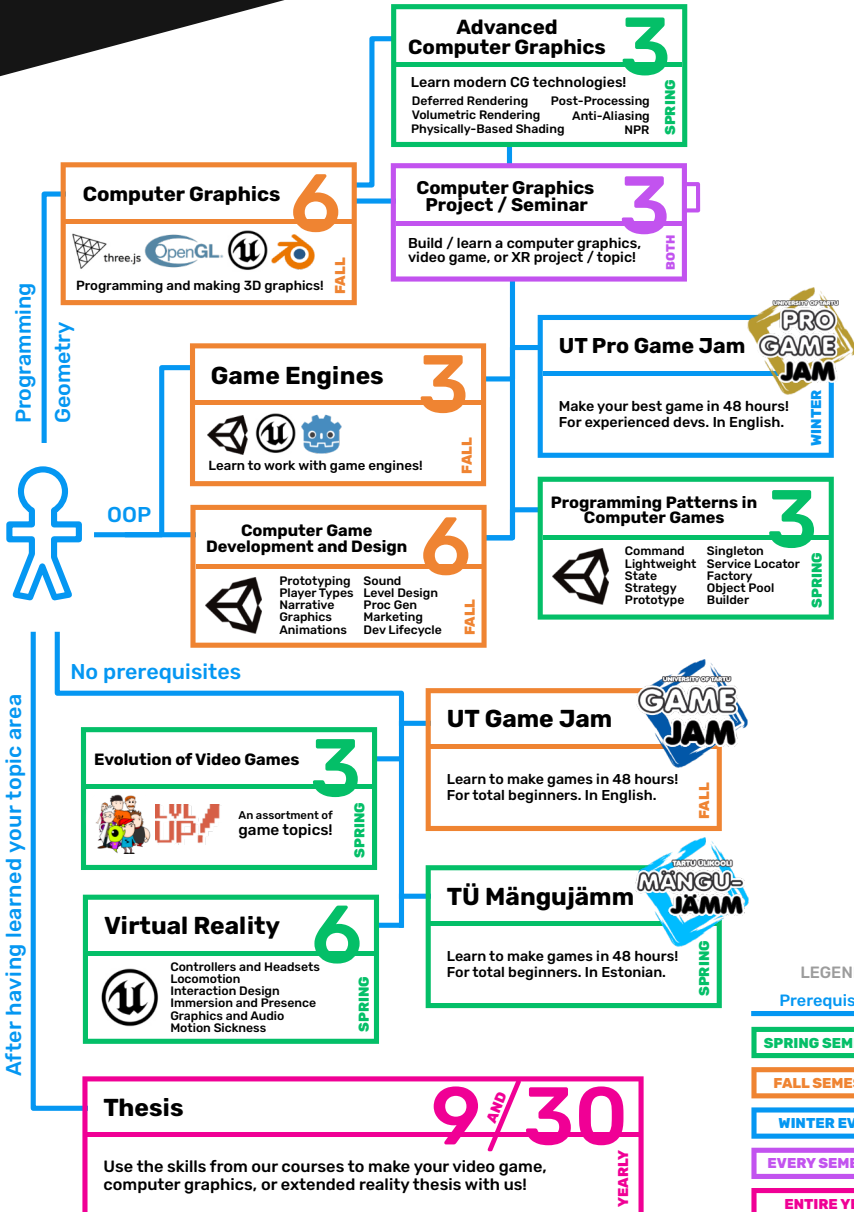


The Courses Map

in the CGVR Lab



Computer Graphics [MTAT.03.015](#)

In the computer graphics course, you learn the standard graphics pipeline that the GPU runs to render cool 3D pictures. We delve into the linear algebra of vectors and matrices to understand the geometry transformations. You learn how the computer shades 3D objects according to the light sources. In the second half of the course, you can learn 3D modeling and graphics in Unreal Engine or dig deeper into the algorithms behind ray tracing, global illumination, procedural generation, and shadow rendering.

Advanced Computer Graphics [NEW](#)

Modern computer graphics relies on different techniques and tricks that utilize the standard graphics pipeline in useful ways. In this course, you will learn about modern techniques that include deferred, volumetric, and non-photorealistic rendering, physically-based shading, post-processing effects, and more. You will experience both developing and studying the algorithms and techniques that produce amazing modern renders.

Computer Game Development and Design [MTAT.03.263](#)

From narrative and sound to graphics and level design, this course covers many aspects of game production. You start by learning how to design your game's identity according to the target audience. You then learn to build your awesome game with the Unity game engine. At the end of the course, you will learn how to successfully market your game to the aimed audience. This course will ready you to pursue your dream as a video game developer!

Game Engines [LTAT.02.018](#)

Often creating a big project like your thesis requires you to use a game engine. Mastering one from scratch can take a lot of time and pulls you away from actually working on your project. In this course, you learn 3 very popular game engines: Unity, Unreal Engine, and Godot. You will understand what makes each of those engines unique and how to choose the best engine for your project. Learning is hands-on by building practical applications in each engine.

