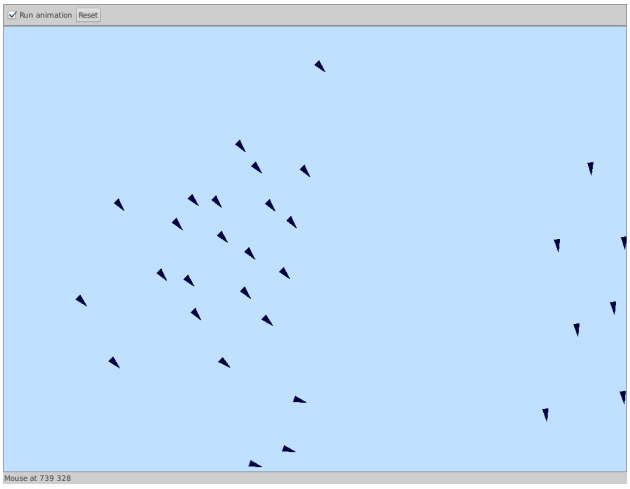
Boids!

Homework #1, CSCI 322, Winter 2016



Demo boids: I have placed an animated boids demo in the class repo called `boids00.rkt`. Currently it uses no explicit threading, tha animation loop uses busy waiting when there is no animation, and it uses `sleep/yield` in order to collaborate with the GUI.

Project: Create two new versions of this program:

1. A program called `boids01.rkt` that uses a single thread for the animation loop: calculate forces, move the boids, refresh the view, sleep, repeat.
   - Clicking the checkbox should suspend and resume this thread.
   - There should be no busy waiting when the animation is suspended.
   - Use `sleep` instead of `sleep/yield`.
   - Also, kill (not suspend) the animation thread when the window is closed. You do this by augmenting the `on-close` method of the top level `frame%`. See my `onclose.rkt` example in the repo. (You may notice that my version does not quit properly.)
2. A program called `boids02.rkt` that uses a new thread for each new boid.
   - Each thread will take care of calculating the forces and updating the position of its own boid.
   - Every time the mouse is clicked, a new boid and a new thread are created. The new threads should be contained in the boid objects themselves (a new field in the object).
   - There should be one more thread that takes care of refreshing the view (theres no point in each boid refreshing the view). Note that the view refreshing does not have to happen at the same frequency as the boid updating. For example, given enough cycles, we can update the boid positions more frequently than we refresh the view, to get a more accurate simulation.
   - Clicking the checkbox suspends and resumes all of these threads.
   - Again, no busy waiting and no sleep/yield.
   - Also, kill all threads when the window is closed.

Due date: Monday, January 25, at midnight. Remember to upload both programs. Each should be stand-alone and run without the other. 50% of this assignment's grade points for each stage.