Artificial Intelligence
Part II

CMPUT 250
Fall 2007
October 18
Nathan Sturtevant

AI in Games

- [If we were to ask developers what are] “the most common A.I. techniques applied to games, undoubtedly the top two answers would be A* and Finite State Machines. Nearly every game that exhibits any A.I. at all uses some form of an FSM to control character behavior, and A* to plan paths.”
  - Jeff Orkin, Monolith Productions

Lecture Overview

- More Finite State Machines
  - F.E.A.R.
  - Pacman
  - Pathfinding (A*)
    - Dragon Age
  - Case Study -- if time permits
    - Halo

Finite State Machines

- A **finite state machine** is a model of behavior composed of a finite number of states, transitions between states and actions
Algorithm

- A detailed set of actions to perform or accomplish some task
- Evaluate game algorithms according to:
  1. Does it meet our time constraints?
  2. Does it meet our memory constraints?
  3. Does it solve the task at hand?
  4. Does it do so in an acceptable/realistic manner?

Finite State Machines

- Can build complexity into FSM
  - More/complex actions
  - More/complex states
- What is a state?
  - The properties of the environment
  - The current plan being executed

Algorithm

- A detailed set of actions to perform or accomplish some task
- Evaluate game algorithms according to:
  1. Does it meet our time constraints?
  2. Does it meet our memory constraints?
  3. Does it solve the task at hand?
  4. Does it do so in an acceptable/realistic manner?

Finite State Machines

- Can build complexity into FSM
  - More/complex actions
  - More/complex states
- What is a state?
  - The properties of the environment
  - The current plan being executed

Algorithm

- A detailed set of actions to perform or accomplish some task
- Evaluate game algorithms according to:
  1. Does it meet our time constraints?
  2. Does it meet our memory constraints?
  3. Does it solve the task at hand?
  4. Does it do so in an acceptable/realistic manner?

Finite State Machines

- Can build complexity into FSM
  - More/complex actions
  - More/complex states
- What is a state?
  - The properties of the environment
  - The current plan being executed

If two units are near each other:
- Find cover
- Shoot from cover

What if we want our agents to collaborate?
- If two units are near each other:
  - Find cover
  - Shoot from cover
Environment State

- Weapons / Enemies
  - Have Weapons, Near Enemies
  - Have Weapons, No Enemies
  - No Weapons, Near Enemies
  - No Weapons, No Enemies
Pacman Ghosts

- Possible States
  - See pacman (N/S/E/W)
  - Pacman direction (N/S/E/W)

- Possible Actions
  - Move randomly
  - Move fixed pattern
  - Moved direction

AI in F.E.A.R.

- First Encounter Assault Recon
- Released October 2005
  - Gamespot's PC Shooter of the Year

- Three States and a Plan: The A.I. of F.E.A.R.
  - Jeff Orkin, GDC 2006

F.E.A.R.

F.E.A.R. FSM
F.E.A.R. FSM
- **Goto**
  - Physical movement to a new location
- **Animate**
  - In-place animation
- **Use Smart Object**
  - Special-case of animate

F.E.A.R. State Transitions
- How do you decide when you switch states?
- How do you decide *where* to go?
- F.E.A.R. uses a planning system

F.E.A.R. AI Goals
- Two possible goals:
  - Patrol
  - Kill Enemy
- Each goal implemented differently in different player types
  - Soldiers patrol in the open
  - Assassins hide and use melee attack

F.E.A.R. AI Actions
- If you see an enemy, shoot
- If you are shot at, dodge (shuffle or roll)
- Melee attack if you get close
- Potentially take cover
- If hit while in cover, shoot blindly
- If cover is lost, find new cover
- *Use dialogue to communicate to user*
“A gamer posting to an internet forum expressed that they he was impressed that the A.I. seem to actually understand each other’s verbal communication. ‘Not only do they give each other orders, but they actually DO what they’re told!’ Of course the reality is that it’s all smoke and mirrors, and really all decisions about what to say are made after the fact, once the squad behavior has decided what the A.I. are going to do.”

AI Psychology in F.E.A.R.

- People will perceive the AI as smarter if they know what it is doing
  - If only one unit remains, say “I need reinforcements”
  - Introduce conversations instead of someone talking to themselves
    - (I’m shot! v. What’s your status?)
  - If AI is stuck, say “I’ve got nowhere to go!”

Pathfinding
FSM for grid problem

- What FSM can we use for grid pathfinding?
  - Actions:
    - N, S, E, W, NE, NW, SE, SW
  - States: (Goal Location)
    - Above (+ left/right)
    - Below (+ left/right)
    - Right
    - Left

Pathfinding FSM

Pathfinding

- Pathfinding is a global problem
  - Need global knowledge of the world to make correct choices
  - Easy for our visual systems to see this global information
  - New approach
    - Search algorithm / Planning

Grid-Based Pathfinding

- Given a start and goal in a grid
  - Compute all 1-step moves
    - Label with cost
  - Compute 2-step moves
    - Label with cost
  - Continue until goal is reached
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
### Diagrams

#### Diagram 1:
```
<table>
<thead>
<tr>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Start</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
```

#### Diagram 2:
```
<table>
<thead>
<tr>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Start</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
```
Pathfinding

