Prof. Dr. Leif Kobbelt
J. Born, M. Ibing,
J. Möbius, P. Schmidt

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Basic Techniques in Computer Graphics

Assignment 2

Date Published: November 7th 2017, Date Due: November 14th 2017

- All assignments (programming and text) have to be done in teams of 3–4 students. Teams with less than 3 or more than 4 students will receive no points.
- Hand in **one solution per team per assignment**.
- Every team must work independently. Teams with identical solutions will receive no points.
- Solutions are due 18:00 on November 14th 2017. Late submissions will receive zero points. No exceptions!
- Instructions for **programming assignments**:
  - Download the solution template (a zip archive) through the L^{2}P course room.
  - Unzip the archive and populate the `assignmentXX/MEMBERS.txt` file. Any team member not listed in this file will not receive any points! (Also see the instructions in the file.)
  - Complete the solution.
  - Prepare a new zip archive containing your solution. It must contain exactly those files that you changed. **Only change those files you are explicitly asked to change in the task description.** The directory layout must be the same as in the archive you downloaded. (At the very least it must contain the `assignmentXX/MEMBERS.txt`.)
  - Upload your zip archive through the L^{2}P before the deadline.
  - Your solution must compile and run correctly **on our lab computers** using the exact same `Makefile` provided to you. If it does not, you will receive no points.
- Instructions for **text assignments**:
  - Prepare your solutions on paper.
  - If you write your solution by hand, write neatly! Anything we cannot decipher will receive zero points. No exceptions!
  - If you hand in more than one sheet, staple your sheets together. (No paper clips!)
  - Put the names and matriculation numbers of all team members onto every sheet.
  - Unless explicitly asked otherwise, always justify your answer.
  - Be concise!
  - Put your solution into the designated drop box at our chair before the deadline. (1st floor, E3 building.)

Prof. Dr. Leif Kobbelt
J. Born, M. Ibing,
J. Möbius, P. Schmidt

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 1 Implicit & Parametric Representations [2 Points]

In the lecture you were introduced to parametric and implicit representations of objects such as lines and planes. In this exercise you should derive both representations for the surface of a sphere with center $c \in \mathbb{R}^3$ and radius $r \in \mathbb{R}$.

(a) [0.5 Points]

Derive a parametric representation for the surface of a sphere described above.

(b) [0.5 Points]

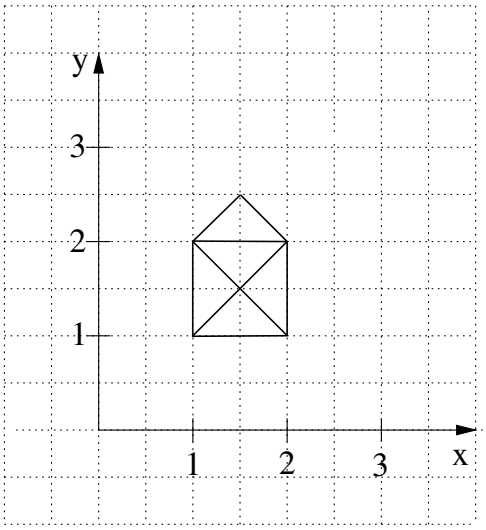
Derive an implicit representation for the surface of a sphere described above.

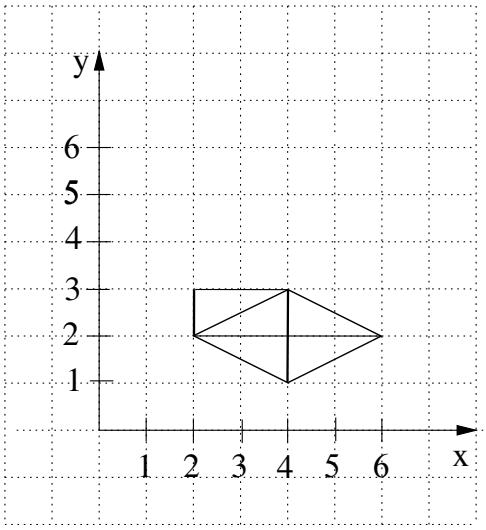
(c) [1 Point]

For which use cases would you prefer using the parametric representation? When would you rather use the implicit representation?

