Modeling & Simulating

Flocking Behaviour of Agents

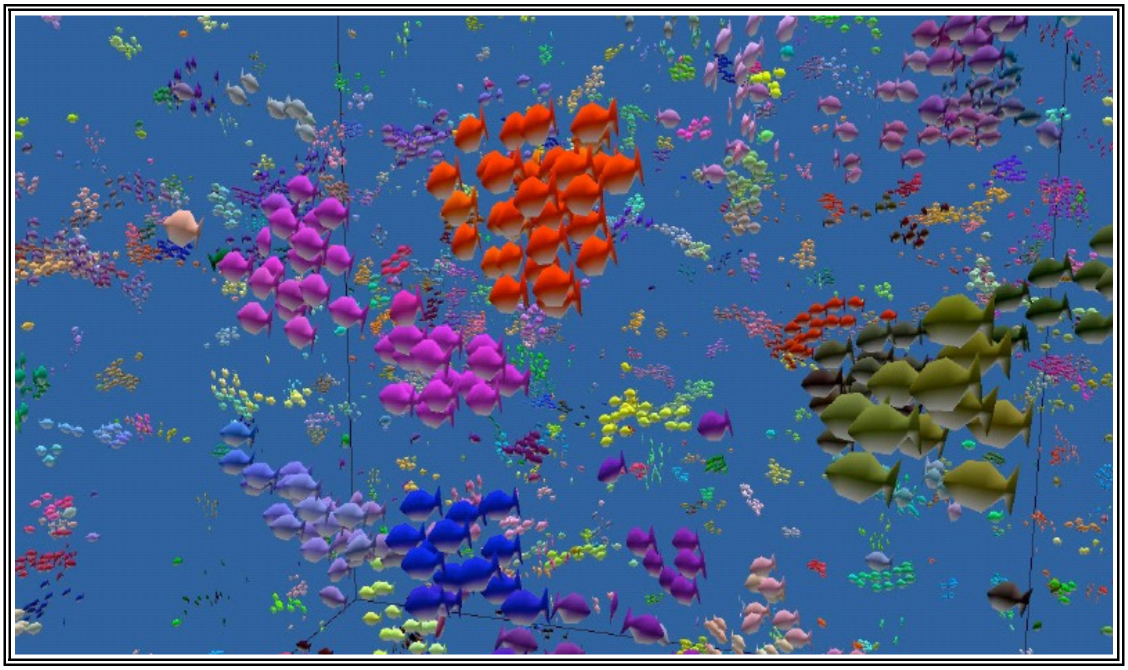
Project Report

Fall 2006

COMP 522

Sanghamitra Dutt
sanghamitra.dutt@mail.mcgill.ca

Table of Contents



1. Introduction

Flock is a group of objects that exhibit the general class of aligned, non-colliding, aggregate motion
Agent is a simulated bird like object i.e. it exhibits this type of behaviour. It can be a fish, cow, etc.

1.1 Abstract

This project deals with the world of agents where each agent has its own intelligence to control its movement. The agents facilitate flocking and obstacle avoidance. The agents in the system avoid the other agents of the flock as they avoid obstacles in the environment. Each agent is capable of deciding its own path to take depending on where the rest of the flock is and how bunched up its neighbours are.

1.2 Motivation

Problem: How do we simulate the motions of a flock in computer animation?

Possible Solutions:

1. Manually scripting each individual object path. But this would be very tedious and error prone. Also, it would be almost impossible to do without collisions and since people are generally used to seeing the natural movements of animal groups, such as flocks of birds, they can quickly tell when something is not behaving naturally.
2. In order to deal with the above problem this paper discusses another approach. It assumes the flock is the result of interactions between the behaviors of different agents. So, we must simulate the behavior of an individual agent (the flock associated behavior). This requires also simulating perceptual mechanisms and some aspects of aerodynamic flight. If this is done correctly, then to create a flock we just create multiple instances of the agents.

This project is based on the paper "Flocks, Herds, and Schools: A Distributed Behavioral Model", by Craig W. Reynolds. There are a few things that have been done differently but the fundamental ideas are the ones that have been described in this paper.

