Manual Advanced Tools

Term 2.3

Module-coordinator	Yvens R. Serpa [y.reboucasserpa@saxion.nl]
Lecturers	Yvens R. Serpa [y.reboucasserpa@saxion.nl] Daniel Valente de Macedo [d.valentedemacedo@saxion.nl]
CMGT roles	**Engineer**

Course explanation

Why this module?

The industry of interactive and graphical applications is an ever-evolving one. New techniques are being developed to achieve better results, such as more realistic virtual environments or the real-time Rendering of highly complex 3D models. (Xue et al., 2019)

Basic knowledge of various tools, techniques, and technologies will allow the students to jump into modern industry solutions more efficiently and actively take their part as a CMGT professional. This knowledge will also be used in their experience through the Minor, Internship, SSS, and Graduation.

This module will cover the basics of various Tools/Technologies/Techniques (TTT) such as:

- Source Control with GIT;
- Collision Detection in 2D/3D;
- Visibility Culling;
- Embedding Languages (examples use Lua in C++);
- Forward / Deferred Render;
- Advanced Shader Techniques (Shadows, Bloom, etc.);
- Artificial Intelligence (Cost Evaluation Function, Regression Models, etc.);
- Machine Learning and Deep Learning.

While this course will not go in-depth in all subjects, it will lead the student in a comfortable position to continue his/her studies on the subject with a good starting point.

Reference

Xue, Junjie, Xiang Zhai, and Huiyang Qu. "Efficient rendering of large-scale CAD models on a GPU virtualization architecture with model geometry metrics." 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE). IEEE, 2019.

Connection other modules

It is expected that the student has followed the C++, 3D Rendering, and 3D Math modules; to be able to understand more advanced programming concepts (pointers and references) and work with OpenGL systems to display graphics on a window. Most of the contents in this module are directly linked to graphical applications, and the primary language and platform to show them will be C++ and OpenGL, backed up by a solid base of 3D Math.

It is also expected that the student has followed the Software Architecture module to understand more complex systems and be able to project a practical framework for the assignment. Since the assignment requires modularity for testing, logging, and evaluation, it is expected that the student knows how to use programming patterns as a basis for it.

The acquired knowledge will be further used in the project 4th Project and prepare the student for possible Minors, Internships, and Graduation projects based on modern interactive and graphical applications.

Besides that, it is expected that the student will use the TTT taught during the course in future projects and work in general.

Assessment

To pass the module, the student must submit and present the module assignment, achieve at least sufficient in all Rubric's criteria, and attend all its requirements. One insufficient criterion is enough to fail the student, even if the overall grade is above 5.5. The assignment submission deadline will be 1 hour before the presentation time (if the presentations start at 9, they can be uploaded until 8 in the morning). The assignment cannot be submitted afterward.

The assignment consists of developing a prototype to evaluate a specific Tool / Technology / Technique (TTT) related to CMGT. The TTT must be part of one of the selected groups mentioned below:

- Collision Detection Simulation: You will either program or use tools to evaluate different types of collisions in a system and evaluate its performance.
- Visibility Culling: You will either program or use tools to evaluate different approaches to visibility culling in a system and evaluate its performance.
- Embedding Languages: You will program a system that embeds two different languages (ideally, one of them is an interpreted language) and evaluates its performance.
- Rendering Techniques: You will either program or use tools to evaluate different rendering techniques in a system and evaluate its performance.
- Artificial Intelligence: You will either program or use tools to simulate Artificial Intelligence behavior in a system and evaluate its performance.
- Game Engines: You will either program a small Game Engine or compare various Game Engines and evaluate their performance in different scenarios.
- XR Techniques: You will either program or use tools to evaluate different scenarios using XR techniques (for instance, FPS comparison in VR with multiple objects and real-time lighting) and evaluate their performance in various methods.

1.1 Selection and Evaluation Proposal

The student is free to select any TTT in the list above and even combine them as he/she wants. After choosing a TTT, the student must write down an Evaluation Proposal consisting of one line of text explaining what the student wants to evaluate with the given TTT. The module teacher must approve both the TTT selected and the Evaluation Proposal by the end of the last week of the module (Week 8). If the student has not received approval until then, he/she is excluded from presenting the assignment during the exam weeks and must submit for a REDO.