Hint: Remember that deriving a representations implies explaining why your formula does indeed describe a sphere. As in all the following assignments and exercises, you will not get any points for simply writing down a formula without any explanation (except we explicitly ask you to only specify the formula)! Also, don't forget to explain your answer for part (c).

Exercise 2 Linear & Affine Transformations [3 Points]





(a) [0.5 Points per matrix]

Derive the four transformation matrices (translation, rotation, scaling and translation) that transform the house depicted in the left image to the house depicted in the right image. Also specify the final transformation matrix. Remember to use extended coordinates! By convention, points are multiplied from the right side!

(b) [1 Point]

Specify how the standard basis and the origin are transformed by this mapping.

Hint: Remember that $\sin 45° = \cos 45° = \sin 135° = -\cos 135° = 1/\sqrt{2}$.

Prof. Dr. Leif Kobbelt
J. Born, M. Ibing,
J. Möbius, P. Schmidt

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 3 Programming: Transformations [5 Points]

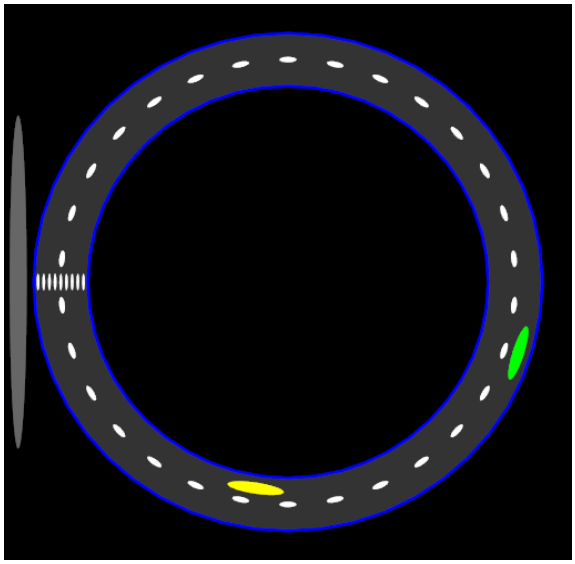
In this hands-on exercise, your task is to write a computer program simulating a simple race track. We provide you with a function that draws a circle in an arbitrary color, all you have to do is place different transformed circled into the scene.

The draw routine is implemented in `assignment.cpp`. Only make modifications to that file! Do *not* modify any other file in the folder!

Each time the draw routine is called, you will have to compute the respective transformation matrices *by hand* (GLM does already offer function to create translation, rotation etc. matrices but you are *not allowed* to use them in this exercise but enter the values into the matrices yourself to show that you understood how those work. You can of course write your own function to build them. You can use the build-in matrix multiplications from GLM.).

Of course, you can also use the matrix and vector classes from `glm`, e.g. `glm::mat4` or `glm::vec3` for colors.

The `drawScene()` routine gets two parameters, `scene` and `runTime`. You can neglect the first parameter as it is not used in this exercise. The second parameter holds the number of seconds passed since the first call (as floating point representation).



(a) [1 Point]

Draw the track itself as a blue circle in which you draw a grey circle which is a little smaller, than another blue one and a black in the center to only let the outline of the blue one be visible.

(b) [1 Point]

Draw a stand for the spectators on the left of the track in grey.

(c) [1 Point]

Draw a start / finish line as nine white ellipses.

(d) [1 Point]

Add a white dotted line between the two lanes as ellipses.

(e) **[1 Point]**

Add the two race cars which race around the track. The outer car should be twice as fast as the inner car. The cars race clockwise.

Notes:

- A non-transformed circle has a radius of 1.
- The local coordinate system is defined from -1 to 1.
- Be sure to follow the instructions for programming assignments on the first page! In particular, don't forget to add all team members' names and matriculation numbers into the `MEMBERS.txt` file, put all changed files (i.e. `MEMBERS.txt` and `assignment.cpp` in this assignment) into a zip archive and submit it via the L^{2}P.
- When `b` is pressed, the time passes faster, undo this by pressing `a` (useful for debugging).