2. Overview

2.1 AnyLogic By XJ Technologies

AnyLogic is a professional simulation tool for complex discrete, continuous and hybrid systems. Powerful and flexible, it is used to model, simulate, visualize and analyze a diverse range of real-world problems. AnyLogic is innovative software based on such advanced technologies as UML, Java, hybrid systems theory, and best numerical methods. AnyLogic supports virtually all existing approaches to discrete event and continuous modeling, such as process flow diagrams, system dynamics, agent-based modeling, state charts, equation systems, etc.

2.2 Theory

Flocks are a generalization of particle systems, which are used to simulate dynamic fuzzy objects, such as fire, clouds, etc. In a flock, the particles (which are dot-like, eg., maybe small spheres) are replaced by full geometric entities with an orientation. The behaviour of agent particles is more complex, plus they interact with each other whereas simple particles don't.

Flocking Algorithms are based on behavioural patterns in nature found in bees, flies, frogs, birds, fish, ants, and any other flocking animal. This individual based model also used in case of emergency evacuations of human crowds. Many consulting companies have used this kind of a model to simulate crowds that help to plan safe public spaces.

Each agent has an internal state and a set of behaviours. It is easiest to encapsulate these into an object, in the sense of object oriented programming. Then each new agent is an instance.

The behaviours were expressed in rules:

1 ***Separation***: Steer to avoid local flockmates or environmental obstacles
2 ***Alignment***: Steer towards the average heading of local flockmates
3 ***Cohesion***: Steer to move towards the average position of flockmates

When an agent follows cohesion behaviour with its neighbors, then it will probably avoid collisions. So static collision avoidance tends to establish minimum distances between agents.

Cohesion means moving to the center of the nearby neighbors, i.e., it is a localized model, not global. If a agent is close to the center of the flock this will have little effect (since the agent density will be uniform), but if it is on the edges then it will have a greater effect. Since each agent is looking primarily at its neighbors, the flock can split to go around an obstacle. That is, if a large part of the flock splits off, most of the agents in the two flocks are unperturbed since their surroundings remain homogeneous. This is actually a better model of herds or schools of fish, who have a limited vision of the entire school or herd, since it assumes that the agents can only detect other agents a limited distance away.

Psuedo-Code

The agents program has the following structure:

```
initialize_positions()

LOOP
    draw_agents()
```

```
        move_all_agents_to_new_positions()
    END LOOP
```

2.3 Simulated Perception

In this model, the agents do not take into account all of the other agents but only their neighbours. This corresponds to a limited perception of the outside world. This is actually more accurate than assuming total perception. For herds and schools of fish, especially in murky waters, there is a limited perceptual range. Even for birds, some of the birds will be occluded by others. An earlier model used a central force for the flock centering. This resulted in the entire flock trying to converge on one spot. These simulations showed that the real world flocking behavior actually depended upon a limited perception.

The sensitivity of an agent to its neighbors is governed by two parameters: it is only sensitive to neighbours within a sphere of a certain radius (centered on the agent itself), and the degree of sensitivity is an inverse exponential of distance, so it is dependent upon the value of the exponent. An improved model would increase the sensitivity in the forward direction, and by an amount proportional to the agents speed. In the current model, agents in the front are overly distracted by agents behind them.

The magnitude of attraction or repulsion was found to be best weighted by an inverse square relationship (similar to gravity). This gave a more realistic behavior than a linear relationship, i.e., the agents are much more influenced by close neighbors. Fish sense their neighbors both visually, which tends to be an inverse square relationship, and by pressure waves, which is an inverse cube of the distance, so their behavior would be a combination of the two.

2.4 Avoiding Environmental Obstacles

Agents in an empty environment quickly settle down to a steady state, i.e. all at a close distance and with matched velocities. Introducing environmental obstacles makes the motion much more complex.

