CS/INFO 4154: Analytics-driven Game Design

Lecture 7: Artificial Intelligence
Alpha Prototype

- Thursday, October 1st
- Three playable levels
Artificial Intelligence!
The Dream

AI

Play games automatically

Procedural Content Generation

Design games automatically
What makes an AI good?
What are examples of good AI?
Today

- Scripts
- Planning
- Pathfinding
Today

- Scripts
- Planning
- Pathfinding
Scripts

1. IF <condition is true> THEN <perform action>
2. IF <condition is true> THEN <perform action>
3. IF <condition is true> THEN <perform action>
Example: Tic Tac Toe

1. If I can win, win
2. If opponent can win, prevent it
3. If center is available, take it
4. If corner is available, take it
Doom II
Final Fantasy VII
Final Fantasy VII

- If (Count == 0 OR Count == 2) Then
  - SelectedTarget = random opponent
  - Use Search Scope on SelectedTarget
  - Count = Count + 1

- If (Count == 1 OR Count == 3) Then
  - With probability 2/3:
    - If Self HP < (Self Max HP / 2) Then
      - Use Scorpion Tail on SelectedTarget
    - Else
      - Use Rifle on SelectedTarget
  - With 1/3 Chance:
    - Use Scorpion Tail on SelectedTarget
  - Count = Count + 1
“Cheating”
Advantages/Disadvantages

- Easy to implement
- Can express complex behaviors
- “Smart” behavior can be very complex
- Not so scalable
- No natural way to vary difficulty
Today

- Scripts
- Planning
- Pathfinding
Today

- Scripts
- Planning
- Pathfinding
Adversarial Search

Now

I move

Opponent moves

I move
Importance of search depth in Chess

- Novice: ~4
- Master: ~8
- Grandmaster: ~12
- Deep Blue: 6 – 40
Advantages/Disadvantages

Potentially *much* smarter
Natural way to vary difficulty
State-space explosion
Game must have certain properties
Unclear what to do if you can’t “see” the end
Heuristics: Chess

- Pawn: 1 point
- Knight & Bishop: 3 points
- Rook: 5 points
- Queen: 11 points
Heuristic Planning

Now

I move

Opponent moves
Heuristic Planning

Now

I move

Opponent moves
Heuristic Planning

Now

I move

Opponent moves
Heuristic Planning

Now

I move

Opponent moves
Minimax

MAX

MIN
Advantages/Disadvantages

+ Scalable
+ Can be rational without “seeing” the endgame
- Strength depends a lot on the heuristic
- Still only works for some games
Simultaneous Actions
A beats B beats C beats A

<table>
<thead>
<tr>
<th>Heimerdinger is Weak Against</th>
<th>Heimerdinger is Strong Against</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syndra</strong></td>
<td><strong>Kassadin</strong></td>
</tr>
<tr>
<td>![Syndra]</td>
<td>![Kassadin]</td>
</tr>
<tr>
<td>Mid</td>
<td>Mid</td>
</tr>
<tr>
<td>5,806</td>
<td>3,518</td>
</tr>
<tr>
<td>1,597</td>
<td>1,413</td>
</tr>
<tr>
<td>6 Comments</td>
<td>2 Comments</td>
</tr>
</tbody>
</table>

| **Ziggs**                   | **Katarina**                  |
| ![Ziggs]                    | ![Katarina]                   |
| General                     | Mid                           |
| 2,814                      | 3,233                         |
| 1,364                      | 1,263                         |
| 3 Comments                 | 2 Comments                    |

| **Lux**                     | **Diana**                     |
| ![Lux]                      | ![Diana]                      |
| Mid                        | Mid                           |
| 2,320                      | 2,573                         |
| 1,095                      | 1,294                         |
| 2 Comments                 | 4 Comments                    |

| **Xerath**                  | **Fizz**                      |
| ![Xerath]                   | ![Fizz]                       |
| Mid                        | Mid                           |
| 1,621                      | 1,925                         |
| 593                        | 1,158                         |
| 2 Comments                 | 0 Comments                    |

| **Vel’Koz**                 | **Akali**                     |
| ![Vel’Koz]                  | ![Akali]                      |
| General                     | Mid                           |
| 1,920                      | 1,608                         |
| 958                        | 868                           |
| 3 Comments                 | 2 Comments                    |

| **Malzahar**                | **LeBlanc**                   |
| ![Malzahar]                 | ![LeBlanc]                    |
| Mid                        | Mid                           |
| 1,264                      | 1,522                         |
| 499                        | 785                           |
| 1 Comments                 | 0 Comments                    |
A beats B beats C beats A
Opponent always plays rock...
Opponent plays…
Idea: Mixed Strategy

\[
\begin{array}{c|c|c}
2 & 3 & 1 \\
\hline
3 & 3 & 3 \\
\end{array}
\]
Idea: Mixed Strategy

\[
\begin{array}{ccc}
\frac{2}{3} & ? & \frac{2}{3} \\
\frac{1}{3} & ? & 0 \\
\end{array}
\]
Idea: Mixed Strategy

