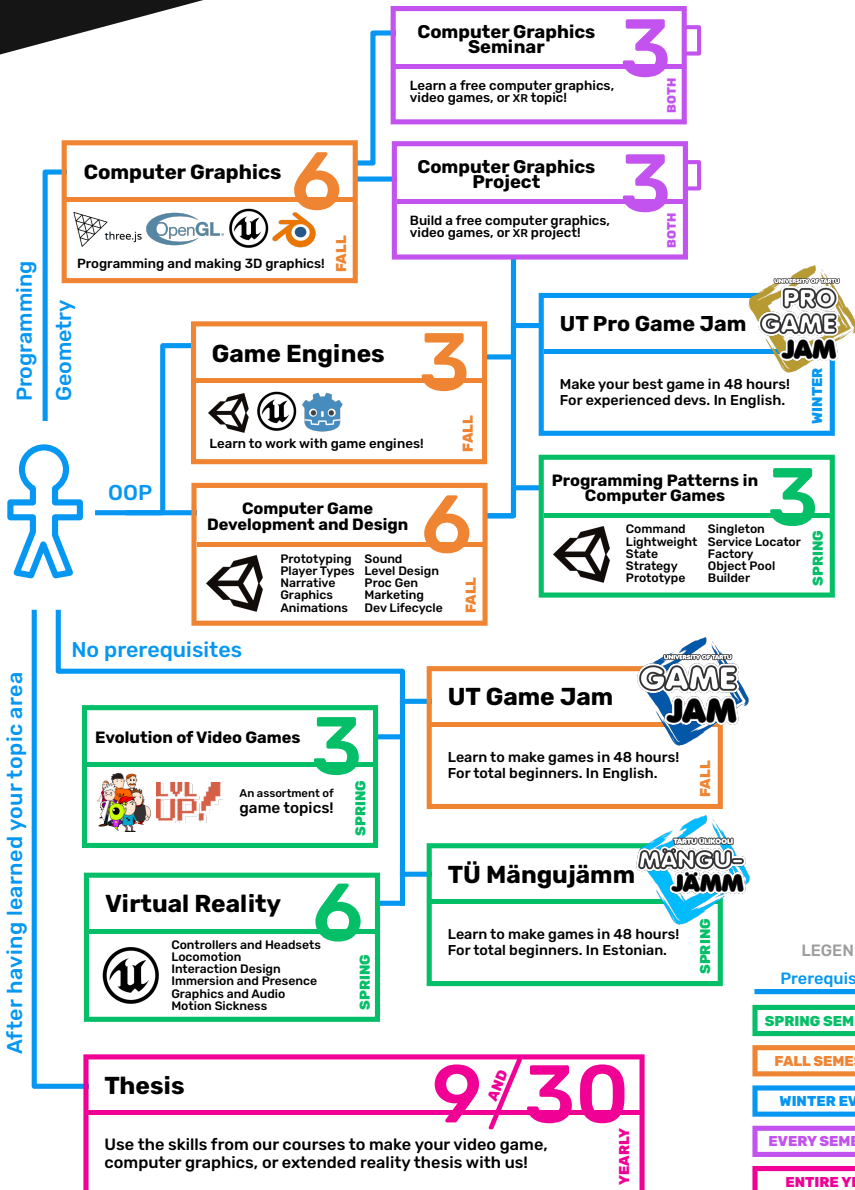


# The Courses Map

## 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 to the screen. 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.

## Computer Game Development and Design MTAT.03.263

This course focuses on a myriad of topics from different disciplines. For example, in the beginning of any game development it is important to define and know your player types and the target audience. During the latter half of the production phase it is important to know how to market your game to the aimed audience. Many of the topics focus on how to design and implement different aspects of a game. From narrative and sound to graphics and level design. During the course you build your own games with the Unity game engine.

## Game Engines LTAT.02.018

Often creating a big project like your thesis requires you to use a game engine. Learning one can take a lot of time and takes you away from actually working on the said project. So in this course you learn 3 very popular game engines: Unity, Unreal Engine and Godot. After this course you have skills in each of those to build your projects. You also acquire understanding about what makes each of those engines unique and how to choose the best engine for your project. Learning in this course 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 quite difficult if you have little experience with it. In this course we advance on the object-oriented programming paradigm by learning a good number of programming patterns. These patterns are common, useful, readable, and scalable solutions to standard problems in programming. We learn how not to make a mess in the code when developing a game. This course gives very useful programming skills and also critical thinking capabilities into your own and other people's code.

## 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 and much more. The course is an advancement course highly focused on big project development and thus is very learning-by-doing.

## 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 we have instructors to help out and teach all the participants in the different fields related to game development. These jams coincide with the global Ludum Dare game jam such that participants also submit their games there.

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

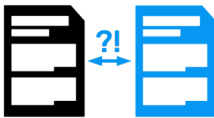
*Make something novel.*



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.*



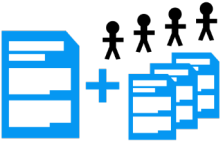
You design, implement and test software. Your software has a target audience and goals that you set. It relates to some existing software already in the market. The thesis can emphasize the user experience design, implemented interesting algorithms, the architecture, used programming patterns, or something 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.

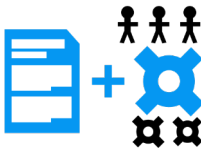


## Learning Materials

*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



# 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



ULNO

VIRTUAL  
REALITY

INDUSTRY  
COLLABORATION

3D PRINTING

IOT



cgvr.cs.ut.ee