CSE 466 Flock Project

Preliminary design outline-- 11/14/03

- Goals for the Flock:
  – Sing the same song for a little while
  – Songs start, then spread, then die out

1

Flock Process Flow:

a. Initialize stuff; select x = random(0-15)
b. Radio off; Sing birdsong[x]; Radio on
c. Listen for Random(min1--max1) sec.
d. SendMessage "I sang song x“
e. Listen for Random(min2--max2) sec.
f. Decide which song to sing next
g. go to b.

2

Flock Details: Sing

- Turn the radio off– we can't handle 416 usec interrupts while generating sound
- Sing the song using Timer1 for PWM and Timer2 for tempo and ADSR control
- Turn the radio on and do any housekeeping required

Flock Details: Listen

- Arriving packets need to be time-stamped
- Packets from Node 0 must be specially treated– they may contain global parameters
- Arriving packets must be strength-stamped for RSSI value– special radio stack required

Flock Details: Decide

- Need algorithm for what song to sing next
- Similar to Cellular Automata, like Conway's Game of Life

- Goals:
  - Sing the same song for a little while
  - Songs start, then spread, then die out

What are Cellular Automata?

- Computer simulations which emulate the laws of nature
- Discrete time/space logical universes
- Complexity from simple rule set: reductionist approach
- Deterministic local physical model
- Rough estimation of nature: no precision
- Does not reflect 'closed sphere' life: can achieve same end results given rules and initial conditions

History

- Original experiment created to see if simple rule system could create “universal computer”
- Universal Computer (Turing): a machine capable of emulating any kind of information processing through simple rule system
- late 1960’s: John Conway invents “Game of Life”

7

Game of Life

- Simplest possible universe capable of computation
- Basic design: rectangular grid of “living” (on) and “dead” (off) cells
- Complex patterns result from simple structures
- In each generation, cells are governed by three simple rules
- Which patterns lead to stability? To chaos?

8

Simulation Goals

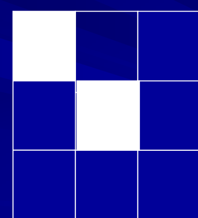
- Avoid extremes: patterns that grow too quickly (unlimited) or patterns that die quickly
- Desirous behavior:
  - No initial patterns where unlimited growth is obvious through simple proof
  - Should discover initial patterns for which this occurs
  - Simple initial patterns should grow and change before ending by:
    - fading away completely
    - stabilizing the configuration
    - oscillating between 2 or more stable configurations
  - Behavior of population should be relatively unpredictable

Conway's Rules

- Death: if the number of surrounding cells is less than 2 or greater than 3, the current cell dies
- Survival: if the number of living cells is exactly 2, or if the number of living cells is 3 (including the current cell), maintain status quo
- Birth: if the current cell is dead, but has three living cells surrounding it, it will come to life

To work out the future colour of any square…

We need to look at the eight squares around it….



To fully specify all possible combinations the rule would need to show 256 cases!

11

The rules for "Life"

- If a square is black *("on")* then it will be **black** at the next step if **2 or 3** of its neighbouring squares are black
- A white *("off")* square will become **black** only if it has **exactly 3 black neighbouring squares**
- Otherwise a square will be white the next step (*overcrowded or lonely*)

12

- So, we can have birth

- Or death…

- A nice implementation is at: http://www.math.com/students/wonders/life/life.html

Types of behaviour in the *Game of Life*…

1. Still life objects – unchanging (*e.g. four-block*)
2. Simple repeating patterns (oscillations)
3. Part of the system can leave the rest and travel (movement - *gliders*)
4. The system can **die out** completely
5. The system grows randomly before stabilising to predictable behaviour
6. The system grows forever *(quite rare and difficult to find)*

Chaos…

- All behaviour in the Game of Life is chaotic – it depends extremely sensitively on the **starting state** and is completely altered if the system **changes** a little (e.g. *just like the weather)*

15

Flock Details: Decide

- Goals:
  - Sing the same song for a little while
  - Songs start, then spread, then die out
- Algorithm?
  - Determine nearest songs
  - If our song = any of nearest n, then
  - repeat song
  - If all same, switch to different song
  - If none same, switch to different song

16

Flock Details: Decide

- Algorithm?
  - Determine nearest songs
  - If our song = any of nearest n, then
  - repeat song
  - If all same, switch to different song
  - If none same, switch to different song
- How do we evaluate this?
- How can we predict it's effectiveness?

17

Flock Process Flow:

a. Initialize stuff; select x = random(0-15)
b. Radio off; Sing birdsong[x]; Radio on
c. Listen for Random(min1--max1) sec.
d. SendMessage "I sang song x“
e. Listen for Random(min2--max2) sec.
f. Decide which song to sing next
g. go to b.

- How do we evaluate overall effectiveness?

18

Homework Assignment:

- Invent a method of calculating whether 50 birdies will ever sing the same song at approximately the same time.
- Make three suggestions for improvement to any aspect of the flow or decision algorithm to improve chances of success
- Due in class on Friday, Nov. 21.
- NOT HAND-WRITTEN– use a computer!

19