projekt do předmětu **PGR – Počítačová grafika**

Realistické zobrazení oblohy

řešitelé: **Filip Zapletal**, xzaple27
Václav Pacholík, xpacho03
Lukáš Piják, xpijak00

Zadání

- Vybrat vhodnou technologii pro vykreslování nepravidelných těles, jako jsou mraky.
- Vytvořit scénu s mraky, které se náhodně rozmístí, ale zároveň nevytvoří přílišné shluky mraků, které dle vizualní stránky by nemuseli vypadat pro lidské oko „náhodné“.
- Vykreslení probíhá přímo na obrazovku.
- Vytvoření náhodně vypadajících mraků ve scéně, které budou vypadat různě pro pozorovatele.
- Pokusit se o co nejvěrnější ztvárnění oblohy.

Použité technologie

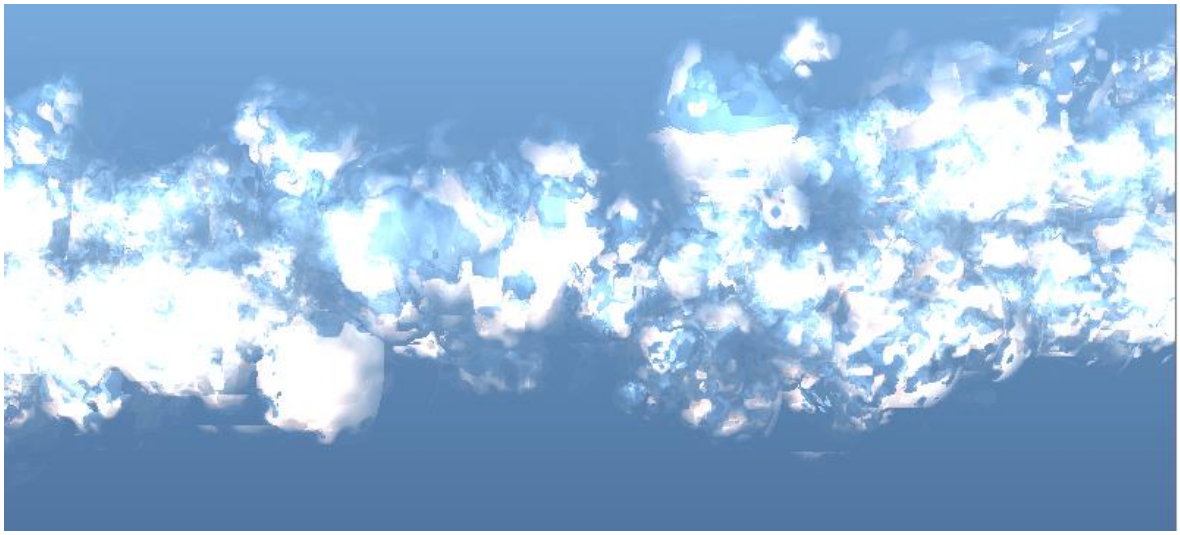
- K vykreslování použit Ray-tracing.
- Pro generování textur použit generátor Perlinova šumu.
- Okno je vykreslováno pomocí SDL.
- Pro vektorové výpočty jsou použity typy z GLM.
- Částicové efekty v OpenGL (nakonec od tohoto řešení upuštěno).
- Rozčlenění prostoru do gridu pro rychlejší vykreslování, grid je procházen za pomoci 3D verze DDA.

Použité zdroje

- http://www.markmark.net/PDFs/RTCloudsForGames_HarrisGDC2002.pdf
- http://devmaster.net/posts/2836/raytracing-theory-implementation-part-1-introduction
  - a navazující díly + inspirace samotnou implementací raytraceru
- http://freespace.virgin.net/hugo.elias/models/m_perlin.htm
- http://freespace.virgin.net/hugo.elias/models/m_clouds.htm

Nejdůležitější dosažené výsledky

- Prvním úspěchem je jistě vytvoření (tedy spíše reimplementace) fungujícího raytraceru.
- Dále se nám přeci jen podařilo vytvořit něco, co alespoň vzdáleně připomíná mraky a vhodně je rozmístit do scény.



- A nejdůležitější také je, že jsme i při níž popsaných obtížích dotáhly projekt alespoň do stavu, ve kterém je.

Ovládání vytvořeného programu

- Klávesa ↑ Přidá jeden mrak pro vykreslení.
- Klávesa ↓ Odebere jeden mrak pro vykreslení.
- Klávesa Enter spustí překreslení.
- Klávesa F přidá do scény mlhu.

Zvláštní použité znalosti

- Vektorová matematika
- SDL a ostatní použité knihovny
- 3D DDA, dělení prostoru

Rozdělení práce v týmu

Filip Zapletal:

- Reimplementace raytraceru (přizpůsobení našim potřebám).
- Implementace primitiv a základní kostry (reprezentace scény, objektů, primitiv, kamery…)
- Modifikace tvaru tělesa texturou.
- Implementace původního vykreslování v OpenGL (ve výsledku nepoužito).

Václav Pacholík:

- Generování textury za pomoci Perlinova šumu.
- Mapování textury na primitiva.
- Tvorba generátoru scény.

