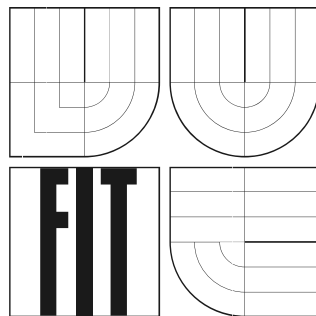


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Vytváření mraků a atmosférických efektů v OpenGL

Ročníkový projekt

Vytváření mraků a atmosférických efektů v OpenGL

© Vít Kučera, 2006.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Prohlášení

Prohlašuji, že jsem tento ročníkový projekt vypracoval samostatně pod vedením pana Ing. Jana Pečívky.
Uvedl jsem všechny prameny, ze kterých jsem čerpal.

.....
Vít Kučera
Datum

Abstrakt

Mraky můžeme spatřit kdykoliv se podíváme na oblohu. Jsou nedílnou součástí počasí na Zemi, součástí koloběhu vody v přírodě a také významným indikátorem změn počasí. Jsou významnou částí vytváření realisticky vypadajících venkovních scén. Zobrazení dynamicky měnící se oblohy je velmi náročné, vzhled ovlivňuje mnoho faktorů . Každý uživatel leteckých simulátorů by chtěl prolétávat kolem realisticky vypadajících oblak.

Práce ukazuje na tvorbu realisticky vypadajících oblak, která využívá Perlinova šumu.

Je kladen důraz na rychlost zobrazení, proto je potlačeno mnoho aspektů.

Klíčová slova

OpenGL, Coin3D, SimVoleon, simulace, 3D scéna, modelování mraků ,tvorba oblohy, Perlinův šum, volume rendering

Poděkování

Rád bych poděkoval svému vedoucímu, kterým byl pan Ing. Jan Pečiva, že mi při práci na projektu pomohl.

Abstract

Clouds are a ubiquitous and ever-changing feature of the outdoors. They are an integral factor in Earth's weather system, component of water circulation in a nature, and a strong indicator of weather patterns and changes. Clouds are an important component of the visual simulation of any outdoor scene, but the complexity of cloud formation, dynamics, and light interaction makes cloud simulation and rendering difficult in real time. In an interactive flight simulation, users would like to fly in and around realistic, volumetric clouds.

I Present, the cloud realism modeling, which use the Perlin noise. It's put more emphasis on speed of graphic representation and so is put down many aspect of physical factors which influence .

Keywords

OpenGL, Coin3D, SimVoleon, simulation, 3D scene, modeling clouds, Perlin noise, volume rendering

OBSAH

1	Úvod	8
2	Fyzikální podstata mraků	9
2.1	Mraky	9
2.1.1	Vznik mraků	9
2.1.2	Druhy mraků	11
3	Coin3D a SimVoleon	16
3.1	Coin3D	16
3.2	SimVoleon	16
4	Metody modelování mraků	16
4.1	Particle systém	16
4.2	Metaballs	17
4.3	Vizuální simulace	17
4.4	Pomocí buněčných automatů	18
4.5	Procedural Noise	19
5	Modelování mraku	20
5.1	Perlinův šum	20
5.1.1	Tvorba Perlinova šumu	20
5.1.2	Jádro perlinova šumu	22
5.1.3	Vyhlazení šumu	24
5.1.4	Interpolace	24
5.2	Tvorba mraků v 3D	26
5.2.1	Vytvoření základního pole	26
5.2.2	Filtr šumu	27
5.2.3	Problém s interpolací (mraku)	27
6	Letecký simulátor	30
6.1	Model letadla	30
6.1.1	Naklápění a pohyb letadla	30
6.1.2	Ovládání letadla	31
7	Závěr	31

1 Úvod

Tento ročníkový projekt a jeho praktická část se zaměřují na vytvoření realistický vypadajících mraků, kdy je důraz kladen na real-timeové zobrazení. Kde z jedné strany je snaha o co nejdokonalější zobrazení oblohy a na druhé straně použitelnost v leteckém simulátoru. Rychlost aplikace musí být natolik vysoká, aby umožňovala plynulé ovládání leteckého simulátoru.

Ročníkový projekt můžu rozdělit do několika celků:

Zkoumání vzniků a vzhledu oblaků v přírodě: Při snaze o realistické zobrazení je důležité pochopit fyzikální principy, které zapříčiňují vzhled

Metody modelování mraků: Popisuje několik základních metod tvorby a zobrazení mraků, kdy jsou metody rozděleny do dvou základních směrů.

Grafický systém: Pro zobrazení se používá knihovny Open inventoru s nástavbou pro volume rendering SimVoleon.

Perlinův šum: Popisuje základní principy perlinova šumu, který je použit při tvorbě mraků.

Model mraků: Vytvoření trojrozměrné matice pomocí perlinova šumu a její transformace na výsledný mrak.

Letecký simulátor: Vytvoření leteckého simulátoru a jeho ovládání.

2 Fyzikální podstata mraků

Při vytváření přírodních scén v počítačové grafice je důležité reálné zobrazení oblohy, které je taky velmi důležité při vytváření leteckých simulátorů. Když budeme pozorovat oblohu složenou z množství mraků, můžeme spatřit jak jednotlivé mraky mění svou velikost, tvar, ale také barvu v závislosti na poloze slunce a také v závislosti na výšce mraku.

Mrak je ve své podstatě viditelné obrovské množství malých vodních kapiček nebo ledových krystalů (záleží na výšce mraku). Kdy barvu mraku určuje procházející nebo odrážející se světlo. Kdy jednotlivé kapičky mraku se vypařují a znovu srážejí a tím vytvářejí dynamiku mraků, která má za následek rozmanité množství tvarů.

V minulosti bylo vyvinuto několik metod pro tvorbu jak statických mraků tak na tvorbu dynamických měnících se mraků, které vychází z fyzikální podstaty vzniků mraků, umožňují animaci tvorbu, pohybu tak také i zániku mraků.

V této práci budou popsány obě skupiny těchto metod a také implementace vlastní statické metody, která je založena na Perlinově šumu a generování 3Dtextur.

Při snaze o realistické zobrazení je důležité pochopit fyzikální principy, které zapříčiňují vzniku a také jeho životního cyklu.

