Realtime Computer Graphics Exercises
Today short lecture (45mins)
- Group and Project setup
Questions next week
Required Hardware/Software:
- Windows/Linux PC
- AMD/Nvidia/Intel GPU
General Information

One exercise per week
Exercises contain practical and theoretical tasks
- Math
- Theory about (realtime) computer graphics
- Vulkan knowledge
  https://khronos.org/registry/vulkan/specs/1.3-extensions/html/index.html
- Programming tasks
At least 50 % of points of the exercises to pass
Register account on [https://git.cg.cs.tu-bs.de](https://git.cg.cs.tu-bs.de)
- Use your *first* and *last* name **not** your y-number and **not** a random nick name

Form groups of 2 – 3 students (preferably 3)
- Send the group info to [ecg@cg.cs.tu-bs.de](mailto:ecg@cg.cs.tu-bs.de)

Register on our website if not done already
- [https://graphics.tu-bs.de/teaching/students](https://graphics.tu-bs.de/teaching/students)
Project Setup
Install Software and Libraries

Recursively Clone Mtstudio and Exercise Project

Exercise Project Handling

Common Issues/Pitfalls
Install Software and Libraries
CMake: [https://cmake.org](https://cmake.org)

IDE/Editor
- Visual Studio Community: [https://visualstudio.microsoft.com](https://visualstudio.microsoft.com) (Windows)
- Clion: [https://www.jetbrains.com/clion/](https://www.jetbrains.com/clion/) (Windows/Linux)
  - Student license: [https://www.jetbrains.com/de-de/community/education/#students](https://www.jetbrains.com/de-de/community/education/#students)
- Text Editor of your choice (No Support from us!!) (Windows/Linux)
  - E.g. (N)Vi(m), Emacs, VS Code, ...

GIT: [https://git-scm.com](https://git-scm.com)

On Linux GCC (compiler) and make
Vulkan SDK: https://www.lunarg.com/vulkan-sdk/

Windows:

Linux (through system package manager):
- GLFW
- GLM
Clone Mtstudio and Project
Clone Mtstudio

git clone --recursive gogs@git.cg.cs.tu-bs.de:ECGSS2022/mtstudio.git

- Make sure that the following folders exist and are populated
  - mtstudio/lib/{ImGuizmo, imgui, vkrenderer}
  - mtstudio/lib/vkrenderer/libs/VulkanMemoryAllocator
  - mtstudio/lib/vkrenderer/libs/include/{container, gli, stb}

cd mtstudio/projects
git clone gogs@git.cg.cs.tu-bs.de:ECGSS2022/hello_world.git
Exercise Project Handling
Create a new folder `mtstudio/build/` as your working directory

(Windows only:)
- Extract `ThirdParty.zip` into `mtstudio/` dir
- Copy `glfw3.dll` (from the `.zip` file) to `build/` (working dir)

Run CMake to generate the binaries for your IDE
- Source: `mtstudio/`
- Binaries: `mtstudio/build/`
Exercise Project Handling

Make sure that your working directory is the build folder
- More details in “Pitfalls” later

To run a project:
- Pass the path to the .prj file in the project folder to the mtstudio executable (relative to the working directory) as an argument
- Example

  *Bash*: ./mtstudio ../projects/hello_world/hello_world.prj
  *Visual Studio*: mtstudio > Settings > Debugging > Arguments:
  ../projects/hello_world/hello_world.prj
Common Issues/Pitfalls
Make sure *everything* is compiled

- in Visual Studio and Clion use “build all”
- While working on the project, it is ok to just (re)build that project

Execute the project from the correct working directory

- From command line `cd` into the correct `build` folder
  - `cmake-build-debug` in Clion
  - Just `build` in Visual Studio *NOT* `build/Debug`
- In Clion/Visual Studio you might have to change the project settings to the correct working directory
Test the setup

Execute the hello_world project

Make sure that the output matches the screenshot

Make yourself familiar with the development tools including

- Build system
- Debugger
- Version control

Questions and trouble shooting next week

Until next week
Hello, World!