University of Southampton	*School of Electronics and Computer Science*	*Coursework (4 of 4) Instructions*
Module: COMP3214	Title: Principles of Computer Graphics.	Lecturer: Dr. J N Carter
Deadline: see hand-in.	Feedback: Deadline + 20 working days.	Weighting: 25%

Instructions

3D World Simulation – A 3D Space War Demo or Moons, Asteroids and Planets

Introduction

If any rules or requirements are in conflict with the general document on the coursework page then this document takes precedence.

This coursework is all about building a 3D animation of a modestly complex scenario. In the past themes such as Moon Landing, Mars (as in bars and rovers). The Sea, Robots and Clangers have also been used in the past.

This year the theme is ‘Space Wars’ Here are some pictures I found when I searched on Google.

PDP-1 Spacewar

Taking a hit
A Big Battle.

Home made BEM's and saucers

There are no other restrictions on how you interpret this title as long as there is a relationship. This can be tenuous, very tenuous as long as there is some visual connection.

There are three different modes.

Demonstration.	This should run as a **demonstration** and aside from **controlling the view point** (mouse or keyboard is OK) and starting and stopping the tour, no further interaction is allowed and may result in a loss of marks. What is important is that in your animated world you should have • Many static objects. • At least two moving object. • An interesting landscape or seascape, or space backgrond • User controls, see later. • A viewing position controlled by the user, see later. • And an animated tour, see later. Please remember that it is more important to produce something that looks good than something that is realistic[1]. To ensure this, as many marks are allocated for visual and artistic style as pure technical achievement. For example in the past excellent marks have been achieved by people who exploited a simple technical style, but using it to produce coherent and

[1] If you know the Hitch Hikers Guide to the Galaxy then you will understand that the power of art over powers that of gravity and that objects can float in mid-air as long as they look good.

	believable animation. An abstract or surrealistic design would also attract good marks if it was well executed.
Tour.	There should be no interaction with the animation other than having a tour to show off all the features of the world. Outside the tour you should be able to control the position and direction of the camera to explore the world and have a number of fixed viewpoints activated by keys. The tour must show off all the different techniques you used, even if it's just lighting.
Game (optional).	This is optional, there are no extra marks, it will be judged as a replacement for Demonstration mode. The following might apply: • Camera mounted on a space ship. • You control the orientation of the spaceship, see keys later. • You have a gun and can only fire one shell/missile/laser bean at a time, reloading. • One adversary who aggressively moves towards you. Takes no account of intervening structures, get stopped by them. Fires automatically at fixed range. Does not take your velocity into account. If you kill it, another appears randomly.

The program must start in 'Demonstration'.

Expanding on this, your coursework submission should be built on what you did for Coursework I/P along with the following additional elements:

1. A pleasing/interesting scene, your interpretation of my theme. There must be motion in the scene.
2. An automatic tour of the scene, triggered by pressing the "T" key. This should take the viewer through the scene showing **all** its features to best advantage, and should last at least 20 seconds, generally much longer. The motion of the camera must be smooth and slow-enough to allow the viewer to appreciate what he or she is seeing. The scene animation should continue to run during this period. If a feature is not shown off in the tour we may not see it and give you credit for it. A catmul-Rom spline is a good way to describe the tour.
3. The user must be able to control the view point via keys detailed below.

The following user interface is mandatory

1. Pressing the "P" key must position the camera at the viewpoint from which your submitted picture was taken (see later.)

2. Optionally pressing ‘L’ or ‘O’ should take you other viewpoints
3. Pressing ‘M’ should return you to the camera.
4. The camera is moved as specified below.
5. Pressing 'Q' or ‘Esc’ should exit the program.
6. ‘T’ for tour, ‘G’ exit tour.
7. Optionally 'H' should either cause a help screen to be displayed (advanced) or for it to be printed on *stdout* in an attached DOS or Shell window.

The scene should contain

8. An interesting landscape/planets (a mesh of triangles and buildings for example). If appropriate a height field might be used.[2]
9. Lights, materials and texture mapping are all allowed but not required.
10. Likewise loading models is nice but not required.
11. The user must be able to move the point of view around the environment., see later

Submission

You must include a file ‘config.lua’ to indicate to the test and checking systems how to build your program. See the Coursework web page for details. The location of this, when your submission is unpacked is taken to be your ‘root’ directory.

Please do not include any files that are part of GLM, GLEW or GLFW/freeGlut, any operating specific files like windows.h or any other files that you used to build your coursework that my build system provides.

