AIIDE 2011

Artificial Intelligence for
Interactive Media and Games

Professor Charles Rich
Computer Science Department
rich@wpi.edu

Artificial Intelligence and
Interactive Digital
Entertainment Conference

2011

October 12-14, 2011
Stanford University
Stanford, California, USA

http://www.aiide.org

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
- about 130 attendees (guess: 85% academic, 15% industry)
- 17 papers presented (all academic)
- 6 technical sessions
- 5 invited talks (all from industry)
- 1 panel
- 19 poster/demos
- 2 workshops (immediately before main conference)
- 1 StarCraft AI competition (before conference)

Workshops

- Intelligent Narrative Technologies and Non-Player Character AI (2 days)
  - automatically generating stories
  - making believable virtual characters
- AI in the Game Development Process (1/2 day)
  - automated playtesting
  - partially automating game design and development

Technical Sessions

1. Planning in Games
2. Character Agents
3. Story
4. Learning
5. Authoring
6. Recognizing Player Activity

1. Planning in Games

Build Order Optimization in StarCraft

- David Churchill, U. Alberta
- Michael Buro, U. Alberta

Build Order Optimization in StarCraft

- In the opening phase of RTS games, players usually don't interact with each other because:
  - their starting locations are spread over large maps
  - player visibility is limited to small regions
- The main sub-goals in this game phase are to:
  - establish a sufficient income flow by producing workers
  - quickly build structures that are prerequisites for other structures or can produce combat units
  - build a minimal force for defense or early attack
  - send out scouts to explore the terrain and search for enemy
- The order in which units and structures are produced is called a **build order**

Build Order Optimization in StarCraft

- The choice of initial build order often decides the game outcome
- Build order optimization:
  - *given*: a desired set of units and structures
  - *compute*: shortest possible sequence of game actions to achieve that goal
- Corresponds to a standard operations research problem: "constraint resource allocation with concurrent actions and makespace minimization"

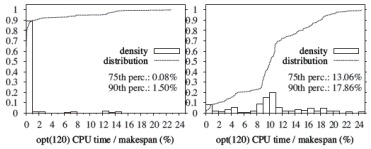
Build Order Optimization in StarCraft

Algorithm 2 Compare Build Order

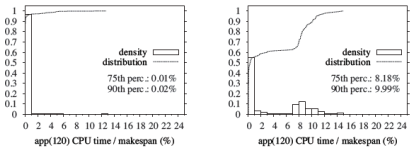
```
Require: BuildOrder B, TimeLimit t, Increment Time i
 1: procedure CompareBuildOrder(B,t,i)
 2:     S ← Initial StarCraft State
 3:     SearchPlan ← DFBB(S,GetGoal(B,0,∞),t)
 4:     if SearchPlan.timeElapsed ≤ t then
 5:         return MakeSpan(SearchPlan) / MakeSpan(B)
 6:     else
 7:         inc ← i
 8:         SearchPlan ← ∅
 9:         while inc ≤ MakeSpan(B) do
10:             IncPlan ← DFBB(S,GetGoal(B,inc-i,inc),t)
11:             if IncPlan.timeElapsed ≥ t then
12:                 return failure
13:             else
14:                 SearchPlan.append(IncPlan)
15:                 S ← S.execute(IncPlan)
16:                 inc ← inc + i
17:             end if
18:         end while
19:         return MakeSpan(SearchPlan) / MakeSpan(B)
20:     end if
21: end procedure
```


Build Order Optimization in StarCraft

A) CPU time statistics for search without macro actions:



B) CPU time statistics for search with macro actions:



Build Order Optimization in StarCraft

- Algorithm produces plans in real-time which are comparable to professional StarCraft players
- Integrated into StarCraft playing agent which competed in 2010 AIIDE StarCraft AI Competition

