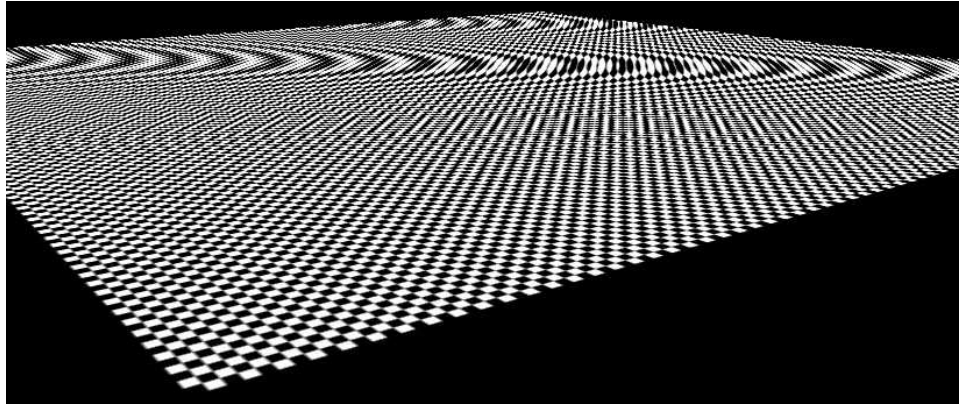


CSCI-1200 Computer Science II — Spring 2006

Homework 2 — Moiré Patterns (a.k.a. Fun with Strings)

A moiré pattern is a visible distortion that can result from a variety of interference conditions. The term comes from the French “moirer” (to water) and is used to describe a rippled, water-like look, which is often a desired artistic effect. The effect can be seen when two geometrically regular patterns (such as two sets of parallel lines or two halftone screens) are superimposed, especially at an acute angle. *Definitions from <http://webster.com> and <http://answers.com>.* Unintentional, distracting moiré patterns can be seen in computer games and other graphical applications when intricate textures (such as a checkerboard below) are displayed in perspective on a computer screen. Various signal processing techniques can be used to reduce the appearance of Moiré patterns.



In this homework you will work with command line arguments, file input and output, and the C++ string class to create *ascii art* with simple moiré patterns! Please read the entire handout before beginning to program the assignment.

File I/O

You will read string patterns from a input text file. Each line of the file will have first a string (of 1 or more non-whitespace characters) followed by a non-negative integer, n , which indicates the size of the finished moiré polygon. Here is a sample input file, `in_patterns.txt`:

```
abcde 9
__hi!__ 21
```

Command Line Arguments

Your program will expect 3 command line arguments. The first is the name of the input file. The second is the name of the output file where the program should output the finished moiré imagery. The third argument will be a string (`square`, `right_triangle`, or `isosceles_triangle`) specifying which type of polygon should be created. Here are examples of valid command lines for your program:

```
moire.exe in_patterns.txt out_square.txt square
moire.exe in_patterns.txt out_right_triangle.txt right_triangle
moire.exe in_patterns.txt out_isosceles_triangle.txt isosceles_triangle
```

Implement very simple error checking to ensure that 3 arguments are provided and that the input and output file streams are successfully opened. Your program should exit gracefully with a useful error message sent to `std::err` if there is a problem with the arguments.

