RESEARCH DISSERTATION & ARTEFACT

Version 1.0
BSc Computing for Games
COMP360

Dr Michael Scott

Introduction

In this assignment, you are required to **design** and **implement** a C++ program using SDL and OpenGL which demonstrates the type of 3D computer graphics techniques that appear in a modern game engine.

Graphics technology is one of the most obvious areas in which innovation has driven gaming technology in recent years. Modern gaming PCs and consoles contain powerful graphics processing units (GPUs), and gamers expect modern games to push this hardware to its full potential. In this assignment you will practice the use of advanced graphical effects. Your final product will be a portfolio piece, which you can use in future to demonstrate your mastery of these techniques. This portfolio piece will be built up using a series of Worksheets, worksheet A and B will give you a foundational framework for the rest of the assignment which will be completed in C and D.

"Because of the nature of Moore's law, anything that an extremely clever graphics programmer can do at one point can be replicated by a merely competent programmer some number of years later."

— John Carmack

"Currently computer graphics are used a great deal, but it can be excessive."

— Hayao Miyazaki

This assignment is formed of several parts:

(A) **Write** a 2-page handout that will:
  (i) **outline** the concept of your demo;
  (ii) **explain** how your demo satisfies the requirements of the contract (provided as an appendix at the end of this document);
  (iii) **describe** at least **two** graphical or simulation effects your demo will include;
(B) Worksheet A - **Project Framework**
(C) Worksheet B - **Camera & Basic Scene**
(D) Worksheet C - **Demo specific work**
(E) Worksheet D - **Demo specific work & Viva**

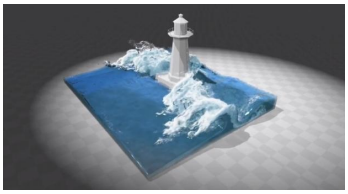
Assignment Setup

This assignment is a **programming** task. Fork the GitHub repository at the following URL:

`https://github.com/Falmouth-Games-Academy/comp220-portfolio`

Use the existing directory structure and, as required, extend this structure with sub-directories. Ensure that you maintain the `readme.md` file.

Modify the `.gitignore` to the defaults for **Visual Studio**. Please, also ensure that you add editor-specific files and folders to `.gitignore`.



A demo of fluid simulation with NVIDIA's PhysX. Recent advances in GPU technology have enabled a wide range of high-fidelity real-time rendering and simulation effects.

Part A

Part A consists of a **single formative submission**. This work is **individual** and will be assessed on a **threshold** basis. Answer the following questions to pass:

- What is the title and high concept of the demo?

- What is the intended aesthetic?
- For each of two graphical or simulation effects:
  - What is the effect?
  - How will the effect contribute to the aesthetic?
- Is the scope appropriate for the product development time-frame?

To complete Part A, prepare the handout using any word processing tool. To help illustrate your intended aesthetic, your handout may include images and/or links to online videos.

Show the handout to your **tutor** for immediate **informal feedback**.

Once you have completed Part A, you should move onto the Worksheets. See the worksheets on the Learning Space for more details.

Additional Guidance

As always, avoid underestimating the effort required to implement even simple software; always consider scope. From the proposal stage, you should consider very carefully what is feasible.

Your code will be assessed on **functional coherence**: how well the finished product corresponds to the user stories, and whether it has any obvious bugs. Correspondence to user stories runs both ways: implementing features that were not present in the design ("feature creep") is just as bad as neglecting to implement features.

Unlike your previous assignments, you will be assessed on the **performance** of your solution. Real-time graphics and simulation are not just about creating aesthetically pleasing effects, but doing so whilst maintaining a smooth and consistent framerate free of any lag or glitches that might frustrate the player. It may be necessary to trade-off the complexity or fidelity of an effect in order to achieve acceptable performance.

Your code will also be assessed on **sophistication**. To succeed on a project of this size and complexity, you will need to make use of appropriate algorithms, data structures, libraries, and object oriented programming concepts. Appropriateness to the task at hand is key: you will **not** receive credit for complexity where something simpler would have sufficed.

Maintainability is important in all programming projects, but doubly so when working in a team. Use **comments** liberally to improve code comprehension, and carefully choose the **names** for your files, classes, functions and variables. Use a well-established commenting convention for **high-level documentation**. The open-source tool Doxygen supports several such conventions. Also ensure that all code corresponds to a sensible and consistent **formatting style**: indentation, whitespace, placement of curly braces, etc. Hard-coded **literals** (numbers and strings) within the source should be avoided, with values instead defined as constants together in a single place. Consider allowing some literal values, where appropriate, to be "tinkered" without changing the source code, e.g. by defining them in an external file read at startup.

As with all assignments on this course, you are expected to display a level of **innovation and creative flair** befitting Falmouth University's reputation as a world-leading arts institution. One approach to promoting creativity is **divergent thinking**: generating ideas by exploring many possible solutions. Often the most interesting ideas are **subversive**: they deliberately go against convention or obvious solutions.

