The University of Melbourne
Department of Computing and Information Systems
COMP30019 Graphics and Interaction

Set: Tuesday 6th September

Project 2, 2016

Electronic Submission 1 (proposal): 4 pm, Friday 16 September
Electronic Submission 2 (project): 8am, Monday 17 October
Demonstration: Week commencing 17 October
Final Electronic Submission 3 (project): 4 pm, Friday 21 October
Marks: This project counts towards 30% of the marks for this subject

Participants: This can be done either individually or in groups of 2-3 people.
Assessment: Marking will be the same regardless of the number of participants.

Note we are asking you to submit the project twice, once *before* the demo and once *afterwards* in case you need to fix something up that you found out during the demo.

Aim

The purpose of the project is to expose you to user interfaces and three dimensional graphics programming. You will develop a **Windows 10 App** to submit to the Windows Store on the tablet computers. Since Unity is designed for multi-platform support, you might also like to make your App compatible with other platforms, such as IOS or Android, but your assessment will be based on the Windows Store version.

Choosing/creating an App

You are free to choose any application that your group would like to develop, although you will be required to submit a *proposal* to enable us to provide feedback and authorise your project to assist you in achieving your goal.

Your Task

Your task will involve questions of

1. how to facilitate user interaction,
2. how to create and render objects/entities,
3. how to elegantly provide a camera which sensibly displays the action,

4. how to effectively manage direct X/3D so that it runs without substantial lag.

Possible projects

We are happy to discuss possible projects that you might choose to adopt and adapt to your App. Since the (natural) user interface version of your tablet application or game might differ significantly from traditional graphical user interface (GUI) form, you are free to adapt your App to suit.

In particular, if you choose an existing or known application you are free to completely change the operation or interaction modes required to achieve natural user interface operation.

Specification and Marking Criteria

As stated above, we have provided you with a great deal of freedom in what you make. In particular, you are not necessarily required to implement *all* of the entities and functions present in your intended application/game if you choose not to. For example, you may choose to only utilise a core subset of functionality. However, the App must still be usable/playable.

An App that meets all of below will receive 30 marks:

Gameplay [8 marks]

- Controls are well specified and respond as expected [2 marks]
- The game has a clearly defined objective, with the player being able to progress towards and achieve that objective [2 marks]
- Gameplay is well executed, bug-free, and operates at a reasonable frame rate (sufficient to play the game) [2 marks]
- The gameplay and control scheme are polished, easy to use, enjoyable and suit the design of the game [2 marks]

Computer graphics & App Compliance [6 marks]

- Objects and entities clearly visible, clearly distinguishable, and suit the style of the game (e.g., different shapes, colours and lighting) [2 marks]
- Correct three dimensional transformations for object/camera motion [3 marks]
- An App certification test has been conducted, showing no errors. The compliance report has been attached [1 mark]

Lighting & Shading Calculations [11 marks]

- At least two clearly distinct custom Cg/HLSL shaders are used which appropriately enhance the game's visuals. At least one shader should produce an effect *not* explored in the labs. For example, cel shading, fog, water effects, or another artistic effect as desired. Descriptions of how the shaders work are clearly detailed in the readme file.[8 marks]
- A particle system or geometry shader is used to create effects within your game [2 marks]
- Shadow volumes are implemented to allow at least some objects to cast shadows [1 mark]

User interface using touch/tablet orientation [5 marks]

- Touch sensing is used to control at least some elements of the game [1 mark]
- At least one other sensor (e.g. accelerometer) is used to control some elements of the game [1 mark]
- Interaction management is well handled, with controls behaving reasonable and as expected [3 marks]

Please note that it is not necessary for the lighting and shading effects described above to be implemented for *all* objects within your game. It is, however, required that they be clearly visible within the game and well documented within your readme file.

Consultation via Discussion Forum

You are encouraged to ask questions, answer questions where possible and share examples of pseudocode and/or small examples of code that highlight the correct invocation of Unity commands or algorithmic/graphical/interaction techniques.

