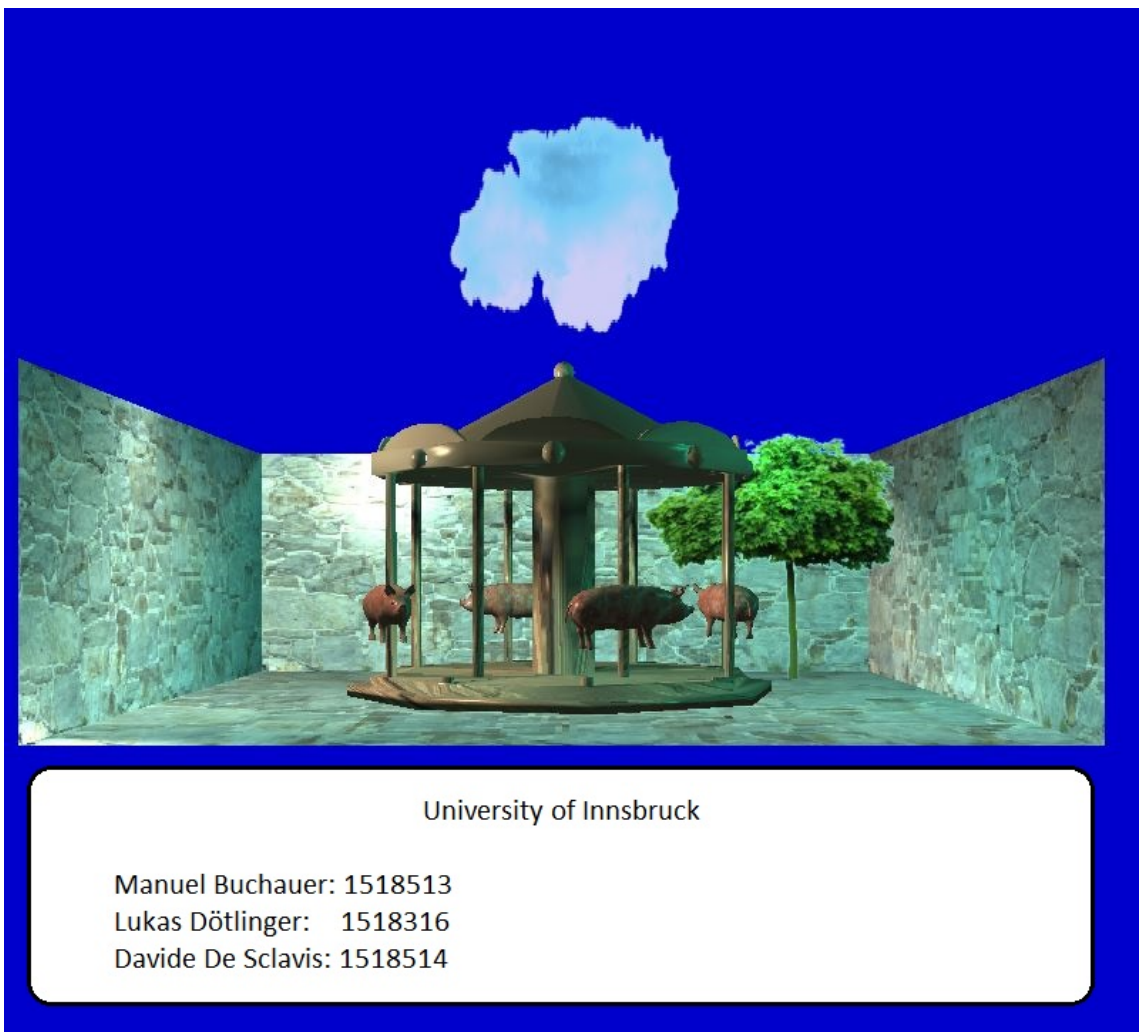


# Project Summary

Proseminar Computer Graphics  
SS 2017



University of Innsbruck

Manuel Buchauer: 1518513  
Lukas Döttinger: 1518316  
Davide De Sclavis: 1518514

## 1 Outline

The goal of this project during the semester was to create a shader based OpenGL program showing an animated merry-go-round with several effects. We were given 5 programming assignments, which we had to implement during the semester.