2. Character Agents

All the World's a Stage:
Learning Character Models from Film

- Grace Lin and Marilyn Walker,
  U. California Santa Cruz

Learning Character Models from Film

- *Long term goal:* automatically generate character dialogue that mimics the style of existing characters
- Internet movie script database (IMSDb)
  - 862 film scripts
  - 7,400 characters
  - 664,000 lines of dialogue
  - 9,599,00 words

Learning Character Models from Film

- *What does "mimic the style" mean?*

SCENE: LOBBY of Sports Club
ALVY: Uh ... you-you wanna lift?
ANNIE: *Turning and aiming her thumb over her shoulder*
Oh, why-uh ... y-y-you gotta car?
ALVY: No, um ... I was gonna take a cab.
ANNIE: *Laughing* Oh, no, I have a car.
ALVY: You have a car?
Annie smiles, hands folded in front of her
So ... *Clears his throat.* I don't understand why ... if you have a car, so then-then wh-why did you say "Do you have a car?"... like you wanted a lift?

Annie Hall (1997), Woody Allen, Director

Learning Character Models from Film

automatically extract these features from scripts using natural language processing techniques:

Set:Description
Basic: number of sentences, sentences per turn, number of verbs, number of verbs per sentence
LIWC Word Categories. Anger (hate, kill, pissed), Social processes (talk, us, friend), Friends (pal, buddy, coworker), Causation (because, know, ought), Discrepancy (should, would, could), Assents (yes, OK, mmhmm), Tentative (maybe, perhaps, guess), etc.
Dialogue Act: Accept, Bye, Clarify, Continuer, Emotion, Emphasis, Greet, No-Answer, Reject, Statement, Wh-Question, Yes-Answer, Yes-No-Question, Other
First Dialogue Act: Same as DA but only look at first sentence of each turn.
Pragmatic Markers: Word counts and ratios, plus word category counts: p-taboo, p-seq, p-opinion, p-aggregation, p-softeners, p-emphatics, p-ack, p-pauses, p-concession, p-concede, p-justify, p-contrast, p-conjunction, p-ingroup, p-near-swear, p-relative
Polarity: overall polarity, polarity of sentences, polarity for concessions
Merge Ratio: merging of subject and verb of two propositions
Tag Question Ratio: number of sentences with tag questions out of all sentences
Average Content Word Length: content words are noun, adjective, adverb, and verb; average words' length
Verb Strength: average sentiment values of verbs
Passive Sentence Ratio: number of passive sentences out of all sentences

Learning Character Models from Film

1. Automatically extract features from existing scripts using natural language processing
2. Use machine learning learn a "character model"
3. Use the character model to control parameters of an automatic dialogue generator

Learning Character Models from Film

- Character model for Annie:
  - many disfluencies, such as *um* and *uh*
  - says *yes* and *yea* a lot, especially at the beginning of her utterances
  - produces short sentences, but talks a lot
  - uses a lot of tag questions
  - does not use long words
  - uses *really, sort of* and *I think* a lot
  - etc.

Learning Character Models from Film

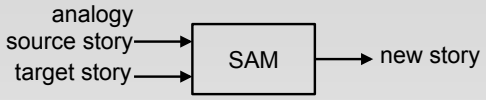
***Annie Hall*: Alvy**	***Indiana Jones*: Indy**
- I don't know. People say Cartmill is st-strange, alright? Err... on the other hand, I don't rush to judgment.	- I don't rush to judgment, but people say Cartmill is strange.
- Right, I am not sure, would you be? I will tell something you because you br-brought me cabbage.	- I will tell something you since you brought me cabbage.
- Oh I am not sure. Wolf wears a hard shell. On the other hand, he is ge-ge-gentle, isn't he?	- Wolf is gentle but he wears a hard shell.
- I see, I don't know. I respect Wolf, wouldn't you? He, however, isn't my close friend.	- Wolf isn't my close friend. But I respect him.
- Yeah, I don't know. Sparrow conveys excitement to my life, so I am fr-fr-friends with her.	- I am friends with Sparrow since she brings excitement to my life.