There are no restrictions on the names or the number of ‘cpp’ files submitted. Other than that, names should not be case sensitive, i.e. names like `X.cpP` and `x.cpp` must not be used together. The automatic system will attempt to build an executable program by compiling all of the files together. Make sure you only have one main() function as having two or more confuses compilers.

There are no restrictions on how many times you may submit your coursework. Normally only the latest submission before the submission deadline will be considered.

Your submission should be in a zip file and should contain all your code, shaders and images/textures used. You may organise this any way you like but your code must be able to find shaders etc. on a path relative to ‘config.lua’.

All the source files must be in the ‘root’ of your archive. If it contains a folder called ’include’ the test software will add this to the search path for header files. Be careful #include <test.h> and #include “test.h” look in very different places. This folder must also contain the mandatory screen shot and text files.

You must not include binary files, executables, libraries or object files

Names must not clash with system files. Otherwise there are no restrictions on the names or the number of ‘h’ or ‘cpp’ files submitted.

Failure to conform to this structure is the most common cause of submissions being rejected.

[2] One of the objects in Bullet is a height field so your vehicle should be able to navigate it.

In some cases you may be asked to resubmit your coursework after the deadline has passed, usually when I judge that there is a problem, there will normally be no penalty. However re-submission must be by the Hand-in system. I will not accept whole or partial coursework submissions by e-mail, disk or other media.

The usual plagiarism rules apply, and automatic plagiarism detection may be used. However, you are free to borrow from the many examples available, on the module web site or on the WWW, provided you quote your source, both in your code and if applicable in the accompanying 'credits.txt' file.

There are NO marks for anything other than the graphics and the programs animation sequence. For example, you might submit a complex physics driven game format like Quake or Doom, in this case there would be no marks awarded for effort spent on a home grown AI/Physics engine to control the environment. The simple AI of other spaceship simply moves a step towards you.

You might chose to use 'bullet' but no marks will be lost if you chose not to. In fact using bullet may increase the complexity of your code to a point where you fail to meet the goals and lose marks. If you score 24/25 you may be awarded the bullet mark as well.

There is no need to have a realistic behaviour. You could pre-compute the trajectory of all objects. There is no need for the trajectory to be accurate.

Key Functions

Key, tour, normal or demo, optional game.

Key	Function
<ESC>, Q, q	Exit the program, (all Modes).
P, p	Move to predefined location where screen shot was taken, (all Modes).
L, l, O, o, M, m (optional)	Switch to alternative view point 1, (all Modes). M returns to default view.
G,g	Start Game mode.
T, t	Start the tour, ignoring all key presses except T, t, E, e, Q, q and <ESC> T pauses the tour, T again restarts it.
E, e (optional)	Exit the tour or game mode otherwise ignored.
<LEFT>	Turn camera view to the left.
<RIGHT>	To the right.

<PAGE UP>	The view pitches up
<PAGE DOWN>	The view pitches down
<UP>	Increase the forward speed of the camera.
<DOWN>	Decrease or make the camera moves more slowly. If the speed decreases to zero it should stay in the same place. The camera/tank not have a reverse mode unless it turns through 180 degrees.
<SPACE> (optional)	Fire the gun (game only)
R, r (optional)	Reset all animation
Key functions (mandatory). They must be case insensitive	

The basic user interface to control the camera must use the arrow keys, LEFT and RIGHT to turn, UP to speed up and DOWN to slow down. If not in game mode then optionally the SPACE bar can be used as a brake. PAGE UP/DOWN may be used to control the tilt of the camera. The camera position must respond to P, L and M. T should start the tour which should run until finished unless E is implemented. Other keys may be used as long as they do not change/replace the above.

The letter commands should be case insensitive.

The user or point of view must occupy physical space in the world, and should normally not be allowed to penetrate the walls or static objects. Collision detection and interaction with moving objects is not required. You might like to think of the viewer as a camera equipped oil drum. Of course if you use B*ullet* you get this.

Any other interaction is optional and you will not gain extra credit for it.

In game mode where we are controlling a space ship <LEFT>, <RIGHT>, <PAGE UP> and <PADE DOWN> act like thrusters rotating the spaceship about its centre of mass. The <UP> and <DOWN> keys control the amount of thrust applied along the long axis of the space ship. So to move in a new direction one first rotates the space craft so one is pointing in the desired direction then <UP> applies units of thrust while <DOWN> reduces it. When no thrust applied the spacecraft continues to move as per Newton’s laws.

Deliverables.