2.1 Mraky

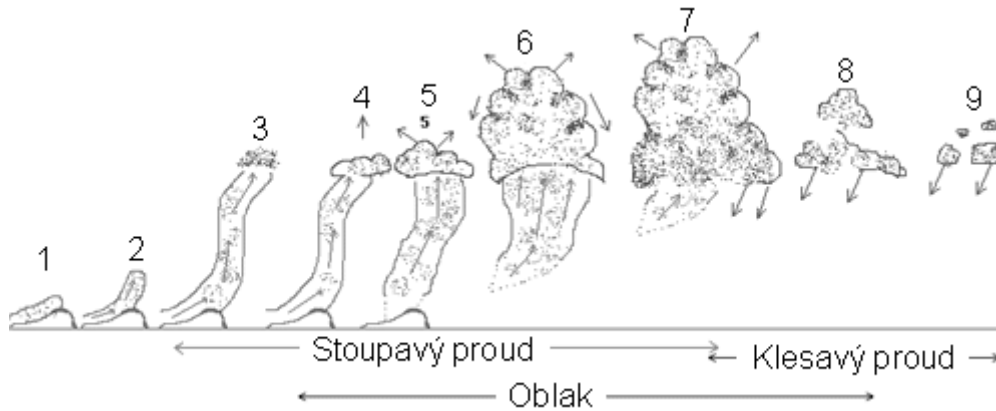
2.1.1 Vznik mraků

Kupovitá oblaka také nazýváme oblaka s vertikálním vývojem. Je to proto, že jejich vznik je bezprostředně vázán na vertikální pohyby v atmosféře, které nazýváme termickou konvekcí. Jiným způsobem kupovitá oblačnost nevzniká.

Sluneční paprsky, které dopadají na zem ohřívají vzduch který proudí po povrchu. Tento ohřátý vzduch je nasycen vodní parou, jestliže se tento proud vzduchu se ohřeje na určitou teplotu a relativní rozdíl teplot je velký, vznikne "komín", kterým se začne odsávat celá zásoba ohřátého přízemního vzduchu nahoru, a následně začne studený vzduch z vrchu nahrazovat odsátý. Tímto způsobem mohou vzniknout víry, jestliže množství je větší mohou vzniknout i tornáda. Celý proces velmi urychluje zemský povrch díky jeho nerovnostem. Aby celý proces odsávání mohl začít musí vzniknout impuls který zapříčiní rozdíl teplot. Tímto impulsem jsou především pohoří.

Tento odsátý teplejší vzduch stoupá a dostává se do tlakových hladin s nižším a nižším atmosférickým tlakem. Díky tomu se vystupující vzduch rozpíná a ochlazuje se. Relativní vlhkost

uvnitř stoupavého proudu neustále vzrůstá a teplota vzduchu se pomalu přibližuje k teplotě rosného bodu. Jakmile se obě teploty ztotožní, nastává kondenzace vodní páry, tedy přeměna z jejího plynného stavu do stavu kapalného. Drobné kapičky tvoří právě oblak



1. Hromadění teplého vzduchu při zemském povrchu.
2. Teplý vzduch se začne odtrhávat od povrchu a začne stoupat.
3. Začíná se vytvářet kolmý stoupavý proud; jestliže je zásoba teplého vzduchu při zemi malá, stoupavý proud se přerušuje a oblačnost se nevytvoří.
4. Jestliže vrchol stoupavého proudu dosáhne kondenzační hladiny, vytvoří se jemná mlhovina.
5. Oblak se začíná vytvářet v nepravidelných kapkách, které rostou a navzájem se spojují.
6. Oblak má stále výraznější a ostřejší okraje; v místech kde je proud nejsilnější, vybíhá mrak nejvýše, má hladké bílé okraje, odspodu je základna nejtmaší.
7. Oblak roste dokud je dostačující zásoba teplého vzduchu při zemi; pokud se tato zásoba vyprázdní, začne se oblačnost rozpadávat.
8. Obrysy oblaku se začínají rozpouštět; oblak se po částech rozpadává.
9. V sestupných pohybech vzduchu se oblak začíná od vrchu rozpouštět.

2.1.2 Druhy mraků

Mraky se dělí na 10 základních druhů. Kdy pro každý mrak je specifická výška .

Můžeme je rozdělit do třech základních pater vysokého(6 až 10 km předpona cirro), středního (od 2 do 6 km předpona alto)nebo nízkého patra (do 2 km předpona cirro).

2.1.2.1 Cirrus

Cirrus - řasa - oblač vysokého patra, což znamená, že se běžně nachází ve výškách 7-10 km. Je složen s ledových krystalů, má vláknitý vzhled a často hedvábný lesk, Nevypadávají z něho srážky a jeho výskyt na obloze bývá často příznakem blízkosti atmosférické fronty.



Obrázek 2.1 - Cirrus

2.1.2.2 Cirrocumulus

Cirrocumulus má podobu tenkých, menších nebo větších skupin nebo vrstev bílých oblaků bez vlastního stínu, složených z velmi malých oblačných částí v podobě zrněk nebo vlánek apod. Tyto jednotlivé části mohou být buď navzájem oddělené, nebo mohou spolu souviset a jsou více méně pravidelně uspořádány.



Obrázek 2.2 - Cirrocumulus

2.1.2.3 **Cirrostratus**

Cirrostratus je průsvitný bělavý závoj oblaků, vzhledu vláknitého nebo hladkého, který úplně nebo částečně zakrývá oblohu



Obrázek 2.3 - Cirrostratus

2.1.2.4 **Alto cumulus**

Oblaky druhu Alto cumulus jsou menší nebo větší skupiny nebo vrstvy oblaků, barvy bílé nebo šedé, popř. obojí, mající vlastní stíny. Skládají se z malých oblačných částí podoby vln, oblázků nebo valounů apod., které mohou být buď navzájem oddělené, nebo mohou spolu souviset. Mnohdy mají částečně vláknitý nebo rozplývavý vzhled.



Obrázek 2.4 - Alto cumulus

2.1.2.5 Altostratus

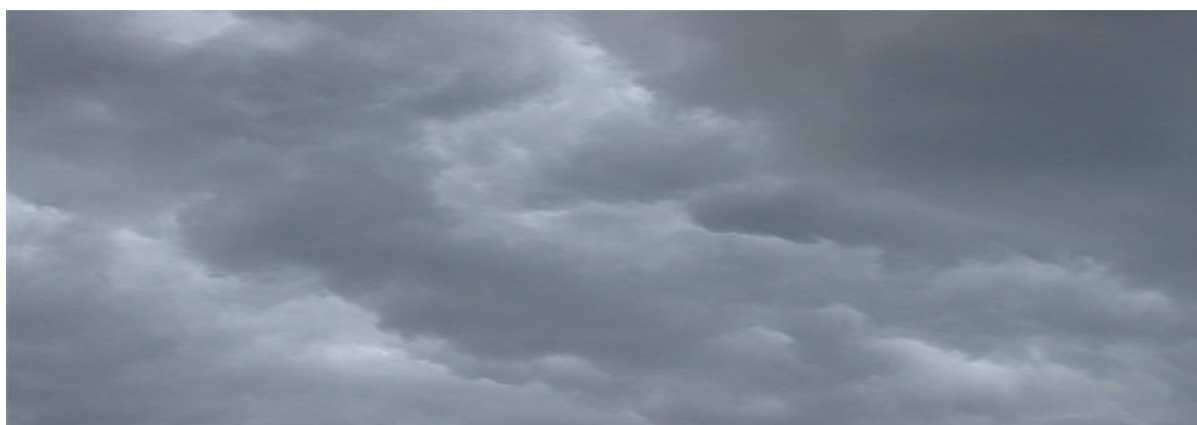
Oblaky druhu Altostratus mají vzhled šedavé nebo modravé oblačné plochy nebo vrstvy se strukturou vláknitou nebo žebrovitou nebo též bez patrné struktury, pokrývající úplně nebo částečně oblohu. Je tak tenká, že místy jsou patrné alespoň obrysy Slunce jako za matným sklem. U altostratu se halové jevy nevyskytují.



