

# CS4052 Computer Graphics

## Assignment 5 - Interactive Game Project

A. Gerdelan, Trinity College Dublin

Demonstrations to be done **11AM-2PM**, Tue 9 Dec 2014  
Course Work %: approx. 40%



## 1 Outline

- The purpose of this assignment is to allow you to put the theory from lectures and reading into practice to create an appealing interactive game.
- You are required to create a 3d game using modern, shader-based OpenGL (specifications at the end of this document).
- On the due day, we will use the full 2-hours, **including our normal lecture time**, to view brief demonstrations of projects in the labs. All projects will be reviewed later by all markers to keep grades consistent.
- You are also required to submit to Blackboard, by the next Tuesday, **16 December**:
  - a PDF report that explains your design and technical choices, with images.
  - all code source files and assets in a ZIP file on Blackboard
  - a short demonstration video recording of your game, in any format (you might find a tool like FRAPS, or Nvidia Shadowplay, quicker to use than writing your own). Put this in the ZIP file.
- You may use any libraries that you like for windowing, interaction, asset handling, special effects, physics, audio, mathematics etc. so long as you acknowledge these.
- It is not allowed to use a full graphics engine. This is a test of your ability to programme the basic 3d graphics functionality covered in class, so no higher-level libraries or engines (OGRE, Unity, Unreal, etc.) are allowed.

- You may use any operating system that you like, so long as you are prepared to demonstrate it on the day - prepare a fall-back in case your laptop stops working. If you want help porting to Windows/Linux/OSX or making a video - ask!
- Be sure to attend labs and ask demonstrators for help. Start well ahead of time, and use discussion boards on Blackboard to ask questions as you go. It's a good idea to ask for as much help as possible before the last week, as the demonstrators will have more time.
- If you fail to show up for the lab or to submit your report on time, you will be reported as absent and will receive a grade of 0%.
- This assignment is strictly individual (no group work).
- Demonstrating code that was not created by you as your own is considered cheating, and will be reported as such.

## 2 Specifications

Your game can be any kind of game that you like, but must have the following functionality:

- user interaction and camera-control
- scoring and winning/losing
- 3-dimensional objects and views
- at least one hierarchical, animated creature or object
- at least one texture-mapped object
- some form of lighting calculation that suits your game style, programmed in shaders

This basic functionality is worth approximately 50% of the project mark, but you must have made a solid attempt at some advanced features to obtain the full 50%.

The final 50% (approx) will be given for advanced features, with approximately 10% per perfectly-executed feature - simple features will receive less, elaborate or well-designed features will receive more. You have some scope to put effort into your favourite aspects of 3d graphics, or experiment with new ideas:

### 2.1 Advanced feature ideas

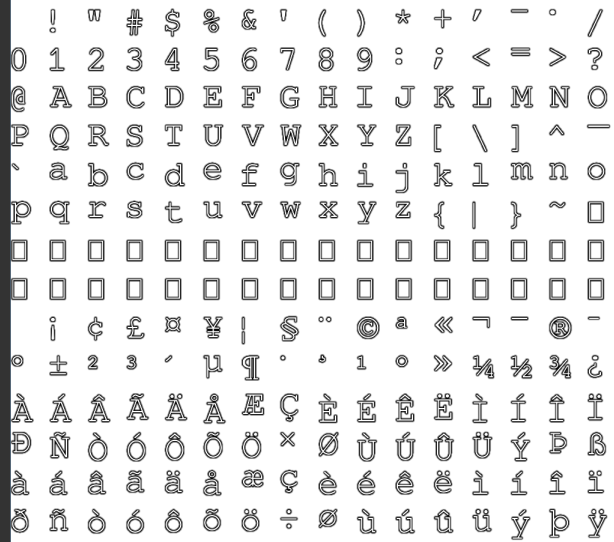
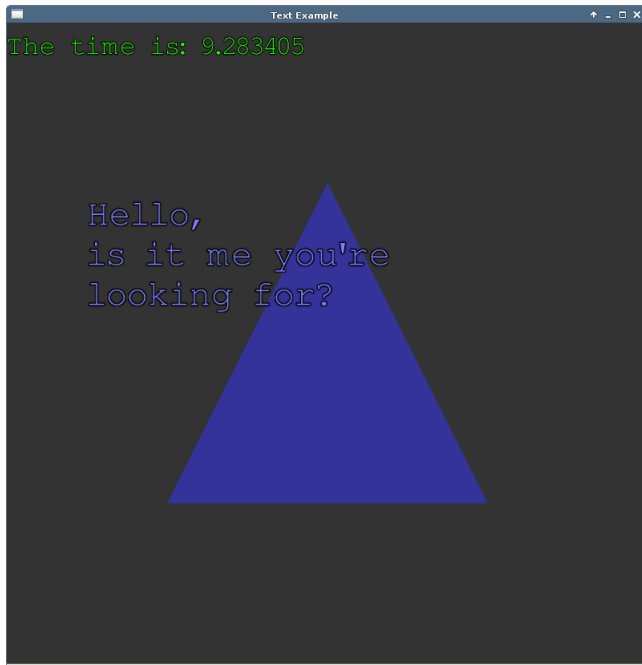
- particularly imaginative game-play or narrative
- cut scenes or in-game cinematic that you made yourself
- menus, start, end screens, etc.
- collision detection
- simulated dynamics or physics of any sort
- game level editor
- level-of-detail control, frustum culling, or occlusion culling
- great models that you made yourself
- height-mapped terrain

