AIIDE 2014

Artificial Intelligence for Interactive Media and Games

Professor Charles Rich
Computer Science Department
rich@wpi.edu

Artificial Intelligence and Interactive Digital Entertainment Conference 2014

October 3-7, 2014
North Carolina State University
Raleigh, North Carolina, USA

http://www.aiide.org

Proceedings online at: http://www.aaai.org/Library/AIIDE

CS/MGD 4100 (C 16)
Mission Statement

AIIDE is the definitive point of interaction between entertainment software developers interested in AI and academic and industrial AI researchers. Sponsored by the Association for the Advancement of Artificial Intelligence (AAAI), the conference is targeted at both the research and commercial communities, promoting AI research and practice in the context of interactive digital entertainment systems with an emphasis on commercial computer and video games.

By the Numbers

- 3 days
- 120-150 attendees (typically 85% academic, 15% industry)
- 14 papers presented (12 university / 2 joint w. same game co)
- 6 technical sessions
- 4 invited talks (2 industry / 2 academia)
- 15 posters
- 7 “playable experiences” demos
- 5 workshops (immediately before main conference)
- 1 StarCraft AI competition (before conference)
Workshops

1. 3rd Workshop on Games and Natural Language Processing (GAMNLP-14) [full day]
   • NL generation: of speech... to narrative structure
   • NL understanding: of speech... to words... to conversations

2. AI in the Adversarial Real-Time Games Development Process [full day]
   • heavily algorithmic
   • search, optimization, etc.
   • StarCraft

3. First Diversity in Games Research Workshop
   • encourage students from under-represented groups to engage in graduate training games research
   • with support from:
     – CRA Committee on the Status of Women in Computing Research
     – Coalition to Diversity Computing

4. Experimental AI in Games Workshop
   • bunch of stuff not yet ready for “prime time”
   • using web search as a game mechanic
   • generating games using crowd sourcing
Workshops

5. 3rd International Workshop on Musical Metacreation
   - computer programs that write music
   - human/computer collaborative performances
   - games that create or modify music

Technical Sessions

1. Human Modeling
2. Procedural Content Generation
3. Strategy AI
4. Narrative
5. NPC Behavior
6. Gameplay Analytics
1. Human Modeling

Toward Personalised Gaming via Facial Expression Recognition

- Pars Blom, Sander Bakkes, Shimon Whiteson, Diederik Roijers, Robert Valenti, Theo Gevers, Intelligent Systems Lab, U. Amsterdam
- Check Tan, U. of Technology, Games Studio, Sydney, Australia

Personalisation via Facial Expressions

- personalization of level difficulty is standard:
  - novice
  - intermediate
  - expert, etc.

- but usually interact with player to select difficulty before game begins or between levels

- because it would be too disruptive to interrupt player during play

- but, could we do this dynamically and unobtrusively?
Personalisation via Facial Expressions

- off-the-shelf facial expression recognition software: INSIGHT (sightcorp.com)

**INFINITE MARIO BROS**

- open-source clone of classic game
- procedurally generated levels and dynamically added segments

Currently being played segment

Segment still to be generated based on assessments of the player experience
Personalisation via Facial Expressions

Algorithm 1 Facial Expression-based Gradient Ascent Optimisation

1: procedure GA\textsc{Optimize}(e_t, e_{t-1}) \quad \triangleright \text{Emotion vectors of current and previous segment}
2: \quad \alpha \leftarrow 5 \times (1 - \text{Var}(e_1)) \quad \triangleright \text{Calculate } \alpha, \text{ scale to action space}
3: \quad \text{for each } chunk \text{ do}
4: \quad \quad \text{if playerDied() then}
5: \quad \quad \quad \phi = \text{round}(5 \times \alpha \times e_t[\text{Anger}])
6: \quad \quad \quad \text{chunk.decreaseChallengeLevel}(\phi)
7: \quad \quad \text{else if segmentFinished() then}
8: \quad \quad \quad \text{if } e_t[\text{Neutral}] \leq 0.8 \times \alpha \text{ then}
9: \quad \quad \quad \quad \text{chunk.decreaseChallengeLevel}(1)
10: \quad \quad \text{else}
11: \quad \quad \quad \epsilon \leftarrow \arg\max_c |e_t - e_{t-1}|
12: \quad \quad \quad \text{nextAction} \leftarrow \text{round}(\epsilon \times \alpha)
13: \quad \quad \quad \text{if } \epsilon \in \{\text{angry, neutral}\} \text{ then}
14: \quad \quad \quad \quad \text{nextAction} \leftarrow -\text{nextAction}
15: \quad \quad \quad \text{nextChallengeLevel} \leftarrow \text{previousChallengeLevel} + \text{nextAction}
16: \quad \quad \quad \text{return newChallengeLevel}

Next step?
- more accuracy
- other inputs
  - gaze
  - body
  - ...