Obrázek 2.5 - Altostratus

2.1.2.6 Nimbostratus

Nimbostratus má podobu šedé, často tmavé oblačné vrstvy, která vlivem vyprázdňování více méně trvalých dešťových nebo sněhových srážek má matný vzhled. Srážky většinou dosahují země. Vrstva je všude tak hustá, že poloha Slunce patrná není.



Obrázek 2.6 - Nimbostratus

2.1.2.7 Stratocumulus

Oblaky druhu Stratocumulus mají vzhled jako šedé nebo bělavé, popř. obojí barvy, menší nebo větší skupiny nebo vrstvy oblaků, které téměř vždy mají tmavá místa. Oblak se skládá z částí podobných dlaždicím, oblázkům, valounům apod., nemívá vláknitý vzhled (s výjimkou zvláštního případu virga). Jednotlivé části oblaku buď spolu souvisí nebo mohou být oddělené.



Obrázek 2.7 -Stratocumulus

2.1.2.8 Stratus

Stratus je oblačná vrstva, obvykle šedá, s celkem jednotvárnou základnou, z níž vypadává mrholení, ledové jehličky nebo sněhová zrna. Prosvítá-li vrstvou stratu Slunce, jsou jeho obrysy zřetelně patrné a nikoliv rozplizlé jako v případě **altostratu**



Obrázek 2.8 - Stratus

2.1.2.9 Cumulus

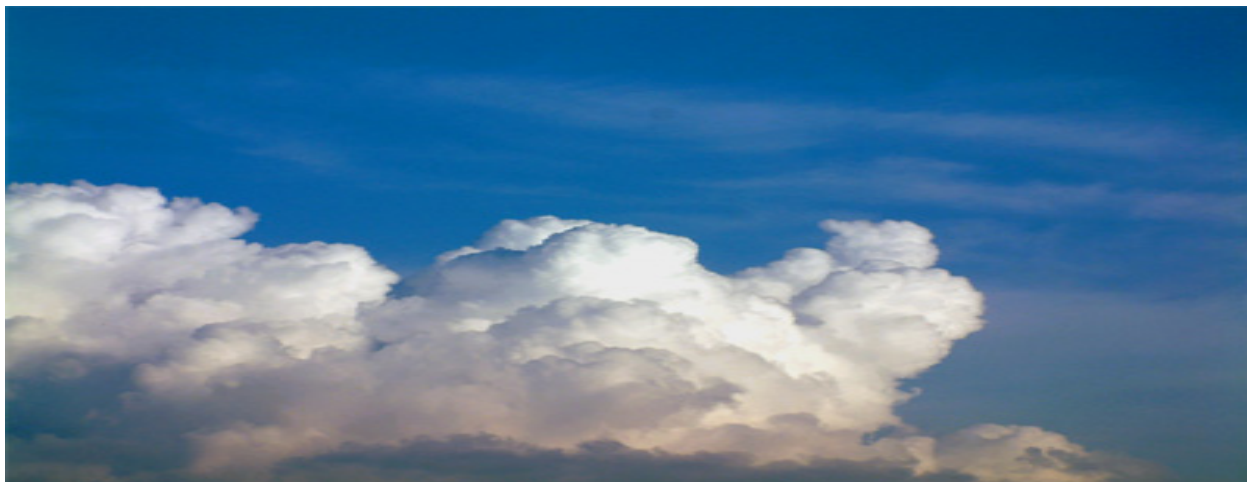
Cumulus je osamocenou zářivě bílou až našedlou kupou s ostře ohraničenými obrysy ve fázi rozvoje. Cumulus se rozvíjí směrem vzhůru a formuje se do podoby kup a věží Cumulus prochází několika stádií rozvoje. V počátcích svého vzniku má podobu neuspořádaných chomáčů s dosud ne zcela srovnanou spodní základnou. Za příznivých podmínek se tato oblaka dále rozvíjejí do tvarů s ostře ohraničenými obrysy a původně plochý oblak získává jasně definovaný vertikální rozměr. Oblak má oslnivě bílou barvu a má tendenci spojovat se do větších celků. Oblak se skládá z vodních kapiček a je-li vertikálně dostatečně vyvinutý, obsahuje jeho vrchní část i ledové krystalky. Je zdrojem krátkodobých srážek-přeháněk.



Obrázek 2.9 -Cumulus

2.1.2.10 Cumulonimbus

Cumulonimbus bouřkový oblak. Mohutný, hustý, často hrozivě vyhlížející kupovitý oblak velkých horizontálních a hlavně vertikálních rozměrů. Vrchol se často zplošťuje a rozlévá do podoby vějíře, závoje, nebo kovadliny. Tento jev je způsoben výškovým prouděním a prorůstáním vrcholů cumulonimbů až do spodních vrstev stratosféry. Spodní základna se obvykle nachází ve spodním patře, avšak vrcholky nezdávka dosahují vysokého patra a nacházejí se ve výškách 7-9 km. Základna je obvykle velmi tmavá a často se pod ní vyskytují protrhané oblaky.



Obrázek 2.10-Cumulonimbus

3 Coin3D a SimVoleon

3.1 Coin3D

Coin3D je nadstavbou grafického systému OpenGL. Využívá stromové struktury, pro srozumitelnější vyjádření vztahů objektů ve scéně. Kdy ke každému objektu (uzlu) scény můžeme definovat jeho chování vůči otci (nadřazeného uzlu). Postupným vytvořením závislostí uzlů, můžeme vytvořit komplexní scénu .

3.2 SimVoleon

Sim voleon je nástavbová knihovna vytvořena pro Open Inventor pro podporu volumetrického rendrování. Je to velmi nová a úplně neprozkoušená knihovna, která je ve stádiu vývoje a zkoušení.

4 Metody modelování mraků

Obecně metody pro modelování mraku lze rozdělit do dvou základních směrů, kdy na jedné straně leží metody založené na fyzikální podstatě využívající meteorologických a fyzikálních procesů , které vedou k tvorbě a zániku mraků. Na druhé straně je vytvářen směr tzv. ontogenický, který se snaží zachytit vizuální podstatu a neřeší výpočtově časově náročné operace některých jevů, které mají na vzhled nepatrný vliv. Tento směr je tlačen především počítačovými hrami a leteckými simulátory kde je kladen velký důraz na rychlost zobrazení.