A single zip file containing

- A configuration fie – ‘*config.lua*’. If this is not I the correct format you will be asked to resubmit again.
- All source code and data files for the project. The automatic evaluation procedure will compile all “.cpp” files it finds and combine them into a single application. These <u>must</u> reside in the root directory.
- Shader/texture files may reside in the root directory. 1 mark lost if not found and I have to *fix* your submission.
- Text files ‘credits.txt’ ‘*readme.txt’* (1 pages each maximum).
- ‘*readme.txt’* details:
    - How to operate the program(s) i.e.: which keys do what, and how the mouse is to be used. No need to duplicate information in this document
    - A list of all your files with brief comments on their contents.
    - A brief description of how to build the program on your chosen platform.
    - A very brief description of how your program works.
- The optional file ‘*credits.txt’* must list the source of all models, external code and pictures that you have used, if any. You can chose to include this information in your source code
- These text files should be included in your submission, be formatted as simple text. Word or other word processor documents will not be read. In themselves they are not worth any marks, so don’t go overboard. However their absence or if the submission does not contain the above will lose you 1 mark.
- One screen dump of your world, in jpg format, this should show your world to best advantage. Its absence will lose you a mark. The file must be called ‘*screenshot.jpg*’.

The screen shot must be in jpeg format and must be called ‘*screenshot.jpg*’ and the descriptive text file must be called ‘*readme.txt*’, and must be plain text without any formatting. You should also provide a file called *‘credits.txt’*. This should contain lists of any external pictures, models and code giving details of where you found them. When your code is inspired by an external source a simple comment line will suffice. The marking system will not look for files with other names.

Finally, please use jpeg images for texture, and use sensible sizes. In the past some submitted textures 4 or 5 times greater than my screen size. These did not run on my test machine.

Marking

Penalties

If your submission will not compile or run because of missing files then you will be asked to provide the files and a 5 mark penalty may be applied.

In the past some people have captured the mouse. If this happens this year, a 5 mark penalty **WILL** be applied. If you don't know how to stop this happening then don't use the mouse.

Leaving out the screenshot, readme or credits file will result in a one mark penalty per item. These must be in the root directory as the penalty is applied automatically.

If the 'config.lua' file is not provided then your coursework will not be compiled and not be run. There is a generic 'config.lua' file on the coursework web pages. This will score zero marks.

Marks will be awarded for achieving the above goals, handing in all the required elements, technical merit and artistic interpretation. A possible mark scheme is as follows.[3]

Information on the hand in system takes priority, and should be regarded as correct.

Activity	**Note**	**Mark out of 100/25**
User Control	Based on the ease of controlling the viewpoint.	8/2
The tour	How interesting it is?	8/2
Static Complexity	Based on appearance of static landscape.	12/3
Dynamic Complexity	Depends on the detail in the animation. i.e. more than lots of things bobbing up/down	16/4
Technical Difficulty	Use of OpenGL features, not necessarily advanced ones or a complex scene. This is not a check list of techniques used.	28/7
Bullet[3]	Using the Bullet Physics engine in a simple way to manage collisions and motion.	4/1
Artistic merit	This is very subjective, and it is why the coursework is double marked.[4]	24/6

[3] If you achieve 24/25 without using Bullet then this mark may be awarded.

To calibrate the lower bound to the marks, consider that all of the marks above are based on added value with respect to Coursework 1. If you submitted your coursework 1 submission or something equivalent in action and appearance you would score zero.

This coursework will be double marked and the average mark used. In particular 'Technical Difficulty' and 'Artistic Merit' are very subjective.

Game play will count to no more than 50% of the artistic mark, i.e. 3 out of 6. In the absence of game mode this will be given for structural elements of the scene.

Submission

Please submit your program in a zip file as specified above, to the C-Bass system. There is no penalty for multiple submissions.

Relevant Learning Outcomes (LOs)

1. Show that you can go beyond simple concepts and generate a complex animation.
2. Demonstrate a familiarity with the capabilities of OpenGL and what can be achieved with them. This does not mean use every advanced trick. Simple is often better
3. Show an understanding of advanced concepts in Computer Graphics.
4. Demonstrate some creativity in content.
5. If game mode, some thought as to best interaction within rules.

Marking Scheme

See above.

Late submissions will be penalised as per Faculty/University rules.

No work can be accepted after feedback has been given.

To achieve a good mark you should expect to spend up to 25 hours on this assignment.

Make sure the marks are worth the time spent on the project

Please note the University regulations regarding academic integrity.

[4] You will score better marks using your own models rather than downloading them from the internet. Consider using blender to create them.