You are not allowed to exchange complete methods or classes. Remember that copying code from the Internet or from your colleagues will be considered cheating. Note that via electronic submission, your code will be checked for similarity between submissions and with code available over the Internet.

Proposal & registration (electronic submission 1)

You are required to submit a proposal which details the project that your group aims to accomplish. Submitting a written proposal should help you clarify your goals in this subject and manage your time more effectively. Having a written proposal will also help us ensure that your attempts are broadly in line with the themes of the subject, and flag potential problems or challenges early in the process. **After submission of your proposal, we will read through it and provide feedback**. If necessary, we may suggest changes to the project. You should aim to cover the following points in your proposal:

Type of application

- Briefly identify the genre of application (e.g. first-person shooter game, stellar cartography visualisation, etc.)
- Don't forget to register at https://www.dreamspark.com/Student/Windows-Store-Access.aspx if you intend to submit your application to the Windows Store (marketplace).

Hardware inputs

- Indicate how you intend to use input from the device in your application. Recall the Surface Pros have touch input and accelerometers (and also a light sensor and a compass).

Visualisation

- Broadly describe the visualisation of the application when it is running. Consider 3-D graphics, the camera, lighting, visual effects, and an estimate of the number of polygons in the scene.

Milestones

- Establish a set of intermediate goals for your application (e.g. detailing the polygon mesh, camera transformations, handling touch input, etc.) Break down the set into core functions and extras which you can attempt if time permits.

Submission of the proposal

One member of the group must submit the proposal electronically to the LMS, in a plain text format. There should also be a plain text file listing your group members' logins and student numbers, one group member per line.

You must create a .zip archive of your proposal text file `proposal.txt` and group file `group.txt` and submit to the LMS by the due date.

The `group.txt` file must (strictly) include a list of lowercase login names and student ids, one per line for all people in your Project group (or just one login for individuals). For example,

```
>cat group.txt
>login1 studentid1
>login2 studentid2
```

App project submission (electronic submission 3)

Your code must compile and run on the provided hardware and software configuration.

You must use an online git repository to store all your source code required to run your programs, readme.txt, certification report and link to a gameplay video (all described below). **Please use a .gitignore so that only essential files are tracked.** Comprehensive .gitignores for Unity are readily available online. You will need to submit a link to the repository via the LMS by the due date. After the due date, you should not make any further commits to the repository to avoid late penalties.

Readme File: You must include a `readme.txt` file which describes your application, specifically what it does and how to use it in *two pages* of plain text. Several paragraphs of text under each of the following headings should be sufficient:

1. What the application does,
2. How to use it (especially the user interface aspects)
3. How you modelled objects and entities,
4. How you handled graphics and camera motion, and
5. A statement about any code/APIs you have sourced/used from the internet that is not your own.

Brief descriptions are sufficient and concise descriptions are preferred over long, verbose descriptions.

Certification Report: Whether or not you intend to submit your project to the store, you must attach an app certification report produced by the Windows App Certification Kit. This report verifies that your app meets the required standards for submission to the store. You can download the Windows App Certification Kit at https://developer.microsoft.com/en-us/windows/develop/app-certification-kit.

Gameplay Video: In order to properly assess your project in action, we request that you submit a short gameplay video to YouTube demonstrating the key features of your game. You can set the video to unlisted if you wish to prevent it from being viewed publicly, but please ensure that we can access it from the provided link without requiring any special permissions. Your git repository should contain a clearly labelled link to this video.

Important: if your project contains code from other sources, in particular from other web sites, you have to clearly indicate this in `readme.txt`. Identify which classes or methods are your own and which are from a different source. Remember that copying code from the Internet or from your

colleagues will be considered cheating. We will be checking for similarity between submissions and with code available over the Internet.

Submission of project to Windows Store

If you would like to submit your application to the Windows Store (marketplace) please contact Alex Zable, Chris Ewin or Adrian Pearce.

You will need to register with *https://www.dreamspark.com/Student/Windows-Store-Access.aspx* to receive free access to the Windows Store.