

2nd year advanced project (2008)

Ben Porter, PhD Student at CEMA (Rm144/Bldg63), Monash University

Benjamin.porter@infotech.monash.edu.au

Here are some ideas for a second year advanced project in computer graphics / 3d modelling. The projects are fairly general and can be tailored depending on your strengths. Some knowledge of computer graphics would be good, but not necessary as I will provide support in this area. These projects would be great introductions to 3rd year computer graphics.

An intuitive and fun modelling environment

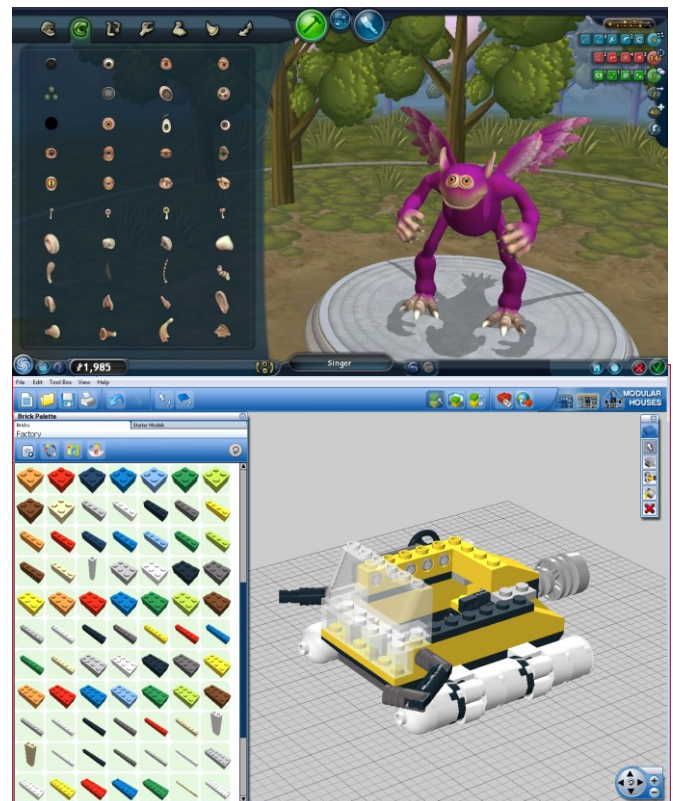
Build a fun, intuitive, and simple editor for creating a very specific type of shape.

For example, the Lego Digital Designer and Spore Creature Creator offer a tightly constrained domain, and simple set of parts which the user can add via a drag-and-drop interface. The system has some intelligence that infers where the user is placing a part, what orientation it has, whether symmetry should be preserved, etc. Both these tools provide a nicely rendered representation of the shape, which is very beneficial to understanding the form being created (via shadows and animation in Spore, to transparency and specular highlights in Lego).

This project would involve looking at a very specific domain (e.g., castles, chairs, caricatures, robots) and building a modelling tool that is interactive, fun to use, and interesting.

Some activities this project would involve are:

- Analysing the domain and designing a general model of the shape, e.g.,
 - Robots have bodies, heads, arms and legs. Heads can be cylindrical or cuboid, of any dimensions. Robots can have any number of arms. Robots can be shiny, rusty, or plastic.
- Consideration of the computational shape representation, e.g.,
 - Robots can be built from primitives such as cubes, spheres, and cones.
 - Caricatures can be built from lines, curves of varying stroke width, cross-hatching.
- Developing some prototypes, looking at different supporting frameworks
 - Does this tool integrate into existing modellers (like Blender), or
 - Does it stand alone? (If so, can you export to a standard file format?)
- Looking at different interaction methods, tools and visual cues within the domain, e.g.,



- A user should be able to make a robots head bigger simply by dragging a handle.
- Is a shiny robot easier to manipulate that a matte one?
- Can a user paint rust directly onto the robot?

The outcome of this project would be the working software.

