# Major/Minor Project 2012-13 • Student Project Suggestion

We will soon be sorting out supervisors and project topics for the Major/Minor projects (CS39440, CC39440, CS39540, CS39620). Lecturers and other members of staff will make a list of project suggestions, which will be released as soon as possible.

We also allow students to suggest project topics. If you would like to suggest a topic for your project, please complete this form and return it to Neil Taylor ([nst@aber.ac.uk](mailto:nst@aber.ac.uk)). He will identify possible project supervisors that you can talk to about the project.

*Making a project suggestion does not commit you to the project, or guarantee that we think that the project is suitable for the module. It is a starting point for a conversation about your idea.*

For other questions about the Major/Minor Project, please see the website at: <http://www.aber.ac.uk/~dcswww/Dept/Teaching/CourseNotes/current/CS39440/>

This form is for any student taking the modules CS39440, CC39440, CS39540, CS39620.

## Student Details

*Please enter the following details.*

|  |  |
| --- | --- |
| Name | Hristoz Stefanov Stefanov |
| Email | hzs9@aber.ac.uk |
| Degree Scheme | Graphics, Vision and Game |

## Suggested Project Title/Topic

*Enter a brief (up to 100 characters) title or topic area.*

3D Volumetric Engine for Real-Time Applications

## Project Description

*Enter a description of your project description. Enter between half a page and a whole page for your description. In your description, consider any technologies that you may investigate as part of the solution.*

Traditional 3D graphics renderer such as Direct3D and OpenGL represent 3D objects as a collection of textured polygons (triangles or sometimes quads). One alternative to the polygon based method is the volume based one. Volumetric renderers represent 3D objects as a collection of 3D points (or voxels) much like how 2D raster graphics use pixels to build up images. They have the following advantages:

* The data represents real objects more accurately allowing for more complex shapes to be rendered without any performance penalties, which in turn produces nicer and more realistic results.
* Objects are composed of “atoms” rather than surfaces, which makes the data dense, operations such as ray casting and point-perfect collision detection faster and more accurate.
* The representation is more natural, which means it is easier to generate data procedurally.

Volumetric engines however also have disadvantages:

* The data requires a lot more memory which makes it very inefficient and not practical for most applications. For instance a just for a 3D scene composed of 256x256x256 and 32-bit voxels the memory required is 67,108,864 Bytes (= 64 MB)
* The size of an object affects the performance and the size of the data, which is not the case with its polygon based alternative.
* There is no integrated hardware support; all modern video cards are designed to operate on polygon data sets, which means at this stage voxel graphics can never have the same performance.

Volumetric engines are now primary used in medical imagery software, as they can show the internal structure of a volume and represent complex natural shapes more accurately; also scanners have to do less work when or after they are collecting the data. There have been several attempts to use volumetric engines in games but the technology is still undeveloped and underappreciated. Now volumetric graphics are gaining more popularity with their use in procedural terrain generators. The purpose of this project is to implement and optimize a volumetric engine for use in real-time applications such as games, and find a solution for its biggest drawback – the amount of memory that it requires.

Once the basics of the engine are clear there are number of other interesting issues that could see more attention: 3D lighting and shadows, ray-casting and picking, skeletal animation and morphing, 3D image processing and special effects, (dynamic) level of detail and other optimizations.

## Relevance of the project to your degree scheme

*Your project needs to relate to your degree scheme. Some schemes have more flexibility than others, e.g. G400 Computer Science, but there is still plenty of scope to choose a project that will be interesting and suitably challenging for your scheme. Briefly explain the particular relevance of this idea to your scheme and/or your future work plans.*

I believe this project fits perfectly in the “Graphics, Vision and Game” scheme. It would require the use of OpenGL (or Direct3D) for the actual rendering and biting to an output device, 3D transformation techniques, image processing techniques (and more specifically image compression and decompression) might and probably will involve implementation of a dedicated scene graph, and may involve the use of shaders to gain optimal performance from the hardware. These are all disciplines that taught or closely related to G451.

## Work with a company

*Does this project relate to work with a company, e.g. your Industrial Year Employer? If so, what is the company and what is your relation to them? What discussions have you had with the company? What discussions have you had with them about the Intellectual Property Rights in the outputs of the project?*

N/A

## Any previous investigations

*Have you investigated any aspects of this project so far? For example, have you been reading about the topic or making some prototypes? If you have, please describe what you have done.*

I have not done any substantial work on the project, nor have I spent enough time on research, but I have experimented with rendering of simple voxel buffers and applying 3D camera transformations to them. I also have thought about how I could reduce the amount of memory required for storing and rendering of volumes and came up with a number of techniques that can be used – use of a frame and depth buffers (and issues arising from them such as handling of transparency), use of sparse data structures and object reuse. Unfortunately I haven’t read much on the subject and I don’t remember author or article names I have, but I could name a recent voxel engine in existence – Unlimited Detail.

- END -