



High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

High performance encapsulation in Casanova 2

Abbadi Mohamed

Ca' Foscari University (Venice, Italy)
Tilburg University (Tilburg, Netherlands)



Introduction

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Talk topics

- Introduction
- Tools and languages
- Encapsulation
- Problem
- Idea
- Details
- Evaluation
- Conclusions

Introduction

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Games in our society

- Games huge



Introduction

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Games in our society

- Games huge
- Games are not meant only for entertainment
 - Serious games
 - Indie games
 - Research games





Introduction

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Costs in game development

- Arising from game content, coding, debugging, maintenance, etc.
- Serious games and research projects do not enjoy the same budgets of the entertainment game industry
- Costs must stay in-check



Introduction

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Games are complex

- Complexity arising from the difficulty of expressing game dynamics



Introduction

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Games are complex

- Complexity arising from the difficulty of expressing game dynamics
- How do we reduce such difficulty?



Introduction

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Tools and languages

- Try to provide support to reduce difficulties in game development



Introduction

High
performance
encapsulation
in Casanova 2

Abbadì
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Tools and languages

- Try to provide support to reduce difficulties in game development
- Tools, limited
- Languages, expressive



Introduction

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Tools and languages

- Try to provide support to reduce difficulties in game development
- Tools, limited
- Languages, expressive
- Our research focus on languages



A domain programming language

High performance encapsulation in Casanova 2

Abbadi
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Casanova 2

- A domain specific language



A domain programming language

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Casanova 2

- A domain specific language
- Casanova 2 is:
 - OO, Functional, and Declarative
 - Entities
 - Rules
 - Effect system
 - Only way only to change the game state



A domain programming language

High performance encapsulation in Casanova 2

Abbadì
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Design strategies in Casanova 2

- Take specific shapes in Casanova 2



A domain programming language

Design strategies in Casanova 2

- Take specific shapes in Casanova 2
- In particular, we have focused on the consequences of encapsulation on code structure and run-time behavior

High performance encapsulation in Casanova 2

Abbadì
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

A domain programming language

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

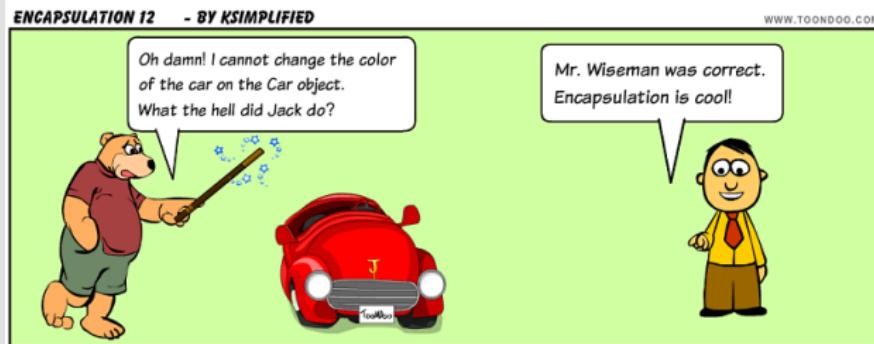
Details

Evaluation

Conclusions & future works

Design strategies in Casanova 2

- Take specific shapes in Casanova 2
- In particular, we have focused on the consequences of encapsulation on code structure and run-time behavior
- **Encapsulation**, helps developers keeping code readable and maintainable, hence help to reduce development costs





Problem statement

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

First problem statement

- To what extent can we take advantage of encapsulation to reduce the difficulty of making games?



Problem statement

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Encapsulation issues

- A game is run-time applications, hence game code must be performant/fast
- Performance becomes an issue when dealing with encapsulation
- Developers avoid encapsulation for highly-coupled and high-performance code



Problem statement

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

```
class Route
    Planet Start, Planet End,
    List<Fleet> TravellingFleets ,
    Player Owner
    void Update()
        foreach fleet in TravellingFleets
            if End.AttackingFleets.Contains(fleet)
                this.TravellingFleets.Remove(fleet)
class Planet
    List<Fleet> DefendingFleets ,
    List<Fleet> AttackingFleets
    void Update()
        foreach route in GetState().Routes
            if route.End = this then
                foreach fleet in route.TravellingFleets
                    if distance(fleet.Position , this.Position) < min_dist && fleet.
                        Owner != this.Owner then
                            this.AttackingFleets.Add(fleet)
```

Nice, it's clear! -:) but ... can't we make it go faster?