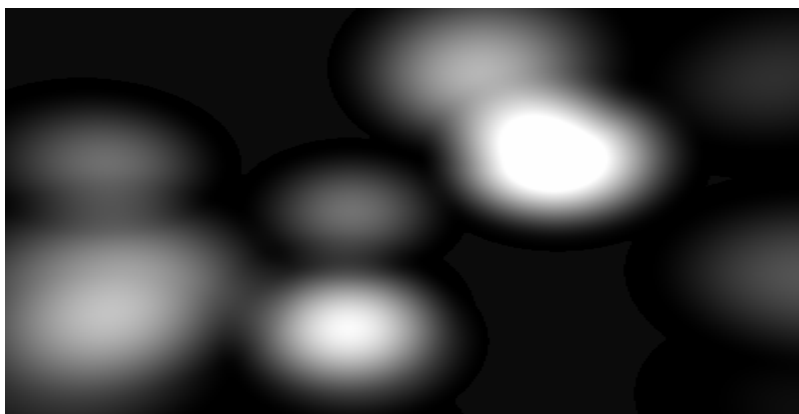
4.1 Particle systém

Objekty vytvořené tímto systémem jsou složeny ze souboru malých částí , které jsou reprezentovány několika jednoduchými částmi jež mají několik atributů jako jsou barva, poloměr a textura. Jednotlivé částičky jsou buď vytvořeny ručně v některém z grafických programů, nebo pomocí procedurálního generátoru, popř. kombinací obou.

Výhodou této metody je jednoduchost a časová nenáročnost na výpočet, která se ale postupně zmenšuje s požadavkem na detail a tím se stávají ostatní metody výhodnější.

4.2 Metaballs

Metaballs jsou reprezentovány množinou zdrojů, kdy každý tento bod má definovaný střed, poloměr a sílu. Samotný způsob metaballs není příliš vhodný pro tvorbu mraků, využívá se především pro tvorbu základu, který se dále pokrývá buď metodou fraktálu nebo texturou .



Obrázek 4.1 – Tvorba mraku pomocí metaballs (základ mráků)

4.3 Vizuální simulace

Je to metoda, která je spíše vhodná pro tvorbu statických modelů mraků. Základním stavebním kamenem této metody je elipsoid, který slouží k vystižení základního tvaru mraku. Na tento elipsoid je implementována texturovací funkce, která slouží k vytvoření detailu mraku pomocí modelování intenzity stínování a průsvitnosti. Implementace této funkce je využita spektrální funkce, které jsou založeny na furierově teorii, která říká , že každou funkci lze vyjádřit ve frekvenční oblasti součtem sinů a cosinů. Kdy při implementaci v OpenGL lze využít Perlinova šumu na místo součtu sinů a cosinů



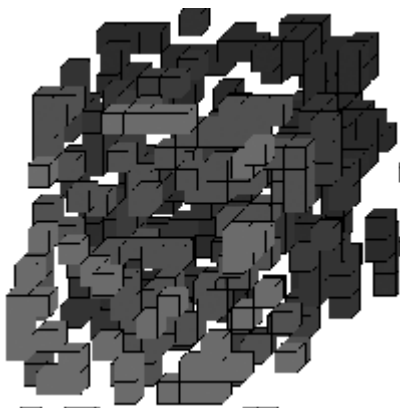
Obrázek 4.2 Jeden elipsoid s použitím texturovací funkce (perlinova šumu) a funkce průhlednosti



Obrázek 4.3 – Spojení několika elipsoidů

4.4 Pomocí buněčných automatů

Výhodou této metody jsou dobré vizuální výsledky, jednoduchost a tím spojená vysoká rychlost výpočtu.



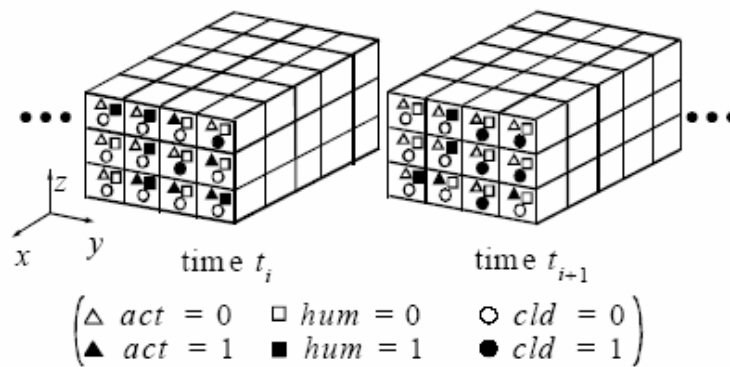
Obrázek 4.4 – buněčný automat

Nevýhody:

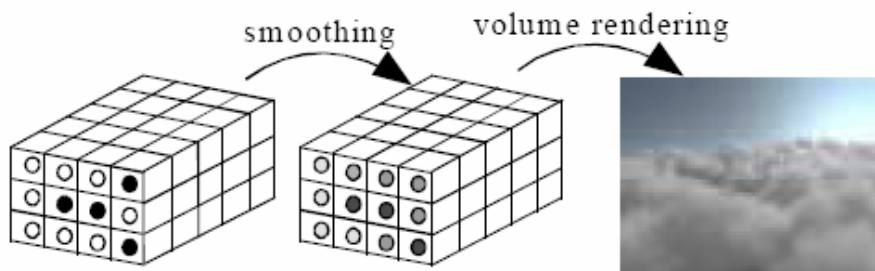
Schopnost zobrazovat pouze mraky typu cumulus (nadýchané velké mraky), dále tato metoda není příliš vhodná pro reálné zobrazování fyzikálních procesů vzniku a zániků mraků. Jak vidíme na obrázku prostor je rozdělen na buňky (voxeli), kdy každá je reprezentována trojrozměrným prvkem obdoby pixelu ve dvojrozměrném prostoru. Jednotlivé voxely jsou umísťovány do mřížky právě pomocí buněčného automatu. Jednotlivé buňky - voxely mají tři základní proměnné: vlhkost (hum), mrak (cld), act, která indukuje, že v daném místě může proběhnout změna (fázová přeměna). Kdy

jednotlivé proměnné nabývají pouze hodnoty TRUE (1) nebo FALSE (0) , což umožňuje bitové operace s použitím boolovy algebry.

Pro reálněji vypadající mraky je nutné vyplnit i prázdný prostor mezi buňkami jejíž proměnná *cld* je TRUE, kdy tato jednobitová hodnota je nedostačující. Prostor lze zjemnit vypočtením spojitého rozložení hustoty v každé buňce.



(a) Simulation process.



Obrázek 4.5 – Ukazuje jakým způsobem se dá využít buněčného automatu

4.5 Procedural Noise

Procedural noise je metoda která patří také mezi důležité metody při tvorbě mraků. Kdy využívá šumu jako základu na nějž aplikuje další metody pro vznik celistvého tvaru jež ve výsledku připomíná model mraku

5 Modelování mraku

Mnoho lidí při snaze vytvořit něco nepředvídatelného využívá náhodných čísel, aby výsledek se přiblížil přírodnímu vzhledu, jež je nám (jako pozorovateli) mnohem bližší než přesně definované tvary. Jestliže se podíváme na svět kolem sebe, můžeme spatřit nepřeberné množství tvarů, objektů jež se skládají z ještě většího množství detailů, kdy ať vezmeme jakékoli dvě věci kolem sebe nikdy nebudou stejné, budou se vždycky lišit v detailech, něčím co předem nemůžeme popsat ani přesně definovat. Můžeme říct že vše co příroda kolem nás vytváří se skládá z těchto nepřesností. Kdy výsledek je na pohled mnohem přirozenější.

