Adaptive Game AI

Pieter Spronck
Tilburg University / TiCC
Open University
What Is Game AI All About?
AI in Games
Game Architecture

- **Game State**
- **Renderer**
- **Simulator** (Physics+Animations)
- **Agents** (controlled by Game AI or Human)

Connections:
- Game State sends updates to Simulator.
- Simulator sends information to Game State.
- Simulator sends actions to Agents.
- Agents provide information to Simulator.
Is This AI?

- Idle Behaviours
- Patrol Formations and Tactics
- Reinforcements
- Taking Cover
- Kill Reactions
Move-knife
To South.
Drambuie:
Get-a knife
"Stop Thief!!!"
Drambuie:
¥
The Illusion of Human Behaviour

- Game AI is about the illusion of human behaviour
  - Smart, to a certain extent
  - Unpredictable but rational decisions
  - Emotional influences
  - Body language to communicate emotions
  - Being integrated in the environment
- Due to the increasing realism of game worlds, evoking the illusion becomes harder every day
Changes in Gaming
AI Tech Situation

- As fidelity of worlds grows, challenge of “just keeping up” ratchets up
  - Pathfinding
    - On a 2d tilemap with 90 degree walls easy
    - On an arbitrary polygon mesh, not as easy
  - Combat posture
    - switching between idle and combat sprites easy
    - managing 100+ bone model, not as easy
- Just keeping old features working is hard
Evoking the Illusion

- No obvious cheating
- Being unpredictable
- No obviously stupid behaviour
- Using the environment
- Self-correction
- Creativity
- Being human
The Complexity Fallacy

- It is a mistake to think that complex AI equals better character behaviour
  - Complex AI might come out looking stupid
  - While simple AI might come out looking perfect

“Clever AI was programmed into the game. The ghosts would group up, attack the player, then disperse. Each ghost had his own AI.”
The Perception Window

- Make sure that a characters’ AI matches their purpose and the attention they’ll get from the player.
- Adding more might look like bugs or sloppy programming.
- A change in behaviour is far more noticeable than the behaviour itself.
- For incidental characters, best is to have only two behaviours: normal and player-spotted.
Non-Obvious Cheating

- Is very common
  - Adding markers to game world
    - For path-planning
    - For cover recognition
  - Looking through obstacles
  - Awarding small advantages
  - Teleportation
  - ...

- Is acceptable from the point-of-view of entertainment
  - If you can think of an easy way to achieve a certain effect, for game AI that is preferable to implementing the “right” (academic) solution
Adaptive Game AI
Games with Learning

- Black & White
- Creatures
- Colin McRae Rally 2.0
- Max Payne
Goals of Adaptive AI

- Self-correction
- Creativity
- Scalability
Why the Hesitation?

- Rarely needed
- Learning the wrong lessons
- Difficulty to get desired results
  - Suitable fitness function
  - Increased entertainment
- Difficult to modify, test, debug, and understand
- Low efficiency of common techniques
  - Neural networks
  - Evolutionary algorithms
- It does not always make sense
Problem of Complexity

- Huge state-action space
- Uncertainty
  - Non-deterministic
  - Incomplete information
  - Multiple parallel agents
- Often real-time
Requirements

- Online computational requirements
  - Speed
  - Effectiveness
  - Robustness
  - Efficiency
- Functional requirements
  - Clarity
  - Variety
  - Consistency
  - Scalability
Necessities

- Use as much prior knowledge as possible
  - Improves all computational requirements
- Good performance measure (fitness)
- Learning by
  - Optimisation
  - Reinforcement
  - Imitation
- Avoid overfitting
- Minimise dependencies between behavioural aspects
  - E.g. learning of “kill hotspots” and “weapon preference”
- Balance between exploration and exploitation
Reinforcement Learning with Dynamic Scripting
Dynamic Scripting

Knowledge Base A
- Generate script
- Script A
- Script control
- Weight updates

Knowledge Base B
- Generate script
- Script B
- Script control

Computer-controlled team

Human-controlled team

Combat
0.3.2: Chromatic Orb hits Blue Wizard B. Saving throw fails (1 < 12). Blue Wizard B receives 3 points of damage.
0.3.3: Blindness hits Red Fighter B. Saving throw fails (11 < 14).
0.3.3: Blindness hits Red Wizard A. Saving throw fails (9 < 12).
0.3.4: Red Fighter B hits Blue Wizard B’s Minion 3 for 3+3 points of damage. Red Fighter B hits Blue Wizard B’s Minion 3 for 2+1 points of damage.
0.3.8: Blue Fighter A hits Red Fighter B for 5+3 points of damage. Blue Fighter A hits Red Fighter B for 5+1 points of damage.
Validation Experiments

