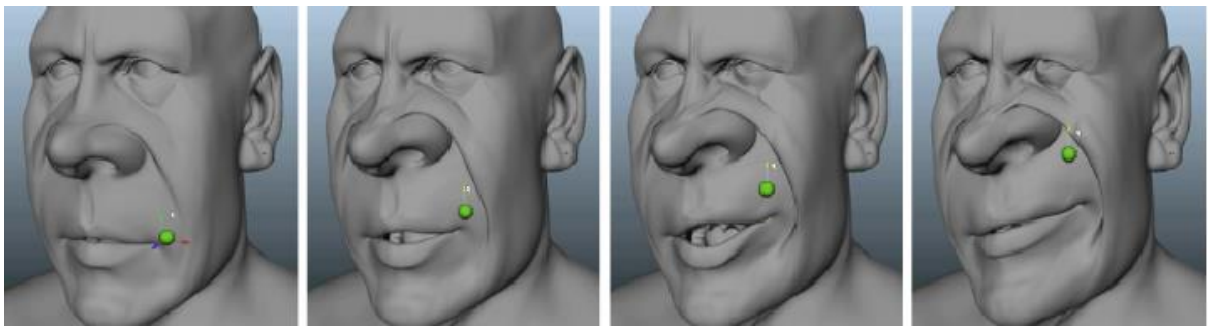
Assignment #03: Direct Manipulation Blendshapes

Due Date: 18th March

Coursework %: approximately 20%

The purpose of this lab is to implement blendshape based facial animation. Blendshape activation can be either controlled by setting blendshape weights or by dragging manipulation controllers.



- This assignment is strictly **individual** (no groupwork).

1. You are required to create a project that can display a 3D facial model, and has a picking function, which allows you to select vertices on the mesh, using the mouse. Picking can be implemented using back-projection in combination with a linear search of the closest vertex. Following this, you should add the functionality for a manipulator to be created after picking a vertex of the mesh. If a manipulator already exists for the vertex, the user should be able to select and move the manipulator in space.

2. You are also required to create a loading routine that loads a set of facial expressions. The facial expressions should be saved internally as delta-blendshapes together with the neutral expression. Implement two additional methods:
   - Compute a facial expression based on the neutral face, the delta-blendshapes and the blendshape weights.
   - Compute blendshape weights by solving a least squares problem based on the positions of the manipulators and the delta-blendshapes. Update afterwards the facial expression as well as the positions of the manipulators.

- You must use shader-based OpenGL with C++ and any freely available blendshape model, as long as it has at least 10 blendshapes.
  - You can use any external library for GUI, model loading, linear algebra and solving the least square problem. We recommend choosing one of the following options:
    i. A combination of GLWF (OpenGL), Anttweakbar (UI) and GLM/Eigen (linear algebra) allow you to build a lightweight 3D viewer. (Eigen will be strongly recommended for the second part of the exercise to solve the linear system but converting vectors from Qt/GLM to Eigen is straightforward).
    ii. Qt LGPL version (4GB download) is a powerful cross-platform framework including OpenGL wrappers, linear algebra functions and UI interfaces. Starting with one of the OpenGL examples and build on top of that might be the best way to manage the complexity of the framework.

- We strongly recommend using Eigen for solving the linear system and using the provided blendshape models (©https://www.meryproject.com/). You will be supplied (on Blackboard) with:
    i. code snippets to help accessing the Eigen library
    ii. low-res blendshape model (99 vertices) for debugging your application
    iii. a high-res blendshape model (ca. 5000 vertices) for final testing
    iv. a file defining the blendshape weights frame by frame for the extra part

- You will be required to submit your **report**, **code** and **youtube video link** on Blackboard by the due date. Your submission should include a pdf report with a short written description and screen shots (same format as before), along with the accompanying zipped code (cpp and h files used). Please submit report and code as separate files, and do not submit the entire Visual Studio project. If you fail to show up for the lab or to submit your report on time, you will be reported as absent and will receive a grade of 0%.

- Be aware that demonstrating a project that was not created by you is considered **cheating** and will be reported as such. The demonstrator will check if you have an understanding of the code that you have written.

Examination

Your program should have the following features:

- An interface for creating, selecting and moving vertex manipulators (~20%)
- A user interface to set blendshape weights (~10%)
- Load facial expressions, save them internally as delta-blendshapes and update a facial expression after blendshape weights have been manipulated (~20%)
- Compute blendshape weights from manipulators (~30%)
- Extra marks for animation playback (~20%)

This exercise is strongly based on the following publication:

Direct-Manipulation Blendshapes, by J.P. Lewis and Ken Anjyo
http://scribblethink.org/Work/DirectManipBlendshapes/DMBpreprint.pdf

Note: The [approximate] marking scheme provided shows the maximum marks that can be obtained for each section if completed perfectly. Merely attempting a section does not imply the full score indicated.