Study guide 2IV60 Computer graphics 2015/2016

Jack van Wijk, TU/e
j.j.v.wijk@tue.nl

The course 2IV60 Computer Graphics provides an introduction in 2D and 3D computer graphics. After following the course, students understand the basic concepts of 2D and 3D computer graphics: the graphics pipeline, transformations, viewing, meshes, geometric modelling, shading, hidden surface algorithms, texturing. Also, they can apply this knowledge to develop interactive graphical applications, using OpenGL.

In this study guide information about the course is given.

Course Materials

The course is based on the book

Donald Hearn, M. Pauline Baker, Warren R. Carithers.

This book provides a good overview of computer graphics. This fourth edition appeared in 2011. The book uses OpenGL, an important library for 3D computer graphics. Thanks to the many examples one can directly experiment with different methods and techniques. In the lectures of 2IV60 the focus is on the theory of graphics, but also the translation into OpenGL is covered.

In a following overview is indicated which chapters have to be studied. This material can also be found in earlier editions of the book, but the numbering of chapters has changed. Also, the same material can be found in many other different books and on the web.

Furthermore, via OASE and the course website the following information is distributed:
- The slides used during the lectures;
- Exercises for home work;
- Practical assignments with additional material.

Video recordings of the lectures for 2013/2014 can be found at http://videocollege.tue.nl, search for 2IV60.

Lectures

In the lectures an introduction to computer graphics is given. The aim is to give an overview, and to elaborate on the more complex mathematical and algorithmic details. Furthermore, the corresponding OpenGL functions are described, and the more difficult homework exercises will be discussed. Finally, there is always space to ask questions about the material during the lectures.

Instructions

During instructions students can work on:
- Mondays: Homework exercises, as preparation for their exam;
- Thursdays: Practical assignments.
The instructors and student-assistants are available to help with problems and give advice. The following table gives an overview of the distribution of student groups over rooms and instructors for the instructions:

<table>
<thead>
<tr>
<th>Group</th>
<th>Mon 3+4</th>
<th>Thu 7+8</th>
<th>Instructors and student-assistants</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gemini-Zuid</td>
<td>AUD 10</td>
<td>Andrei Jalba, Paul van der Corput, Dennis Dingen, Niels Rood</td>
</tr>
<tr>
<td>2</td>
<td>Gemini-Zuid</td>
<td>MF 08</td>
<td>Huub van de Wetering, Wilco Brouwer</td>
</tr>
</tbody>
</table>

Note:
- **Students are requested to select an instruction group via OASE before Tuesday November 18.**
- Groups are limited in size. If no more room is available in your first choice, please select another one.
- Assignments should be done in pairs, forming pairs is left to students. If you have issues finding a partner, you can post requests via Forum under Discussions in OASE.
- Instructions take place on:
  - Mondays, 3rd and 4th hour, 10:45-12:30;
  - Thursdays, 7th and 8th hour, 15:45-17:30.

**Homework exercises**

Each week a set of homework exercises is distributed. Students are **strongly** encouraged to work on these to improve their understanding and to prepare for the intermediate and final exams, but no formal check is done. Students can get guidance during instructions, versions with answers will be provided at a later stage, and the more difficult questions will be discussed during the lectures.

**Intermediate Exam**

To encourage students to study the material, half way the course an intermediate exam is scheduled. This will be held after 5 weeks, on Monday December 14, 2015, during the instruction hours. The grade G₁ for the intermediate exam contributes 10% to the final score. The code for this part is 2IV63.

**Assignments**

To obtain practical experience assignments have to be done, and also for these guidance and advice is offered during the instructions. The final challenge of the practical assignments is to show a scene of robot-like figures that move around a track. The assignments should be done by pairs of students. The strict deadlines for submission are:

- Round 1: At last Monday November 30, 2015 (after 3 weeks);
- Round 2: At last Monday January, 11, 2016 (after 7 weeks).

The assignments must be done using NetBeans, Java, and JOGL. A template project is offered to make a quick start. A complete description of the assignments can be found in a separate document in OASE. For round 1 100 points can be scored, for round 2 200 points, hence in total maximally 300 points. The grade Gₐ for the assignments is calculated as

\[
Gₐ = 1 + 9 \times \frac{Pₐ}{300},
\]
where \( P_A \) is the number of points scored in total for all assignments. \( G_A \) is rounded to one decimal. The code for this part is 2IV62.

