

# CS4499

## Project 1 - turtle graphics

### Due Sept 11, 5:00 pm

## Overview

For this assignment you will be implementing an interpreter for a very simple turtle graphics language. See [http://en.wikipedia.org/wiki/Turtle\\_graphics](http://en.wikipedia.org/wiki/Turtle_graphics) for information on turtle graphics. Our language will have the following commands:

1. `up` (pen up)
2. `down` (pen down)
3. `forward dist`
4. `left degrees`
5. `right degrees`
6. `loop num`
7. `color r g b`
8. `origin` (place turtle at canvas center pointing up)

See the files `example1.txt` and `example2.txt` for example usage. The default color is black and default position/orientation is (0, 0) and facing up. Your program will read commands and render the graphics.

The purposes of this assignment are threefold:

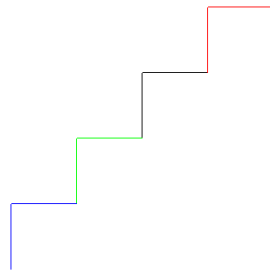
1. Get familiar with basic OpenGL commands
2. Get familiar with JavaScript
3. Review and solidify trigonometry and vector algebra concepts

## Instructions

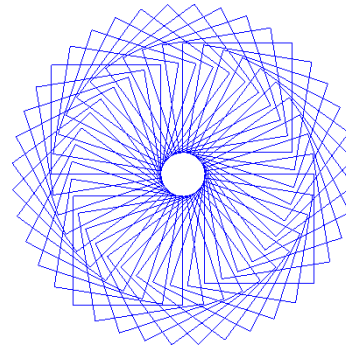
1. Load `turtl-soln-obf.html` (an obfuscated version of the solution) into a web browser and get familiar with it, including loading and executing `example1.txt` and `example2.txt`.
2. Run the skeleton code by loading `turtle.html` into a web browser.
3. Look for all TODO comments in `turtle.js` and follow the instructions.
4. Your final product should produce renderings exactly equal to what is shown in figure 1.
5. Write your own turtle graphics program commands in a file called `design.txt` and render it with your program. You should make this design interesting and creative. Your 10% creativity points will come from this. You are not permitted to expand the language – you can use only commands listed in the overview section above.
6. Take a screenshot of your `design.txt` program and save it as `design.png`.
7. Submit your work by typing `./submit` at the command-line. A report will be written that you can consult to make sure it submitted correctly.

## Scoring

1. 80% - Correctly render `example1.txt`, `example2.txt` and other test files
2. 10% - Create an interesting and creative `design.txt`
3. 10% - Coding quality and style given by report from check-code and visual inspection



(a)



(b)

Figure 1: (a) Output from `./turtle example1.txt`. (b) Output from `./turtle example2.txt`.

## Notes

1. You may use the standard C++ library (e.g. `vector`). You *may not* use any other third-party libraries.
2. Run `./check-code *.cpp *.h` frequently as you program. Fixing all errors at the end is a real drag.
3. You can submit your work as frequently as you like – only the most recent submission will be retained. Suggestion: submit first thing to get familiar with how it works and submit occasionally during development. This way there won't be any surprises when you're up against the deadline.