**Assignment 1: DrawSVG**

**Assignment 1 Instructions**

Project repository: [**https://github.com/cmu462/DrawSVG**](https://github.com/cmu462/DrawSVG)

Assignment 1 is due Sept 25th at 10:00pm. Late days may be used on this assignment (see Course Info).

This assignment is to be completed individually. You may discuss the material in general terms with your peers, but the code you submit must be entirely your own.

**NOTE**: Tasks 5-8 depend on concepts that will be discussed in Week 4, so it is not expected that you understand those requirements at the time this assignment is released.

Remember that making your solution public now or in the future is a violation of the academic integrity policy. We are using Github to distribute code, but do not work in a public fork!

**Grading**

Your code must run on the GHC 5xxxx cluster machines as we will grade on those machines. Do not wait until the submission deadline to test your code on the cluster machines. Keep in mind that there is no perfect way to run on arbitrary platforms. If you experience trouble building on your computer, while the staff may be able to help, but the GHC 5xxx machines will always work and we recommend you work on them if any troubles arise.

Each task will be graded on the basis of correctness. You are not expected to completely reproduce the reference solution pixel-by-pixel as slight differences in implementation strategy or even the order of floating point arithmetic will causes differences, but your solution should not be very far off. (While we provide a diff view in the application to help you debug, we will mostly be using it as a reference when we grade your code.) If you have any questions about whether your implementation is "sufficiently correct", just ask.

The assignment consists of a total of 100 pts. The point breakdown is as follows:

* Task 1 (Hardware Renderer): **5**
* Task 2 (Drawing Lines): **5**
* Task 3 (Drawing Triangles): **15**
* Task 4 (Supersampling): **20**
* Task 5 (Transforms): **10**
* Task 6 (Scaled Images): **15**
* Task 7 (Trilinear Filtering): **20**
* Task 8 (Alpha Compositing): **5**
* Task 9 (Draw Something!): **5**

The project description includes some additional challenges marked as extra credit. The point value next to these problems indicates a typical number of points that would be earned for successful completion, though we may award more or fewer points for these problems based on your particular level of effort. In general, we encourage you to push the envelope and explore extensions that make your program more powerful or more efficient, and we may even award additional points to you for doing so. If you do anything extra, please explain your work in the writeup! However, no matter how much extra work you do, your overall score on this assignment is capped at 110 points.

**Writeup**

Additionally, you will submit a short document explaining how you believe your implementation performs on each of the tasks. The project directory contains a template in writeup.md. For each task, if you believe that you have correctly implemented the requested functionality, simply leave the text "Fully implemented." If you were not able to achieve the requested functionality, write a short explanation of what your submission does and does not do correctly, and (optionally) briefly explain your strategy and how you got stuck. If your explanation indicates significant understanding of the problem, we may award partial credit even though the code does not work.

**Failure to submit this writeup will incur a 10 pt penalty on the assignment.**

**Code Environment**

This codebase should compile on Linux, macOS, and Windows on a typical environment. The build instructions given on the project page will walk you through installing dependencies and building the code. If you have difficulties running the code on your local machine, the GHC 5xxx cluster machines have all the packages required to build the project.

**Handin Instructions**

As with the previous assignments, we will be submitting on Autolab. You should create a tar archive of your entire srcsubdirectory along with the writeup (writeup.txt) and Task 9 submission (task9.svg).

$ pwd

> (...)/DrawSVG

$ tar cvzf handin.tgz src/ writeup.txt task9.svg

> a src

> a src/cmake

> a src/CMakeLists.txt

(...)

> a writeup.txt

> a task9.svg

The Autolab system will run a simple script that attempts to compile your code and checks for the extra files. If the output indicates the code does not compile or that something is missing, fix it or risk losing points!

Copyright 2017 Carnegie Mellon University