5.1 Perlinův šum

Perlinův šum se stal základem pro vytváření realistických přírodních textur a tvarů. Ken Perlin vyvinul perlinův šum (perlin noise) na počátku 80 let a v dalších letech dále pokračoval na vývoji. Velmi rychle se stal základem, bez kterého se v dnešní grafice neobejdeme, především pro jeho jednoduchost a velmi příjemný výsledek. V roce 1997 dostal ocenění za jeho práci a přínos v oboru počítačové grafiky.

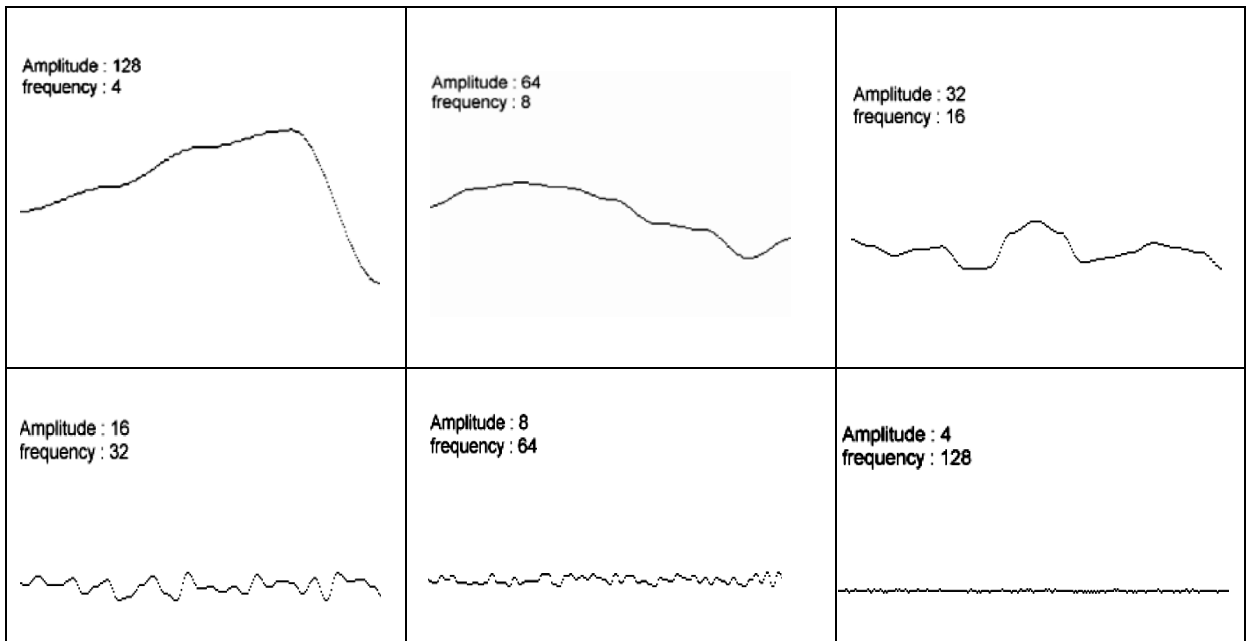
5.1.1 Tvorba Perlinova šumu

Perlinův šum napodobuje přírodní fenomén pomocí spojování šumů, které vznikli s různou amplitudou a frekvencí.

Pro nejlepší pochopení mějme příklad skalnatých hor. Jestliže se na pohoří díváme z velké dálky je tvořeno jednoduchou křivkou bez větších detailů (v našem případě jedna vrstva šumu) , ale pokud se podíváme na dané pohoří blíže spatříme spoustu detailů , kdy s každým dalším přiblížením se tato množina zvětší a zjemní a právě na této myšlence je založen Perlinův šum.

Na obrázcích níže jsou jednotlivě znázorněny funkce s různou frekvencí (počet opakování na reálné číslo) a amplitudou (výškou "vlny") , kde spodní funkce je součet všech předchozích.

Jestliže spojíme vyhlazovací funkci s funkcemi s různou amplitudou ,frekvencí a dáme (sečteme) vše dohromady získáme perlinův šum.

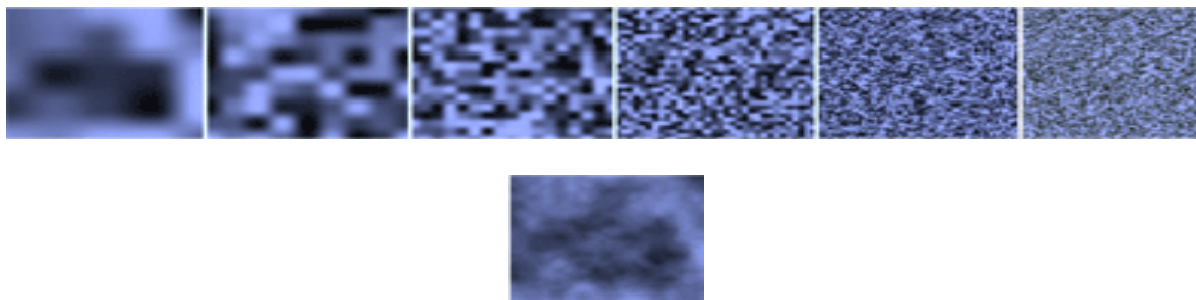


Sum of Noise Functions = (Perlin Noise)



Obrázek 5.1 – Perlinův šum s různou frekvencí a amplitudou, ale je vyobrazen součet těchto funkcí

Jestliže to samé provedeme ve dvou rovinách a začneme s nízkou frekvencí a rozlišením obrázku $16 * 16$ a postupně danou frekvenci zvyšujeme a tím nutně i rozlišení až na hodnotu $256 * 256$. Kdy jednotlivé snímky interpolujeme na stejné rozlišení (viz interpolace níže) pomocí bilineární interpolace. Kdy první snímek nám udává základní tvar výsledného mraku, postupně další snímky nám přidávají detaily. Nakonec získáme příjemně vyhlížející šum, který jako v předchozím případě je součtem všech .

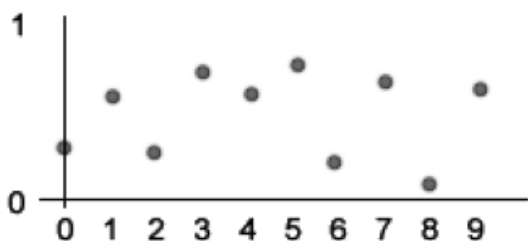


Obrázek 5.2 – Perlinův šum ve dvourozměrném poli.