In this model, an agent is only affected by objects directly in front of it. Once it detects an object, it finds the silhouette edge closest to its intersection point. The actual obstacles encountered in an animation might be very complex geometric entities, eg., a tree, bridge, etc., so the agent's angle is changed to change the direction it is heading towards and the speed is slowed down.

The obstacles do not have to be static, they might be other moving entities, eg. a herd of elephants. They could also be predators, which would necessitate the addition of a higher priority avoidance algorithm.

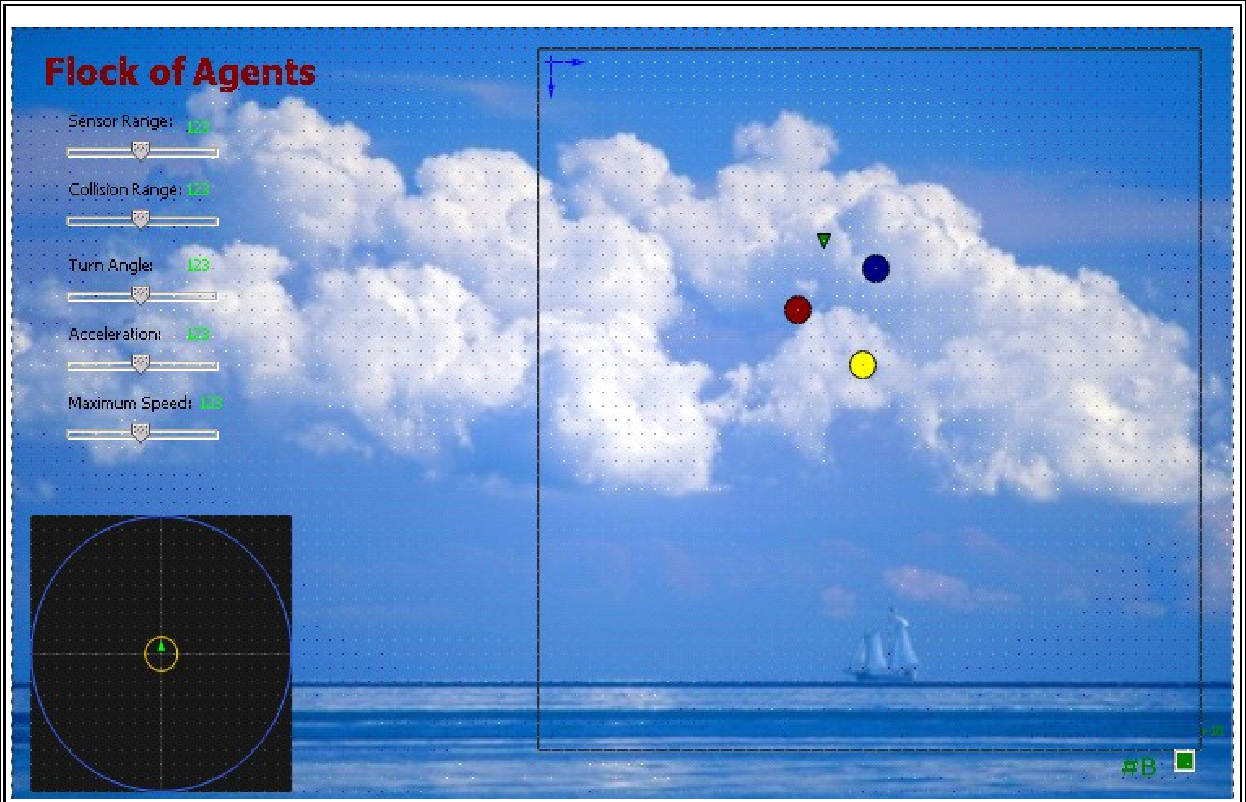
3. Simulation And Results

The Agent class has attributes such as width, height and current position in the frame given by x and y at time dt. This class also has attributes that have operations or functions that help the agent to follow the Separation, Alignment and Cohesion rules.