**Exam**

In the final exam students are evaluated on their knowledge (describe concept), insight (given a practical problem, which method is preferred), and skill (give a solution for a given problem). During the lectures example questions will be made available. The exam consists of four questions, where for each question 10 points can be scored, hence in total 40 points. The grade \( G_E \) for the exam is calculated as

\[
G_E = 1 + 9 \times \frac{P_E}{40},
\]

where \( P_E \) is the number of points scored for the exam. \( G_E \) is rounded to one decimal. The code for this part is 2IV61.

**Final grade**

The final grade \( G_F \) for the course is determined as follows:

\[
G_F = \begin{cases} 
0.4 \times G_A + 0.1 \times G_I + 0.5 \times G_E & \text{if } G_E \geq 5.0 \\
\min(5.0, 0.4 \times G_A + 0.1 \times G_I + 0.5 \times G_E) & \text{if } G_E < 5.0.
\end{cases}
\]

In other words, the final grade is based on the results for the assignment (40%), the intermediate exam (10%) and the final exam (50%), and with an exam grade lower than 5, the final grade is at most 5. \( G_F \) is rounded to an integer number from 1 to 10. Half points are rounded up.

**Website**

All materials will be distributed via OASE; if this gives unforeseen problems then they will also become available on the website [http://www.win.tue.nl/~vanwijk/2IV60](http://www.win.tue.nl/~vanwijk/2IV60).

**Appendix A: Topics and sections**

The following list gives an overview of the topics discussed and the corresponding parts of the book to be studied. Note that the order of treatment does not strictly follow the book.

**Introduction**

1 A Survey of Computer Graphics  
3-1 Coordinate Representations  
3- 2 Graphics Functions  
3-3 Software standards  
3-5 Introduction to OpenGL

**Basic Math**

A-1 Coordinate Reference Frames  
A-2 Points and Vectors  
A-5 Matrices
Two-Dimensional Geometric Transformations
7-1 Basic Two-Dimensional Geometric Transformations
7-2 Matrix Representations and Homogeneous Coordinates
7-3 Inverse Transformation
7-4 Two-dimensional Composite Transformations
7-5 Other Two-Dimensional Transformations
7-8 Transformations between Two-Dimensional Coordinate Systems
7-9 OpenGL Functions for Two-Dimensional Geometric Transformations
7-10 OpenGL Geometric-Transformation Programming Examples
7-11 Summary

Three-Dimensional Geometric Transformations
9-1 Three-Dimensional Translation
9-2 Three-Dimensional Rotation, without quaternions
9-3 Three-dimensional Scaling
9-4 Composite Three-Dimensional Transformations
9-5 Other Three-Dimensional Transformations
9-6 Transformations between Three-Dimensional Coordinate Systems
9-7 Affine Transformations
9-8 OpenGL Geometric-Transformation Functions
9-9 OpenGL Three-Dimensional Geometric-Transformation Programming Examples
9-10 Summary

11-1 Basic Modeling concepts
11-2 Modeling Packages
11-3 General Hierarchical Modeling Methods
11-5 Summary

Viewing
8-1 The Two-Dimensional Viewing Pipeline
8-2 The Clipping Window
8-3 Normalization and Viewport Transformations
8-4 OpenGL Two-Dimensional Viewing Functions (up to Creating a GLUT Display Window)
8-5 Clipping Algorithms
8-6 Two-Dimensional Point Clipping
8-7 Two-Dimensional Line Clipping (up to Liang-Barsky line clipping)

10-1 Overview of Three-Dimensional Viewing Concepts
10-2 The Three-Dimensional Viewing Pipeline
10-3 Three-Dimensional Viewing-Coordinate Parameters
10-4 Transformation from World to Viewing Coordinates
10-5 Projection Transformations
10-6 Orthogonal Projections
10-8 Perspective Projections (partially)
10-9 The Viewport Transformation and Three-Dimensional Screen Coordinates
10-10 OpenGL Three-Dimensional Viewing Functions
10-13 Summary

Graphics Primitives and Attributes
4-3 OpenGL Point Functions
16-2 Back-Face Detection
16-3 Depth-Buffer Method
16-6 Depth-Sorting Method
16-10 Ray-Casting Method