5.1.2 Jádru perlinova šumu

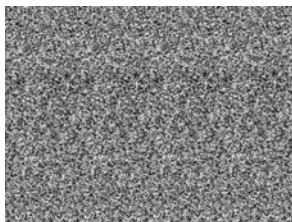
Jádru Perlinova šumu je matematická funkce, která na základě parametrů v podobě celých nezáporných čísel generuje pseudo náhodná čísla v rozmezí -1 a 1 . Jestliže do této funkce vložíme stejné parametry dostaneme vždy stejně “náhodné” číslo, což je velmi důležité.

Jednorozměrná funkce, nám ukazuje dvě hodnoty, jedna hodnota udává uplívající čas (diskrétní) a druhá náhodnou hodnotu jež je výsledkem funkce.



Obrázek 5.3 – Generátor náhodných čísel

Jak je vidět na dalším obrázku každý bod (pixel) má teď definovanou hodnotu šumem v rozmezí 0 až 1 jež je výsledkem funkce kdy parametry tvoří souřadnice x a y (pro 2D). Tento “rozsypaný čaj” je základ pro Perlinův šum.



Obrázek 5.4 - “Čistý” šum (převeden na šedou stupnici)

5.1.2.1 Oktáva

Oktáva je jedna “vrstva“ šumu viz obrázek výše. Jinak řečeno je to množina šumů, která vznikla pomocí stejných parametrů (frekvence, atd.) Jestliže vezmeme danou množinu (obrázek) a položíme ji nad ten stejný a sečteme, získáme tak mnohem větší turbulenci provedeme tak dvě oktávy (opakované sečtení).

Při provedení několika oktáv a změnou parametrů (frekvence, atd.) získáme perlinův šum.

Je mnoho různých způsobů jak měnit parametry.

Jedna z nejpoužívanějších je metoda kdy frekvence začne 1 a postupně jí násobíme 2 pro každou oktávu, počet oktáv záleží na uživateli (obecně 6 stačí).

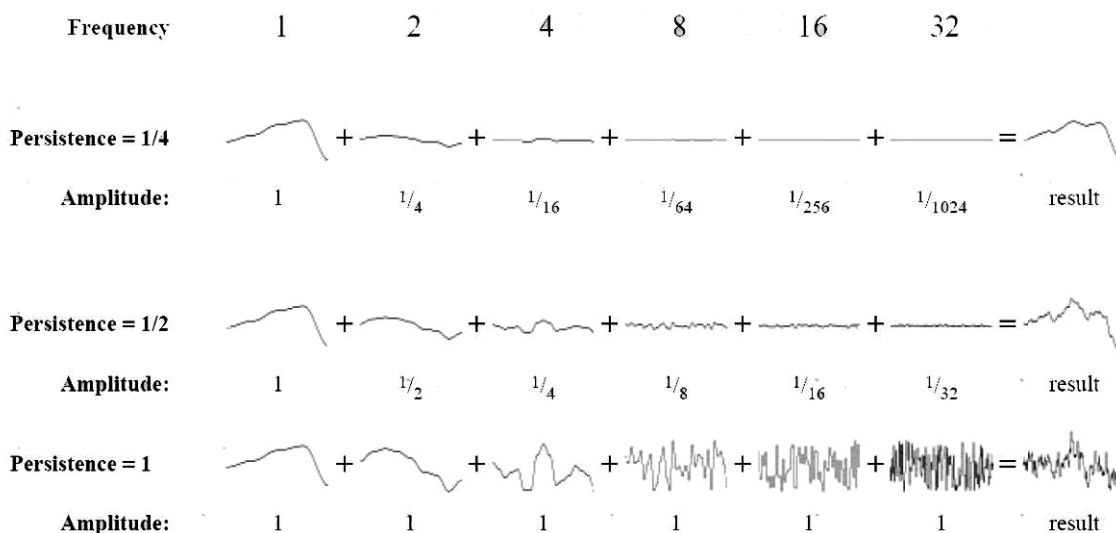
5.1.2.2 Persistenceence

Nyní víme jak vytvořit perlinův šum, ale pro vhodný výsledek je nutné použít perzistence.

Je to hodnota v rozmezí 0 – 1, jež reprezentuje zlomkovou hodnotu (váhu) dané oktávy. Používá se z důvodu, že čím větší oktáva tím má na výsledek větší vliv. Tím že snížíme perzistenci získáme mnohem přijatelnější výsledek, jak ukazuje obrázek.

Pro každou oktávu

Frekvence = frekvence * 2; Persistenceence = persistenceence * Persistenceence;



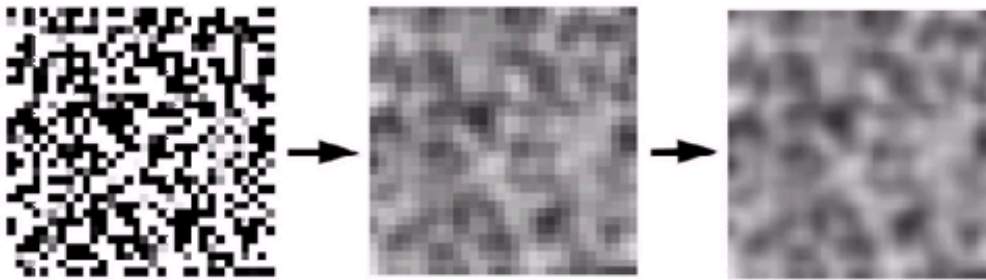
Obrázek 5.5 –Různé šumy při různých vstupních parametrech

5.1.3 Vyhlazení šumu

Čistý šum vyjadřuje náhodné hodnoty bez jakýchkoliv přechodů (sekvencí), spojení mezi jednotlivými hodnotami, můžeme jej přirovnat k diskretnímu signálu jež je na pohled příliš kostrbatý a vytvářejí zobrazení podobné rozsypanému čaji, kdy mezi jednotlivými pixely (sousedy) není žádný vztah, každý vzniká náhodně. a pro použití je nutné tyto hodnoty propojit.

Jeden z přístupů je metoda zkoumání sousedů, získáme hodnoty sousedů, spočítáme jejich průměr a tím získáme výsledný bod (pixel) jež je závislý na hodnotách vedlejších pixelů.

Pro názornost přikládám ještě jeden obrázek, kde na prvním snímku je vidět čistý šum, na následujícím snímku je provedeno vyhlazení původního šumu a na posledním snímku je převeden na výsledné rozlišení pomocí bilineární interpolace.



Obrázek 5.6 – Převod čistého šumu na vyhlazený

5.1.4 Interpolace

Interpolace (převzorkování) je metoda, která nám slouží k zvětšení nebo zmenšení rozlišení obrázku (objektu), hojně se používá u digitálních fotoaparátů jako digitální zoom. V dnešní době většina grafických programů nabízí několik základních metod převzorkování, které nabízí jak zvětšení tak i zmenšení rozlišení obrázku.