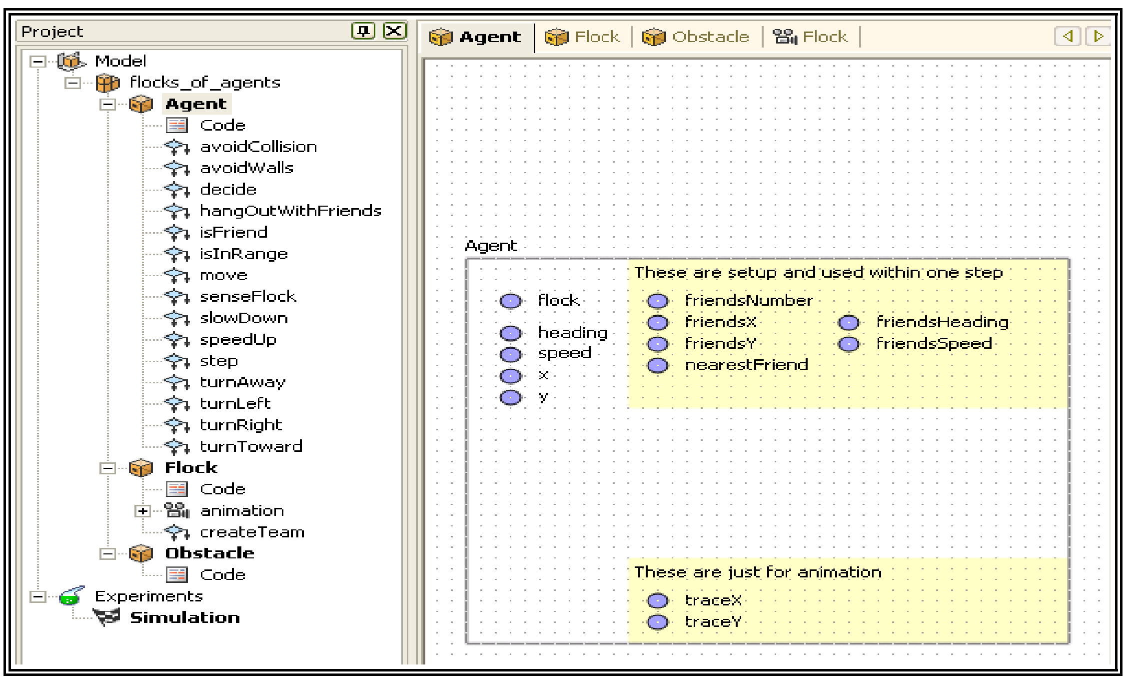
The Flock class creates a team of agents that is it creates instances of agent objects. Each object has its own set of attributes and operations.

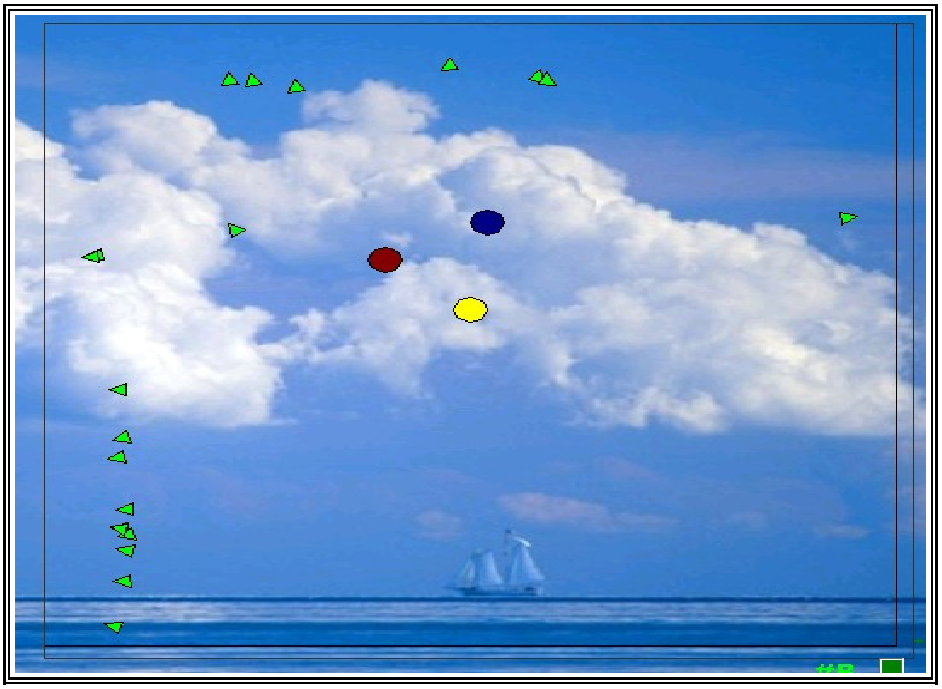
The model has a step timer that iterates in a step by step manner with time. At every step the timer iterates through each agent instance present and updates the attributes and calls all the functions that help the agent to follow the three rules i.e. alignment, separation and cohesion. This project implementation uses a time slicing model.