3. Story

The SAM Algorithm for
Analogy-Based Story Generation

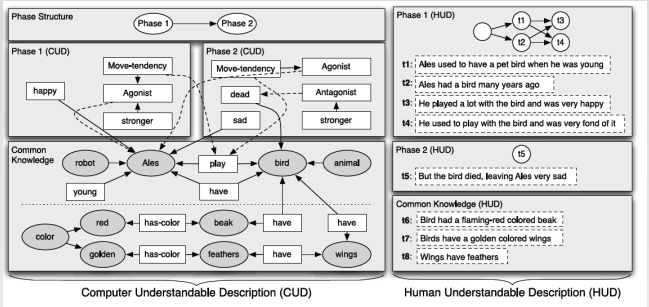
- Santiago Ontanon, Spanish Council for Scientific Research
- Jichen Zhu, U. of Central Florida

Analogy-Based Story Generation



- *Goal:* Automatically generate a new story from given analogy source and target stories
- *Challenges:*
  - how to represent stories in a computer
  - the analogy algorithm (uses previous work by Forbus & Gentner)

Analogy-Based Story Generation



Analogy-Based Story Generation

Analogy Source Story

Ales remembered the garage in which he had his first oil change, it was all red. His owners said he was rusty, and forced him to change his oil; he was a fool to accept. Ales felt very awkward afterwards, and decided that he would have to be really rusty before the next time he gets an oil change. He wondered why no one ever complained about oil changes.

Analogy-Based Story Generation

Target Story

One day, Ales was walking in an alley, when he saw a cat in front of him. Ales hesitated about what to do with the cat since he was late for work. Ales played with the cat.

Analogy-Based Story Generation

New Story

One day, Ales was walking outside, when he saw a cat in front of him. Ales hesitated about what to do with the cat since he was late for work. Ales played with the cat. Ales felt very awkward afterwards, and decided that Ales would have to be really rusty before the next time Ales played with the cat. Ales wondered why no one ever complained about that.

Analogy-Based Story Generation

- Conclusion
  - not ready for prime time ☺
  - authors working on improvements

4. Learning

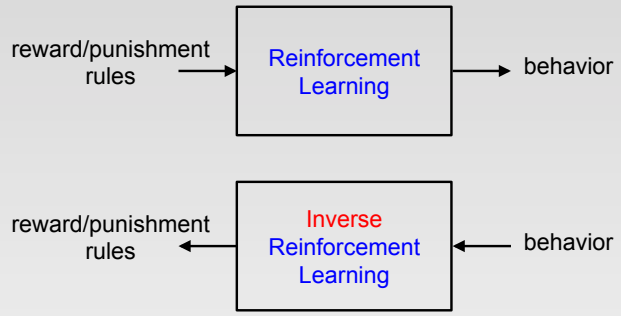
Learning Policies for
First Person Shooter Games
Using Inverse Reinforcement Learning

- Bulent Tastan and Gita Sukthankar, U. of Central Florida

Learning Policies for FPS Games

- A policy is a mapping from situations to actions
  - situations described by features
- *Goal:* To learn policies from examples of human play
  - "learning from demonstration"

Learning Policies for FPS Games



Learning Policies for FPS Games

- Given a set of (learned) reward/punishment rules, you can easily compute a policy by always choosing the action with the greatest reward
- Used to build a bot for Unreal Tournament based on human players' behaviors

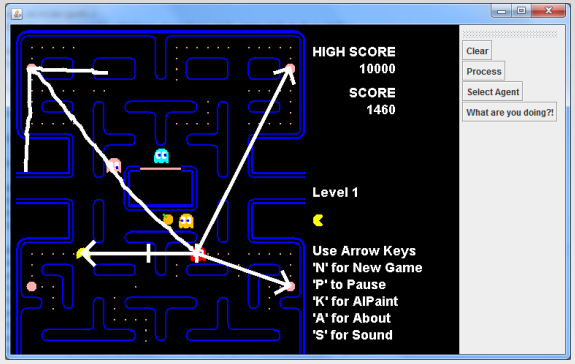
Learning Policies for FPS Games

- Experimental Evaluation
  - goal is "human-ness" of play, not optimal play (no fun in game to always be beat ☺)
  - trained system on about 45 minutes of human play
  - 13 experimental participants played two games
    - one against bot trained with inverse reinforcement learning
    - one against a hand-coded bot
  - participants told they were playing against one human and one bot and supposed to figure out which is which!
  - participants significantly chose the IRL bot as the human

