

projekt do předmětu PGR – Počítačová grafika 2023

Zobrazení modelů automobilů pomocí OpenGL

řešitel: **Josef Kotoun**, xkotou06

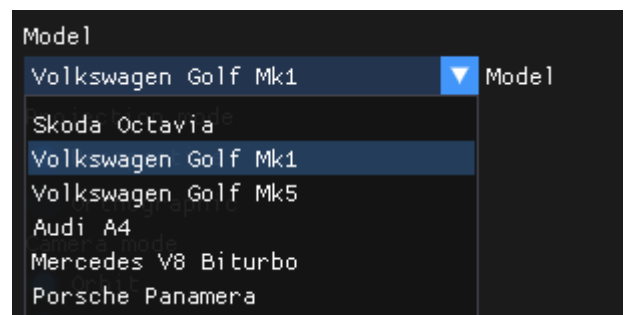
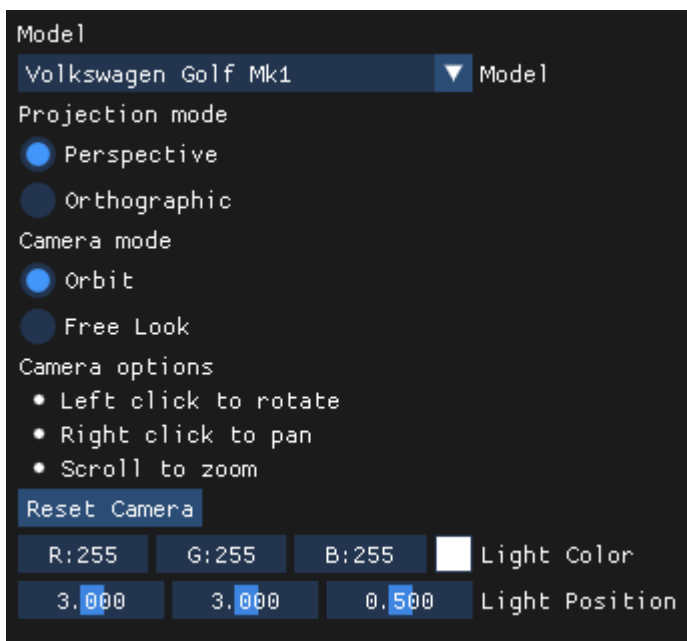
Zadání

- Získání modelů automobilů
- Načtení modelů do Opengl
- Vykreslení modelů
- Vytvoření uživatelského rozhraní pro různé nastavení prohlížení objektu
 - Výběr konkrétního modelu
 - Typy kamer
 - Typy projekcí
 - Nastavení osvětlení
- Funkčnost s různými formáty objektů – testováno na .obj a .glb

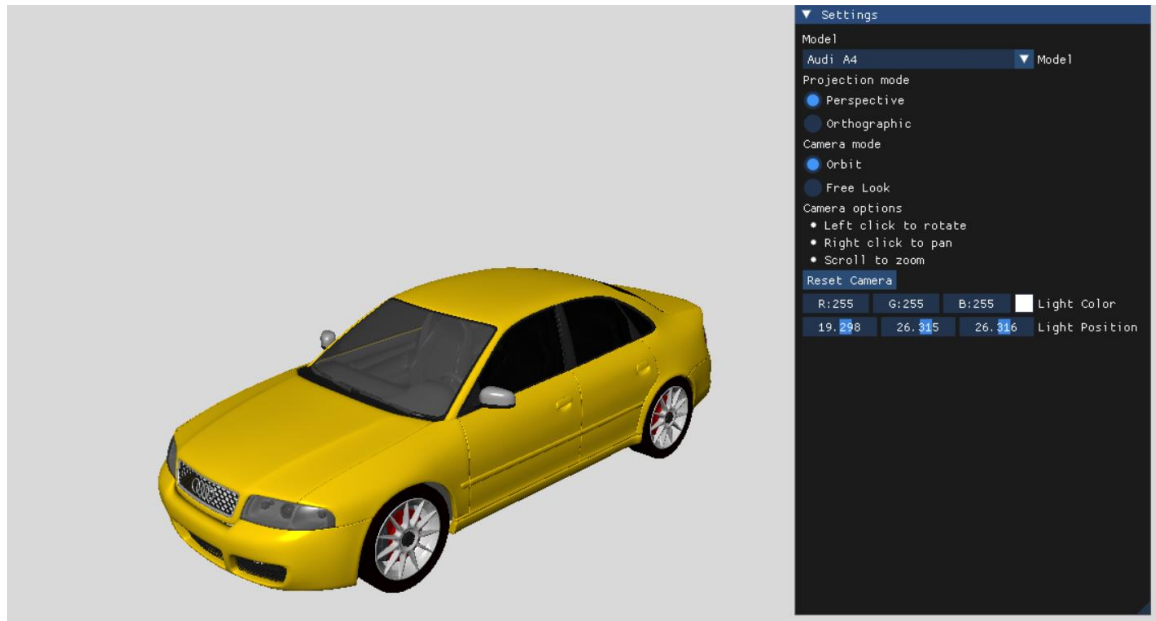
Nejdůležitější dosažené výsledky

- Podpora různých formátů modelů – s využitím knihovny assimp, testováno primárně na formátech gltf a obj, ale program načte i model v .fbx nebo .glb
- Jednoduché uživatelské rozhraní vytvořené pomocí knihovny ImGui, pomocí kterého lze vybírat zobrazený model, upravovat pozici světla, typ projekce, typ kamery a další + obsahuje popis k ovládání kamery
- Podpora 2 typů kamer
 - orbit kamera
 - rotace a posun kamery po scéně
 - free look camera
 - pohyb scénou z první osoby
- Podpora 2 typů projekce – perspektivní a ortografická
- Normalizace modelů do stejných rozměrů a pozice pomocí model matice
- Funkční průhledné materiály – např. okna, světla...
- Phongův osvětlovací model

