Games AI
Lecture 6

Topics covered this week:

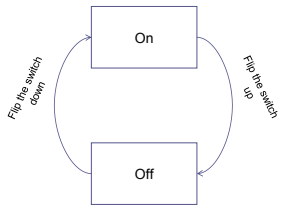
- Finite State Machines
- Scripting
- Path-finding
- Flocking / steering
- AI in games

© Derek Stobbs 2010

Finite State Machines

- What is a finite state machine (FSM)?
  - A model that defines a finite number of states
  - Any input to the model either produces an output, or a change in state
- Example FSM – a light bulb
  - States: On; Off
  - Input: Flipping the switch toggles state On > Off; Off > On

Light bulb Example



Programming the light bulb using a simplified state machine

```
class lightBulb
{
    private int state = 0;
    private Image bulbOff;
    private Image bulbOn;
    public lightBulb()
    {
        // load the images …
    }
    public void changeState(int newState)
    {
        state = newState;
    }
    public void paint(Graphics g) …
}
```

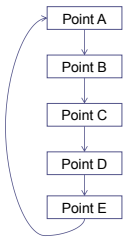
```
public void paint(Graphics g)
{
    switch(state)
    {
        case 0: // off
            g.drawImage(bulbOff, 0, 0, this);
            break;
        case 1: // on
            g.drawImage(bulbOn, 0, 0, this);
            break;
        default:
            System.out.println("Unsupported state!");
            break;
    }
}
```

- Note: This is Java code, but easily reproducible in C++

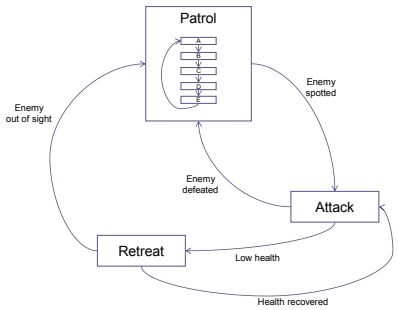
Scripting

- What is a script?
  - A set of instructions to be carried out in order … like a computer program
- Why use a script?
  - Bring NPCs to life
  - Waypoints: A > B > C > D > E > A > B > C …
  - Can be combined with FSM for extra realism

Scripting
Path following example

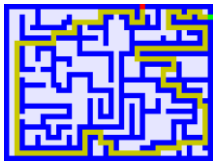


Scripting
FSM combination



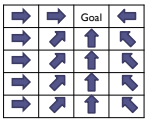
Path-finding
A-star

- Find a path between any two points on a map
- Dynamically calculated online search
  - Select a start position
  - Select a goal position
  - Check each position using cost and heuristics to evaluate which path to take



Path-finding
Fast march

- Find a path to the goal from any other point on the map
- Pre-calculated offline search
- Minimal overheads
  - Check current position
  - Move in direction indicated by Fast march data



Flocking / Steering Behaviours

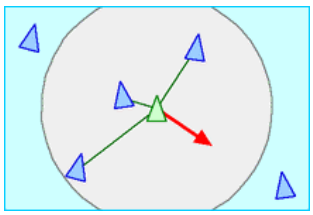
- Emulation of the flocking behaviour of birds, or similarly behaving animals (fish, insects...)
- Emergent behaviours as there are no rules governing the whole flock, just simple rules for each animal
- Craig Reynolds' Boids (1986) – first computer simulation

Steering behaviours

- Each boid follows three simple rules to make sure they stay at about the same velocity as their neighbouring boids, while staying close, but not too close to each other
  - Separation
  - Alignment
  - Cohesion

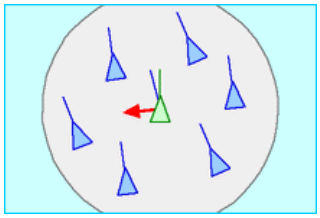
Separation

- Steer to avoid crowding local flockmates



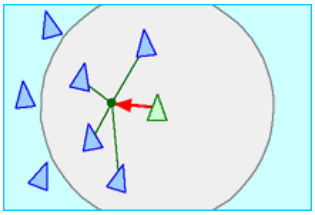
Alignment

- Steer towards the average heading of local flockmates



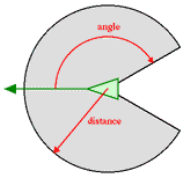
Cohesion

- Steer to move toward the average position of local flockmates



Neighbouring flockmates

- Each boid should be aware of the entire scene, but only respond to flockmates within the immediate area



Taking things further

- Algorithm can be expanded by other game-play elements
  - Collision avoidance
  - Path-finding
  - Fleeing from an enemy
  - Chasing an enemy
  - Boids can die

Genetic Algorithms

- Use some basic principles of genetics to pass behaviours from generation to generation
- The best of each generation are used to create the next generation, so moving towards a better solution
- Mutation is used to add variance so the final solution can be better than the best solution from the first generation
- Game Example: Life simulation, passing character traits on from parents to children

Neural Networks

- Computational model that tries to simulate the biological neural networks of a brain
- Artificial neurons that each have simple processes are interconnected to provide a more complex process / behaviour
- The system is adaptive to changing inputs, meaning a change of its outputs as it "learns"
- Game Example: Realistic path navigating as enemy "learns" to avoid obstacles

Simulating Emotions

- Autonomous agents given different ratings on emotion scales (e.g. Happy->Sad)
- Actions dependent on current emotions
- Actions of others cause change of emotions
  - He stole my food, I don't like him as much
  - I have plenty of food, that makes me happy

AI in modern games

- Many AI algorithms can be, and frequently are, used in games
- As more resources become available, more complex and realistic AI can be performed in Real-Time
- Can combine different AI methods to give unpredictable results

Combining methods

- Colony of humans arriving on a new world
  - FSM useful for performing different actions
  - Scripted behaviour to perform certain actions / tasks
  - Use neural networks to learn about their surroundings, including path-finding
  - Pass on traits from parents to children using genetic algorithms
  - Form groups based on their likes and dislikes
  - Groups use flocking behaviours when navigating around the world