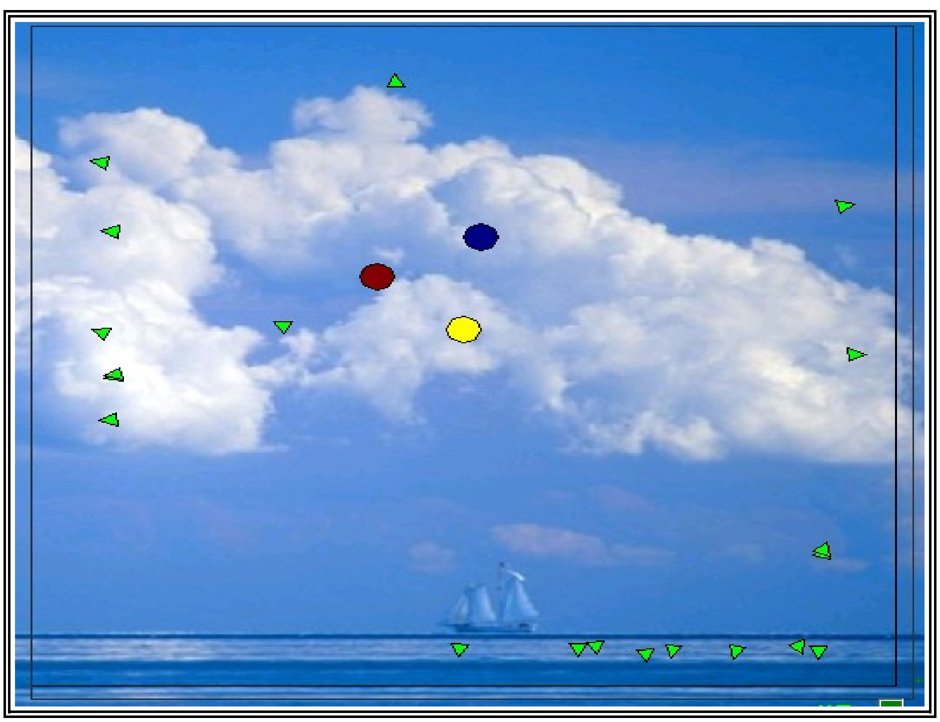
Screenshots:



Animation







4. Future Work

One aspect of agent behavior not considered in this paper is drafting, i.e., one agent following another agent to lessen wind resistance. In real life, birds draft behind each other, just as bicyclists do in a race. Birds that are in front, or on the outside, eventually move to the rear, or the inside, of the flock so that the wind-breaking is shared equally. This leads to an oval shaped flock (or bicycle pack).

A simple addition to the project would be to add groups to the agents such that individual groups will tend to interact with each other, but tend to avoid other groups. This would allow for a world of multiple types of animals to create a more complete virtual environment. We would be able to emulate a number of other animals including fish, different species of birds, squirrels, prey, and predators.

5. References

[1] Flocking (behavior)

http://en.wikipedia.org/wiki/Flocking_%28behavior%29

[2] AnyLogic
http://www.xjtek.com/

[3] Craig W. Reynolds
http://www.red3d.com/cwr/boids/