- pilot user study with 10 participants:
  - P = personalized system preferred
  - S = static preferred
  - B = both preferred equally
  - N = neither preferred

<table>
<thead>
<tr>
<th>Participant</th>
<th>Easy</th>
<th>Normal</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P</td>
<td>P</td>
<td>S</td>
</tr>
<tr>
<td>2</td>
<td>P</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>S</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>4</td>
<td>P</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>6</td>
<td>P</td>
<td>P</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>P</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>9</td>
<td>P</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>10</td>
<td>S</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

Totals: 70% P, 70% P, 40% P
30% S, 30% S, 30% S
6% B, 6% B, 6% B
0% N, 0% N, 30% N
2. Procedural Content Generation

Generative Methods for Guard and Camera Placement in Stealth Games

- Qihan Xu, Jonathan Tremblay, Clark Verbrugge
  School of Computer Science, McGill U., Montreal, Quebec, Canada

Guard and Camera Placement

- Stealth Games
  - e.g., *Mark of the Ninja, Metal Gear Solid*
  - more puzzle than combat
  - *placement* of guards (NPCs) and cameras greatly affects challenge
  - a lot of manual effort to design levels that are believable and challenging
  - can we automate this placement?
Guard and Camera Placement

Contributions (quoting authors):

- A heuristic approach to camera placement based on weakening a solution to the well known “art gallery problem” for simple polygons.

- The design of a flexible, grammar-based method for defining roadmap-based guard patrol routes.

- Application of quantitative metrics that demonstrate how different parameterizations affect the existence of level solutions and player perception of difficulty.
Guard and Camera Placement

- The design of a flexible, grammar-based method for defining roadmap-based guard patrol routes.

Guard and Camera Placement

- Application of quantitative metrics that demonstrate how different parameterizations affect the existence of level solutions and player perception of difficulty.
3. Strategy AI

Game Tree Search over High-Level Game States in RTS Games

- Albert Uriarte and Santiago Ontanon, Computer Science Dept., Drexel

High-Level Game Tree Search

- “Classic” AI algorithms
  - search trees in state space (based on “next move”)
    - alpha-beta search
    - Monte Carlo tree search (MCTS)
  - successfully applied to chess, checkers, cards, ...
  - but for RTS games, state space gets really big

- Basic solution approach:
  - apply abstraction to state space to get smaller searches
High-Level Game Tree Search

Algorithm 1: MCTS Considering Durations

1. function MCTSSEARCH(n₀)
2.    n₀ ← CREATE_NODE(n₀)
3. while within computational budget do
4.    nᵢ ← DEF_Policy(nᵢ)
5.    Δᵢ ← DEFAULT_Policy(nᵢ)
6.    BACKUP(nᵢ, Δᵢ)
7.    return (BEST_CHILD(n₀)).action
8. function CREATE_NODE(a, n₀)
9.    n₀.parent ← n₀
10.   n₀.lastSimult ← n₀.lastSimult
11.   noplayer ← PLAYER_TO_MOVE(a, n₀, lastSimult)
12. if BOTH_CAN_MOVE(a) then
13.   n₀.lastSimult ← noplayer
14. return n
15. function DEFAULT_POLICY(a)
16.   lastSimult ← n₀.lastSimult
17.   s ← n₀.s
18. while within computational budget do
19.    p ← PLAYER_TO_MOVE(s, lastSimult)
20.    if BOTH_CAN_MOVE(a) then
21.       lastSimult ← p
22.       simulate game s with a policy and player p
23. return s.reward
High-Level Game Tree Search

- Experimental Evaluation using StarCraft

![Graph 1](image1.png)

Figure 5: Average search time for each algorithm grouped by the number of groups in the high-level state.

![Graph 2](image2.png)

Figure 4: Average branching factor grouped by the number of groups in the high-level state.

