‘What Have the Computer Scientists Ever Done for Us?’

*Algorithms and Networking for Computer Games*

Jouni Smed

*University of Turku*
Three academic perspectives to computer games
Three academic perspectives to computer games

Humanistic perspective
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
Three academic perspectives to computer games

- **Humanistic perspective**
- **Technical perspective**
- **Administrative/business perspective**
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective
- Game design
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

Game design

Game programming
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective
  - Game design
  - Game programming
  - Software development

GAME
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

**Game design**
- rules
- graphics
- animation
- audio

**Game programming**

**Software development**
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective
  - Game design
  - Game programming
  - Software development

GAME
Three academic perspectives to computer games

Humanistic perspective

Administrative/business perspective

Technical perspective

Game design

Software development

Game programming

- gfx & audio
- simulation
- networking
- AI
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

GAME

Game design

Technical perspective

Game programming

Software development
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

GAME

Game design

Technical perspective

Game programming

Software development
- design patterns
- architectures
- testing
- reuse
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

GAME

Game design

Technical perspective

Game programming

Software development
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

Game design ➝ Game programming

Software development ➝ Technical perspective ➝ Humanistic perspective ➝ Administrative/business perspective ➝ Game design ➝ Game programming
Three academic perspectives to computer games

- **Humanistic perspective**
- **Administrative/business perspective**
- **Technical perspective**

**GAME**

- Game design
- Game programming
- Software development
Three academic perspectives to computer games

- Humanistic perspective
- Administrative/business perspective
- Technical perspective

GAME

- Game design
- Technical perspective
- Game programming
- Software development
Scientific articles on computer games

source: Inspec Database, June 2006
...well, I have to plug it at some point...
Part I: Decision-making for synthetic players

OR

How to make intelligent bots?
Components, relationships and aspects of a game
Components, relationships and aspects of a game
Components, relationships and aspects of a game

- rules
- goal

Players agree on rules and their goal to face challenges.

- definition
- motivation
- agreement
Components, relationships and aspects of a game

- Rules
- Goal
- Player
- Opponent
- Definition
- Agreement
- Motivation

CHALLENGE
Components, relationships and aspects of a game

- **rules**
  - definition
  - agreement

- **goal**
  - obstruction
  - motivation

- **player**
  - indeterminism

- **opponent**
  - CHALLENGE
  - CONFLICT
Components, relationships and aspects of a game

- **rules**: definition
- **goal**: obstruction
- **representation**: agreement
- **opponent**: indeterminism
- **player**: CHALLENGE
  - **motivation**: CONFLICT

Components, relationships, and aspects of a game.
Components, relationships and aspects of a game

- **player**
  - representation
  - agreement
- **rules**
  - definition
  - motivation
- **goal**
  - obstruction
  - indeterminism
- **opponent**
  - correspondence
  - concretization

**PLAY**

**CHALLENGE**

**CONFLICT**
Definition for ‘computer game’

- a game that is carried out with the help of a computer program

- roles:
  - coordinating the game process
  - illustrating the situation
  - participating as a player

→ Model–View–Controller architectural pattern
Model–View–Controller

model

controller

view
Model–View–Controller

- **model**
- **controller**
- **view**

- state instance
- core structures
Model–View–Controller

- model
  - state instance
  - core structures
- controller
- view
  - proto-view
  - rendering
Model–View–Controller

- **Model**
  - State instance
  - Core structures

- **Controller**
  - Control logic
  - Configuration
  - Driver

- **View**
  - Proto-view
  - Rendering

- **Inputs**
  - Instance data
  - Input device
  - Action

- **Outputs**
  - Output device
  - Perception
  - Options

- **Human Player**
  - Action
  - Perception
Model–View–Controller

**Model**
- state instance
- core structures

**Controller**
- control logic

**View**
- proto-view
- synthetic view
- rendering

**Instance Data**
- instance data
- input device
- script

**Output Device**
- output device

**Options**

- perception

**Human Player**
- action

**Image**

GAME RESEARCH
GDC WORKSHOP
Synthetic player

- computer-generated actor in the game
- displays human-like features
  - human traits and characteristics: fear, bravery, cowardice, devotion…
  - a computer game comprising only synthetic players can be interesting in itself!
- stance towards the human player
  - enemy, ally, neural
- games are anthropocentric!
Decision-making

