Dokumentace IPA projekt

- Autor: **Radim Dvořák** (xdvorar00)
- Datum: **2025-05-12**

Projekt byl vypracován pomocí rožšření AVX pro x86 na operačním systému Windows.

Volání hlavní funkce

Prvním krokem je zpracování argumentů a jelikož jsem pracoval na Windows tak první čtyři celočíselné argumenty byly v registrech *rcx*, *rdx*, *r8* a *r9*.

Postup optimalizace

Všechny výpočty jsem se snažil zpracovávat vektorově v registrech ymm0-ymm15 pomocí vektorových instrukcí AVX.

Největší problém bylo rozdělit vektor *vertex* na x, y a z komponenty. K načítání x a z souřadnic jsem použil funkci *vgatherdps*. Problém stále setrvával s y souřadnicí, který jsem nakonec vyřešil přes ukládaní jednotlivých floatů pomocí indexu, který od bázové adresy měl offset *i*3 + 1* floatů, tedy *(i*3 + 1)*4* bytů.

Vypocet sinu a cosinu

Byl použit algoritmus CORDIC [1]. Počet iterací byl nastaven na 15, protože další iterace už nepřinášeli velkou změnu výsledku. Protože CORDIC používá větvení na základě, jestli je úhel kladný úhel nebo záporný, tak v implementaci to bylo řešeno pomocí masky, která umožnovala výpočet jenom na daných floatech ve vektoru. Jelikož AVX nedokáže přímo pracovat s číselnými konstantami tak hodnoty funkce *atan()* a hodnoty záporné mocniny 2 (2^-1)pro všehcny iteracce jsou uloženy v datovém segmentu.

Shrnutí optimalizace

```
Ocean::update took 46128600ns CPU cycles: 110698431
Ocean::update (SIMD) took 4613100ns CPU cycles: 11062866
```

Naimplementovaný kod v assembleru je přibližně 10x rychlejší než kod v čistým C++.

Zdroje

[1] FIT VUT, 2024.
https://moodle.vut.cz/pluginfile.php/871554/course/section/86864/inp2024_09fp_iter.pdf