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>(\frac{2}{3})</th>
<th>(\frac{2}{3})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\frac{2}{3})</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\frac{1}{3})</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\frac{1}{3})</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluating a strategy

• Idea: compute the *expected reward* for a strategy
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Reward Matrix

\[
\begin{pmatrix}
0 & 1 & -1 \\
-1 & 0 & 1 \\
1 & -1 & 0
\end{pmatrix}
\]
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Expected Reward

\[ \text{Expected Reward} = \text{probability of event}_1 \times \text{reward of event}_1 \\
+ \text{probability of event}_2 \times \text{reward of event}_2 \\
+ \text{probability of event}_3 \times \text{reward of event}_3 \\
\ldots \]
Expected Reward

\[ = p_{me}(\text{stone}) \times p_{you}(\text{stone}) \times R(\text{stone}) \]

\[ + p_{me}(\text{tile}) \times p_{you}(\text{tile}) \times R(\text{tile}) \]

\[ \ldots \]
\[
\begin{bmatrix}
\frac{2}{3} & \frac{1}{3} & 0 \\
-1 & 0 & 1 \\
1 & -1 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 1 & -1 \\
1 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]
\[ \pi_{you}^T \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{3} \pi_{me} \]
Expected Reward

$$\text{reward} = \pi_{you}^{T} R \pi_{me}$$
Minimax!

$$\max_{\pi_{me}} \left( \min_{\pi_{you}} \pi_{you}^T R \pi_{me} \right)$$
Linear Programming

feasible region
Calculating Reward

\[
\max_{\nu, \pi} \nu \text{ such that } \sum_{i} \pi_i = 1, \quad \pi \geq 0, \quad \nu \leq R \pi
\]
Optimal Strategy

\[
\begin{align*}
\text{Rock} & : \frac{1}{3} \\
\text{Paper} & : \frac{1}{3} \\
\text{Scissors} & : \frac{1}{3}
\end{align*}
\]
Rock Paper Scissors

Pathfinding
Make a grid!
Pathfinding: Depth-First
Pathfinding: Breadth-First
Pathfinding: Breadth-First
Breadth-First is Slow!

Idea: use heuristics
A* Algorithm

- Score $f = g + h$
  - $g$: distance on best path
  - $h$: naïve distance to goal

Manhattan distance = $30 + 20 = 50$
Pathfinding: A* Algorithm

A

B
Pathfinding: A* Algorithm
Pathfinding: A* Algorithm
Pathfinding: A* Algorithm
Pathfinding: A* Algorithm

In case of tie, use most recently added
Pathfinding: A* Algorithm

<table>
<thead>
<tr>
<th></th>
<th>f: 94</th>
<th>f: 80</th>
<th>f: 74</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>g: 24</td>
<td>h: 70</td>
<td>g: 20</td>
<td>g: 24</td>
<td>h: 70</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 80</td>
<td>f: 60</td>
<td>f: 54</td>
<td></td>
</tr>
<tr>
<td>g: 20</td>
<td>h: 60</td>
<td>g: 10</td>
<td>g: 14</td>
<td>h: 60</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 94</td>
<td>f: 74</td>
<td>f: 60</td>
<td></td>
</tr>
<tr>
<td>g: 24</td>
<td>h: 70</td>
<td>g: 14</td>
<td>g: 10</td>
<td>h: 70</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 80</td>
<td>f: 60</td>
<td>f: 54</td>
<td></td>
</tr>
<tr>
<td>g: 20</td>
<td>h: 60</td>
<td>g: 10</td>
<td>g: 14</td>
<td>h: 60</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 94</td>
<td>f: 74</td>
<td>f: 60</td>
<td></td>
</tr>
<tr>
<td>g: 24</td>
<td>h: 70</td>
<td>g: 14</td>
<td>g: 10</td>
<td>h: 70</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 80</td>
<td>f: 60</td>
<td>f: 54</td>
<td></td>
</tr>
<tr>
<td>g: 20</td>
<td>h: 60</td>
<td>g: 10</td>
<td>g: 14</td>
<td>h: 60</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 94</td>
<td>f: 74</td>
<td>f: 60</td>
<td></td>
</tr>
<tr>
<td>g: 24</td>
<td>h: 70</td>
<td>g: 14</td>
<td>g: 10</td>
<td>h: 70</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f: 80</td>
<td>f: 60</td>
<td>f: 54</td>
<td></td>
</tr>
<tr>
<td>g: 20</td>
<td>h: 60</td>
<td>g: 10</td>
<td>g: 14</td>
<td>h: 60</td>
</tr>
<tr>
<td>f: 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram showing the pathfinding algorithm with nodes A and B.
A* Mario

FPS: 24
Attempt: 1 of 1
AStarAgent
Selected Actions:

RIGHT
SPEED
Reinforcement learning
Civilization II

Wins 78% of games!
Group Activity Choice

1. Think about AI and write some scripts
2. Discuss how an AI might solve your game
3. Just work on your games