Programming Patterns in Computer Games [MTAT.03.315](#)

Developing a real-time application with a graphical component can be very difficult if you do not have enough experience. In this course, you advance your object-oriented programming skills by learning very useful code design patterns that allow you to build standard, useful, readable, and scalable solutions. You will learn how not to make a mess in your code when developing a computer game!

Evolution of Video Games [LTAT.02.019](#) / [LTAT.TK.017](#)

To design a good video game, one has to know what makes video games good. In this course, you learn the evolution story of video games, from the time video games first became a thing to modern virtual reality and beyond. Seeing the evolutionary scope will give insight into the design, development, and business choices made in the video game industry. This course is fully online, so you get to learn with like-minded people regardless of your physical location. The course materials are made together with the LVLup! Video Game Museum.

Virtual Reality [LTAT.06.019](#)

In the Virtual Reality course we learn what makes virtual reality distinct from just regular 3D software. The course covers important topics like immersion, presence, interaction design, motion sickness, graphics, and audio in VR. The course is very learning-by-doing and ready for anyone to come and build their VR dream project.

University of Tartu Game Jams

We organize game jams three times a year. In April and October, we have learning-oriented game jams dedicated for high school and beginner-level university students. In these jams, instructors help out all the participants in the different game development fields. These jams are local sites of the global Ludum Dare game jam.

In January we have a highly competitive game jam organized for advanced level university students and other game developers from the field. The Pro Game Jam has great prizes for the best participating teams, and it is a great chance to show your skills to industry representatives in the jam.

This jam coincides with and is organized as a site for the Global Game Jam.



The 7 Thesis Types

in the CGVR Lab



Independent Study

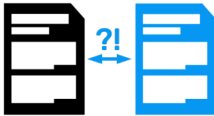
Make something novel.



You devise your own new algorithm, theory, design, solution, which is in some sense novel and potentially helps others in the field. The thesis describes your contribution. You base your creative work on existing research papers, theories, and other discoveries, which you compare your solution against. The appendix can include a demo application as well.

Article Review

Analyze an article from a new perspective.



You take one existing article and analyze it thoroughly. It could be about theory, an algorithm, some study, or something else you feel that your review would contribute to. The thesis should bring out the pros and cons you discovered with your work, not just reiterate the existing results.

Survey

Make a map to orient people among many existing objects.



You investigate many existing things, which could be different software, hardware, algorithms, formulae, theories, etc. You come up with a novel system for categorizing these objects. The thesis describes these many things from an important perspective and clearly provides your new systematic categorization.

Hardware or Software Experiments

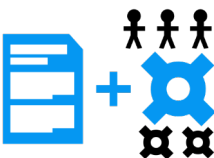
Try out several things and see how they compare.



You take several different hardware or software and try to determine, which of these are best for some certain situation. The difference from the survey thesis is that you take a lot fewer objects and go more in-depth (experiment) with them. The thesis also describes the methodology used in the experimentation.

Author's Software Solution

Make a software product.



You design, implement and test software. Your software has a target audience and goals that you set. It relates to some existing software already in the market. The thesis can emphasize the user experience design, implemented interesting algorithms, the architecture, used programming patterns, or something else that is key to your software. The testing could be performance testing, usability testing, stress testing, etc.



User Manual

Write an effective user manual.

You analyze some existing piece of technology and write a user manual for it. It could be for complicated software, programming library or language etc. You assess the created user manual with some actual users.



Learning Materials

Teach people.

You design and write some learning materials for a course. You can even design the course itself too. The thesis describes the didactic principles and your approach used to create of the materials. You use the materials yourself to teach and assess their effectiveness, or other parameters.

Make a new thing



Author's Software Solution



User Manual



Learning Materials



Independent Research

Practical



Theoretical

See existing things



Hardware or Software Experiments




Survey



Article Review

 - you make during thesis work

 - already exists in the world



cgvr.cs.ut.ee

Thesis Supervisors

in the CGVR Lab



MARK



EDUCATION

VIDEO GAMES

YOUR OWN IDEAS

LEARNING MATERIALS

DANIEL



VIDEO GAMES

GAMES PROGRAMMING

MADIS



VIRTUAL
REALITY

AUGMENTED
REALITY

NEUROSCIENCE

THE ENVIRONMENT

THESIS TOPICS

THESIS TOPICS



THESIS TOPICS

THESIS TOPICS

COMPUTER
GRAPHICS

GEOMETRY

PROC GEN

MATH

MATHIAS



JAAANUS

BLASTRONAUT

COMPUTER GRAPHICS

GODOT GAME ENGINE

PROC GEN



RAIMOND

ONLY
CONSCIENTIOUS
STUDENTS PLZ!



VIDEO GAMES
COMPUTER GRAPHICS
EDUCATION
STUDIES
DESIGN



cgvr.cs.ut.ee