



REYNEP's Vulkan "Adventure Guide"

Where, you adventure on your own 😊, I only 'guide', showing you the roadmap

Chapter 0: Prerequisites 🏠

📖 Suggested Reading (before embarking on this journey)

1. <https://paminerva.github.io/docs/LearnVulkan/01.A-Hello-Window>
 - Read the **1 - Introduction** part from here only 😊 [untill **1.2. Why Vulkan?** end]
 - 😊 [00-Introduction-and-prerequisites.pdf](#)
 - 😊 [01.A-Hello-Window.pdf](#)
2. Alternatively:- you can give this page a try too:-
 - <https://vkdoc.net/chapters/fundamentals>
 - that is, if you are into "official formal-documentation" [i sure am not....]

👤 The 5 Questions

1. What is **Vulkan** ?🤔.... Why **Vulkan** ?
 - **Suggested Reading 1:-** **p.a.minerva**
2. Why should **'you'** learn/use **Vulkan** ?
 - i. 5-10% Faster
 - ii. More Control
 - iii. Lower Level API
 - iv. You can ask and know 'what actuaaallyyy happens under the hood of the gpu'
3. Why is this Important?
 - Well if you are planning on becoming a game dev, then yeah, this kinda is important!
 - otherwise, if you are just here for **Creating Shaders**:- **OpenGL** is fine enough
 - a. **Shader Enthusiast:-** <https://www.shadertoy.com/>
 - a. <https://www.youtube.com/playlist?list=PL9Zb80ovNLWGRFZVL4LcckTWnEGN73dFS>
 - b. https://www.youtube.com/playlist?list=PLGmrMu-lwbguU_nY2egTFmlg691DN7uE5
 - c. <https://www.youtube.com/playlist?list=PLCAFZV4XjzP-jGbTke6Bd3PNDpP1AbIKo>
 - d. <https://www.youtube.com/playlist?list=PLGmrMu-lwbgtMxMiV3x4lrHPIpmg7FD-P>
 - e. https://www.youtube.com/watch?v=5J-0sy2pu_8&t=357s&pp=ygUVc2hhZGVyVG95IHJheW1hcmNoaW5n
 - f. <https://www.youtube.com/watch?v=kHbIXafu7iA&pp=ygUjc2hhZGVyVG95>
 - b. **Making an App/UI :-** doing everything with **OpenGL** -> would be just fine
 - a. [TheCherno OpenGL Playlist \[YT\]](#)
 - b. [TheCherno Game Engine Playlist \[YT\]](#)
4. When will "You" need **vulkan** ?
 - kinda never -> unless you have grown tired of OpenGL
 - kinda yes -> when you wanna understand "How the heck does the GPU Work?"
 - but yes, Big AAA games would need **vulkan** for even that last 5-10% performance
5. How does **vulkan** work?
 - Rest of this entire guide is dedicated to answer this question 😊

1. grab `vulkan-sdk`, `cmake`, `amGHOST`

1. if you don't have `vscode` & `C++ Compiler`

- see [4.guide.CH0.vscode.md](#)

2. <https://vulkan.lunarg.com/sdk/home>

- make sure `VULKAN_SDK` & `VK_SDK_PATH` environment variables are set
- restart `vscode` after installing

3. <https://cmake.org/download/>

- [optional] <https://enccs.github.io/intro-cmake/hello-cmake/>
- [optional] OR: Watch 6/7 videos from this playlist:- <https://www.youtube.com/playlist?list=PLK6MXr8gasrGmliSuVQXpfFuE1uPT615s>
- restart `vscode` after installing

• `REY_DOCS`

```
cmake_minimum_required(VERSION 3.25 FATAL_ERROR)

project("idk_PROJECT" VERSION 0.1)

set(CMAKE_CXX_STANDARD 23)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

# -----
set(SRC
    "main.cpp"
)

set(INC
    ${CMAKE_CURRENT_SOURCE_DIR}
)

# -----

# -----
# set_source_files_properties(main.cpp PROPERTIES COMPILE_FLAGS "/P /C")
# Output Preprocessed File
add_executable(idk ${SRC})
target_include_directories(idk PUBLIC ${INC})

# -----amGHOST-----
add_subdirectory(amGHOST)
target_link_libraries(idk PUBLIC amGHOST)

# -----install-----
install(TARGETS idk
    DESTINATION ${CMAKE_CURRENT_SOURCE_DIR})
```

4. `amGHOST`

- amateur's Generic Handy Operating System Toolkit
 - [secretly inspired by `blender's GHOST` xP 🤖]
- `git clone -b win32-intro https://github.com/REYNBP/amGHOST`
- Open it with `VSCode`
- `F1 --> CMake: Configure`
- `F1 --> CMake: Build`

- `F1 --> CMake: Install --> .install dir`
- check's *amGHOST's Usage Example* inside `amGHOST/README.md`
 - **Option 1** :- use `cmake` for your project too.... using `add_subdirectory(amGHOST)`
 - **Option 2** :- use `libamGHOST.lib` after installing & `#include amGHOST/<header>`
- just copy paste *amGHOST's Usage Example* into a `main.cpp` for your program

```
#include "amGHOST/amGHOST_System.hh"

int main(int argumentCount, char* argumentVector[])
{
    amGHOST_System::create_system();    // initializes amG_HEART

    amGHOST_Window* W = amG_HEART->new_window_interface();
    W->create(L"Whatever", 0, 0, 500, 600);

    REY::cin.get();    // wait for terminal input
    W->destroy();
}
```

- [shorter than `readme ex. 1`]
- now you shall have a OS-Window 😊