Osnova

1. Úvod
2. Seznámení s problematikou
3. Popis dříve používaných způsobu „renderingu“
4. Popis dnešního způsobu „renderingu“
5. Rozbor projektu (viz Teoretická část v dokumentu Maturitní témata z IVT)

Úvod

Tématem mé maturitní práce je 3D hra v Javě. Hra je vytvářená na způsob staré střílečky Wolfenstein 3D. Důvody, proč jsem si vybral tohle téma jsou celkem tři. Chci se zabývat programováním a vývojem video her. Hry typu DOOM, Warcraft I a hlavně The Elder Scrolls: Daggerfall jsem si opravdu oblíbil. A zároveň algoritmus, na základě kterém Wolfenstein funguje je v zásadě dost jednoduchý, tudíž je to dle mého názoru úctyhodný start pro nadšeného programátora a matematika.

Cílem práce je naprogramovat jednoduchou hru s funkčním systémem renderingu.

Typy Renderingů

**Ray Casting**

Ray casting je způsob renderingu, ve kterém posíláme paprsky ve 2D prostoru a zjišťujeme kolizi/protnutí s objekty. Tento proces je poměrně rychlý a ne tolik náročný, protože vysíláme paprsek pouze za každý pixelový sloupec na obrazovce z kde každý paprsek počítá pouze jednu jednoduchou matematickou rovnici, díky čemuž byl velice populární v dobách prvních 3D her. Je však omezený a kvůli geometrickým úpravám, které se v průběhu dělají, aby obraz vypadal realistický ztrácí schopnost vykreslovat všechny tvary a tak je limitován na víceméně jen jednodušší prostředí.

Studium Problematiky

**Herní Enginy**

Co se vývoje 3D hry týče existují zde enginy, ve kterých se hry tohoto typu dají vytvořit bez řešení jakýchkoliv problémů, týkajících se matematiky renderingu. Mezi takové dva největší enginy bych vybral Unity Engine a Unreal Engine. Tyto programy už v sobě mají zabudované tyto systémy a vývojář se může prakticky soustředit jen na ty „jednodušší“ úkoly a nemusí se zaobírat vytvářením už několikrát použitých algoritmů, které se jen opakují. Na svůj projekt jsem však žádný herní engine nepoužil, protože mým hlavním cílem bylo naprogramovat právě algoritmus na renderování prostředí.