WPI

AIIDE 2014

Artificial Intelligence for
Interactive Media and Games

Professor Charles Rich
Computer Science Department
rich@wpi.edu

CS/MGD 4100 (B 14)   1

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 (B 14)   2

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.

WPI CS/MGD 4100 (B 14)   3

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)

WPI CS/MGD 4100 (B 14)   4

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

CS/MGD 4100 (B 14)   5

Workshops

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

CS/MGD 4100 (B 14)   6

Workshops

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

CS/MGD 4100 (B 14)   7

Technical Sessions

1. Human Modeling

2. Procedural Content Generation

3. Strategy AI

4. Narrative

5. NPC Behavior

6. Gameplay Analytics

CS/MGD 4100 (B 14)   8

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

CS/MGD 4100 (B 14)   9

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***?

CS/MGD 4100 (B 14)   10

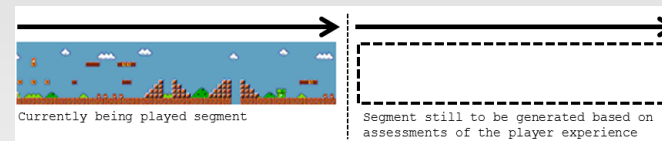## Personalisation via Facial Expressions

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

CS/MGD 4100 (B 14)   11
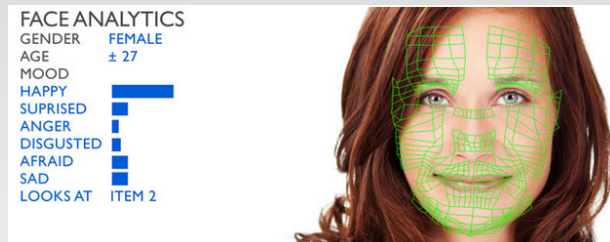
Personalisation via Facial Expressions

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

CS/MGD 4100 (B 14)   12

Personalisation via Facial Expressions

Algorithm 1 Facial Expression-based Gradient Ascent Optimisation

1: **procedure** GAOPTIMIZE(e_t, e_{t-1})   ▷ Emotion vectors of current and previous segment
2:    $\alpha \leftarrow 5 * (1 - Var(e_1))$ ▷ Calculate α, scale to action space
3:    **for** $each : chunk$ **do**
4:        **if** playerDies(t) **then**
5:            $\phi = round(5 * \alpha * e_t[Anger])$
6:            $chunk.decreaseChallengeLevel(\phi)$
7:        **else if** segmentFinished(t) **then**
8:            **if** $e_t[Neutral] <= 0.8 * \alpha$ **then**
9:                $chunk.decreaseChallengeLevel(1)$
10:            **else**
11:                $\epsilon \leftarrow argmax_e |e_t - e_{t-1}|$
12:                $nextAction \leftarrow round(\epsilon * \alpha)$
13:                **if** $e \in \{angry, neutral\}$ **then**
14:                    $nextAction \leftarrow -nextAction$
15:                $nextChallengeLevel \leftarrow previousChallengeLevel + nextAction$
16:                **return** $newChallengeLevel$

Personalisation via Facial Expressions

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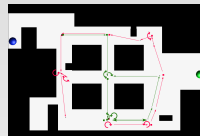 - B = both preferred equally
  - N = neither preferred

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

Participant	Easy	Normal	Hard
1	P	P	S
2	P	P	N
3	S	P	P
4	P	P	N
5	P	P	P
6	P	P	S
7	P	S	S
8	S	S	N
9	P	S	P
10	S	P	P
Totals	70% P	70% P	40% P
	30% S	30% S	30% S
	0% B	0% B	0% 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 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 parametrizations affect the existence of level solutions and player perception of difficulty.
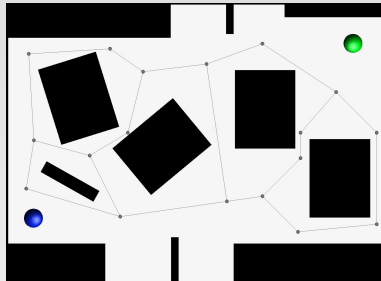
CS/MGD 4100 (B 14) 17

