University of St Andrews

General information: For your benefit please take note of the following information:

- Assignment title: 3D Rendering.
- Submission deadline: 9pm (2100) on the 26th April 2019.
- Assignment weight: 60% of the coursework (which itself contributes 40% to the total module grade, the remaining 60% being covered by the exam).
- Lateness penalty: Scheme B, 1 mark per 8-hour period, or part thereof.
- Required submission content: a single file report (in PDF format), a standalone Java application, and the full source code.
- Mark descriptors: `https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html`
- Good academic practice: `https://www.st-andrews.ac.uk/students/rules/academicpractice/`

Aims: The aim of this practical is to help you understand the key principles behind various techniques frequently used for the rendering of 3D objects, and give you hands-on experience with their implementation and manipulation.

Basic specification (up to 16 marks): Your task involves the creation of a Java application which facilitates interactive modelling of faces in 3D.

All input data you need can be found zipped in the archive `CS4102_2019_P2_data.zip`. The archive contains files named `sh_xxx.csv` and `tx_xxx.csv` (where '`xxx`' stands for a zero-padded integer between 0 and 199), `sh_EV.csv`, `tx_EV.csv`, and `mesh.csv`. Each line in `mesh.csv` corresponds to a triangle in a mesh used to represent a face; specifically, each line contains three integer indices. These index the 3D coordinates and colours of face mesh vertices in respectively `sh_xxx.csv` and `tx_xxx.csv`.

Your application should start by drawing a triangle in the main window. Each of the corners of this triangle correspond to a rendered face. The n-th (of 3) of these will have the 3D coordinates of its vertices computed by adding the coordinates in `sh_00.csv` (the average face shape) summed with the coordinate offsets in `sh_00n.csv` multiplied by the n-th weight in `sh_EV.csv`. The corresponding colours are similarly computed by adding the colours in `tx_000.csv` (the average face colour) summed with the colour offsets in `tx_00n.csv` multiplied by the n-th weight in `tx_EV.csv`.

The interactive design performed by the user is simple: a click within the screen area communicates depending on how similar (close) the synthetic face is desired to be from the three reference faces. The synthetic face is generated by interpolating 3D shape and colour between the three reference faces, and is displayed on the side (or in a different window if more convenient).

You can use flat shading and orthographic projection, and assume that faces are perfectly matte with a unity diffuse coefficient and that there is a single directional light source aligned with the viewing

direction. You are free to use any library you wish for basic mathematical operations but you must implement all relevant computer graphics techniques from scratch.

Advanced specification (up to 18 marks): For additional marks, there are several extensions which you may wish to consider (you are also free to come up with your own). For example, you may extend your application to allow the user to interpolate between possibly all 199 possible reference faces by replacing one or more of the initial reference faces. Note that the contribution of any replaced reference faces should not be lost but rather kept unchanged while other contributions are added. Consider allowing the user to preview the possible reference faces before making the choice.

Highly advanced specification (up to 20 marks): Assuming that you have successfully met the advanced specification requirements, you may wish to add further interactive elements such as the ability to rotate faces, change the direction or the number of light sources, the shading model, etc.

Hints: The following suggestions should help you prevent common mistakes, and save time and effort:

- Make sure that your submission is complete i.e. that the application can be executed on different machines without the need for tinkering by the marker. It is not reasonable to expect the marker to debug and fix your code (e.g. hard-coded paths).
- In your report, focus on the quality of content. Do not be overly verbose – well formed, succinct explanations are easier to read and more convincing than convoluted and excessively long verbiage. Aim for up to two pages of text but feel free to include images or screenshots to complement this content and illustrate your work better.
- If you are interested in learning more about the model use here, you may find it useful to have a look at the following paper: `https://gravis.dmi.unibas.ch/publications/Sigg99/morphmod2.pdf`.

Ognjen Arandjelović (Oggie)
`ognjen.arandjelovic@gmail.com`