Lukáš Piják:

- Vykreslování výsledku na obrazovku (tvorba okna, ovládání…).
- Reprezentace mraku, jeho generování pomocí Gaussova rozložení.
- Dokumentace.

Co bylo nejpracnější

Nejvíce nám zabralo vytvoření nepravidelných objektů a jejich použití v raytraceru. Nelze u nich totiž exaktně počítat průsečíky. Použití velkého množství částeček jednotlivých mraků (částicových effektů) extrémně prodlužuje dobu renderování až do nereálných časů, museli jsme tedy použít složitější objekty – vychází z geometrie koule, jejíž tvar je upraven texturou, ale tato aproximace má za následek zkreslení reality. Tento problém jsme bohužel nedořešili.

Hodně času nám také sebralo ubírání se špatným směrem, kde jsme si vzali moc velké sousto a chtěli oblohu generovat realtime v prostředí OpenGL. Nedostatek zkušeností nám však v tomto nedovolil pokračovat a byli jsme nuceni projekt předělat do raytracingu (otázkou zůstává, zda se to vyplatilo a zda by výsledek dosažený v OpenGL nevypadal lépe).

Zkušenosti získané řešením projektu

Naučili jsme se používat jak HW metody vykreslování (OpenGL), tak i ty čistě SW (Raytracing). Zjistili, že u Raytracingu pro vykreslení jednoduché scény není třeba až tak složitý kód, ale pokud se odprosíme od základních matematických tvarů k realitě, řešení se značně komplikuje. Také jsme se poučili, že v budoucnu je třeba nejdříve vše nastudovat a navrhnout, slepá ulička a reimplementace nás připravili o potřebný čas.

Autoevaluace

Technický návrh: 10% (analýza, dekompozice problému, volba vhodných prostředků, …)

Bohužel jsme pro prvotní implementaci nezvolili vhodné řešení a i to následné asi nebylo pojato tak, jak by bylo třeba. Z toho nám vyplynulo mnoho zbytečných problémů.

Programování: 70% (kvalita a čitelnost kódu, spolehlivost běhu, obecnost řešení, znovupoužitelnost, …)

Kód je psán objektově, jednotlivé části scény a použitých prvků rozděleny do tříd, které lez použít nezávisle. Raytracer psán obecně i pro jiné scény. V kódu by neměli být chyby způsobující pády aplikace.

Vzhled vytvořeného řešení: 20% (uvěřitelnost zobrazení, estetická kvalita, vhled GUI, …)

Požadovaného realistického vzhledu nebylo bohužel dosaženo, při použitém (a implementovaném) řešení se nám nepodařilo docílit věrohodné podoby mraků (je moc exaktní), nehledě na to, že jsme nenašli způsob vytvoření „pruhů světla“, které jsou pro realističnost potřebné.

Využití zdrojů: 60% (využití existujícího kódu a dat, využití literatury, …)

Jako inspirace sloužil kód raytraceru, který byl upraven pro konkrétní potřeby. Základní ideu renderování oblohy jsme získaly z článků.

Hospodaření s časem: 30% (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části řešení, …)

Projekt nebyl zcela dokončen z důvodu počátečního špatného rozhodnutí (použití OpenGL a realtime renderingu), poté již nebylo dostatek času na implementaci tohoto řešení.

Spolupráce v týmu: 75% (komunikace, dodržování dohod, vzájemné spolehnutí, rovnoměrnost, …)

Komunikace probíhala poměrně dobře mezi všemi členy. Bylo použito IM, ale také se konala spousta osobních schůzek. Obecně jsme neměli moc sporů. Jako vedoucí týmu by se dal určit Filip Zapletal, který měl asi nejucelenější pohled na věc a nejvíc zkušeností s tímto odvětvím. Všichni sme se snažili přispět do řešení projektu.

Celkový dojem: 40% (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, …)

Zvolené zadání ne zcela odpovídalo našim původním představám – generování oblohy nakonec bylo obtížnější, než jsme si zpočátku mysleli. K tomu nepomohla ani změna použité technologie v průběhu práce a nutnost začít téměř od začátku. Ale i přes špatný konečný výsledek si myslíme, že nám řešení projektu poměrně dost rozšířilo znalosti jak v oblasti OpenGL, tak i Raytracingu a již to pro nás nejsou jen nějaká abstraktní témata. Také pro příště jistě zvolíme hned zpočátku to „jediné správné řešení“ (pokud bude zřejmé), protože jeho změna nás stála drahocenný čas.

Doporučení pro budoucí zadávání projektů

Do budoucna bychom (tedy spíše naši následníci) ocenili možná přesněji specifikované zadání a alespoň obecný možný návrh daných řešení. V některých zadáních jsou totiž použity věci, které nejsou až tak detailně probírány v rámci přednášek. Samozřejmě je to subjektivní, každému může vyhovovat něco jiného.

Dále je dle nás také problém v tom, že drtivá většina zadání je z oblasti raytracingu a radiozity, které jsou jako témata probírány až ke konci semestru, pokud tedy na tyto znalosti čekáme do doby, než je získáme na přednáškách, tak už je pak pozdě (vzhledem k tomu, že v této době pak jsou i projekty/půlsemky z jiných předmětů).