Find details about controls and usage in our **README.md** file.

## 2 General structure

Due to the large amount of code we needed to implement the individual programming assignments we decided on outsourcing some parts of our code into different header-files with their respective c-files. Otherwise our main-file would have been grown to some thousands of lines of code.

The **Carousel.c** is our main file including the initialisation of OpenGL and GLEW. This file also combines functions from other files and includes our main-function.

In the **models-folder** we have stored our objects as for example the pigs and the merry-go-round which we load with our **OBJ-Parser.c**. The **textures-folder** stores our textures as for example for the walls and the pigs which are loaded with the functions in the **LoadShader.c**.

Other important functions are to be found in **Setup.c**. For loading the billboard textures we used another picture loader, **stb\_image.h**, since the given picture-loader can't load images with transparent parts.

## 3 First assignment

The goal of the first assignment was to create a 3D model of a merry-go-round and add some animation to it. To create the model of the merry-go-round we calculated the vertices on a sheet of paper and then transferred it into our program. Within the second assignment we changed that and loaded our merry-go-round model with the **OBJ-Parser** to save some lines in our main program. More informations at *Second assignment*. As animation we decided for a rotation. To add that rotation to the merry-go-round we created a matrix using the given method **SetRotationY()** and multiplied it with our transformation matrix. We individually created transformation matrices for every model, for example the pigs with their sliding animation.

## 4 Second assignment

In the second assignment the first task was to improve the model of the merry-go-round. We decided to construct a new merry-go-round obj file with the help of <https://www.tinkercad.com/> and load it with the **OBJ-Parser.c**.

The second and third tasks were to add camera modes. So we first implemented a function called **Keyboard()** and a function called **Mouse()** for the user-interaction. In our **OnIdle()** we set the rotations and translations for the *ViewMatrix* as we multiplied the view matrix with the **RotationMatrixAnimView** and the **TranslationMatrixView**.

## 5 Third assignment

For the third assignment multiple light sources had to be implemented. First we defined the position of the light sources and its colors. In the **Display()** function the data of the light sources have to be passed to the shader. For the keyboard control of the light sources the **Keyboard()** function is used where the different light sources can be switched off and on, modes can be changed (for example into ambient) and hue and color of the light sources can be changed. To change the hue and value of the colors also the **hsv2rgb()** function is used, where the hsv values, edited by the user, are converted into suitable rgb values. In the **onIdle()** function another matrix multiplication is used to create the rotating light source. We also implemented a full functioning Phong-Shader as described in the lecture. To make this possible we first needed to calculate the vertex normals. We wrote a function to do this, to find in **Setup.c**, since most OBJ-files we came across had wrong normals. We then use those to calculate the color for every vertex with the shading effect in our **fragment Shader**.

## 6 Fourth assignment

The fourth assignment was about textures. So the first task was to add textures to our scene. To make this possible for all obj files, we implemented functions in the **Setup.c** to match the right vertex coordinates to the texture coordinates, since the **OBJParser.c** can't do this. Without those functions, we would not be able to display textures correctly. We added textures to the room and to our pigs. The textures are loaded with the functions in the **LoadTexture.c**. We have also implemented two billboards in our scene. One cloud, which hovers over the room and a tree to clarify the using of billboards. In our **OnIdle()** we give the billboard the necessary rotations

and translations as needed for the task. This leads to the effect that the billboards are always facing to the camera view.

## 7 Fifth assignment

The fifth assignment was about the finalization of our scene. The task was to include at least one advanced visual effect. We decided to implement fog in our scene. For the implementation we used our shader in combination with the fog blending from the lecture. We also modified the the **Keyboard()** function to enable or disable the fog effect.