



Group Assignment 2

Deadline: **February 25th, 2021**

In this assignment, we will look into the computer graphics pipeline. In particular, generating meshes, creating shaders and achieving visual coherence.

1. Mesh generation

PREPARATION

- Download simple unlit shader from the Assignment page on Blackboard.

TASKS

- a) **3pt** Create a new scene. Create a spaceship of any shape by building a mesh from scratch. The spaceship has to consist of at least 16 different triangles. Make sure to include some basic type of wings on both sides.

- Access the [Mesh class](#) to generate your meshes.
- You can use `mesh.RecalculateNormals()` to easily set the normals of a mesh after setting vertices and indices. Be careful with reusing vertices (i.e., indexing the same vertices multiple times), because sharing normals between triangles is not always desired. For instance, you can create vertices with the same position, but different normals to create sharp edges.

- b) **2pt** Use the simple unlit shader (vertex color shader using CG PROGRAM) that renders the spaceship, including vertex color.

Explain: How does your shader work? Explain the declarations and functions in the CG PROGRAM.

- c) **2pt** Let the spaceship fly between two image targets (of your own choice) by interpolating its position between the targets. As it flies between the targets, animate the wings of the spaceship (or any other part that makes sense) by changing parts of the mesh. The moving mesh parts must continuously transition between at least two states.

- Remember to clear your mesh before updating it every frame.
- d) **BONUS 2pt** Make parts of the spaceship blink by changing the vertex colors.
- Change the color programmatically in a script, *not* in the shader code. This can be changed in the color array, which is a property of the [Mesh class](#).

Min / Total = **3.5pt / 9.0pt**

2. Visual coherence

PREPARATION

- You will need to use a 3D model for this part of the assignment. Use the penguin model, which can be downloaded from the Assignments page on Blackboard.
- Use two image targets, one for the penguin model and one other image target (e.g., beer).
- Download the phantomShaderBasis for implementing the phantom technique from the Assignments page on BlackBoard.
- You will be using OpenCV, a computer vision library. Download *OpenCVForUnity*, a Unity port of OpenCV, and *MatDisplay* from the Assignments page on Blackboard.
- To get *OpenCVForUnity* to work with your Unity project you have to do the following: On Windows and Mac place *OpenCVForUnity* and *MatDisplay* in the Assets folder.

TASKS

- a) **3pt** Create a new scene and add the penguin model to its respective image target. Add a quad to the second image target and add the PhantomShaderBasis to this quad. Implement the phantom technique (using [ColorMask](#)) to let an image target correctly occlude virtual objects by adapting the PhantomShaderBasis. Observe the phantom technique in action by moving the quad image target in front of the Penguin. You should see the penguin getting occluded by the quad image target.

Explain: How does your phantom shader work? Explain the declarations and functions in your **CG PROGRAM**.

- b) **BONUS 2pt** Adapt the code in the phantom shader. Allow for any 2D shape (i.e., cutouts) to occlude virtual objects correctly. The occlusion mask can be given as a black and white texture (e.g., the beer mask in the auxiliary material on BlackBoard). Use `discard` in your shader to omit pixels.
- c) **2pt**

Explain: What is the data structure of a depth buffer and how it is used in the CG pipeline?

- d) **3pt** Create a new scene and again add the penguin model to its respective image target. Implement a stylization of the scene: Draw the contours of real objects in the environment in the camera video feed. Create a shader that renders the contour of the penguin model by computing the dot product between the normals of the triangle faces and the camera view direction.

TIPS

- For the video feed, the `Imgproc.adaptiveThreshold` OpenCV function can be used for simple stylization (please note that this is generally not the purpose of adaptive thresholding).

Min / Total = **4.0pt / 10.0pt**

3. Paper Analysis [Individual 10-minute task]

TASKS

- a) **BONUS 1pt** Download the paper you were assigned for the paper analysis (see [list of assigned papers](#)). Spend a few minutes reading your paper - at least look at the title, abstract, and the figures to get a first idea of what this paper is about. Before the assessment session, make a brief post on Slack in the "paper-analysis" channel: in 1-2 sentences briefly explain the main topic and contribution of the paper. DO NOT simply copy the title or a sentence from your abstract - you should write this in your own words.

Min / Total = **0.0pt / 1.0pt**