Games Research at U of A

Michael Buro, Ph.D.
Department of Computing Science
University of Alberta, Edmonton

May-6-2003

Outline

- The GAMES Group at U of A
  - Classic Games Research
  - Commercial Games Research

- ORTS: An Open RTS Game Environment
  - RTS Games
  - Issues: Client Hacks & Weak A.I.
  - Project Roadmap, Current State
  - Demo

GAMES Group at U of A
Game-playing, Analytical Methods, Minimax search, and Empirical Studies

- Classic Games
  - Early A.I. test domain
    - Heuristic Search
    - Evaluation Functions
    - Opponent Modeling ...
  - Making computers strong(er)
  - Goal: World-champion level

- Commercial Games
  - Addressing A.I. problems game developers face
    - Pathfinding
    - Learning
    - Believable behaviour
    - Low/high level A.I. ...
  - Goals: smart NPC or opponent behaviour - efficient algorithms

Classic Games Research

- Minimax Search Enhancements
- Evaluation Function Learning
- Imperfect Information
- Opponent Modeling
- Single Agent Search

- Chinook - Checkers World-champion
- Logistello - super-human Othello program
- IS Shogi - computer Shogi World-champion
- OptiBot - an “optimal” Poker program
Commercial Games Research

- Tackling A.I. issues games developers face
  - Efficient Pathfinding
    - What topology? incremental, real-time
    - How to find paths for formations?
  - Learning in Sports Games
    - Finding/Avoiding “Sweet Spots”
    - Increasing replay value
  - Scripting Languages for role playing games
  - A.I. for RTS Games

Ties to Computer Games Companies

- Currently Electronic Arts and Bioware
- Get source code access to current titles: E.g.
  - FIFA Soccer
  - Baldur’s Gate
  - Neverwinter Nights
- Run experiments with state-of-the-art game engines
- Interact with game developers
- Solve problems that matter
- Internships

Real-Time Strategy (RTS) Games

- Players build and command armies
- Real-time object motion usually on 2.5D battlefield
- Imperfect information (“Fog of War”)
- Resources
- Technology tree

+ Fancy graphics = Million sellers
  WarCraft, StarCraft, AoE, AoM, Homeworld ...
A Typical RTS Game
StarCraft (tm)

RTS Game Wish Lists

Player Perspective:
- Smarter unit level AI
- Better computer opponents/allies
- Multiple-view GUIs
- Hack-free game environment

CS Perspective:
- Better AI (low- and high-level)
- AI interfaces
- Man/Machine + Machine/Machine competitions

ORTS - An Open RTS Game Environment

- Test domain for real-time AI research
- Abstract RTS game
- Hack-free server-side simulation
  - Only server maintains entire game state
  - Local player views are sent to clients
  - Clients send actions back to server
- Portable: C++ & SDL
- Free software - GPL

Server-Side Simulation Issues

- Downstream data: \( \sim C \times \#\text{visible objects} \)
  (10 KB/sec 4x250 objects @ 5fps)
- Upstream data: \( \sim D \times \#\text{own objects} \)
  (\( \sim 1 \) KB/sec)

- Bottlenecks: CPU + network latency
- Need dedicated server
- Can the server be trusted?
Server-Side Simulation Benefits

- All unit commands are generated in clients
  - Command for each unit in every frame
  - No fixed unit behaviour! Micro actions are sent
- Users can roll their own client software
  - GUIs with multiple views, resolutions etc.
  - Low-level unit behavior (a la Quake’s AimBots)
- Client hacks pointless

Project Roadmap

- First:
  - Finish server code, optimize it
  - Implement platform independent GUI
  - Client software, AI plugins for low-level unit behavior
  - “Advanced” RTS competitions (humans+AI plugins)
- Then:
  - AI arms race commences
  - Machine RTS game competitions
- Ultimate Goal:
  - High-level AI replaces human general

Current State

- StarCraft-like terrain features almost complete
- Efficient object motion + collision test
- Not so efficient tile-based view computation
- Incremental / compressed data transmission
- Performance
  - worst case: all objects visible
  - on P3/1GHz: ~15 fps  1200 moving objects

Current Projects

- Summer Students:
  - optimize server/client code
  - implement GUI
- Thesis Topics:
  - Learning low-level behavior
  - Heuristic search, abstraction, and planning
  - Scripting in RTS games
Demo