Guard and Camera Placement

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

CS/MGD 4100 (B 14) 18
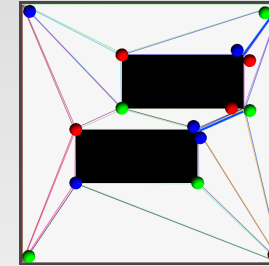
Guard and Camera Placement

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

CS/MGD 4100 (B 14) 19

Guard and Camera Placement

- Application of quantitative metrics that demonstrate how different parametrizations affect the existence of level solutions and player perception of difficulty.
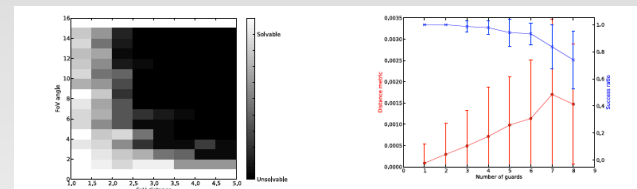

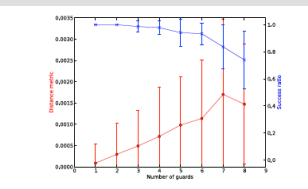Figure 5: RRT success ratio from 0% to 100% for different camera FoV angles and ranges.

Figure 6: Success ratio (×'s, blue) and distance metric (o's, red) vs. number of guards in the test level.

CS/MGD 4100 (B 14) 20

5

3. Strategy AI

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

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

CS/MGD 4100 (B 14)   21

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

CS/MGD 4100 (B 14)   22

High-Level Game Tree Search

CS/MGD 4100 (B 14)   23
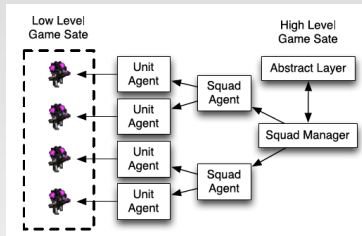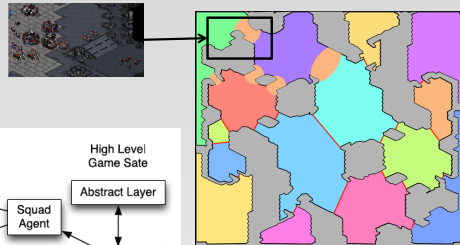
High-Level Game Tree Search

Algorithm 1 MCTS Considering Durations
```
1: function MCTSSEARCH(s₀)
2:    n₀ ← CREATENODE(s₀,∅)
3:    while withing computational budget do
4:       n_l ← TREEPOLICY(n₀)
5:       △ ← DEFAULTPOLICY(n_l)
6:       BACKUP(n_l, △)
7:    return (BESTCHILD(n₀)).action
8:
9: function CREATENODE(s,n₀)
10:    n.parent ← n₀
11:    n.lastSimult ← n₀.lastSimult
12:    n.player ← PLAYERTOMOVE(s,n.lastSimult)
13:    if BOTHCANMOVE(s) then
14:       n.lastSimult ← n.player
15:    return n
16:
17: function DEFAULTPOLICY(n)
18:    lastSimult ← n.lastSimult
19:    s ← n.s
20:    while withing computational budget do
21:       p ← PLAYERTOMOVE(s,lastSimult)
22:       if BOTHCANMOVE(s) then
23:          lastSimult ← p
24:       simulate game s with a policy and player p
25:    return s.reward
```

CS/MGD 4100 (B 14)   24

High-Level Game Tree Search
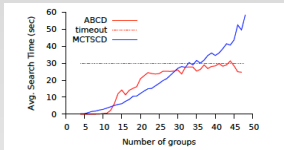
- Experimental Evaluation using StarCraft

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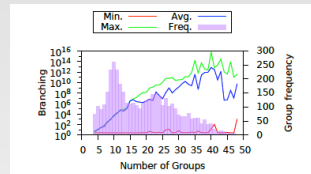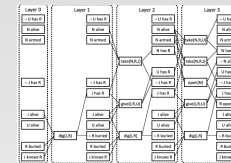

Figure 4: Average branching factor grouped by the number of groups in the high-level state.

