education games are inherently different
- We need *cheap* and *fast* ways to prototype

Casanova language

Basics

- Build a programming language *exclusively for making games*
- Syntax and semantics are focused on the primitives of simulation
  - ✓*Transformation* of the game world as time progresses
  - ✓*Time* as a first class primitive
  - ✓*Parallel update* of many things at once
  - ✓Collection-oriented programming
  - ✓Math-oriented programming
  - ✓Pluggable game engines
  - ✓Saving and loading

Casanova language

Basics

- Build a programming language *exclusively for making games*
- Syntax and semantics are focused on the primitives of simulation
  - ✗IDE integration
  - ✗Lots of samples (for our own debugging)
  - ✗Networking primitives
  - ✗Object-orientation
  - ✗SQL-style optimization
  - ✗Static/dynamic analysis framework to detect errors
  - ✗Graphics-oriented language subset (CNSL)

Basic semantics

- **Rules**: a series of transformations of the world
- Each repeated with a variable time step ($\delta t_{ij} \leq \delta t_{\text{min}}$)
- Merged together automatically

$$r_1 : w_{10} \underbrace{\rightarrow}_{\delta t_{11}} w_{11} \underbrace{\rightarrow}_{\delta t_{12}} w_{12} \underbrace{\rightarrow}_{\delta t_{13}} \cdots \underbrace{\rightarrow}_{\delta t_{1n}} w_{1n}$$

$$r_2 : w_{20} \underbrace{\rightarrow}_{\delta t_{21}} w_{21} \underbrace{\rightarrow}_{\delta t_{22}} w_{22} \underbrace{\rightarrow}_{\delta t_{23}} \cdots \underbrace{\rightarrow}_{\delta t_{2n}} w_{2n}$$

$$\vdots$$

$$r_m : w_{m0} \underbrace{\rightarrow}_{\delta t_{m1}} w_{m1} \underbrace{\rightarrow}_{\delta t_{m2}} w_{m2} \underbrace{\rightarrow}_{\delta t_{m3}} \cdots \underbrace{\rightarrow}_{\delta t_{mn}} w_{mn}$$

Sample 1 - hello world

```
world HelloWorld = {
  Text   : Text
}

let world = {
  Text = Text.Create("Hello␣Casanova!",
           Vector2<pixel>(0.0f, 0.0f),
           Vector2<pixel>(200.0f, 200.0f))
}
```

Sample 2 - animation

```
world SimpleAnimation = {
  Text          : Text
  Position      : Vector2<pixel>
  Velocity      : Vector2<pixel/s>
  Checkpoints   : List<Vector2<pixel>>
} with
  rule Position = self.Position + self.Velocity * dt
  rule Velocity =
    for c in self.Checkpoints do
      yield Vector2.Normalize(c - self.Position) *
          10.0f<pixel/s>
      wait_until (Vector2.Distance(c, self.Position)
          <= 0.1f<pixel>)
```

Case study

A game for dyslexia detection in children

- Pairs of sounds; say if equal or different
- Children get bored really quickly

That's it

Sounds

Case study

A game for dyslexia detection in children

- Idea: create a frame of "game mechanics" around this concept
- More pleasant environment to perform the tests
- No visual help or hints
- Very simple mechanisms

Case study

A game for dyslexia detection in children

- Use a tutorial mechanism for initial training (no lengthy explanations)
- Use lots of animations and visual metaphors to explain the tasks

Dyslexia game

Demo!

Case study

About the implementation

- Huge, nested state machines
- Lots of prototyping (we built three full versions of the game)
- Can be done with other tools, but:
  - Casanova helped immensely
  - Our expertise in Casanova contributed a lot

Dyslexia game

Demo!

Conclusion

Games and research

- Game development is interesting for research and other novel applications
- Huge costs make the realization of this infeasible for most practical purposes
- We propose a novel language built for *fast development* of games

Conclusion

Games and research

- Game development is interesting for research and other novel applications
- Huge costs make the realization of this infeasible for most practical purposes
- We propose a novel language built for *fast development* of games
- We aim at using this language to speed-up development of research games for third parties
  - We already have a case study
- While building case studies we wish to
  - Perfect the quality of our technology
  - Add "killer features" like networking for bigger customers

That's it

Thank you!

References I