The Evaluation Proposal is required to validate if the research topic and Prototype are sophisticated enough for the module. Simple cases (for instance, simply build a lighted scene in Unreal 4 is too simple) or Invalid cases (for example, comparing completely different TTTs without a specific purpose) will be denied.

Additionally, the student can change the TTT and Evaluation Proposal by submitting another one. New submissions abide by the same deadline as the first submissions and cannot be submitted after the end of Week 8.

1.2 Tracking Development

The development must be tracked by either text or using a version control tool. It must show the student's progress into the working version of the Prototype and prove the number of minimal work hours required for this module. Ideally, the progress will be tracked using Git and displayed through the repository's commit history.

1.3 TTT Evaluation

For the TTT Evaluation, the student must provide at least two charts/graphs that are either:

1. Comparing Two Systems with the Same Workload;
2. Comparing Two Versions of the Same System with the Same Workload; or
3. Comparing the Same System with Different Workloads.

The student is free to choose any combination of those types for the charts presented, including two of the same kind. The teacher must previously accept comparisons of other types before the end of Week 8. Each chart must have a small text explaining and analyzing the results and findings.

1.4 Prototype Assessment

The Prototype itself is also subject to Assessment and will be assessed in the aspects of code quality, framework quality, complexity, and reusability.

1.5 Additional Notes

For more information on the assignment, the student can assess the CMGT Advanced Tools - Assignment Suggestions in Blackboard. It contains examples for the TTT, suggestions for Evaluation Proposals, and Charts / Graphs analyses.

The student is allowed to use codebases from previous modules/projects, for instance, the MGE or the GXPEngine, as a base for this assignment. For example, the student can use the MGE and 3D Rendering assignments to establish the basis for his/her assignment.

However, the assignment cannot be the same code, and the final Prototype must show significant change and improvement connected to the TTT.

The presentations will be done in Week 9 or 10 of the current quarter (the specific date will be informed through Blackboard as soon as it is scheduled). The scheduling for presentations will be done via e-mail and a shared Excel file with timeslots. During the presentation, the student will also be questioned about the module's content and assignment. Answering questions incorrectly can demonstrate that the student does not have sufficient knowledge of specific criteria and may result in an insufficient grade. The questions are also a way to validate the ownership of the presented work. In the case of a non-submission, the timeslot reserved will be freed.

Assignment note for more specific information on the assignment, refer to the individual assignment documents in Blackboard.

Procedure

This module has a 2-phase procedure, the Evaluation Proposal (6.1.1) and the Prototype Presentation described in the section below.

1.1 Prototype and Presentation

The presentations will be done in Week 9 or 10 of the current quarter (the specific date will be informed through Blackboard as soon as it is scheduled). For the presentation, the student needs to show the Prototype in execution, demonstrate how the data for the evaluation was gathered, show the report (at least two charts), and explain the results.

The presentations are done individually in a 15-minute time slot, including questions. They do not require a PowerPoint presentation and can be done by merely browsing the report.

The student must submit the Prototype and report via Blackboard with all files in a .zip file. For reports in a version control system, it is advised the student downloads the repository and uploads it on Blackboard.

Not applicable for the boot camps, but serves as a general overview:

Week	Lecture/Lab	Topic(s)
3.4	Lecture 1	Adv. Tools: Introduction & Performance Analysis
3.4	Lecture 2	Adv. Tools: Collisions & Visibility Culling
3.5	Lecture 3	Adv. Tools: Framebuffers & OpenGL
3.5	Lecture 4	Adv. Tools: Light & Shadow
3.6	Lecture 5	Adv. Tools: Introduction to Artificial Intelligence
3.6	Lecture 6	Adv. Tools: Embedding Languages (Lua & C++)
3.7	Lecture 7	Adv. Tools: Unreal pt. 1
3.7	Lecture 8	Adv. Tools: Unreal pt. 2
3.8	Lecture 9	Adv. Tools: Big O Notation
3.8	Lecture 10	Adv. Tools: QA

Which resources do you need?