Uživatelské rozhraní – ImGui



Ukázka s perspektivní projekcí a orbit kamerou:



Ukázka s ortografickou projekcí a orbit kamerou



Ukázka s free look kamerou



Ovládání vytvořeného programu

- Výběr nastavení pomocí rozhraní vytvořeného pomocí ImGui (nastavení barvy světla, pozice kamery pomocí sliderů, ostatní nastavení pomocí radio buttonů a selectů)
- Výběrem z menu lze vybrat model k vykreslení, dosud nenačtený model je načten až po výběru
- Dále lze zvolit typ projekce a typ kamery
- Orbit kamera
 - při podržení pravého tlačítka posun kamery, kolečkem přiblížení, při podržení levého tlačítka rotace ve scéně jak doleva/doprava tak nahoru/dolů
- Free look kamera
 - Pravým tlačítkem přepnutí do módu pohybu po scéně, poté pohybem myši rozhlížení po scéně, pohyb pomocí WASD a opuštění módu stisknutím ESC

Zvláštní použité znalosti

- Nastudování formátu .obj pro pochopení struktury modelů
- Nastudování načtení modelů do reprezentace v c++
- Základy použití knihoven assimp, glm, imgui, SDL2 a stbimage

Použité technologie

- Využívá řadu knihoven, které jsou zabaleny součástí řešení Visual Studio pro windows x64.
- Knihovny
 - SDL2 – pro zpracování vstupů, vytvoření okna apod. – přibaleny lib, dll a .h
 - Assimp – pro načítání modelů – přibaleny soubory lib, dll a .h
 - Glm – pro matematické operace – přibaleny .hpp
 - ImGui – pro načtení textur ze souboru – přibaleny .h
 - Glad – pro zpřístupnění opengl funkcí – přibaleny .h a .c
 - ImGui – pro tvorbu uživatelského rozhraní – přibaleny .h a .cpp
- Projekt je odevzdán jako řešení .sln které je propojeno s těmito přibalenými knihovnami, projekt by tedy měl fungovat po zkompilování řešení na pc se systémem windows 64 bit

Použité zdroje

- Modely dostupné zdarma na sketchfab.com

- Inspirace tutoriálem k načítání modelů pomocí assimp z learnopengl - [LearnOpenGL - Assimp](#)
- Obecné použití OpenGL a SDL - youtube playlist od autora Mike Shah - [\(6\) \[Episode 1\] Learning OpenGL - Modern OpenGL - YouTube](#)

Co bylo nejpracnější

- Správné načítání modelů z různých formátů (zpočátku fungoval jen .obj a pouze s 1 texturou)
- Správné vykreslení průhledných objektů, aby přes ně bylo vidět a zároveň se vykreslily objekty za nimi – tedy aby jejich vykreslení nezahlavil depth test
- Správné fungování kamer – především orbit kamery

Zkušenosti získané řešením projektu

Naučil jsem se především lépe fungování opengl a jeho propojení s knihovnou jako je SDL2. Také jsem se naučil jak funguje reprezentace 3D modelů v různých formátech a lépe pochopil použití a potřebu view a model matic. Dále propojení c++ s knihovnami třetích stran, s čímž jsem neměl zkušenost a zabralo mi to také nějaký čas, než jsem pochopil, co je vše třeba nastavit.

Autoevaluace

Technický návrh: 90% (analýza, dekompozice problému, volba vhodných prostředků, ...)

- Návrh a rozložení do tříd si myslím, že je dobré. Jednotlivé kamery a reprezentace modelu je abstrahována do samostatných tříd, dekompozice main.cpp by byla asi třeba lepší pro větší projekt, ale tento rozsah projektu je dle mého dostačující. Stručně (1-2 řádky) komentujte hodnocení.

Programování: 90% (kvalita a čitelnost kódu, spolehlivost běhu, obecnost řešení, znovupoužitelnost, ...)

- Díky abstrakci do tříd je dle mého znovupoužitelnost a čitelnost kódu dobrá.

Vzhled vytvořeného řešení: 70% (uvěřitelnost zobrazení, estetická kvalita, vzhled GUI, ...)

Gui je vytvoření pomocí ImGui, pomocí kterého lze vytvořit gui jednoduše, ale není vzhledově moc hezké. Zobrazení je celkem dobré díky funkční průhlednosti materiálů a phongovu osvětlovacímu modelu, ale určitě je zde prostor pro zlepšení, jako využití komplexnějších modelů spolu s podporou spekulárních textur, vykreslení stínů apod.

Využití zdrojů: 90% (využití existujícího kódu a dat, využití literatury, ...)

U běžných problémů, jako je načtení modelu do c++ nebo napojení opengl na knihovnu pro zobrazení okna, zpracování vstupů apod. jsem využil existující řešení pro inspiraci

Hospodaření s časem: 60% (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části řešení, ...)

Projekt jsem začal zpracovávat s dostatečným předstihem, ale vzhledem k deadline ostatních projektů jsem mu nemohl věnovat tolik času a výsledek mi tedy určitě mohl být propracovanější, pokud bych začal dříve.

Celkový dojem: 90% (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, ...)

Volbu zadání i získané zkušenosti v poměru k strávenému času nad projektem hodnotím kladně

Doporučení pro budoucí zadávání projektů

Projekt mi vyhovoval, především mi vyhovovala větší volnost zadání