

Project 4. Procedural modelling in urban planning.

This project must be turned in **before 1:00pm on Tuesday June 26th, 2018.**

For this project you have freedom of choosing a development platform of your preference. You may produce, for example:

- a Python script to be run in Blender, or
- a standalone C++ application, or
- a WebGL application to be run inside a browser, or
- other (but in this case, please clear it with me first)

Whichever your choice, it must implement an interpreter for a simplified version of the CGA shape grammar described in the Müller et al. Paper [1]. It should be possible for me to execute it and check its operation in a standard Linux system (not unlike the ones in the labs).

Your code will get as input a simplified CGA grammar (you might want to have the rules hardcoded in the code to simplify the development, but in this case they should be clearly isolated and easy to modify) together with a collection of simple shapes (e.g. cubes, cylinders and predefined roof types). It will create a scene resulting of the interpretation of the grammar rules.

You should support basic rule symbols (instantiate, translate, rotate, scale, push/pop, subdivision, absolute/relative size values, extrusion...) but you do not need to implement advanced features such as occlusion tests.

You have greater leeway in the exact definition of the scope than in previous projects; the exact nature and format of the rules is up to you. For example, it is OK to force the user to rewrite CGA rules as Python classes/functions, provided that this rewriting is straightforward. In other words, you may adapt the way the rules are written to ease your implementation (this is not an exercise in language parsing), but you should include indications clear enough that I can test your program with grammars of my choice.

Correct handling of texture coordinates (you can assume simple shapes are parameterized) is a desirable but optional feature.

You must turn in all source files, including your script(s) or code and .pro or Makefiles if appropriate, and one initial configuration, together with a text file with your comments and instructions to use your implementation. Provide as examples at least two more configurations that can be loaded and evolved into full models using your implementation.

[1] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool (2006). "Procedural modeling of buildings". *ACM Trans. on Graph.* **25**(3) (July 2006), pp.614-623. <http://doi.acm.org/10.1145/1141911.1141931>