

CPS

Repo for CPP to PS software

- See Note for explanation of the two projects in this repo.
- See Other Documents for explanation of other documentation in this repo.
- See below those for documentation of CPS Library.

Note:

There are two parts of this project - the CpsCompiler and the CPS language. The CpsCompiler interprets a new language, cps, into C++, and then interprets that into postscript. This was developed by Jake.

The CPS language is a series of classes following the interpreter design pattern to create a virtual CPS language within C++. These classes are accompanied by a function that iteratively writes their postscript interpretations to a postscript file. This was developed by Hal and Paul

These two parts have not been merged and to some degree cross over in their use. This divergence is mostly due to lack of communication and differing interpretations of the deliverables requested for project 2.

Other Documents:

- See *CpsCompiler/HowToUseCPS.txt* for instructions on the CpsCompiler
- See *Explanation of Testing Procedures* for testing procedures and a critique of the shape language specifications.
- See *readme.txt* for a list of files and their purposes.
- See *ReadMe PostScript_Samples* for an explanation of the postscript files in the top directory of this repo

Building the CPS Library and Main:

- Run [*build.sh*](#), which is located within the *source* directory.

Using the CPS library and Main:

A main method can be found in *source/main.cpp*. Editing this file and running the executable produced is a quick way of writing shapes to a postscript file.

The main entry point however is the *writePSfile* function located in *source/write-shape.cpp*. A vector of pointers to shapes along with a string filename can be passed to this function, producing a valid postscript file. It is the task of the user to include the appropriate headers from the CPS language.