- Compute the cost to all locations
  - Time? Memory?
    - $O(t^2)$
  - Solves the problem
  - Will find the shortest path
- *Breadth-First Search*
What about…

Pathfinding

- In some cases we do much more work than the simpler algorithm
  - Avoid this by improving our algorithm
  - Consider the distance to the goal
    - Assuming no obstacles in the world
<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>6+1</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2+6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1+7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>Goal</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4+6</td>
<td>3+5</td>
<td>4+4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2+6</td>
<td>3+5</td>
<td>4+4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1+7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4+6</td>
<td>3+5</td>
<td>4+4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2+6</td>
<td>3+5</td>
<td>4+4</td>
<td>5+3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1+7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4+6</td>
<td>3+5</td>
<td>4+4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2+6</td>
<td>3+5</td>
<td>4+4</td>
<td>5+3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1+7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>7+1</td>
</tr>
<tr>
<td>6</td>
<td>4+6</td>
<td>3+5</td>
<td>4+4</td>
<td>6+2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2+6</td>
<td>3+5</td>
<td>4+4</td>
<td>5+3</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1+7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Pathfinding in Dragon Age™

- Pathfinding on a large map is too expensive
  - Build a smaller, more “abstract” map
  - Compute paths on smaller map
  - Use as a guide for paths on larger map

A*

- Standard game pathfinding algorithm is A*
  - Combines actual costs with cost estimates
  - In easy cases behaves the same as simple FSM
  - In complicated cases still finds optimal paths
- Many extensions by AI researchers
  - IDA*, SMA*, D*, HPA*, PRA* …
Pathfinding in Dragon Age™

- Memory
  - Abstraction takes about 10-20k memory
  - 1% or less than the rest of the map
  - Keep only abstract path in memory between planning steps

- Time
  - Abstract paths cheap to compute -- small
  - Refining cheap -- short paths
  - Pathfinding can be spread across multiple frames

- Solve task?
  - So far...games has yet to be released
Other AI techniques

- Planning
  - Search similar to A*
- Classical Games
  - High-performance 2-player search
- Learning
  - Reinforcement Learning
  - Decision Trees
  - Neural Networks

Planning

- Like A* for pathfinding, not on a grid
  - List of available actions
    - Conditions on which actions can be performed
  - (List of) goals to achieve
  - Search for “best cost” set of actions to accomplish goal
Game AI

- Most game AI isn’t about beating up the player
  - Challenge the player to just barely win
    - Easy for computers to have perfect aim
    - Easy for computers to cheat
  - Produce a fun (addicting) experience

Case Study: Halo

- GDC 2002 talk covering Halo AI
  - Jaime Griesemer & Chris Butcher
- How did they design the AI?
  - Avoid heavy scripting
  - Avoid masses of enemies

Case Study: Halo

- Building a good AI is a mix of design and programming
  - Designers worked on long-term interactions (~3 minutes)
  - Program/script the short-term behaviors (run from grenade, etc)
- Give the AI the same capabilities as player
Case Study

- **Predictability**
  - Want enemies to be predictable…
  - …give player the joy of beating them
  - Added “breaking point” change of behavior
    - When AI is almost dead, drastically change behavior

- **Unpredictability**
  - Random enemies too unpredictable
  - Try to make human random
  - AI becomes more unpredictable

Player Feedback on AI

- **Stronger enemies perceived as smarter**

<table>
<thead>
<tr>
<th>Weak Enemy Playtest</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Too hard</td>
<td>12%</td>
<td>Very Intelligent</td>
</tr>
<tr>
<td>About right</td>
<td>52%</td>
<td>Somewhat Intelligent</td>
</tr>
<tr>
<td>Too easy</td>
<td>36%</td>
<td>Not Intelligent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tough Enemy Playtest</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Too hard</td>
<td>7%</td>
<td>Very Intelligent</td>
</tr>
<tr>
<td>About right</td>
<td>92%</td>
<td>Somewhat Intelligent</td>
</tr>
<tr>
<td>Too easy</td>
<td>0%</td>
<td>Not Intelligent</td>
</tr>
</tbody>
</table>

Level Design

- **Design levels to show off AI**
  - Not much AI needed to fight in a long hallway

- **Make sure visual cues are obvious**

  “In Halo the Grunts run away when an Elite is killed. Initially nobody noticed so we had to keep adding clues to make it more obvious. By the time we shipped we had made it so not only does every single Grunt run away every single time an Elite is killed but they all have an outrageously exaggerated panic run where they wave their hands above their heads they scream in terror and half the time one of them will say ‘Leader Dead, Run Away!’ I would still estimate that less than a third of our users made the connection”

Design Decisions

- Can handle 20-25 units, 2-4 vehicles
- AI can’t track everyone around them
  - Only track 3-5 important players
- Use sound and animation to convey internal state of character
Technologies

- Build a model of the world
  - Emotional state of units
  - Complex perceptions of world
  - Implemented in a Finite State Machine
- Ray-casting for lines of view
  - 60% of AI code

Summary

- There isn’t true intelligence in most game AI
  - The illusion of intelligence exists
  - An illusion is good enough for most players
    - We don’t start conversations with our enemies
- If we can’t be intelligent, avoid the issue
  - Get other human opponents
    - Internet makes it easy to find opponents
    - MMORPG