

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

REALISTICKÝ MODEL OBLOHY

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

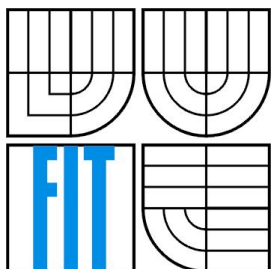
AUTOR PRÁCE  
AUTHOR

BC. ZDENĚK KUSSIOR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# REALISTICKÝ MODEL OBLOHY

REALISTIC SKY MODEL

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

BC. ZDENĚK KUSSIOR

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. MARTIN DRAHANSKÝ, PH.D.

BRNO 2007

## Zadání diplomové práce

Řešitel: **Kussior Zdeněk, Bc.**

Obor: Inteligentní systémy

Téma: **Realistický model oblohy**

Pokyny:

1. Nastudujte literaturu se zaměřením na modelování volumetrických mraků a mlhy (parametry, změny apod.) pro 3D scénu.
2. Navrhněte grafickou reprezentaci volumetrických mraků (a příp. mlhy) vhodnou pro databázi (3D prostředí) firmy E-COM.
3. Navrženou metodu prakticky implementujte a otestujte na databázi firmy E-COM.
4. Provedte experimenty a dosažené výsledky zhodnoťte.

Literatura:

- Dle specifikace školitele a průmyslového zadavatele.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Drahanský Martin, Ing., Dipl.-Ing., Ph.D.**, UITS FIT VUT

Datum zadání: 28. února 2006

Datum odevzdání: 1. května 2007

L.S.

---

doc. Dr. Ing. Petr Hanáček  
*vedoucí ústavu*

# **Licenční smlouva**

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

## **Abstrakt**

Dokument popisuje teoretickou přípravou a řešení realistické vizualizace volumetrických oblaků pro prostředí simulátoru, běžícího v reálném čase. V první části se zabývám meteorologickým základem celého problému a uvádím mezinárodní klasifikaci oblaků do deseti základních kategorií včetně popisu jejich vzhledu, výskytu a okolností vzniku. Druhá část se zabývá zpětnou interakcí oblaků se simulátorem. Ta dává oblačnosti charakter mechanické a elektromagnetické překážky, se kterou je v realistické simulátoru techniky vhodné počítat. Dále popisuji některé technologie i konkrétní implementace vizualizace a hodnotím jejich vlastnosti. V poslední části dokumentu popisuji samotnou implementaci systému zobrazování volumetrických oblaků a pokouším se popsat směr, kterým bude možné projekt v budoucnu dále rozvíjet.

## **Klíčová slova**

volumetrický, mrak, oblak, mlha, oblačnost, klasifikace oblaků, srážky, déšť, sníh, kroupy, námraza, elektromagnetické záření, vizualizace, průhlednost, impostor, osvětlovací model, shader, OpenGL

## **Abstract**

The paper describes a theoretical base and realization of realistic volumetric clouds visualization in an environment of real-time simulator. The first part is concerned with a meteorological background of this problem. I show international classification of ten basic cloud types including a short description and cases of occurrence. The following part is concerned with an interaction between cloudiness and simulation core, which is based on the fact, that each cloud acts as a mechanical or an electromagnetic obstacle. This should be considered on some way in simulation. The next part describes technologies and practical implementations of visualization and evaluates their characteristics. Finally, the last chapter describes my implementation and tries to outline project advancement.

## **Keywords**

volumetric clouds, cloud, fog, cloudiness, classification, precipitation, rain, snow, hail, icing, electromagnetic radiation, visualization, transparency, impostor, shading model, shader, OpenGL

## **Citace**

Zdeněk Kussior: Realistický model oblohy, diplomová práce, Brno, FIT VUT v Brně, 2007

# Realistický model oblohy

## Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením

Ing., Dipl.-Ing. Martina Drahanského, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jméno Příjmení  
Datum

## Poděkování

Rád bych poděkoval Martinu Drahanskému za vedení projektu a Václavu Bílkovi, jakožto zástupci průmyslového zadavatele projektu (firma E-COM simulation & training systems). Dále děkuji Davidu Dosoudilovi, Michaelu Borůvkovi a také internetové komunitě, zabývající se letectvím, za poskytnutí praktických zkušeností s mnohými problémy, popisovanými v tomto dokumentu.

© Zdeněk Kussior, 2007.

