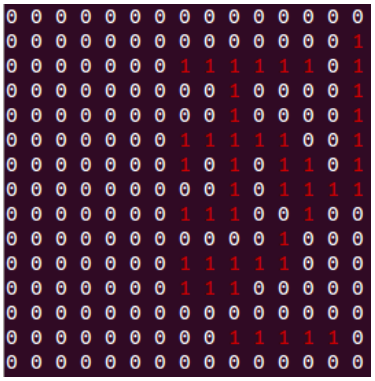
DGI 14: Procedural road network generation
Personal report

Martin Engilberge
engi@kth.se
930118-T070

June 19, 2014

1 Workflows:

We first started from the **'Procedural modeling of cities'** paper [1], in this document a complex L-system is used to generate the road network (parametric with several constraints), not knowing how an L-system was working we started by making a small prototype of a very simple L-system using the turtle graphics method to convert the string produce by the L-system in an actual road.



(a) Screen-shot of the first prototype, displayed in a terminal. The red 1's represent the road

```
S-> SR 0.20     L-> LS 0.7
    SL 0.20         Ls 0.3
    SS 0.50
    Ss 0.10     s-> sS 0.40
                    sR 0.10
R-> RS 0.7          sL 0.10
    Rs 0.3          ss 0.30
```

(b) Grammar of the L-system (S:straight, R:right, L:Left, s:straight invisible

From this prototype it was noticed that to get a more realistic result it will be necessary to use a more complex L-system, we decided to use a Bracketed L-system (allowing us to design road we several "branches")

But with that kind of L-system we have very little control on where the road is heading, to guide the road we decided to combine the L-system with another algorithm (inspired from [2]). By mixing this two methods we get the randomness of the L-system to generate small portions of road network, and we get the control from the other algorithm, its define the primary and secondary goals and generate corresponding parameters that are then use by the L-systems.

Once this global idea of the program define the coding of the project started, it is written in c++, using qmake to build the project, we reuse some library

and function from the lab 1 (GLM and SDL library, interpolate function) to spare some time.

2 Difficulty encountered :

- At the beginning we encountered some difficulty to integrate the library into the project (SDL, GLM).
- Since our L-system has a stochastic grammars (several proposition with the same left hand side and an associated probability) the program has to choose a proposition depending on their probability, and we were not use to implement random selection with non equiprobable elements. Our first implementation wasn't mathematically correct, it took us several attempts to get a correct result.
- Then during the project we spend a bit of time wondering how the population density and altitude should be integrated in our program. We ended up using the population to guided the road network, and the altitude to increase the cost of a road segment.

3 Group work:

To simplify the work in a group we used Git [3] combined with a wiki page where all the information on the project where written (Todo list, Work in progress, and completed part).
I mostly work on the L-system and the main algorithm, as Florian was figuring out how was working the png library to read data from png files (population density, height-map) and to store the final result as a png.

4 Perceptual user studies:

A perceptual user studies could be beneficial in many different ways for our project. First a perceptual study could be used to evaluate the realism of our results it could be done by asking participant to grade several maps representing road networks (some real other generated by our program). Since there is a lots of parameters that can be change in the program (like road segment length, angle of branching roads, size of squares when the map is subdivided ...) and that changing this parameters can give various results a study on this parameters could be interesting: This could be done in a more interactive way by giving some control over this parameters to the participants (the keyboard could be use to increment/decrement some variables in an hidden way the participants would only see the final result) then it will be ask to the participants to try to find the most realistic result (or find something close to an existing pattern). It could be good way to find the best sets of parameters.

5 Possible improvements:

- To improve our project, we would need to add more constraints when we add a new point to the network to get something more coherent at a global scale, A possibility to get that is to implement a more complex L-system (using parametric grammar) and to add the constraints directly into the grammar.
- Another improvement would be to deal with inaccessible areas, (water, mountain, park ...) and to create special road sections like bridge our tunnel with a corresponding cost (higher than a simple road segment).
- If I had to restart this project now the first thing I would do differently would be to choose another way to display the result, currently the road are drawn pixel by pixel, and the result is not really smooth (it's complicated to keep a consistent road width or to gets curved roads). A more high level way (Why not in 3D?) to represent the road network would be a big improvement.

6 Conclusion:

To conclude even if the results are far from perfect and could be improve with more time, we still get a plausible road network (corresponding to our input density map and height map). More importantly this project was an occasion to learn several things:
it was a good introduction to procedural generation of content, I learn a great deal about L-systems, and get more familiar with C++ syntax.

References:

[1] Yoav I. H. Parish and Pascal Müller. 2001. Procedural modeling of cities. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01)
[2] http://www.newton64.ca/blog/?p=747
[3] https://code.google.com/p/road-network-generator/source/browse/