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 15th May 2020.
- Assignment weight: 70% of the *module*.
- 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 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.

Specification: Your task involves the creation of an application which facilitates interactive modelling of faces in 3D.

All input data you need can be found zipped in the archive `CS4102_2020_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_000.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) using Painter's algorithm (4 marks).

The user should be able to select flat or Gouraud shading (3 marks). Similarly, there should be a choice of orthographic or perspective projection, with the focal length parameter of the latter also adjustable

by the user and expressed as relative to the distance of the camera from the face (4 marks). Compare visually (2 marks) how the shading and projection choices affect the output, and then analyse the same question quantitatively by measuring the difference in the resulting renderings (3 marks). Specifically, if you are comparing two rendered colour images represented by matrices $\mathbf{I}_1, \mathbf{I}_2 \in \mathcal{R}^{h \times w \times 3}$, their appearance difference can be quantified as follows:

$$d(\mathbf{I}_1, \mathbf{I}_2) = \frac{1}{3hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{3} |\mathbf{I}_1(i,j,k) - \mathbf{I}_2(i,j,k)|^2 \tag{1}$$

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 and low level graphics operations but you must implement all relevant computer graphics techniques covered in the lectures from scratch.

For the final marks:

- implement the light attention model covered in the lectures, again allowing the user to set and change its three free parameters (2 marks), and
- include Phong's specularity term in the reflectance model (2 marks).

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`