World
Decision-making

World

Primitive events and states
Decision-making

World → Primitive events and states → Pattern recognition
Decision-making

World

Primitive events and states

Previous primitives

Pattern recognition
Decision-making

*World*

Primitive events and states

Pattern recognition

Previous primitives

Observed events and states
Decision-making

- **World**
- **Primitive events and states**
- **Pattern recognition**
  - Previous primitives
  - Observed events and states
- **Decision-making system**
Decision-making

World

Primitive events and states

Possible actions

Pattern recognition

Previous primitives

Observed events and states

Decision-making system
Decision-making

World

Requested actions

Primitive events and states

Possible actions

Pattern recognition

Observed events and states

Previous primitives

Decision-making system
Perspectives for decision-making in computer games

- level of decision-making
  - strategic, tactical, operational
- use of the modelled knowledge
  - prediction, production
- methods
  - optimization, adaptation
- modelling uncertainty
  - probabilistic, possibilistic
Key questions for synthetic players
Key questions for synthetic players

- how to achieve real-time response?
Key questions for synthetic players

- how to achieve real-time response?
- how to distribute the synthetic players in a network?
Key questions for synthetic players

- how to achieve real-time response?
- how to distribute the synthetic players in a network?
- how autonomous the synthetic players should be?
Key questions for synthetic players

- how to achieve real-time response?
- how to distribute the synthetic players in a network?
- how autonomous the synthetic players should be?
- how to communicate with other synthetic players – and humans?
Part II: Hiding the communication delay in networked games

OR

To communicate or to compute?
Consistency vs. responsiveness

- **consistency**
  - how similar is the replicated data?

- **responsiveness**
  - how promptly the data gets updated?

**computer games require real-time interaction**

- responsiveness is more important and consistency can be compromised
- ≠ traditional distributed systems
How to achieve responsiveness: Dead reckoning
How to achieve responsiveness: Dead reckoning

$(x_0, y_0)$
How to achieve responsiveness: Dead reckoning

$(x_0, y_0)$
How to achieve responsiveness: Dead reckoning

\[ (x_0, y_0) \]
How to achieve responsiveness: Dead reckoning
Dead reckoning in networking

Time 3.5:
Position (5.5, 6)

Time 4.5:
Position (8.5, 8)

Current Predicted Path
Dead reckoning in networking

Time 3.5:
Position (5.5, 6)
Velocity (6, 3)

Time 4:
Position (6, 3)

Time 4.5:
Position (8.5, 8)

Current Predicted Path

New Predicted Path
Dead reckoning in networking

Current Predicted Path

Time 4: Position (6, 3)
Velocity (6, 3)

Time 3.5: Position (5.5, 6)

Time 4.5: Position (8.5, 8)

New Predicted Path

Time 4.5: Position (9, 4.5)
Features

- widely known, tried and tested
- provides responsiveness
- tries to maintain consistency
- can be adapted to different models
- simple to understand
- more or less simple to implement
- the *de facto* method in commercial computer games
Features

- widely known, tried and tested
- provides responsiveness
- tries to maintain consistency
- can be adapted to different models
- simple to understand
- more or less simple to implement
- the *de facto* method in commercial computer games

*But is dead reckoning the only choice?*
Local perception filters (LPFs)

- a method for hiding communication delays (Sharkey, Ryan & Roberts, 1998)
- exploits the human perceptual limitations by rendering entities slightly out-of-date locations based on the underlying network delays
  - causality of events is preserved
  - rendered view may have temporal distortions
  - rendered view ≠ real view
Rules of LPFs
Rules of LPFs

1. Player should be able to interact in real-time with the nearby entities.
Rules of LPFs

1. Player should be able to interact in real-time with the nearby entities.
2. Player should be able to view remote interactions in real-time, although they can be out-of-date.
Rules of LPFs

1. Player should be able to interact in real-time with the nearby entities.
2. Player should be able to view remote interactions in real-time, although they can be out-of-date.
3. Temporal distortions in the player’s perception should be as unnoticeable as possible.
Entity types

- **active**: indeterministic, unpredictable (humans)  
  \[\Rightarrow \textit{players}\]
  - local: residing in the same computer
  - remote: connected over a network

- **passive**: deterministic, predictable (projectiles, buildings etc.)  
  \[\Rightarrow \textit{entities}\]
Example: Pong