The student must select a programming language and a proper IDE of its computer (suggested C++ with Visual Studio / Visual Studio Code for Windows users and Visual Studio Code / CLion for Mac users). According to the TTT of choice, the student might need to borrow hardware from Saxion or use its hardware to implement, test, and evaluate the Prototype. For instance, students willing to do a prototype using AR / VR need to get the proper equipment by themselves.

What are you going to learn in this module (learning objectives)?

1. The student creates a prototype to study the performance of a TTT.
2. The student evaluates TTT using graphs, charts.
3. The student understands the principles behind the modern TTT used.
4. The student uses tools to track the progress of the Prototype.
5. The student justifies the choices made and TTT selected.

General overview

Module Name	**Advanced Tools**
Unit code	L.26031
Year and Term	2.3
CMGT roles	Engineer
Credits	3 ECTS
Lessons	Five lectures and five labs
Study load	52 to 80 hours
Responsible lecturer	Yvens R. Serpa (y.reboucasserpa@saxion.nl)
Lesson structure	1.5-hour lecture, twice a week
Module summary	The student is introduced to modern and advanced techniques for interactive and graphical applications, including collision detection algorithms, visibility culling, advanced shader usage, language embedding techniques, and artificial intelligence.
Industry relevance	Modern applications require more advanced and new approaches to reach the quality expected by today's market. Additionally, industry solutions rely on scientific methods for optimization and high-quality graphics. This module will introduce and explain the fundamentals' basis to prepare students to take on an active role in modern solutions in Creative Media & Game Technologies.
Type of exam	Assignment
Exam code	T. 51757
CMGT Competencies	1. Technical research and analysis 2. Designing, prototyping, and realizing 3. Testing and rolling out 9. Working in a project-based way 12. Responsibility
Required prior knowledge and skills/conditions for enrolment.	The student needs to be familiarized with using C++ and OpenGL to follow the lectures. For the assignment, the student also needs to be familiar with using a Game Engine, such as Unity 3D. The student must be used to work with a medium-size code base project.
Preparatory for:	The acquired knowledge will be further used in the project 4th Project and prepare the student for possible Minors, Internships, and Graduation projects based on modern interactive and graphical applications.

Rubric	Insufficient	Sufficient	Good	Excellent
Tool / Technology / Technique (TTT) (30%)	0% The Prototype does not work/execute/compile and/or presents unexpected/wrong behavior.	15% **Simple Prototype:** mainly based on one Tutorial. Simple/Basic Implementation (code structure is simple with mostly hardcoded values).	25% **Structured Prototype:** based on multiple sources (Tutorials, for instance). Structured Implementation (flexible, modular, and structured code).	30% **Advanced Prototype:** The Prototype is an example of state-of-the-art or closely based on an example of one. Structured Implementation (flexible, modular, and structured code).
Evaluation (20%)	0% There is no evaluation, or it was conducted with wrong/invalid data.	10% The evaluation shows **2 (two)** Charts/Graphs and attends to the criteria for the testing. The data used is valid (Random seed is fixed, etc).	15% **Sufficient+:** The evaluation explores the behaviour of the TTT using critical points from the Charts/Graphs.	20% **Good+:** The evaluation includes Big O Notation.
Understanding (20%)	0% Cannot explain the basic principles behind the TTT within the context of the evaluation.	15% Understands the basic fundaments of the TTT within the context of the evaluation.	20% Understands the TTT within the context of the evaluation.	25% Understands the TTT enough to develop further an extension for it within the context of the evaluation. (Extension in the sense of a new algorithm/framework/methodology, etc.)
Justification (15%)	0% There is no or weak justification for the chosen TTT.	10% The student justifies the chosen TTT by using examples of other systems that use the same TTT.	12.5% The student justifies the chosen TTT by using **at least one** valid scientific source.	15% The student justifies the chosen TTT by using **at least one** valid scientific source and can further develop it within the evaluation context.
Report + Structure (15%)	0% Does not provide a report and/or valid progress tracking.	10% The report is written in a PDF and attends to the requirements.	12.5% The report was done on a source control website with a small/simple wiki page.	15% **Good+:** The report shows proper Git usage (Commit Message Convention, Merge/Pull Request, use of Tags, etc.).