



BA:

## Game Engine Architecture in Rust

Game engines are the central part of interactive real-time graphics applications, especially video games. They provide a common core of shared code upon which a wide variety of games and applications can be built. To achieve the requirements of this task, such systems have to be flexible as well as performant.

Rust seems to be especially well suited for game engine development. It offers zero-cost abstractions for many common programming patterns, and safety guarantees which make it more robust against certain programming errors than other low-level languages, such as C++, which is used in current commercial game engines.

We want to design and implement a game engine architecture in Rust and evaluate the viability of Rust's language features for such systems. In particular, the focus of the architecture should be on modularity to provide a framework for experimentation with particular engine subsystems.

As a second part we want to demonstrate the usefulness of this framework by implementing two different renderer subsystems: A simple mesh renderer and a novel GPU-driven Voxel renderer.

**Requirements:** This project will involve programming parts of a game engine in Rust. An interest in interactive real-time graphics and prior knowledge of the Rust programming language are thus advantages. We will have weekly meetings to discuss open questions and determine the next steps.



**Interested? Please contact me for more details!**

### Contact

- Quentin Kniep: [qkniep@ethz.ch](mailto:qkniep@ethz.ch),  
ETZ G95

### References