*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.*

# Obsah

Obsah .....	6
1 Úvod.....	8
2 Meteorologický základ.....	9
2.1 Definice oblaku .....	9
2.2 Vzhled oblaku .....	9
2.3 Vlhkost vzduchu.....	9
2.4 Vznik a vývoj oblačnosti.....	10
2.5 Klasifikace oblaků.....	11
2.6 Typy mlh .....	16
2.7 Vertikální rozložení oblačnosti .....	18
3 Návrh.....	19
3.1 Požadavky v kontextu cílové aplikace .....	19
3.2 Modularita .....	20
3.3 Interakce se simulátorem.....	22
3.3.1 Srážky .....	22
3.3.2 Námraza .....	22
3.3.3 Stínění elektromagnetického vlnění.....	24
3.4 Směr další práce .....	27
4 Vizualizace oblačnosti .....	28
4.1 Implicitní tělesa .....	29
4.2 Harrisův přístup.....	30
4.3 Shluky texturovaných ploch.....	32
4.4 Impostor .....	33
4.5 Další technologie.....	36
4.6 Stín .....	37
4.7 Optimalizace.....	39
4.8 Existující implementace .....	39
4.9 Směr další práce .....	43
5 Implementace .....	44
5.1 Třídní model .....	44
5.2 Simulace .....	46
5.3 Modelování základního tvaru.....	47
5.4 Textury .....	49
5.5 Stínování .....	49

5.6	Geometrická reprezentace oblaku .....	51
5.7	Vykreslovací řetězec scény .....	53
5.8	Rychlost výpočtů a zobrazování.....	54
5.9	Směr další práce .....	58
6	Závěr .....	60
	Literatura .....	61
	Seznam Příloh.....	62



# 1 Úvod

S oblačností a doprovodnými jevy se každý z nás dennodenně setkává. Dobře známe tvary, barvy, ale i časově závislé chování oblaků. Při dostatečných výtvarných schopnostech dokážeme nejzákladnější typy oblaků velmi věrně reprodukovat na papíře a to i bez znalosti jakékoliv podrobnější teorie. Na základě předchozích zkušeností dokonce dokážeme letným pohledem určit, zda nám z daného oblaku nebo celkové situace na obloze hrozí nějaká nebezpečí či komplikace – ať už v krátkodobém horizontu, například lokální přeháňka nebo bouřka, tak i s několikadenním předstihem, například blížící se změna počasí. Přesto pro vytvoření systému pro realistickou vizualizaci oblačnosti na počítači nejsou tyto znalosti dostatečné. Je potřebné znát fyzikální pozadí řešeného problému, uvést závěry pozorování, které byly provedeny a zdokumentovány odpovědnými institucemi, ústavy či vědci. S tímto úzce souvisí i ustálená terminologie. Procesy v atmosféře jsou velmi obsáhlé téma, s vybranými útržky z něj bych se čtenáře pokusil seznámit v první části tohoto dokumentu.

V části druhé bych se rád zaměřil už poněkud konkrétněji na cílovou aplikaci. Tou je virtuální prostředí s realistickou (nikoliv abstraktní) krajinou, domy, komunikacemi, vegetací, technikou a postavami. Cílem projektu je pro toto prostředí připravit věrohodně vypadající oblačnost, ve které bude možné se volně pohybovat, prolétávat jí. Vizualizace ovšem nemusí být jediným požadovaným výstupem. Situaci na obloze je nejprve nutné popsat, nadefinovat nebo vygenerovat. Dále je třeba myslet na zpětnou vazbu směrem zpět do simulačního jádra – typickým příkladem může být vznik námrazy na křídle letounu při průletu skrze oblak. Celou tuto část budu brát v kontextu cílové aplikace, kterou je realistický simulátor pozemní a především letecké (vojenské) techniky. Ne všechna popsaná témata budou implementována, tuto kapitolu lze z velké části brát, jako nástin možného budoucího vývoje projektu.

V další části dokumentu se budu věnovat různým přístupům k vizualizaci. Popíši dnes používané techniky, jejich klady, zápory a limitace. Prostředí, pro které je systém zobrazování oblaků vyvíjen, má charakter počítačové hry, přesněji (leteckého) simulátoru. Proto je na místě ukázat, jak si s řešením problému poradili tvůrci některých známých počítačových leteckých simulací. Mnohdy ještě zajímavějších, ovšem ne vždy prakticky použitelných výsledků dosahují experimentální, univerzitní implementace.

