**04.DISPOSIZIONE DEL PROGETTO**

Per un motore grafico in Vulkan usiamo la seguente gerarchia di cartelle:

* **assets:** contiene texture e modelli 3d che useremo nella guida.
* **bin:**  path di compilazione dell’eseguibile, non usiamo il percorso di base di CMake per

semplificare il pecorso delle shader e dei modelli.

* **Shaders:** conterrà le shader del programma e il loro output compilato.
* **Third\_party:** contiene librerie di terze parti usare per semplificare il codice.

Le librerie usate sono le seguenti:

* Vulkan SDK: kit di sviluppo che include alcune librerie menzionate in lista.
* GLM (Vulkan SDK): libreria per operazioni su vettori e matrici e contiene tipi compatibili con GLSL, il linguaggio di shading. Ad esempio, glm::mat4 è compatibile con mat4 nelle shader.
* SDL (Vulkan SDK): Libreria per processare input, audio e gestire una finestra. Cross platform e in uso da molti engine.
* ImGUI: libreria per programmare in maniera semplice un’interfaccia di debug sul nostro applicativo. Supporta vari renderer ed è cross platform.
* STB Image: libreria per caricare immagini
* fastGLTF: Libreria per caricare modelli 3D, basata su GLTF ma più performante.
* Vk Bootstrap: libreria che permette di semplificare l’inizializzazione del renderer (istanze, swapchain ecc)
* VMA Vulkan Memory Allocator: libreria che aiuta nell’allocazione di memoria automatica per buffer, immagini ecc.
* Fmt: libreria di formattazione delle stringhe, più performante rispetto alle stampe standard con il buffer cout in C++.