- How quick is dynamic scripting able to adapt to an unchanging tactic?
- Or, can dynamic scripting quickly force a (human) player to change tactics?
Actions
// OFFENSIVE WIZARD SCRIPT

if healthpercentage < 50 then
    drink("Potion of Healing");

if roundnumber < 1 then
    cast("Mirror Image");

if distance(closestenemy("Wizard"), furthestenemy("Wizard")) < 200 then
    cast("Fireball", centreenemy("Wizard"));

if distance(closestenemy("Fighter"), furthestenemy("Fighter")) < 200 then
    cast("Stinking Cloud", centreenemy("Fighter"));

cast(strongoffensive, closestenemy);

if distance(closestenemy) > 200 then
    rangedattack(defaultenemy);

meleeattack(closestenemy);

meleeattack(closestenemy);
Party Fitness

- Indicates how well the party functions as a whole
- Takes into account
  - Win or loss
  - Number of surviving party members
  - Total health of surviving party members
- Is used to decide when one party outperforms the other
**Individual Fitness**

- Indicates how well an individual functions in the party
- Takes into account
  - Party fitness
  - Individual’s health at the end of the encounter
  - Individual’s time of death
  - Damage done to enemies
- Determines how the weights in the individual’s rulebase will be changed
Reward/Penalty Determination

- Maximum reward at individual fitness 1
- Maximum penalty at individual fitness 0
- No reward or penalty at fitness \( b \) (break-even)
- Linear extrapolation
Tactics

- Basic
  - Offensive
  - Disabling
  - Cursing
  - Defensive

- Composite
  - Random team
  - Random agent
  - Consecutive
Party Fitness Progression

- Absolute

- Average over last 10 encounters
Comparison DS/MCC

Monte Carlo
Dynamic scripting
Biased Rulebases
Igor: Round 1, ladies and gentlemen!

PAUSED
Blanche the Cleric casting Summon Creature I
Dynamic Scripting Enhancements

- Automatic creation of strong knowledge bases (Marc Ponsen)
- Automatic determination of priority values (Timor Timuri)
- Generalisation to different team configurations (Richard Arnoldussen)
- Effective and diverse (i.e., interesting) rulebases (István Szita)
Player Modeling
Player Model

- Modeling
  - Player actions
  - Player preferences
- Model types
  - Evaluation functions
  - Neural networks
  - Finite state machines
  - Probabilistic models
  - Case-based models
- Model application
  - Strong AI
  - Appropriate AI
Opponent Model Search
GAI-Yannick wins $14 with a Pair of Fives, Jack kicker
PaSSAGE Player-Specific Stories via Automatically Generated Events
Opponent Models in Video Games

- Model types
  - Implicit
  - Explicit

- Model goals
  - Effectiveness
  - Entertainment
  - Suitability
Research Lines

- Opponent modelling for effectiveness
  - Formations
- Player profiling
  - Incongruity scaling
  - Psychological profiling
- Player Preference Modelling
  - Civilization IV
Formations
Dynamic Formations

- General formation framework
- Learning algorithm to create suitable formations
  - Online adaptation of 8 formation parameters
  - Offline learning of suitable parameters against player models
- Fast modeling algorithm to determine a model for the current player
  - Player model based on 3 observations

Note: This research was performed by Marcel van der Heijden, Sander Bakkes, and Pieter Spronck
Open Real-Time Strategy
Michael Buro, University of Alberta
ORTS Challenge #4

- Small-scale tactical combat, formations
  - Flat map
  - Moving obstacles
  - Perfect information
- 5 minutes to destroy 50 opponent tanks
  - Units can fire in a pre-defined range
  - Time-out after firing
- 6 AIs available (AIIDE2007): Blekinge, NUS, UBC, UM, WarsawA, WarsawB
Dynamic Formation Shape

- Centered around leader
- Adapt 8 parameters
- Shape
  - 1-n lines
  - 1-m units per line
  - $\alpha$, $\beta$, $\gamma$
- Speed
- Target
- Low-level behaviour
Replacement of Destroyed Units

- Neighbouring units take place of destroyed units
- Fast and cheap
Movement

- Parameter 6: speed
- Leader moves
- Formation moves with leader
- Units attempt to stick to their place in the formation
- If units fall seriously behind, the speed of the formation is adjusted
Target Selection (Parameter 7)

(a) Relative

(b) Leader

(c) Centre

(d) Shortest
Combat Behaviour (Parameter 8)

- Choice of five
  - Overrun – keep pushing forward
  - Hold – stop when opponent is within weapon range
  - Retreat – hit and run
  - Bounce – hit an run, but return after half of time-out
  - Border- stay out of weapon range during time-out
Performance Measures

- **Absolute performance:** percentage of games won during last 50 games
  - Sequence of 200 games against each opponent
- **Relative performance:** fitness value obtained
  - Range: \([-50,50]\]
- **Turning point:** win-loss ratio of last 20 games becomes 15-5 or better
  - Statistically, it is 98% certain that learning opponent is now better than static opponent
Dynamic Formations w/o Modelling