Problem statement

High performance encapsulation in Casanova 2

Abbadí
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Encapsulation issues: same semantics but without encapsulation

```
class Route
    Planet Start, Planet End,
    List<Fleet> TravellingFleets
    void Update()
        foreach fleet in this.TravellingFleets
            if distance(fleet.Position, this.Position) < min_dist && fleet.Owner
                != End.Owner then
                    this.TravellingFleets.Remove(fleet)
                    End.AttackingFleets.Add(fleet)
```

Finally fast :-) but does this code affect readability, so its maintainability?



Problem statement

High
performance
encapsulation
in Casanova 2

Abbadì
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Encapsulation issues

- A game is run-time applications, hence game code must be performant/fast
- Performance becomes an issue when dealing with encapsulation
- Developers avoid encapsulation for highly-coupled and high-performance code
- PITY!



Problem statement

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Refined problem statement

- To what extent can we take advantage of encapsulation in order to produce a high-performance run-time game code?
- To what extent can we automatize this process?



Idea

High performance encapsulation in Casanova 2

Abbadi
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Optimizing the Planet/Route example

- We maintain an index `FleetIndex` in `Planet`
- `FleetIndex` is the collection of `Fleet` that satisfy the attacking property
- A `Fleet` adds/removes itself from `FleetIndex` depending on its distance from the `Planet`



Idea

High
performance
encapsulation
in Casanova 2

Abbadì
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Optimization generalization

- A predicate P , a conditional statement, is based on fields of a type E_A
- An index $I_A \in E_B \mid \forall f \in I_A, P(f) = True$
- Every entity of type E_A got a reference to E_B . Every entity of type E_A is tasked to update its reference inside of I_A
- An entity of type E_B checks its I_A whenever the entity needs to interact with the objects of type E_A that satisfy the property P



Idea

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Temporal locality of predicates - part I

- Predicates belong to entities that exhibit similar behaviors with respect to the time flow
- We can expect that predicates will exhibit some sort of *temporal locality* on their behaviors



Idea

High performance encapsulation in Casanova 2

Abbadi
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Temporal locality of predicates - part II

- If we manage to evaluate the predicate only when a field that takes part of the predicate changes then we can achieve a further optimization



Idea

High
performance
encapsulation
in Casanova 2

Abbadi
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Temporal locality of predicates - part II

- If we manage to evaluate the predicate only when a field that takes part of the predicate changes then we can achieve a further optimization
- We need:
 - a fast wake-up collection where we store predicates and their block of code
 - to activate those predicates for some time in order to perform the predicate check whenever a predicate changes its state

Details

High performance encapsulation in Casanova 2

Abbadi
Mohamed

Introduction

A domain programming language

Problem statement

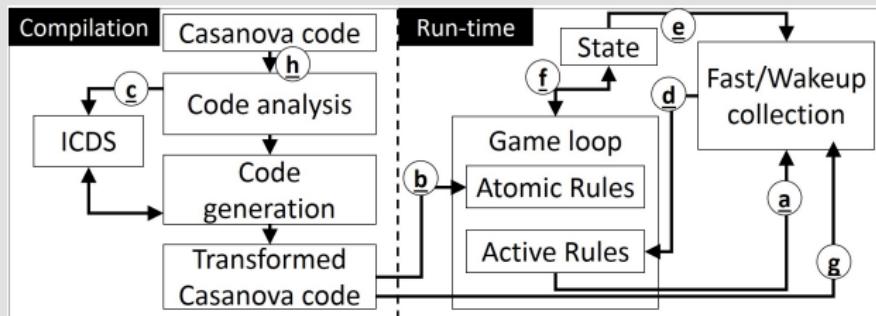
Idea

Details

Evaluation

Conclusions & future works

Process



Details

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

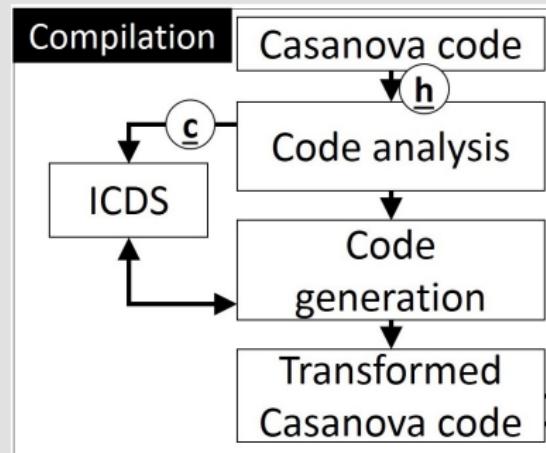
Idea

Details

Evaluation

Conclusions & future works

Recognizing IC's



- First we identify the so called interesting conditions (IC) in code
- Among the all possible IC's we chose only those IC's that are not affected by an atomic rule

