CSC104 Project 2, Fall 2013
Due: Friday November 29th, 11:59 pm

You will complete two simulations, `fractal.rkt` and `fx.rkt`, described below. Your task is to download fx.rkt and fractal.rkt from the course website, under November 29th (right-click on them). Each of these files has comments indicating things you need to fix. The comments begin with three exclamation marks:

```
; !!! <some important work-needing instruction goes here>
```

Your job is to try to fix these, one-by-one, until you have a working simulation. After you fix one thing, run the program to verify that you've made progress.

NB: Start early and leave yourself time to **ask questions** when you're stuck. Also, did I mention that it's a really good idea to fix one thing at a time, and verify that it is actually fixed, before proceeding.

fx.rkt explained

Each glowing dot in the rectangle of dots that makes up an image shines red, green, and blue light into our eyes, the mixture creating the colour we perceive. Typically the intensity of the three colours is not uniform.

This assignment explores what happens when we make one or more of the three colours "average" — just take the sum of all three colour intensities and divide by three (making sure you end up with a whole number, not a fraction). In order to get strong intuition about what's happening, you will make the simulation reversible: typing the ”r” key once will make all red intensities the average of the three colours, but typing ”r” again will restore them to their original values. Similarly with ”g” and ”b”, and you will be able to combine the effects of two, or all three keys.

In order to make this all work correctly, you will have to enter some new `(check-expect ...)` expressions, and some new definitions. If you are shakey on some of these, review the videos on decomposing images or defining functions. And, of course, be sure to ask questions.

fractal.rkt explained

In tutorial #8 you saw code for a fractal I thought of that i called bulls-eye, but which might more aptly be called **eyes**. To make the fractal more interactive, you will make it respond to the ”left” and ”right” arrow keys by making its small base circles smaller or bigger. It will also respond to the ”up” and ”down” arrow keys by making the depth, or complexity, of the fractal greater or smaller.

To make all this work you will make some more `(check-expect ...)` expressions and fix some definitions. Once you're done, you will be able to explore the fractal, although it will get rather large for the screen real-estate (and a little slow) if the depth gets too large. There are solutions for those issues, but they are beyond the scope of this assignment. Another possible enhancement is to make the circles random colours.

You may want to look at some of the fractal code we've had in tutorials, as well as reviewing **structs** in the struct video. And ask questions.

What to hand in

You will submit the following files to MarkUs:

- `fx.rkt`
- `fractal.rkt`

You may work in groups of no more than 3 in preparing your project. To set up a trio or pair, one group member should log on to MarkUs and invite the other one or two. You should submit your files early and often. The first time you create a file with meaningful content, submit it. You may re-submit the same file as many times as you wish, and only the last submission is stored. A good habit is to re-submit your files each time you improve them.