GHENT UNIVERSITY

# Course Specifications

From the academic year 2019-2020 up to and including the

## GPU Programming (E640063)

Due to Covid 19, the education and evaluation methods may vary from the information displayed in the schedules and course details. Any changes will be communicated on Ufora.

Course size (nominal values; actual values may depend on programme)

| Credits | Study time | Contact hrs |
|---|---|---|
| 3.0 | 90 h | 30.0 h |

Course offerings and teaching methods in academic year 2020-2021

| | | | | |
|---|---|---|---|---|
| A (semester 2) | Dutch | Kortrijk | seminar | 9.0 h |
| | | | project | 12.0 h |
| | | | lecture | 9.0 h |

Lecturers in academic year 2020-2021

| | | |
|---|---|---|
| Saenen, Ignace | TW06 | staff member |
| Lambert, Peter | TW06 | lecturer-in-charge |
| Van Wallendael, Glenn | TW06 | co-lecturer |

| Offered in the following programmes in 2020-2021 | crdts | offering |
|---|---|---|
| Bachelor of Science in Engineering Technology (main subject Electronics and ICT Engineering Technology) | 3 | A |

Teaching languages

Dutch

Keywords

GPU, C + +, Parallel Programming, Visualization, Video games

Position of the course

This course aims to deepen the principles and operation of the GPU with a view to applications in visualization, real-time rendering technology and simulations, using C+ + AMP, CUDA or OpenCL.

This objective is achieved through a project assignment:
Textures used in video games are becoming larger and larger. These textures are typically compressed to save disk space, e.g., using JPEG compression. Yet the GPU requires these textures to be in a compressed format called DXT. As a result, the game textures need to be transcoded from JPEG to DXT on the fly. The main goal of the project is to build a texture encoder which uses the massively parallel GPU to accelerate the DXT encoding steps.

Contents

Part C + +Arrays
- Pointers
- Functions
- Templates
- Classes and Objects
- Constructors
- Overloading
- Inheritance
- Interfaces
- Polymorphism
- Memry Management

Part GPUGPU and GPGPU
- Parallel Programming
- Architectures and APIs overview
- GPU programming with C++ AMP
- GPU Memory Model

- C++ AMP - CUDA - OpenCL
- Project

Initial competences

Has programming experience in C or C #, has knowledge of computer architectures and multimedia.

Final competences

1 carry out in a team assignments and report the results
2 be able to programme in C++
3 process image information using (IT) techniques and algorithms
4 know the operation of a GPU and programme them
5 use parallel programming to build (interactive) real-time multimedia applications
6 look up and process relevant technical and scientific information
7 think creatively and scientifically, judge and act

Conditions for credit contract

Access to this course unit via a credit contract is determined after successful competences assessment

Conditions for exam contract

This course unit cannot be taken via an exam contract

Teaching methods

Lecture, project, seminar

Extra information on the teaching methods

By means of lectures and exercises the student is trained to use the programming language C++ on the GPU-project.
The GPU assigment is worked out in small groups of 2 or 3 students.

Learning materials and price

- C++ slides on the electronic learning environment
- GPU slides on the electronic learning environment
- Project description on the electronic learning environment

References

- Aan de slag met C++, Gertjan Laan, Academic Service 2012, ISBN 9789039526576
- C++ Primer, Stanley B. Lippman, ISBN: 0321714113 DDC: 5 Edition: Paperback, 2012-08-13
- Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series), David B. Kirk (Author), Wen-mei W. Hwu, ISBN-10: 0123814723
- C++ AMP: Accelerated Massive Parallelism with Microsoft Visual C++, Kate Gregory, Ade Miller, ISBN-10: 0735664730
- CUDA by Example: An Introduction to General-Purpose GPU Programming, Jason Sanders, Edward Kandrot, ISBN-10: 0131387685

Course content-related study coaching

Interactive support via the electronic learning environment; guided lab exercises and project coaching; contact with teachers and lab supervisors via e-mail and personally after appointment.

Evaluation methods

continuous assessment

Examination methods in case of periodic evaluation during the first examination period

Examination methods in case of periodic evaluation during the second examination period

Examination methods in case of permanent evaluation

Assignment, skills test

Possibilities of retake in case of permanent evaluation

examination during the second examination period is possible

Extra information on the examination methods

First examination period:
NPE1: The part C + + is evaluated based on submitted programming exercises. For the GPU assignment a report, source code and presentation is required.
Second examination period:

NPE2: The part C + + is evaluated based on a series of submitted programming exercises. For the GPU assignment in a modified form a report, source code and presentation is required.

Calculation of the examination mark

- 50% of the final score is determined by the C++ part
- 50% of the final score is determined by the GPU project
- The non-periodical evaluation can be resumed in the second examinationchance.
- In order to pass the course, the student must score at least 8/20 for both C + + and GPU. If this condition is not met, a score of 10 or more is reduced to 9.