Details

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

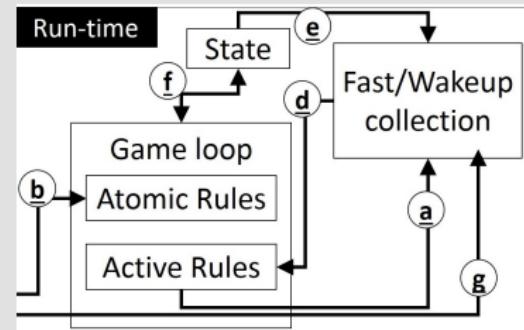
Details

Evaluation

Conclusions & future works

Run-time efficient sleep/wake-up system

- We store an index that represents the block of code that follows the predicate into a collection



- We run the block of code (by simply moving its index into the collection of active blocks) whenever the predicate changes to true
- We keep the block active as long as the evaluation of the predicate yields to true

Details

High performance encapsulation in Casanova 2

Abbadi
Mohamed

Introduction

A domain programming language

Problem statement

Idea

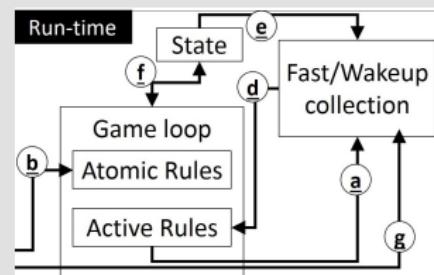
Details

Evaluation

Conclusions & future works

Some considerations - part I

- Every instance should take care about its reference inside the fast/wake-up collection
- In particular it should take into consideration:
 - Creation
 - Deletion



Details

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

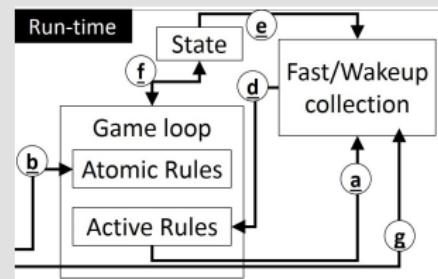
Idea

Details

Evaluation

Conclusions & future works

- Dithering is an important property that an entity should exhibit
- Our system adds a layer of complexity to the run-time system





Evaluation

High performance encapsulation in Casanova 2

Abbadì
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Code lines comparison

| Original language | Generated language | Optimized code | Lines |
|-------------------|--------------------|----------------|-------|
| Casanova | - | - | 45 |
| Casanova | C# | No | 139 |
| Casanova | C# | Yes | 327 |
| C# | - | - | 88 |

Evaluation

High performance encapsulation in Casanova 2

Abbadi Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Running time comparison

| Platform | Language | Optimized | Performance |
|----------|----------|-----------|-------------|
| Monogame | Casanova | No | 0.0159 ms |
| | Casanova | Yes | 0.0098 ms |
| | C# | - | 0.0147 ms |
| Unity3D | Casanova | No | 0.0257 ms |
| | Casanova | Yes | 0.0085 ms |
| | C# | - | 0.1642 ms |

Evaluation

High performance encapsulation in Casanova 2

Abbadi
Mohamed

Introduction

A domain programming language

Problem statement

Idea

Details

Evaluation

Conclusions & future works

Some Casanova games/demos



<https://github.com/vs-team/casanova-mk2/wiki>



Conclusions & future works

High
performance
encapsulation
in Casanova 2

Abbadì
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Some Casanova games/demos

- This work is far to be completed ...
- We showed that:
 - By using encapsulation, game code may be written in a maintainable way
 - By mean of our solution, encapsulated programs suffer less performance
 - Developers can use safely encapsulation in their design/code, hence costs are reduced
 - Making games for serious games developers and researchers is more accessible



This is it!

High
performance
encapsulation
in Casanova 2

Abbadì
Mohamed

Introduction

A domain
programming
language

Problem
statement

Idea

Details

Evaluation

Conclusions &
future works

Thank you! :)