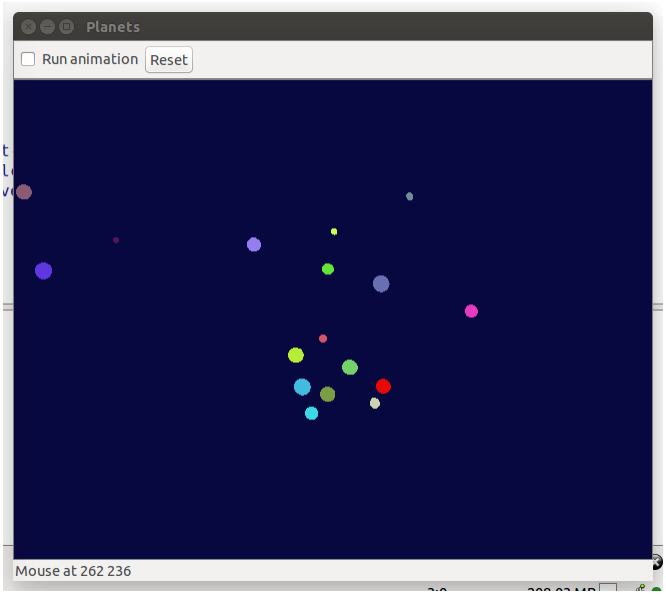
Planets!

Homework #1, CSCI 322, Winter 2015



Description: I have placed an animated planets demo in Racket in the class repo called `planets00.rkt`. Currently it uses no explicit threading, the animation loop uses busy waiting when there is no animation, and it uses `sleep/yield` in order to collaborate with the GUI. Create two new versions of this program:

1. A program called `planets01.rkt` that uses a single thread for the animation loop: calculate forces, move, 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 homework repo.

2. A program called `planets02.rkt` uses a new thread for each new planet. Each thread will take care of calculating the forces and updating the position of its own planet. Every time the mouse is clicked, a new planet and a new thread are created. The threads can be kept in a separate list, or each planet object can contain its own thread.

   There should be *one more thread* that takes care of refreshing the view (there's no point in each planet refreshing the view). Note that the view refreshing does not have to happen at the same frequency as the planet updating. For example, given enough cycles, we can update the planet 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: Friday, January 23, at midnight. Remember to upload both programs. Each should be standalone and run without the other. 50% of this assignment's grade points for each stage.