- two active entities: paddles
  - movements unpredictable
- one passive entity: ball
  - movements predictable
- latency of $d$ seconds
The view of the red player
The view of the red player
The view of the red player
The view of the blue player
The view of the blue player
The view of the blue player
Pong: Summary

- players see a different representation of the same playing field
  - ball accelerates as it approaches the local player’s paddle
  - ball decelerates as it approaches the remote player’s paddle
- rendered position alternates between
  - the current time
  - a past time reference
3½-dimensional temporal contour

- represent each player’s perception as a four-dimensional coordinate system \((x, y, z, t)\)
- \(x, y, z\): the spatial position relative to the local player’s current position
- \(t\): the time associated with rendered information from that position
Temporal contours in Pong

Blue player

Red player
2½-dimensional temporal contour
2½-dimensional temporal contour
$2^{\frac{1}{2}}$-dimensional temporal contour
2½-dimensional temporal contour
2½-dimensional temporal contour
Bullet time

- movies: visual effect combining slow motion with dynamic camera movement
- computer games: player can slow down the surroundings to have *more time* to make decisions
- easy in single player games: slow down the game!
- how about multiplayer games?
Bullet time in multiplayer games

- two approaches:
  - speed up the player
  - slow down the other players

- if a player can slow down/speed up the time, how it will affect the other players?
  - localize the temporal distortion to the immediate surroundings of the player

- but how to do that?
Bullet time in multiplayer games

- two approaches:
  - speed up the player
  - slow down the other players
- if a player can slow down/speed up the time, how will it affect the other players?
  - localize the temporal distortion to the immediate surroundings of the player
- but how to do that?

Yes, you guessed it: with local perception filters!
Adding bullet time to LPFs

- player using the bullet time has more time to react
  ⇒ the delay between bullet-timed player and the other players increases
- add artificial delay to the temporal contour
  - approaching entities slow down
  - receding entities speed up
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Example: A simple shoot-out

Blue view

Orange view
Simple temporal contour and bullet time
Simple temporal contour and bullet time
Simple temporal contour and bullet time
Simple temporal contour and bullet time
$2^{\frac{1}{2}}$-dimensional temporal contour and bullet time
$2^{\frac{1}{2}}$-dimensional temporal contour and bullet time
2½-dimensional temporal contour and bullet time
2½-dimensional temporal contour and bullet time
Bullet-time testbench

<table>
<thead>
<tr>
<th>Action</th>
<th>Alpha</th>
<th>Bullet Time Active</th>
<th>Player Moving</th>
<th>Shoot</th>
<th>Generate Events</th>
<th>Clear Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ply1</td>
<td>Ply2</td>
<td>Ply3</td>
<td>Ply1</td>
<td>Ply2</td>
</tr>
<tr>
<td>Presets</td>
<td>Config</td>
<td>Settings</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active View</td>
<td>Alpha</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ Player 1 (R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ Player 2 (G)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ Player 3 (C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ Realworld</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coords</th>
<th>Mouse</th>
<th>Ply1</th>
<th>Ply2</th>
<th>Ply3</th>
<th>Bullet</th>
<th>DelayTime</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Events</td>
<td>Real</td>
<td>24,462</td>
<td>438,900</td>
<td>520,564</td>
<td>440,254</td>
<td>438,608</td>
<td>86</td>
</tr>
<tr>
<td>View</td>
<td></td>
<td>438,900</td>
<td>520,564</td>
<td>440,254</td>
<td>438,625</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Further work

- non-linear temporal contours
  - how to compute quickly?
  - noticeable benefits (if any)?
- numerical evaluation
  - measuring the distortion and its effects
- practical evaluation
  - how well does it work?
  - does it allow new kinds of games?
Conclusion:

So what do the computer scientists do?
We...
We…

- …have the leisure to study both new ideas and old ideas.
We...

- ...have the leisure to study both new ideas and old ideas.
- ...compete with one another to find new fields of research.
We...

- ...have the leisure to study both new ideas and old ideas.
- ...compete with one another to find new fields of research.
- ...collaborate with one another – and of course with the game industry.
We...

- ...have the leisure to study both new ideas and old ideas.
- ...compete with one another to find new fields of research.
- ...collaborate with one another – and of course with the game industry.
- ...teach the next generation of game programmers.