(a) Blekinge
(b) NUS
(c) UBC
(d) UM
(e) WarsawA
(f) WarsawB
Player Model in 3 Parameters

- Number of formations
  - Determined with k-means clustering
- Unit distribution
  - Proportion of width and height of rectangle around all units
- Unit distance (density)
  - Average distance of each unit to its closest neighbour
3 Base Models

- Blekinge
- UBC
- WarsawB

- Image: Normalized Gaussian distribution of all three parameters for all three models
Player Modelling During Game

- Measure the three parameters ~100 timesteps after the game started
- Calculate the combined likelihood of observed parameters for each known model (Bayes’ theorem)
- Select the model that best explains the observations
  - For the first few trials
  - After that, do regular adaptation
### Performance Comparison

<table>
<thead>
<tr>
<th>Opponent</th>
<th>No player modelling</th>
<th>With player modelling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abs.perf.</td>
<td>Rel.perf.</td>
</tr>
<tr>
<td>Blekinge</td>
<td>100%</td>
<td>37</td>
</tr>
<tr>
<td>NUS</td>
<td>12%</td>
<td>-31</td>
</tr>
<tr>
<td>UBC</td>
<td>94%</td>
<td>15</td>
</tr>
<tr>
<td>UM</td>
<td>66%</td>
<td>4</td>
</tr>
<tr>
<td>WarsawA</td>
<td>44%</td>
<td>-1</td>
</tr>
<tr>
<td>WarsawB</td>
<td>98%</td>
<td>17</td>
</tr>
</tbody>
</table>
Adapting to Entertain
Learning and Enjoying Chess

Deep Blue (1997)

Chess Challenger (1978)
Iida’s Theory of Refinement

Seesaw game:
Optimal length of time outcome is uncertain

Refinement

\[ \frac{\sqrt{B}}{D} = 0.07 \]

Complexity
Noble uncertainty:
New tactics are possible

Fairness
Draw ratio:
Matching opponents
Enjoying Games

- Computer should *be able* to play stronger than the human player
- Computer should *adapt* to the level of skill of the human player
- Computer should constantly offer *new challenges*

In short: Computer and human increase their playing skill in *parallel*
- Manual
- Coarse
- Simple
Automatic Scaling of Game AI

- High-fitness penalising
  - Award the highest fitness to the “most equal” AI, instead of to the “best” AI

- Weight clipping
  - Increase AI variety when the computer plays too well
  - Decrease AI variety when the computer plays badly

- Top culling
  - Remove the currently “best” knowledge when the AI plays too well
  - Reactivate the “best” knowledge when the AI plays badly
Difficulty Scaling Tests

- Against five different basic tactics
- Against three different composite tactics
- Against Neverwinter Nights game AI

Novice tactic
- Imitates novice player
- Knows obvious good tactics
- Does not know subtle good tactics
Scaling Results

- *Without automatic scaling*, dynamic scripting wins against all tactics
- With *high-fitness penalising* an even game is achieved against 2 out of 8 tactics
- With *weight clipping* an even game is achieved against 7 out of 8 tactics
- With *top culling* an even game is achieved against 8 out of 8 tactics, combined with the lowest standard deviation
Difficulty Scaling

- Adaptive effectiveness can be easily converted to difficulty scaling
- Often not useful against strong players
  - Losing a sense of accomplishment
  - Best results against novices
- For highest enjoyment
  - Should the AI play stronger than the human player?
  - How much stronger?
  - To what extent does this depend on the type of game?
  - Is this equal for all human players?
Incongruity-Based Adaptive Game Balancing
Incongruity

high contextual complexity

positive incongruity

negative incongruity

low contextual complexity

context (game + AI)

internal (mental) complexity

system (human)

frustration

satisfaction

boredom
Press F1 for information on how to play.
## Incongruity in Glove

### Contextual Difficulty Manipulation

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Health vs. Progress</th>
<th>Incongruity level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy difficulty</td>
<td>~75% 50%</td>
<td>Negative (boredom)</td>
</tr>
<tr>
<td>Balanced difficulty</td>
<td>~50% 50%</td>
<td>Low (pleasure)</td>
</tr>
<tr>
<td>Hard difficulty</td>
<td>~25% 50%</td>
<td>Positive (frustration)</td>
</tr>
</tbody>
</table>
Experimental Setup

- N=24
- 1 practice game
- 3 games
  - 1 of each difficulty setting
  - random order
- A questionnaire after each game
  - 26 questions
  - Measuring boredom, frustration, pleasure, concentration, and curiosity

Note: This research was performed by Giel van Lankveld and Pieter Spronck, assisted by Jaap van den Herik, Matthias Rauterberg, and Maarten Schadd
Confirmation