Závěrečná část bude zaměřena na samostatnou implementaci. Popíši strukturu celého systému, řešení zásadních problémů a ukáži grafický výstup systému při různých podmínkách scény. Dokument uzavřu shrnutím, co vše bylo vytvořeno, s jakými omezeními a jaký bude směr dalšího vývoje projektu.

## 2 Meteorologický základ

Před jakýmkoliv návrhem řešení zadaného problému, tedy přípravy a vizualizace oblačnosti, je třeba si ujasnit fyzikální pozadí dějů souvisejících s vývojem oblačnosti. Vědou, která se tímto tématem zabývá, je meteorologie, část fyziky studující složení, stavbu, vlastnosti a jevy probíhající v atmosféře. S meteorologií je pevně spjata i terminologie, o které se zmíním v souvislosti s klasifikací oblaků.

### 2.1 Definice oblaku

Oblak, jakožto meteorologický útvar, je chápán jako v ovzduší se vznášející viditelný shluk malých částic vody, případně ledu. Obecně lze jako oblak považovat shluk i jiných částic, jako např. viditelné plyny, kouř, prach, v extrémním případě větší objekty, jako je kamení a sopečné produkty (pyroklastický oblak) [Skr06].

### 2.2 Vzhled oblaku

Vzhled oblaku je dán povahou, rozměry, množstvím a prostorovým rozložením částic, intenzitou a barvou světla, které na oblak dopadá a vzájemnou pozicí pozorovatele, oblaku a zdroje světla. Vzhled oblaku se popisuje veličinami rozměr, forma, struktura, stavba, jas, barva, přičemž poslední dvě jsou nejdůležitější [Skr06].

**Jas** závisí na množství odraženého, rozptýleného a propouštěného světla oblakem. Zdroj světla je typicky bodový s téměř paralelními paprsky (od Slunce a Měsíce), dále rozptýlený, přicházející od oblohy nebo jiné oblačnosti, a konečně odražený – sluneční nebo měsíční světlo odražené typicky od rozsáhlých zasněžených ploch. Jas oblaků za dne je dostačující, v noci je k jejich pozorování potřeba svit alespoň čtvrtiny měsíčního kotouče.

**Barva** oblaku závisí především na barvě světla. Zákal, tedy vysoká úroveň vlhkosti v atmosféře, řídká mlha či jiné částice v ovzduší, může barvu vzdálených oblaků změnit do žluta, oranžova až červena. Pokud je Slunce vysoko nad obzorem, jsou osvětlené části oblaků bílé nebo šedé. Části osvětlené rozptýleným světlem z modré oblohy jsou modrošedé. [Skr06]

### 2.3 Vlhkost vzduchu

Vlhkost vzduchu je veličina vyjadřující množství vodní páry obsažené ve vzduchu, není do ní započítáno množství vody v kapalném a pevném skupenství, které se ve vzduchu může vznášet. Pro

každou teplotu existuje maximální hodnota vlhkosti vzduchu, kterou nelze překročit – tento stav se nazývá stav nasycení.

Teplota [°C]	-30	-20	-10	0	10	20	30
Množství vodní páry na jednotku objemu vzduchu [g / m <sup>3</sup> ]	0.3	1.1	2.3	4.9	9.4	17.3	30.4

Tab. 2.1: Množství vodní páry v nasyceném vzduchu.

Vlhkost vzduchu vyjadřujeme čtyřmi způsoby [Vas06]. **Absolutní vlhkost vzduchu** v jednotkách g / m<sup>3</sup> udává aktuální množství vodní páry ve vzduchu, při dané teplotě nemůže překročit hodnotu množství vodní páry v nasyceném vzduchu. **Rosný bod** je teplota, při které by se současným množstvím vodní páry ve vzduchu nastal stav nasycení. Teplota rosného bodu je jednou ze základních a nejdůležitějších vlastností atmosféry, zjistí se měřením. Klesne-li teplota pod rosný bod, začíná pára kondenzovat. **Relativní vlhkost** je poměr aktuálního (e) a maximálního možného (E) tlaku vodní páry v totožných podmínkách. Jde o známé vyjádření vlhkosti v procentech. Konečně **tlak vodní páry** vyjadřuje tlak páry, pokud by se v daném objemu vzduchu vyskytovala sama. Tlak vodní páry dosahuje maximální hodnoty (E) při stavu nasycení. Tlak vodní páry při nasycení je závislý na teplotě a skupenství vypařující se vody. Při vypařování z ledu je tlak oproti vypařování z vody o několik desítek procent nižší při teplotách pod bodem mrazu.