- procedurally generated terrain or meshes
- high-quality texture artwork that you made yourself
- 3d “picking” to select game objects with the mouse cursor
- more advanced texturing effects; multi-texturing, specular-mapping, etc.
- partially transparent geometry using alpha-blending
- consistent Phong lighting, using either spotlights or point lights with distance attenuation, or otherwise very high-quality lighting
- distance fog
- cube mapping for sky-boxes, environment mapping, or refraction
- on-screen control panels
- good animated texturing effects
- particle systems, Boids, or other interesting animations using shaders
- overall style and consistency
- interesting shader effects of other types
- interesting camera animations or manipulations of perspective
- ??? - your own feature ideas

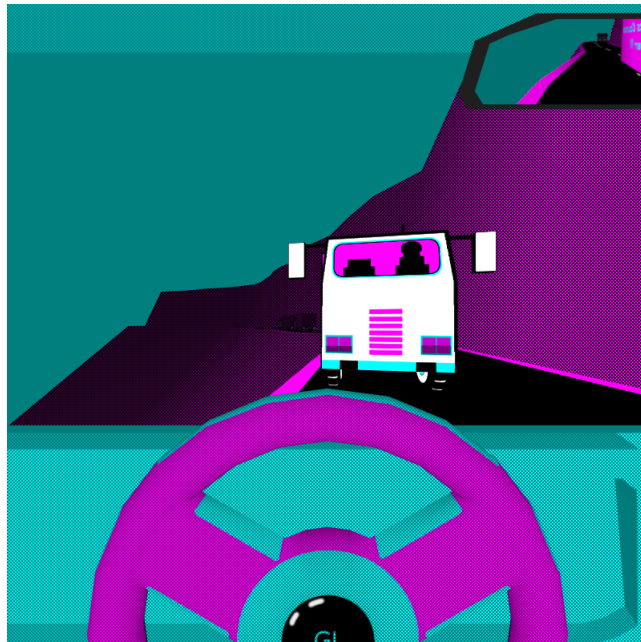
### 3 Notes

Your main problem with this assignment will be project **time estimation and management**. Learning new techniques is a good idea, but can be very time consuming - we are available for advice or to help you with a new topic - **don't be afraid to ask for help and discuss your plans!** Try to create or find a very simple game design that allows you to play with some new graphics techniques or extend your previous assignments. It's easier to add value to a simple design than to finish a complex design!

One very difficult aspect of modern OpenGL is that there is no built-in text rendering, which would be useful for updating a score. The typical approach is to create a texture containing all of the characters required, and then to change the texture coordinates to display a desired character on triangles. There are libraries available to do this, but they can be a time-sink to learn and integrate. I made some simple text-rendering functions for you to get you started quickly. You can find `text.h` and `text.cpp`, and font files `freemono.png`, `freemono.meta` at [https://github.com/capnramses/opengl\\_text\\_rendering](https://github.com/capnramses/opengl_text_rendering). You will also need `stb_image.h` if you don't already have it. Let me know if you need a hand getting started.



The demonstrators have both made game projects in OpenGL before, and can give you some thoughts on the feasibility or challenges with different project ideas. Two of Tim's game projects are pictured at the top. You can find a video of "Space BeTrail" at [http://youtu.be/L3F8k\\_9T7jM](http://youtu.be/L3F8k_9T7jM).



I tend to do a remake of older Atari 2600 or console games for projects like this. I have started a game project example to show you how I would do a project like this in OpenGL - by remaking 1987 IBM-Compatible game "Test Drive". You can find my posts about this on the discussion forum, and my code is on GitHub at [https://github.com/capnramses/opengl\\_test\\_drive\\_clone](https://github.com/capnramses/opengl_test_drive_clone). Yes, the colour choices are deliberate - it's a reference to the garish CGA colour palette from the original game. If you want some more detail about how I handled a certain part I'm happy to talk about that.