Procedural terrain



Figure 1 (C) Yaroslav Dynda 2006 “Hill Watcher” (Created with Volumetrics 0.9 and MojoWorld Pro 3.1.1) from http://www.pandromeda.com/gallery/still_large.php?screen=790

Terrain is a necessary component of any virtual world and there is much research into efficient and interesting tools for the creation, texturing, destruction, and rendering of virtual terrain. Manual creation of terrain geometry and texture has been overshadowed by *procedural methods*. These methods use various algorithms to synthesize geometry. Texturing & Modelling (Ebert and Musgrave) is one of the definitive books on these methods and fun to read. Also see <http://www.vterrain.org/Elevation/Artificial/> for some refs/articles.

The state of the art software is made by Pandromeda (<http://www.pandromeda.com/>), but numerous open or free tools also exist (e.g., <http://www.bundysoft.com/L3DT/>). Procedural terrain

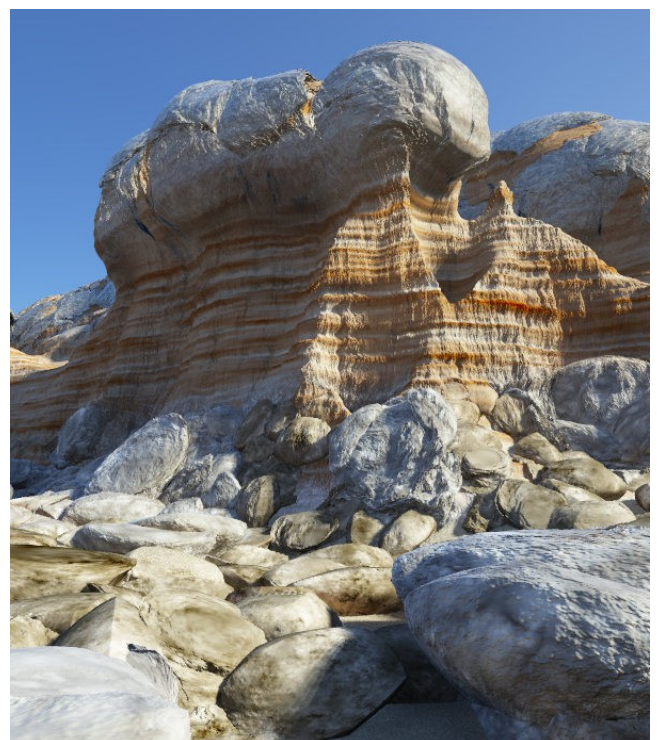


Figure 2 (C) Klas Kruse, from <http://www.planetside.co.uk/gallery/f/tg2/explorersend.jpg.html>

is also used in games (such as <http://www.infinity-universe.com/Infinity/>).

Chapter 1 in GPU GEMS 3 nicely illustrates one method for generating terrain in real-time (see this video: http://www.nzone.com/object/nzone_cascades_videos.html). This project involves looking at this general method and exploring it. It involves concepts such as isosurfaces <http://en.wikipedia.org/wiki/Isosurface> , marching cubes http://en.wikipedia.org/wiki/Marching_cubes , and noise http://en.wikipedia.org/wiki/Perlin_noise .

This project could be geared towards either programmatic or creative outcomes. For example, povray (<http://povray.org/>) has an isosurface renderer, so you could just play around with the models, or you could build a terrain plugin for Blender (<http://blender.org>), or alternatively you build a library for use in real-time applications.

Other Projects

I'm willing to supervise almost any project that involves computer graphics, computer games or physical simulation in some way. Please come to see me if you have any projects of your own and we can discuss it. For more ideas about what I'm interested in, here are some more links:

http://algorithmicbotany.org/papers/ , groimp	Visual plant modelling, L-systems
http://en.wikipedia.org/wiki/Generative_Modelling_Language	Describing shapes via construction
SIGGRAPH (need to login via authcate)	Almost anything published in Siggraph
http://puredata.info/ , http://vvvv.org/ , xfrog	Visual programming of graphics
http://blender.org	Open source 3d modelling tool (scriptable with Python)
http://processing.org	Synthesis of image through process
http://www.cs.cmu.edu/~baraff/sigcourse/index.html	Physically based modelling