CS/MGD 4100 (B 14)

25

4. Narrative

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

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

CS/MGD 4100 (B 14)

26

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
  - produce
    – a *believable* and *interesting* story (sequence of events) that goes from initial to final state

CS/MGD 4100 (B 14)

27

Narrative Planning

- 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
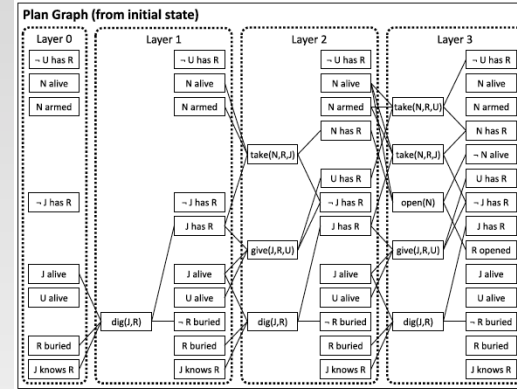
CS/MGD 4100 (B 14)

28

Narrative Planning

- Set up as classic AI planning (search) problem

Domain

dig(?char,?item)

Precons: ?char alive.
?item buried.
?char knows ?item.

Effects: ?char has ?item.
~ ?item buried.

Consent: ?char

give(?gvr,?item,?rcvr)

Precons: ?gvr alive.
?gvr has ?item.
?rcvr alive.

Effects: ?rcvr has ?item.
~ ?gvr has ?item.

Consent: ?gvr ?rcvr

open(?char)

Precons: ?char alive.
?char has ?item.

Effects: R opened.
~ ?char alive.

Consent: ?char

take(?thief,?item,?char)

Precons: ?thief alive.
?char has ?item.
~ ?char alive OR
?thief armed.

Effects: ?thief has ?item.
~ ?char has ?item.

Consent: ?thief

Problem

Initial State: R buried. J alive.
J knows R. J intends U has R.
U alive. U intends U has R.
N alive. N armed.
N intends R opened.
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: give(J,R,N)

Layer 1: dig(J,R), take(J,R,N), give(N,R,J), take(J,R,U), give(U,R,J)

Layer 0: give(J,R,U)

Indiana Jones and the Raiders of the Lost Ark

Narrative Planning

Plan Graph (from initial state)

Layer 0 / Layer 1 / Layer 2 / Layer 3 diagram with states: ~ U has R, N alive, N armed, ~ J has R, J alive, U alive, R buried, J knows R, etc., and actions take(N,R,J), take(N,R,U), give(J,R,U), open(N), dig(J,R).

Indiana Jones and the Raiders of the Lost Ark

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

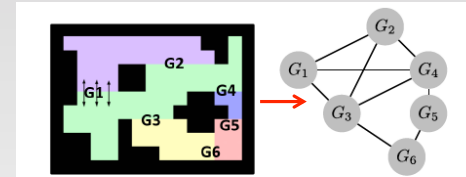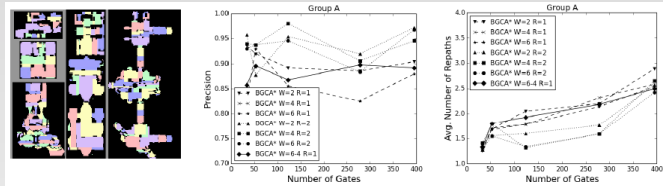- *Solution approach*: apply A* to ***abstraction(s)*** of map

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 arrrow.

Belief-Driven Pathfinding

Experimental results:

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...

CS/MGD 4100 (B 14)   37

Gameplay Analytics

- Player statistics in *Team Fortress 2* (FPS) predicted aspects of their identities expressed their social networking profiles:

  - number of friends
  - number of uploaded screenshots
  - number of uploaded videos

CS/MGD 4100 (B 14)   38

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?

CS/MGD 4100 (B 14)   39

Invited Talks

- Constraint-Based Multitasking in *The Sims 4*
  - Peter Ingebretson, Senior Software Engineer, Electronic Args

- Tracking Sports Players and Understanding Their Movements
  - Peter Carr, Disney Research

CS/MGD 4100 (B 14)   40

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 Eiselroh, GuildHall at the Southern Methodist U.

WPI CS/MGD 4100 (B 14)   41

Questions?   Comments?

- P.S. The other big yearly game AI confab is the yearly AI Summit at GDC
  - March 2-6, 2015, San Francisco
  - organized by the AI Game Programmers Guild (http://gameai.com)
  - approx 85% industry, 15% academic

WPI CS/MGD 4100 (B 14)   42