Goal implement the Boid algorithm on a 2d plane including obstacle avoidance.

What is a boid?

The rules

Emergent behavior

Swarm robotics

Algorithms for obstacle avoidance(obstacle as forces/edge detection)

Edge detection is harder

It makes more sense than a force

When you look a wall you don’t feel pushed away from it you just turn to get out side the edge

Our ray tracing circle strategy (typhoon algorithm)

Concentric circles

Steps in the algorithm

Note try and find a source

Efficiency concerns

Boids \* Boids

Boids \* shapes \* avg distances

Circle estimation may be faster

Drawing time

Some solution ideas

Multithreading in the dead frame

To be possible need to maintain independence in steps ie frame 0 only relies on info from frame -1

Vector math memory

Want to make a few new vectors as possible to save memory

Simple outline of class/packages

Basic outline

Boid

Relies on vector

The object that calculates the boid algorithm

Vector

A class the acts as a vector to do calulations

Ray

relies on vector

interacts with shapes to implement the ray tracing

Shape

relies on vector drawable ray

the obstacles for the boids

could be circle rectangle polygon

Graphics

relies on vector

relies on drawable

a screen jframe that can draw subclasses of drawable

Drawable

Can discrip an object to the graphcis module

relies on vector

Xmlreadin

Used to inialize shape on the screen

relies on vector

relies on shape