T [°C]	-50	-40	-30	-20	-10	0	10	20	30	40	50
E [mm Hg]	0.04	0.13	0.36	0.91	2.11	4.54	9.17	17.51	31.82	55.35	92.57

Tab. 2.2: Tlak nasycené páry v milimetrech Hg při vypařování z vody.

V častých případech dochází k rovnováze lokálního nasyceného vzduchu kolem vodních kapiček a okolního vzduchu s relativní vlhkostí jen 80%. Pokud k tomu dojde, může oblačnost přetrvávat i pokud vlhkost klesne pod 100%.

## 2.4 Vznik a vývoj oblačnosti

Oblaka vznikají přeměnou neviditelné vodní páry na kapalnou nebo pevnou formu. Tento jev se nazývá kondenzace v případě přeměny na kapalinu – malé kapičky vody, desublimace v případě přeměny na pevnou formu – ledové krystalky. Nastává, když je vzduch maximálně nasycen vodními parami, tedy při 100% vlhkosti, a teplota klesne pod teplotu rosného bodu. K jevu dochází při vzestupu výparů z moře, jezer, řek, ale i z půdy, sněhu a ledu do vrstvy vzduchu, kde dojde ke kondenzaci, nebo mísením dvou vzduchových hmot s vysokou vlhkostí a rozdílnou teplotou a podobně. Pokud se vodní kapičky začnou spojovat dohromady, roste jejich průměr na desetiny až jednotky milimetrů. Takto těžké kapky se již neudrží v oblaku a padají k zemi ve formě deště. Při nízké teplotě se neformují kapky, ale sněhové vločky. Oblakem typu cumulonimbus dešťové kapky

stoupají několik kilometrů, ve vyšších výškách mrznou, při pádu směrem dolů na sebe nabalují další kondenzovanou vodu a stávají se z nich až několikacentimetrové kroupy.

Simulace vývoje oblačnosti, obecně počasí, patrně nebude součástí řešení projektu, proto se jen velice krátce zmíním, že při modelování proudění vzduchu a vzniku oblačnosti je třeba brát v úvahu aktuální synoptickou situaci, trojrozměrný model terénu a v neposlední řadě také sadu map – mapa vodních ploch, mapa vegetačního pokryvu a podobně.

## 2.5 Klasifikace oblaků

Prvním, kdo se pokusil o třídění oblaků byl francouzský vědec Jean Baptiste Lamarck. V roce 1802 publikoval klasifikaci s pěti hlavními a několika doplňkovými typy oblaků. O rok později přišel Angličan Luke Howard s názvoslovím, založeným na třech hlavních druzích oblaků: Cirrus – vláknité, Cumulus – kupovité, Stratus – vrstevnaté oblaky. Postupně se objevovaly další obměny, zpřesnění a rozšíření Howardovy klasifikace. V této souvislosti se objevují jména jako E. Renou, Hildebrand M. Hildebrandsson a Ralph Abercromby. Poslední dva jmenovaní koncem 80. let 19. století navrhli skutečně moderní soustavu klasifikace včetně přesné definice, doplněné atlasem věrně zobrazených oblaků. U nás se tématem ve 40. letech zabýval Dr. Antonín Bečvář, který se v roce 1953 spolupodílel na vydání obsáhlého „Atlasu horských mraků“. Konečně v roce 1956 vydala Mezinárodní meteorologická organizace (WMO) „Mezinárodní atlas oblaků“. Rozlišujeme deset základních druhů oblaků. Za jejich latinské označení se přidává konkretizující latinský přívlastek, specifikující tvar, odrůdu, zvláštnost a průvodní oblak. Na několika následujících stranách je přehled latinských označení, zkratk, českých ekvivalentů a popisů oblaků, případně zmínka o některých významných odrůdách. Informace jsem čerpal především ze [Skr06] a [Ast87].

U popisovaných oblaků neuvádím fotografie. Obsáhlou galerii všech druhů oblaků včetně mnoha odrůd, tvarů, zvláštností a průvodních mraků lze nalézt ve zdroji [Atl06] a [Skr06].