- Incongruity theory is partly confirmed
  - Frustration is confirmed
  - Pleasure is partly confirmed
    - For balanced-hard
  - Boredom is not statistically confirmed
    - Game too complex?
    - Few test subjects?
    - Coping?
Player Profiling in Civilization IV
Personalities
AI Preferences

- Aggression (low, medium, high, very high)
- Culture (low, high)
- Gold (low, high)
- Growth (low, high)
- Military (low, medium, high)
- Religion (low, high)
- Science (low, high)
Approach

- Determine player personality from observing in-game behaviour
  - Collect observations (about 70,000 observations from 240 games)
  - Data mining on observations (SVM)
  - Building models of AI character preferences
  - Validating models on different AIs
  - Determining traits of humans by using the models
Features

- 5 × War
- Cities
- Units
- Population
- Gold
- Land
- Techs
- Score
- Economy
- Industry
- Agriculture
- Power
- Culture
- Maintenance
- GoldRate
- Research
- CultureRate
- Etc.
Modified Features

- Diff $v_{0t} - v_{1t}$
- Derivate $v_{0t} - v_{0(t-1)}$
- Trend $(v_{0t} + v_{0(t-1)} + ... + v_{0(t-4)})/5$
- DiffDerivate
- DiffTrend
- TrendDerivate
- DiffTrendDerivate
Modelling of AI

<table>
<thead>
<tr>
<th>Category</th>
<th>Frequency Baseline</th>
<th>Correctly Classified</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggression</td>
<td>52.1156</td>
<td>24.9158</td>
<td>-52.2%</td>
</tr>
<tr>
<td>Culture</td>
<td>99.9939</td>
<td>88.2004</td>
<td>-11.8%</td>
</tr>
<tr>
<td>Gold</td>
<td>69.8837</td>
<td>38.5671</td>
<td>-44.8%</td>
</tr>
<tr>
<td>Growth</td>
<td>69.4979</td>
<td>30.7471</td>
<td>-55.8%</td>
</tr>
<tr>
<td>Military</td>
<td>66.5973</td>
<td>34.5723</td>
<td>-48.1%</td>
</tr>
<tr>
<td>Religion</td>
<td>83.2272</td>
<td>58.9835</td>
<td>-29.1%</td>
</tr>
<tr>
<td>Science</td>
<td>82.9884</td>
<td>70.9675</td>
<td>-14.5%</td>
</tr>
</tbody>
</table>
## Modelling of Humans

- Use the AI classification to model a human

<table>
<thead>
<tr>
<th>Feature</th>
<th>Frequency Baseline</th>
<th>Veteran Feature Set</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggression</td>
<td>78.9784</td>
<td>32.0236</td>
<td>-59.5%</td>
</tr>
<tr>
<td>Culture</td>
<td>50.4912</td>
<td>92.3379</td>
<td>82.9%</td>
</tr>
<tr>
<td>Gold</td>
<td>55.5992</td>
<td>81.3360</td>
<td>46.3%</td>
</tr>
<tr>
<td>Growth</td>
<td>99.8035</td>
<td>97.8389</td>
<td>-2.0%</td>
</tr>
<tr>
<td>Military</td>
<td>79.1749</td>
<td>76.6208</td>
<td>-3.2%</td>
</tr>
<tr>
<td>Religion</td>
<td>50.4912</td>
<td>50.8841</td>
<td>0.8%</td>
</tr>
<tr>
<td>Science</td>
<td>71.7092</td>
<td>43.6149</td>
<td>-39.2%</td>
</tr>
</tbody>
</table>
Classifying Leaders

- New classification: model leaders

![Bar chart showing comparisons between different historical leaders.](chart.png)
Classifying Leaders

- New classification: model alternate leaders

![Bar chart showing comparison of leaders' contributions.](chart.png)
Leader Characteristics

- 191 traits per leader
  - War attitudes
  - Peace attitudes
  - Religion attitudes
  - Trade attitudes
  - Anger values
  - Memory
  - ...

- Some of these seem better able to explain leader behaviour than the “flavours”
Tentative Conclusions

- It is hard to abstract the complex Civilization IV environment to feature sets
- It is very hard to model preferences
  - Feature set insufficient?
  - Insufficient observations?
  - In-game behaviour depends on more than preferences
- How to translate human play to preferences?
- It seems possible to model leaders
  - A player acts “like Alexander”
Conclusions

- Player models can be used to make adaptive game AI learn faster
  - ORTS experiments
  - Thesis Bakkes
- Our data mining approach seems lacking for determining player preference models in really complex games
  - Though high-level player models might be possible
  - In future work, do this for simpler games
The Future
References
Literature

Contact details

Pieter Spronck
pieter@spronck.net
http://www.spronck.net