Existuje několik základních metod interpolace, které se dělí jak výsledkem, tak také časovou náročností. Čím lepší výsledek tím obecně časově náročnější.

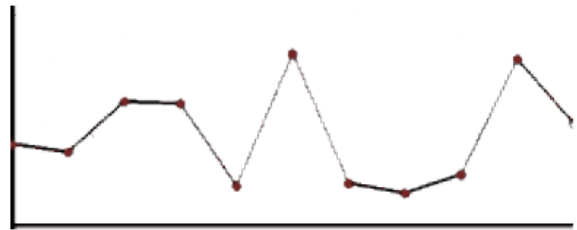
Obecně řečeno interpolace je funkce jež vrací číslo mezi dvěma hodnotami na základě třetí hodnoty (x),

jestliže se x rovna 0 tak výsledná hodnota je A, a naopak jestliže se rovná 1 tak hodnota je rovna B.

5.1.4.1 Lineární interpolace

Je to jednoduchý algoritmus, který je především používá pro realtime výpočet perlinova šumu

```
function Linear_Interp(a, b, x)
    return a*(1-x) + b*x
end
```

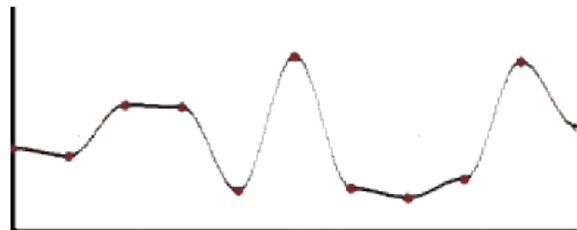


Obrázek 5.7 – Lineární interpolace

5.1.4.2 Kosinova interpolace

Tato metoda dává mnohem lepší (hladší) výsledek. Je o trochu časově náročnější ale vhodná i pro realtime výpočet perlinova šumu.

```
function Cosine_Interp(a, b, x)
    ft = x * 3.1415927
    f = (1 - cos(ft)) * .5
    return a*(1-f) + b*f
end
```

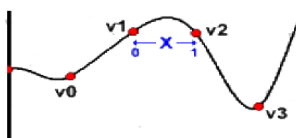


Obrázek 5.8–Kosinova interpolace

5.1.4.3 Kubická interpolace

Tato metoda dává velmi jemný výsledek, ale za cenu výpočetní a časové náročnosti. Ve skutečnosti se její výsledky liší jen v malých detailech od Kosinovy metody. Tato metoda je pro realtime aplikaci nepoužitelná, její výpočet je příliš složitý

Pro tuto metodu je nutné znát pět hodnot namísto třech



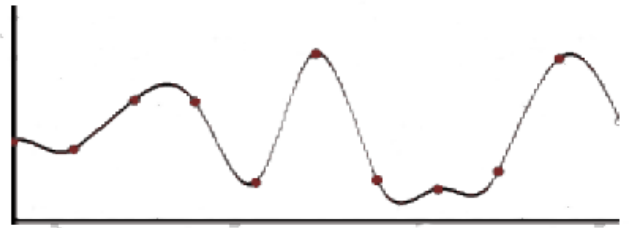
v_0 = bod před **a**, v_1 = bod **a**, v_2 = bod **b**, v_3 = bod po **b**

Obrázek 5.9 – Kubická interpolace

```

function Cubic(v0,v1,v2,v3,x)
    P = (v3 - v2) - (v0 - v1)
    Q = (v0 - v1) - P
    R = v2 - v0
    S = v1
return Px3 + Qx2 + Rx + S
end

```

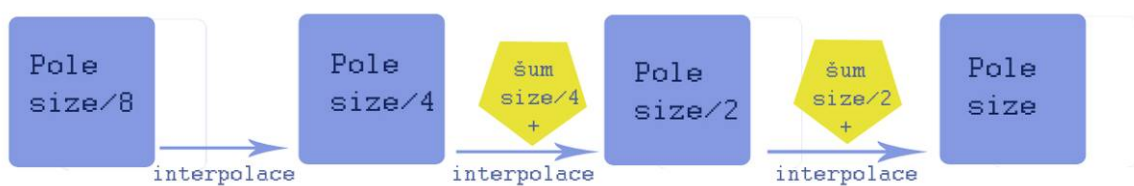


5.2 Tvorba mraků v 3D

5.2.1 Vytvoření základního pole

Abychom mohli zobrazit mrak musíme nejprve vytvořit 3D texturu mraku, kterou následně zobrazíme.

3D textura v našem případě je 3 rozměrné pole Postup vytvoření 3D textury popisuje obr. .



Obrázek 5.10 – Tvorba 3Dšumu

Kdy první pole (size/8) definuje základní tvar mraku a postupnou interpolací a přičítáním šumu mrak získává detailnější rozlišení.

Kdy každý prvek pole obsahuje hodnotu intenzity (průhlednosti – 0 pixel není vidět, 255-pixel je zcela neprůhledný).

Jestliže zobrazíme takto vzniklé pole viz obr. Je výsledek nepřijatelný a z takto vzniklé mlhoviny je nutné vyseparovat samostatný mrak .

5.2.2 Filtr šumu

Pro vyseparování se může použít exponenciální funkci, kdy si definujeme několik poměných, které nám definují co v daném šumu je mrak a co se nebude zobrazovat.

- cloudCover - jak moc velký mrak chceme mít
- cloudSharpness - hrana mraku, jak moc bude ostrá popř. vyhlazená.

```
c = (noise64[x][y][z]*255) - cloudCover;
```

```
if (c < 0) c=0;
```

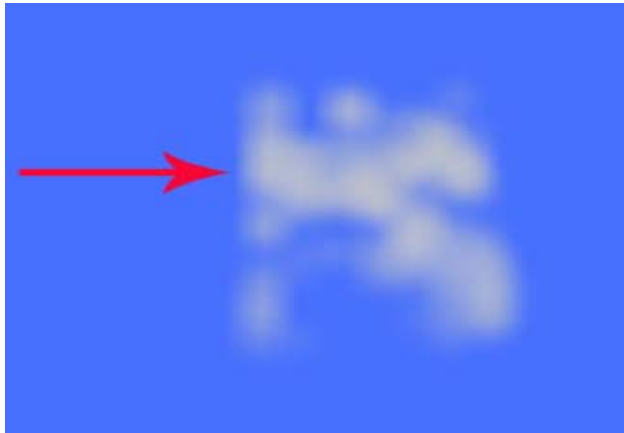
```
cloudDensity = 255 - ((pow(cloudSharpness, c)) * 255);
```



Obrázek 5.11 – Zobrazení šumu bez filtrace

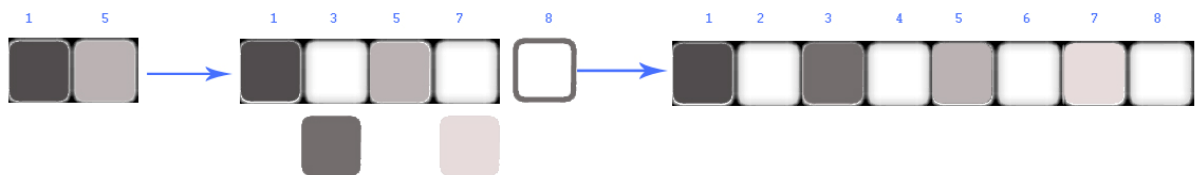
5.2.3 Problém s interpolací (mraku)

Jestliže zobrazíme takto náhodně vygenerovaný mrak (pole), zjistíme závadu s hranicemi jež definují mrak. Na obrázku níže je zřetelně vidět okraj pole ze kterého byl vyobrazen oblak.