You will **not** be judged on the quality of your art assets. It is fine to use meshes and textures found online, as long as they are available under an appropriate license and are properly attributed.

FAQ

- **What is the deadline for this assignment?**
  Falmouth University policy states that deadlines must only be specified on the MyFalmouth system.
- **What should I do to seek help?**
  You can email your tutor for informal clarifications. For informal feedback, make a pull request on GitHub.
- **Is this a mistake?**
  If you have discovered an issue with the brief itself, the source files are available at:
  `https://github.com/Falmouth-Games-Academy/bsc-assignment-briefs`.
  Please make a pull request and comment accordingly.

Additional Resources

- `http://www.opengl-tutorial.org`
- `http://gamedev.stackexchange.com/questions/32876/good-resources-for-learning-modern-opengl-3-0-or-later`
- `https://google.github.io/styleguide/cppguide.html`

Marking Rubric

Criterion	Weight	Refer for Resubmission	Basic Proficiency	Novice Competency	Novice Proficiency	Professional Competency	Professional Proficiency
Project Ouline	10%	No or Late submission of project outline	A poor standard of English and the outline is not clear	A good standard of English and the outline has some merit	A good standard of English and the outline is well justified	A good standard of English and the outline is very good	A good standard of English and the outline pushes the students skill
Worksheet A	10%	No or Late submission of worksheet	See worksheet for details				
Worksheet B	15%	No or Late submission of worksheet	See worksheet for details				
Worksheet C	20%	No or Late submission of worksheet	See worksheet for details				
Worksheet D	25%	No or Late submission of worksheet	See worksheet for details				
Maintainability	15% ‡	There are no comments in the source code, or comments are misleading. Most variable names are unclear or inappropriate. Code formatting hinders readability.	The source code is only sporadically commented, or comments are unclear. Some identifier names are unclear or inappropriate. Code formatting is inconsistent or does not aid readability.	The source code is somewhat well commented. Some identifier names are descriptive and appropriate. An attempt has been made to adhere to thhe PEP-8 formatting style. There is little obvious duplication of code or of literal values.	The source code is reasonably well commented. Most identifier names are descriptive and appropriate. Most code adheres to the PEP-8 formatting style. There is almost no obvious duplication of code or of literal values.	The source code is reasonably well commented, with Python doc-strings. Almost all identifier names are descriptive and appropriate. Almost all code adheres to the PEP-8 formatting style. There is no obvious duplication of code or of literal values. Some literal values can be easily "tinkered" in the source code.	The source code is very well commented, with Python doc-strings. All identifier names are descriptive and appropriate. All source code adheres to the PEP-8 formatting style. There is no obvious duplication of code or of literal values. Most literal values are, where appropriate, easily "tinkered" outside of the source code.
Use of Version Control	5%	GitHub has not been used.	Source code has rarely been checked into GitHub.	Source code has been checked into GitHub at least once per week. Commit messages are present. There is evidence of engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of somewhat meaningful engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of meaningful engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of effective engagement with peers (e.g. code review).

Appendix: British Computer Society Requirements

An individual project is an expectation within undergraduate, integrated masters, and postgraduate masters programmes in computing. Students must be provided with written guidance on all aspects of the project, including selection,conduct, supervision, milestones, format of the report and the criteria for assessment. All projects should reflect the aims and learning outcomes which characterise the programme to which they contribute as set out in the programme specification.

It is expected that within an undergraduate programme, students will undertake a major computing project, normally in their final year and normally as an individual activity, giving them the opportunity to demonstrate:

- their ability to apply practical and analytical skills present in the programme as a whole
- innovation and/or creativity
- synthesis of information, ideas and practices to provide a quality solution together with an evaluation of that solution
- that their project meets a real need in a wider context
- the ability to self-manage a significant piece of work
- critical self-evaluation of the process

Projects must involve the production of a report which should include:

- Elucidation of the problem and the objectives of the project
- an in-depth investigation of the context and literature, and where appropriate, other similar products (this section is likely to be emphasised less for an IEng project)
- where appropriate, a clear description of the stages of the life cycle undertaken
- where appropriate, a description of how verification and validation were applied at these stages
- where appropriate, a description of the use of tools to support the development process
- a critical appraisal of the project, indicating the rationale for any design/implementation decisions, lessons learnt during the course of the project, and evaluation (with hindsight) of the project outcome and the process of its production (including a review of the plan and any deviations from it)
- a description of any research hypothesis
- in the event that the individual work is part of a group enterprise, a clear indication of the part played by the author in achieving the goals of the project and its effectiveness
- references

In the event of this major activity being undertaken as part of a group enterprise, there is a requirement that the assessment is such that the individual contribution of each student is measured against all the above learning outcomes.

For accreditation for CITP, CEng or CSci, the individual project should be worth at least 30 credit points at level 6 or above. The project must be passed without compensation. For accreditation for IEng the individual project should be worth at least 20 credit points at level 5 or above. The project must be passed without compensation.