Artificial Intelligence

CMPUT 299
Fall 2005
November 1, 3

Lecture 1 Overview

- Artificial Intelligence
  - Definition
    - Related concepts
      - Algorithm
      - Time/Memory Cost
  - Finite State Machines

Artificial Intelligence (AI)

- What is intelligence?
- What is artificial intelligence?

Intelligence (*wikipedia*)

- **Intelligence** is usually said to involve mental capabilities such as the ability to reason, plan, solve problems, think abstractly, comprehend ideas and language, and learn.

AI (*wikipedia*)

- Intelligence exhibited by an artificial entity.

Artificial Intelligence

- Any computer action that isn't understood by the user
- The "process" by which objects are controlled in a game environment
- The "process" by which agents make rational actions in an environment

Game AI

- Control for all but first-person entities
  - *Objects/areas*
    - Magical chests (ScriptEase)
  - *Enemies*
    - Computer-controlled teams (real-time strategy games)
    - Computer bots (first-person shooter)
    - Passive enemy units (arcade games)
  - *Allies*
    - NWN Henchman
  - *Your character*

Game AI drives Animation

- AI dictates the behavior of all non-passive objects in the world
  - Animation is determined by game AI
  - Sounds and music might be changed by the AI

Why is AI hard/important?

- Computers cannot easily deal with abstract ideas like we do
  - *Sting's blade glows blue when orcs are near.*
  - *Mars will not be this near to earth again until 2018.*
- We must define concrete rules (an *algorithm*) for the computer to follow

What is an algorithm?

- A detailed set of actions to perform or accomplish some task
- Examples:
  - Make a peanut butter and jelly sandwich
  - Draw a picture of a dragon

Describing Algorithms

- Actions
  - Move forward 1 step
  - Turn 90 degrees
- State
  - Location (coordinates)
  - Health
- Dynamics (State transitions)
  - How does state change when an action is applied

PB&J

- What are the actions?
- What are the states?
- What are the transitions?

Drawing

- What are the actions?
- What are the states?
- What are the transitions?

Evaluating Algorithms

- How can we evaluate an algorithm?
  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?

Evaluating Memory Usage

- How much is used during computations?
  - No less than the solution size
- How much is stored between computations?
- How much memory does our state take?
- How does this scale?
  - Bigger maps, more units

Evaluating Speed

- What is the cost of each operation we perform?
- How many of each operation will we perform?
- How does this scale?

AI Complexity in Games

- We must balance all four needs
- Most resources are dedicated to graphics

Complexity Numbers

- Suppose we have a 3Ghz machine
  - 3 billion cycles/second
- Suppose we run at 30 fps
  - 100 million cycles/frame
- Suppose we have 100 units
  - 1 million cycles/unit/frame
- Suppose world is as complex as all units
  - 500k cycles/unit/frame
- Suppose each unit has 1,000 polygons * 500 ops
  - Time's up!

Simple Task Complexity

- How many paths are there through a grid?
  - Start at one corner
  - Travel to the other corner
  - How many possible paths are there?
- Actions, states, transitions?

2x2 Grid

	Goal
Start	

3x3 Grid

		Goal
Start		

4x4 Grid

			Goal
Start			

How many paths?

Grid Size	# of paths
2x2	2
3x3	6
4x4	20
5x5	70
6x6	252
7x7	924
8x8	3432

AI Approaches

- Ad-hoc
- Finite State Machines
- Search
  - Methodic exploration of environment
- Learning

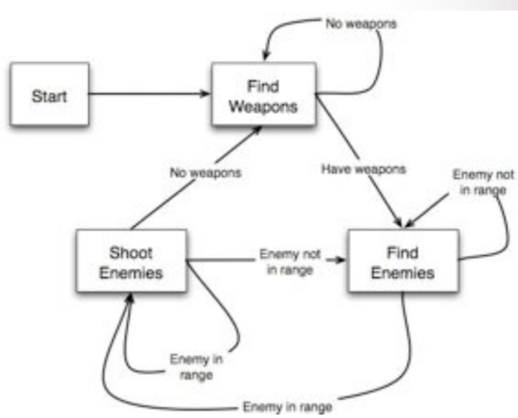
Finite State Machines (FSM)

- What is a state?
  - The context of the environment that is relevant for making decisions
- A FSM can test states and apply actions

FSM Example

- Simple first person shooter (FPS) state:
  - Do I have a weapon?
  - Am I near an enemy?
- FPS Actions
  - Find weapons
  - Find enemies
  - Shoot enemies

Finite State Machine



Trogdor!

Pacman

Pacman ghosts

- How are pacman ghosts controlled?

