# MC39 Homework 2: The Chaos Game

Max Hailperin

Due: February 20, 1996

This homework, which involves a graphical educational demonstration program called "the chaos game," is intended to give you experience in the following areas:

1. reading C++ code, including in particular class-related features
2. reading the appendices at the back of the textbook to find necessary information
3. extending an existing class to support new needs
4. using a class given only its external interface, not its implementation
5. testing components of a program separately ("unit testing") to gain confidence in their correctness and cope with concurrent development of modules by different programmers.

You should familiarize yourself with the files in the directory `~max/MC39/hw2`, which are the starting point for your work. Printouts of these files are attached.

I will be producing one or more implementations of the `DrawingWindow` class; in the meantime you should work on the other parts of the program as follows:

1. The program uses the procedure `makeVectorPolar` as a "constructor" in the informal sense that it constructs `Vector`s, though it is not an actual C++ constructor in the `Vector` class. This procedure still needs to be written, however. Do so.
2. The `chaosGame.C` file includes several places where operations are performed on `Vector`s that, although they make perfect sense, are not provided in the very minimal definition of the `Vector` class that Wang provides. Therefore you need to add some additional member functions and/or operators to `Vector`:
   (a) Identify what members need to be added, and add *declarations* of them to the `Vector` class's *declaration* (i.e., in the `Vector.h` file). Include any comments needed to clarify what the procedures are supposed to accomplish.
   (b) Implement these extensions in your `Vector.C` file.

(c) Now you need to test your extensions to the `Vector` class, and also the `makeVectorPolar` procedure you wrote above. Write a simple "test jig" program that allows the user to enter and operate on vectors and display the results. Use this to demonstrate that the code you have written thus far works correctly.

3. At this point the chaos game should work, but you can't really test it until I get done implementing `DrawingWindow`. But, you shouldn't wait for me. Instead, test the rest of the program independently of my work using a "stub" class as follows:

   (a) Write a file `StubDrawingWindow.C` that `#include`s `DrawingWindow.h` and implements the `DrawingWindow` class, but rather than actually managing graphics windows it should just use normal character output to describe what's happening. For example constructing a `DrawingWindow` should print out "window constructed" and drawing a point should print out the numerical coordinates of the point.

   (b) Use this stub class, together with your `Vector` class and the code I supplied to demonstrate that the system would work correctly once supplied with a real `DrawingWindow` implementation.

Turn in all your code, including the "jig" and "stub" code, which won't be part of the final product, but rather is used only in testing. Turn in your test results as well, providing evidence that your modules and the main `chaosGame.C` module work correctly.

I will release one or more real implementations of `DrawingWindow` next week. You should be able to just re-link your program with one of these modules rather than the stub your wrote in order to have a working system. Since you tested your code carefully, and I'll test my code, it should just work. But try it out and see! (There is nothing to hand in from this "integration testing," which will happen after the homework is due.)