A Multi-Agent System for Autonomous Control of Game Parameters

Luc Pons

Institut de Recherche en Informatique de Toulouse
SMAC Team
www.irit.fr/SMAC
Games and Serious Games

- Growing interest for both games and serious games
- The objective is either learning or entertainment
- The experience is effective when the player reaches the state of flow [Csikszentmihalyi90]
  - Believability
  - Adapted Challenges
Games and Serious Games

- Games are getting more and more complex
  - Large number of parameters
  - Nonlinear
  - Difficult if not impossible to model

- Players are human beings
  - Large populations of players may involve original individuals
  - Human reasoning is unpredictable

- “The holy grail of game design is to make a game where [...] the difficulty curve is perfect and adjusts itself to exactly our skill level.” Raph Koster, Theory of fun, 2004
General Question

- **How to**
  - Adapt a game experience so it is profitable for all players?
  - Preserve the consistency and believability?

- **Given that**
  - Potentially all players are different
  - All game engines are different

- **While keeping**
  - Real time activity
  - General concepts
How it has been done so far?

- **Machine Learning**
  - Q-Learning with Markov Decision Processes [Andrade04]
    - Needs many iterations with a static player
  - Case-Based Reasoning [Sharma07]
    - Experts need to exhaustively list cases
  - Rules Re-ordering [Spronck06]
    - Limited to plan-based agents

- **Optimization**
  - Genetic Algorithms [Zook12] & Artificial Neural Networks [Stanley05]
    - Large amounts of data needed with many iterations
Proposition

- New paradigms to overcome complexity
- Use of collective intelligence
- (Logically) Distributed computation
- Bottom-up approach
Problem Definition: A General Scenario

Inputs
Scenario parameters

Game

Outputs
Measurable values

Example

Enemy 1 health points
Enemy 2 health points
Enemy 1 robustness
Enemy 2 firepower
Player attack power
Player shooting range
...

Success percentage
Completion time
Accuracy

Tower Defense Game
Problem Definition: A General Scenario

Enemy 1 health points
Enemy 2 health points
Enemy 1 robustness
Enemy 2 firepower
Player attack power
Player shooting range
…

Tower Defense Game

- Constraints
  - Enemy 1 health points ∈ [5; 10]
  - Enemy 2 health points ∈ [8; 13]
  - Enemy 2 hp > Enemy 1 hp
  - Player hp > 3 * Enemy 2 hp
  - Etc…

- Objectives
  - Score ≈ 75 %
  - Completion time < 1min30
  - Success percentage
  - Completion time
  - Accuracy
  - …
A Multi-Agent System for Autonomous Control of Game Parameters
A Multi-Agent Modeling

- Parameter Agent 1
- Parameter Agent 2
- Constraint Agent 1
- Constraint Agent 2
- Objective Agent 1
- Objective Agent 2
- Measure Agent 1
- Measure Agent 2

Agents have a local perception of their environment
Agents have a local perception of their environment

Constraints and objectives compute their satisfaction level
Behaviour

MAS

Game engine

observes

Measure-agent

Score

Objective-agent

Objective: 75%

Parameter-agent

Player Resources

Enemy

HP

Enemy

Power

Scenario Control for (Serious) Games using Self-organizing Multi-Agent Systems
Behaviour

Score = 60%

1

Game engine

MAS

observes

Objective-agent

Objective: 75%

Measure-agent

Score

Parameter-agent

Player Resources

Enemy

HP

Enemy

Power

Score = 60%
Behaviour

Score = 60%

1. Game engine

2. Measure-agent
   Score
   value = 60%

MAS

Objective-agent
   Objective: 75%

Parameter-agent
   Player Resources

Enemy
   HP

Enemy
   Power
Behaviour

Scenario Control for (Serious) Games using Self-organizing Multi-Agent Systems

1. **Game engine**

2. **Measure-agent**
   - **Score**
   - Value = 60%

3. **Objective-agent**
   - **Objective**: 75%
   - Not satisfied

4. **Parameter-agent**
   - **Player Resources**
   - **Enemy HP**
   - **Enemy Power**

Score = 60%
**Behaviour**

1. **Game engine**
   - Score = 60%

2. **Measure-agent**
   - Score
   - Value = 60%

3. **Objective-agent**
   - Objective: 75%

4. **Parameter-agent**
   - **Player Resources**
   - **Enemy HP**
   - **Enemy Power**

Not satisfied: request: Go up!

MAS observes.

Scenario Control for (Serious) Games using Self-organizing Multi-Agent Systems
Behaviour

MAS

Game engine

Score = 60%

1

Measure-agent

Score

2

value = 60%

4

request: Go up!

3 Not satisfied

Objective-agent

Objective: 75%

Parameter-agent

Player Resources

5 request: Go up!

5 request: Go down!

Enemy

HP

5 request: Go down!

Enemy Power

Scenario Control for (Serious) Games using Self-organizing Multi-Agent Systems
Score = 60%

1. Game engine

2. value = 60%

3. Not satisfied

4. request: Go up!

5. request: Go up!

6. Send value to Game platform

Objective-agent
Objective: 75%

Measure-agent
Score

Parameter-agent
Player Resources

Enemy HP

Enemy Power

observes

Send value to Game platform
Behaviour

- Works fine where there is no problem

- When a conflict occurs, behaviours need to be cooperative
Behaviour – Parameter Agents (1)
Behaviour – Parameter Agents (2)

Game engine

Measure-agent

Score

Constraint

Enemy

Resources

Parameter-agent

Player Resources

Up, down
At the Global Level

- Local cooperative interactions lead to a global optimization of the satisfaction

- Successfully applied
  - To a population equilibrium simulation
  - To a tower defense game
Conclusion

- Real time adaptivity
- Minimal information from the domain is needed
- Concepts are general enough to be applied to various kinds of games
Thank you


## Dependencies Matrix

<table>
<thead>
<tr>
<th></th>
<th>Player HP</th>
<th>Enemy HP</th>
<th>Defense firepower</th>
<th>Enemy speed</th>
<th>Enemy frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Completion time</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>
Adaptive Value Tracker

CurrentFB

LastFB

\[ \Delta = \Delta \cdot 2 \]
\[ v = v + \Delta \]
\[ \Delta = \Delta / 2 \]
\[ v = v - \Delta \]

\[ \Delta = \Delta / 2 \]
\[ v = v + \Delta \]
\[ \Delta = \Delta \cdot 2 \]
\[ v = v - \Delta \]

A Multi-Agent System for Autonomous Control of Game Parameters