Obrázek 5.12 – Zobrazení šumu s filtrací, ukazující na problém s filtrací

Problém vzniká při interpolaci na větší rozlišení, viz obrázek níže. Kdy první (1) pixel který se náhodně vygeneruje se postupnou interpolací na větší rozlišení přenáší směrem doprava. Kdy interpolujeme (zjemňujeme) směrem do středu, ale kraje se nemají kam zjemňovat. Tento jev se dá odstranit několika způsoby.



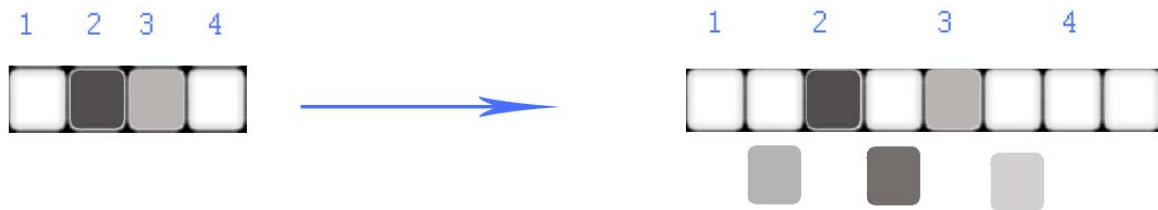
Obrázek 5.13 – Ukáka interpolace ve na řadě pixelů

5.2.3.1 Pomocí funkce SIN

Určíme si poloměr (do jaké míry budeme rohy zaoblovat) a postupně po jednotlivých souřadnicích zaoblíme pomocí funkce SIN. Toto řešení , ale neposkytuje uspokojující řešení.

5.2.3.2 Pomocí většího pole

Tato metoda je založena na principu interpolace kdy daný pixel má možnost interpolovat do všech stran a tím vytvořit kompaktní objekt ve tvaru oblak.



Obrázek 5.14 – Odstranění hran pomocí většího pole

Tato metoda inicializuje vstupní pole (nastaví kraje rovné nule). Vloží první prvky šumu do pole (mimo krajní) provede se interpolace (zvětšení rozlišení). K takto vzniklé matici se přičítá další matice šumů s jinou amplitudou (viz Perlinův šum)jen s tím rozdílem že se nezaplňují krajní pixely. Toto opakujeme dokud nedosáhneme požadovaného rozlišení.

Takto vznikající mrak se inicializuje v celém prostoru a jeho vznik a ani jeho výsledný tvar není nijak omezen. Viz obr.



Obrázek 5.15 – Mrak (šum+filtrace+větší pole)

Takto vzniklí oblak můžeme upravit pomocí měřítka v každé rovině na požadovaný mrak .



Obrázek 5.16– Zobrazení oblouhy

6 Letecký simulátor

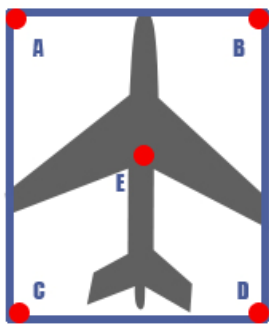
6.1 Model letadla

Model letadla byl exportován s 3DS do iv (podporovaný formát Open Inventoru) za pomoci volně dostupného exportéru 3DS2IV.

6.1.1 Naklápění a pohyb letadla

Pro naklápění letadla slouží 5 výškových bodů (A,B,C,D,E), kde E má funkci těžiště letadla a udává výšku letadla.

S měnícím se směrem jednotlivé výškové body buď stoupají nebo klesají v závislosti na době zmáčkuté ovládací klávesnici. Kdy na základě rozdílu jednotlivých výškových bodů se spočítá naklonění letadla v požadovaném směru.



Obrázek 6.1 – Letadlo s výškovými body

Pohyb letadla je závislý na směru naklopení letadla, které udává jakým směrem se letadlo bude pohybovat.

6.1.2 Ovládání letadla

- Šipka nahoru - klesání – způsobí naklonění špice letadla dolů a jeho následné klesání
- Šipka dolů - stoupaní – způsobí naklonění špice letadla nahoru a jeho následné stoupaní, aby letadlo mohlo stoupat musí mít rychlost.
- Šipka doleva – otočení doleva – způsobí naklonění levého boku dolů
- Šipka doprava - otočení doprava – způsobí naklonění pravého boku dolů
- W – zvýšení rychlosti (do definované maximální rychlosti)
- S – snížení rychlosti (min rychlost 0)

7 Závěr

Výsledný letecký simulátor prezentuje vytvořené mraky. Aplikace umožňuje ve scéně zobrazit rozmanité oblačnosti, zaleží na nastavení jednotlivých parametrů. Kdy pomocí letadla můžeme dané mraky prolétat.

Letecký simulátor je velmi výpočetně náročný a plynule jede pouze s menším rozlišením s tím se otvírá možnost pokusit se některé hodnoty, které se nemění před počítat a následně se na ně pouze odkazovat (klade velké nároky na paměť). Další urychlení aplikace by bylo použití impostorů na místo zobrazování celých volumetrických dat (oblaku)

Kdy by se aplikaci dokázalo urychlit, otevírala by se možnost jak zkvalitnit vizuální vzhled oblohy, který nepůsobí zcela realisticky.

Literatura

- [1] Ing. Pečiva, Jan: Seriál Open Inventor.
<http://www.root.cz/clanky/open-inventor>
- [2] Coin3D, knihovna a informace k ní
<http://ftp.coin3d.org/coin/src/Coin-2.4.0.tar.gz>
- [3] Tvorba Mraků
http://freespace.virgin.net/hugo.elias/models/m_clouds.htm
http://freespace.virgin.net/hugo.elias/models/m_clouds.htm
http://wscg.zcu.cz/wscg2002/Papers_2002/D37.pdf
http://www.gamasutra.com/features/20040114/wang_01-03.shtml
<http://www.markmark.net/cloudsim/harrisGH2003.pdf>
<http://www.vterrain.org/Atmosphere/Clouds/>
<http://www.intel.com/cd/ids/developer/asmo-na/eng/dc/games/20534.html>
http://nis-lab.is.s.u-tokyo.ac.jp/~nis/abs_sig.html
- [4] Perlinův šum
<http://www.noisemachine.com/talk1/11.html>