FSM Pros and Cons

- Pros:
  - How much time does it take to run a FSM?
  - How much memory does a FSM use?
  - Very simple to implement
- Cons
  - May be difficult to reproduce complex behavior
  - May be too predictable

Lecture 1 End

- Any questions?

Artificial Intelligence - Part II

- Review
  - Algorithms
  - FSM
- Today
  - Pathfinding
  - Other Technologies
  - Case Study: Halo
  - Traditional Games

Algorithm

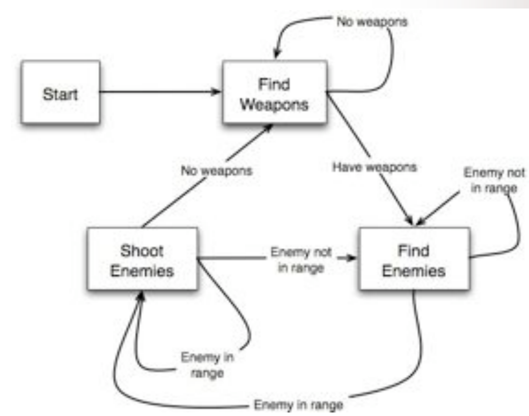
- A detailed set of actions to perform or accomplish some task
- Evaluate game algorithms according to:

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 be built in two ways
  - Boxes are states or actions

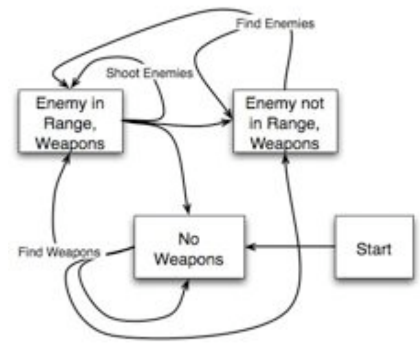
FSM 1



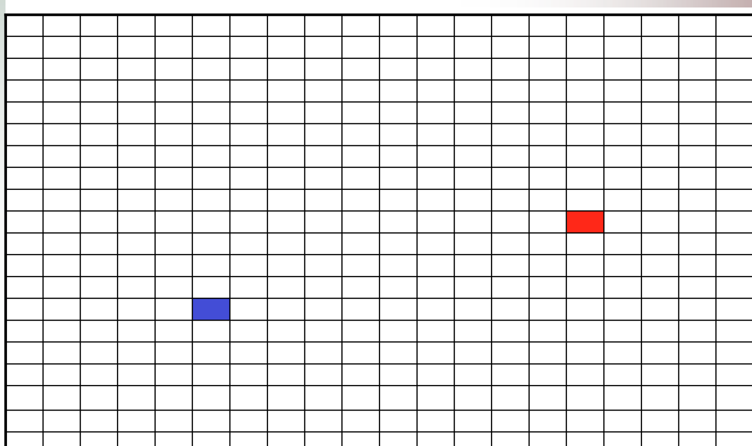
Possible states

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

FSM 2



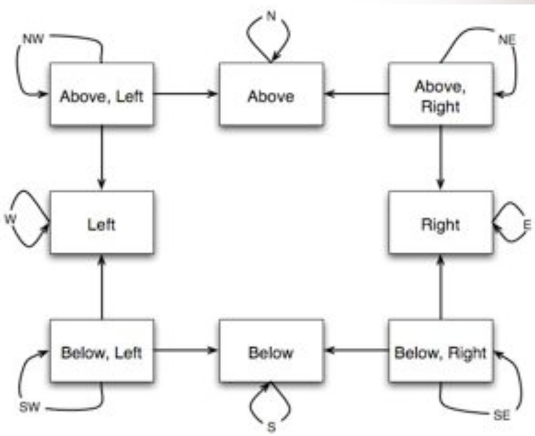
Pathfinding



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*

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

				Goal
Start				

				Goal
1				
Start	1			

				Goal
2				
1	2			
Start	1	2		

				Goal
3				
2	3			
1	2	3		
Start	1	2	3	

4				Goal
3	4			
2	3	4		
1	2	3	4	
Start	1	2	3	4

4	5			Goal
3	4	5		
2	3	4	5	
1	2	3	4	5
Start	1	2	3	4

4	5	6		Goal
3	4	5	6	
2	3	4	5	6
1	2	3	4	5
Start	1	2	3	4

4	5	6	7	Goal
3	4	5	6	7
2	3	4	5	6
1	2	3	4	5
Start	1	2	3	4

Pathfinding

- Compute the cost to all locations
  - Time? Memory?
    - $\pi \cdot r^2$
  - Solves the problem
  - Will find the shortest path
- *Breadth-First Search*