5. Authoring

- AIPaint: A Sketch-Based Behavior Tree Authoring Tool
  - David Becroft, Adrian Mejia, Jesse Basset, Charles Rich and Candace Sidner, Worcester Polytechnic Institute

AIPaint



6. Recognizing Player Activity

Goal Recognition with Markov Logic Networks for Player-Adaptive Games

- Eun Ha, Jonathan Rowe, Bradford Mott and James Lester,
  North Carolina State U.

Goal Recognition w Markov Logic Networks

- *Goal Recognition:* inferring users' goals from sequences of observed actions
- *Application:* automatically adjust (adapt) play in Crystal Island serious game
- *Technique:* Markov logic networks

Goal Recognition w. Markov Logic Networks

- Crystal Island
  - educational game for 8th-grade microbiology
  - overall goal is to identify source of mysterious illness
  - open, exploratory environment
  - could be improved by a little guidance

Goal Recognition w. Markov Logic Networks

- Markov logic network (MLN)
  - combines first-order logic with probabilistic graphical models
  - instead of true/false values, logical constraints have *weights*
  - used to infer *hidden* predicates (goals) from *observed* predicates (actions)

Predicate		Description
Observed	*action(t, a)*	Player takes an action *a* at time *t*.
	loc(t, l)	Player is at a location *l* at time *t*.
	state(t, s)	The narrative state at time t is s.
Hidden	*goal(t, g)*	Player pursues a goal *g* at time *t*.

Goal Recognition w. Markov Logic Networks

- Evaluation
  - collected data from 137 middle school students playing the game for approx 1 hour each
  - data manually labeled and used to train MLN

Total Number of Observed Player Actions	77182
Total Number of Goals Achieved	893
Average Number of Player Actions per Goal	86.4

Running laboratory test on contaminated food	26.6%
Submitting a diagnosis	17.1%
Speaking with the camp's cook	15.2%
Speaking with the camp's bacteria expert	12.5%
Speaking with the camp's virus expert	11.2%
Speaking with a sick patient	11.0%
Speaking with the camp nurse	6.4%

  - trained MLN compared to other goal recognition algorithms (including unigram and bigram)

Goal Recognition w. Markov Logic Networks

- Conclusion
  - technique does a better job of goal recognition than any other existing system
  - next step is to see how knowing players' goals will help guide the game.

* Invited Talks

- The Evolution of RTS AI
  - Bob Fitch, Blizzard Entertainment
- Bringing Physical Characters to Life---Lessons and Challenges from Disney
  - Akhil Madhani, Walt Disney Imagineering Research and Development
- *Darkspore* AI Post Mortem
  - Dan Kline and Lauren McHugh, EA/Maxis

* Invited Talks [cont'd]

- Creating the Enemies of *Dead Space*
  - Louis Gascoigne, Visceral Games, an Electronic Arts Studio
- Social Games and the Role of Simulation in a Social World
  - Robert Zubek, Zynga

* Panel

- Player Modeling: Games, Data and Human Behavior
  - David Roberts, North Carolina State U.
  - Nick Yee, Palo Alto Research Center
  - Mike Carr, Novel Interactive
  - Carrie Heeter Michigan State U.

Questions? Comments?

- P.S. The other big yearly game AI confab is the AI Summit at GDC
  - March 5-6, 2012, San Francisco
  - organized by the AI Game Programmers Guild (http://gameai.com)
  - approx 85% industry, 15% academic