4. Narrative

Glaive: A State-Space Narrative Planner
Supporting Intentionality and Conflict

- Stephen Ware and R. Michael Young,
  Computer Science Dept., NC State U.
Narrative Planning

- Narrative? Another word for “story”
- The minimum story is:
  ...two events and an explanation
- The game AI problem:
  • given
    - a set of characters (and their motivations, etc.)
    - an initial state of the world (including characters)
    - a desired goal state (authorial intent)
  • produce
    - a believable and interesting story (sequence of events) that goes from initial to final state

Why would you want to do this?
• save the effort of manual story writing (get more stories and replayability)
• make story interactive (replan after user actions)

Why is this hard?
• tension between two desires:
  – strong story: ensure coherent plot defined by author
  – strong autonomy: ensure accurate simulation of each character
Narrative Planning

- Set up as classic AI planning (search) problem

Domain

<table>
<thead>
<tr>
<th>Event</th>
<th>Precond.</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>dig(thief, Item)</td>
<td>theif alive.</td>
<td>R has Item.</td>
</tr>
<tr>
<td></td>
<td>theif has Item.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>theif knows Item.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>J alive.</td>
<td>R open.</td>
</tr>
<tr>
<td></td>
<td>J has Item.</td>
<td>~ J alive.</td>
</tr>
<tr>
<td></td>
<td>R open.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R open.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Problem

Initial State: R buried. J alive.
J knows R, J intends U has R.
U alive, U knows U has R.
N alive, N armed.
N intends N has R.
Goal: U has R, N alive.

Key

- J = Indiana Jones
- R = Ark of the Covenant
- N = Nazi Soldiers
- U = US Army

Goal Graph: J Intends U has R.

Layer 2

- take(J, R)
- give(U, R)
- give(U, U)
- give(U, N)
- give(U, J)

Layer 1

- give(J, R)
- give(J, J)
- give(J, U)

Layer 0

- give(J, U)
- give(J, R)

Narrative Planning

Plan Graph (from initial state)
Narrative Planning

- Technical issues
  - resolving conflicts (between characters)
  - heuristics for searching space efficiently
  - many other very technical issues in planning and representation

5. NPC Behavior

Belief-Driven Pathfinding Through Personalized Map Abstraction

- Davide Aversa and Savros Vassos, Dept of Computer, Control and Management Engineering, Sapienza U. of Rome
Belief-Driven Pathfinding

- Pathfinding
  - NPC finding an appropriate path to navigate from current location to desired location
  - essential mechanism in many games
  - crucial for interaction quality and believability
  - A* algorithm most commonly used

- Belief-Driven/Personalized?
  - rather than all NPC’s sharing same pathfinding module
  - each NPC plans path based on what it has observed or been told (beliefs) about environment

---

Belief-Driven Pathfinding

- Technical challenge: reduce expense of doing this for large maps and large number of NPCs

- Solution approach: apply A* to abstraction(s) of map

Figure 1: A small map decomposed in five regions, and a corresponding gate connectivity graph. Gate \( G_1 \) is a set of three portals, each of which is denoted by a double arrow.
6. Gameplay Analytics

Developing Social Identity Models of Players from Game Telemetry Data

- Chon-u Lim and D. Fox Harrell, Computer Science and Artificial Intelligence Lab, MIT

[poster/short paper]
Gameplay Analytics

- Analytics?
  - gathering data (stats) from gameplay
  - player actions, timing, scores, customization, etc.
  - applying statistical analyses, data mining, machine learning, etc.
  - to better understand game design
  - make better games
  - sell more games...

Gameplay Analytics

- Player statistics in *Team Fortress 2* (FPS) predicted aspects of their identities expressed in their social networking profiles:
  - number of friends
  - number of uploaded screenshots
  - number of uploaded videos
Predications from Gameplay Analytics

1. Veteran players with high customization have higher number of friends

2. Offensive-driven players upload more screenshots

3. Stealth or support-driven players upload more videos

Privacy concerns?

Invited Talks

- Constraint-Based Multitasking in *The Sims 4*
  - Peter Ingebritson, Senior Software Engineer, ElectronicArgs

- Tracking Sports Players and Understanding Their Movements
  - Peter Carr, Disney Research
Invited Talks [cont’d]

- Natural Language Dialogue in Interactive Learning Environments
  - Kristy Boyer, NC State U.

- Vegans at Your Barbecue: How to Feed Hungry Game AI Developers
  - Squirrel Eiselerh, GuildHall at the Southern Methodist U.

Questions? Comments?

- P.S. The other big yearly game AI confab is the yearly AI Summit at GDC
  - Feb 27-28, 2017, San Francisco
  - organized by the AI Game Programmers Guild (http://gameai.com)
  - approx 85% industry, 15% academic