				Goal
Start				

				Goal
1				
Start	1			

				Goal
2				
1	2			
Start	1	2		

				Goal
3				
2	3			
1	2	3		
Start	1	2	3	

4				Goal
3				
2	3	4		
1	2	3		
Start	1	2	3	4

4	5			Goal
3				
2	3	4		
1	2	3		5
Start	1	2	3	4

CMPUT 299 - Fall 2005
Artificial Intelligence

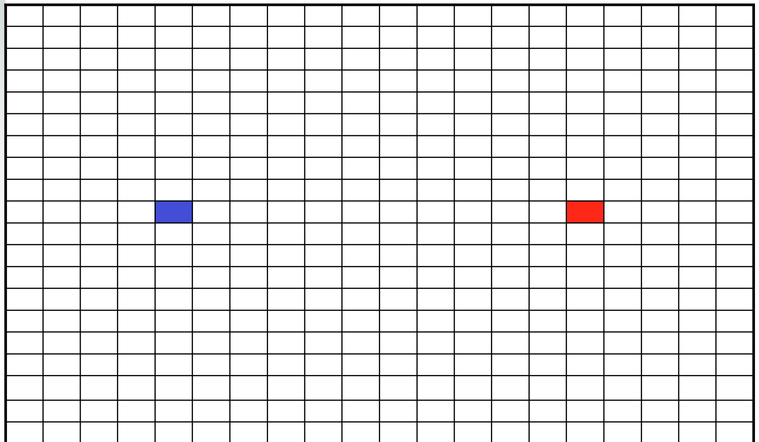
4	5	6		Goal
3				
2	3	4		6
1	2	3		5
Start	1	2	3	4

CMPUT 299 - Fall 2005
Artificial Intelligence

4	5	6	7	Goal
3				7
2	3	4		6
1	2	3		5
Start	1	2	3	4

CMPUT 299 - Fall 2005
Artificial Intelligence

What about…



What about…

			10	9	10														
		10	9	8	9	10													
	10	9	8	7	8	9	10												
10	9	8	7	6	7	8	9	10											
9	8	7	6	5	6	7	8	9	10										
8	7	6	5	4	5	6	7	8	9	10									
7	6	5	4	3	4	5	6	7	8	9	10								
6	5	4	3	2	3	4	5	6	7	8	9	10							
5	4	3	2	1	2	3	4	5	6	7	8	9	10						
4	3	2	1		1	2	3	4	5	6	7	8	9	10					
5	4	3	2	1	2	3	4	5	6	7	8	9	10						
6	5	4	3	2	3	4	5	6	7	8	9	10							
7	6	5	4	3	4	5	6	7	8	9	10								
8	7	6	5	4	5	6	7	8	9	10									
9	8	7	6	5	6	7	8	9	10										
10	9	8	7	6	7	8	9	10											
	10	9	8	7	8	9	10												
		10	9	8	9	10													
			10	9	10														
				10															

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

4	3	2	1	Goal
5	4	3	2	1
6	5	4	3	2
7	6	5	4	3
Start	7	6	5	4

4	3	2	1	Goal
5	4	3	2	1
6	5	4	3	2
7	6	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	2	1
6	5	4	3	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	2	1
6	**3**+*5*	4	3	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	2	1
6	**3**+*5*	**4**+*4*	3	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	2	1
6	**3**+*5*	**4**+*4*	**5**+*3*	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	**6**+*2*	1
6	**3**+*5*	**4**+*4*	**5**+*3*	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	**6**+*2*	**7**+*1*
6	**3**+*5*	**4**+*4*	**5**+*3*	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

4	3	2	1	Goal
5	4	3	**6**+*2*	**7**+*1*
6	**3**+*5*	**4**+*4*	**5**+*3*	2
7	**2**+*6*	5	4	3
Start	**1**+*7*	6	5	4

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* …

Other AI techniques

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

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

Weak Enemy Playtest

Too hard	12%	Very Intelligent	**8%**
About right	52%	Somewhat Intelligent	72%
Too easy	**36%**	Not Intelligent	20%

Tough Enemy Playtest

Too hard	7%	Very Intelligent	**43%**
About right	92%	Somewhat Intelligent	57%
Too easy	**0%**	Not Intelligent	0%

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

Bonus: AI work at UofA

- UofA has one of the largest groups of researchers working on AI
  - Shifting from traditional games to commercial games

Checkers

- Chinook
  - Best Checkers program in the world
    - 1994 played human champion
  - Better than any human
  - Work ongoing to solve the game
  - Jonathan Schaeffer and many others

Othello (Reversi)

- Logistello
  - Uses learning to evaluate positions in the game
  - Beat human world champion in 1997
  - Michael Buro

Poker

- Poker Academy
  - Considered best Poker program on the market
  - Uses game theory, adaptation and other technologies

Other work

- Go, Hex, Hearts, Spades
- EA Soccer
- RTS (ORTS), Pathfinding

- http://games.cs.ualberta.ca/