### ***Cirrus (Ci, řasa)***

Cirry jsou oddělené obláčky v podobě jemných bílých vláken, úzkých pruhů nebo řidčeji hustých šedavých plošek. Vlákna jsou často zakončena vzhůru směřujícími háčky. Tyto oblaky, složené z ledových krystalků, mají hedvábný lesk a žádný stín. Pokud je Slunce vysoko nad obzorem jsou oblaky cirrus bílé, bělejší než kterékoliv oblaky ve stejném místě na obloze. Je-li Slunce na obzoru, zachovávají si svou světlou barvu, přičemž níže položená oblačnost získává žlutý až oranžový nádech. Po západu Slunce se barví od žluté, přes růžovou a červenou až k šedé, při východu Slunce naopak. Cirrus blízko obzoru je často žlutavý až oranžový. Na cirru se mohou vyskytnout neúplně

halové efekty<sup>1</sup>. Celková velikost cirru na obloze je výrazně menší, než u cirrostratu a altostratu, je to jeden z rozlišovacích znaků.

Cirrus vzniká z cirrocumulů, altocumulů nebo z nejvyšších rozšířených částí cumulonimbů. Může vzniknout také z cirrostratu, u kterého se tenčí části vypaří. Cirrus ve formě chomáčků se zaoblenými vrcholy se často objevuje i na jinak bezoblačné obloze.

### ***Cirrocumulus (Cc, řasová kupa)***

Tenké skupiny nebo vrstvy bílých oblaků bez vlastního stínu, složené z velmi malých oblačných částí, podobných zrnkům nebo vlnkám. Tyto jednotlivé části jsou pravidelně uspořádány a mohou být vzájemně oddělené, případně spolu souviset. Velikost jednotlivých částí je velmi malá a obvykle nepřesahuje 200 metrů, právě malá velikost částí je jedním z rozlišovacích znaků od typu altocumulus. Druhým je absence stínu. Cirrocumulus může vyskytnout také v malých plochách v podobě čočky s obvykle ostrými okraji.

Tato oblaka jsou složena z ledových krystalků nebo ve velmi malém množství ze silně ochlazených vodních kapek. Vždy jsou dostatečně průsvitné, lze přes ně dobře určit polohu Slunce a Měsíce.

Cirrocumulus může vzniknout na bezoblačné obloze nebo se vyvinout z typu cirrus a cirrostratus nebo zmenšením rozměrů částí altocumulu. Čočkovitý cirrocumulus vzniká vynuceným zvednutím vrstvy vlhkého vzduchu například nad vysokým horským hřebenem.

### ***Cirrostratus (Cs, řasová sloha)***

Průsvitný bělavý závoj hladkého mlhavého nebo vláknitého vzhledu s případným jemným žebrováním, zakrývající velkou část oblohy. Okraj je někdy výrazně ohraničen a nebo častěji přechází v cirru. Cirrostratus je tak tenký, že předměty na zemi vrhají stín. Změna barvy při různých světelných podmínkách je stejná, jako u cirru.

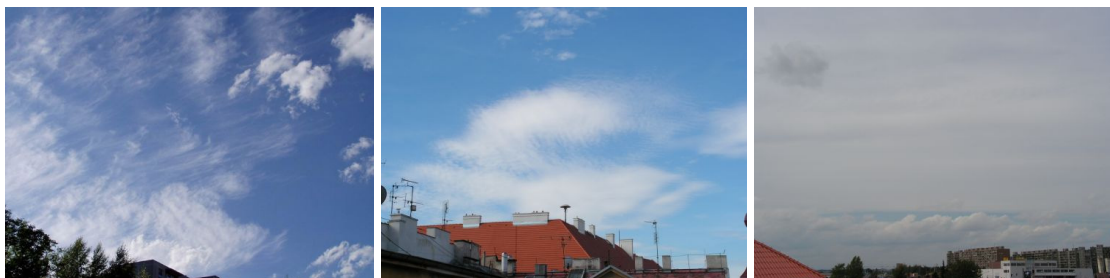
Složení z ledových krystalků dává při průchodu světla skrz vznik halovým efektům. Halové efekty dokonce prozradí přítomnost cirrostratu v případech, kdy je okem prakticky nerozeznatelný od modré oblohy.

Cirrostratus vzniká při pozvolném zvedání vrstvy vzduchu do velkých výšek. Může také vzniknout spojením cirrů, slitím jednotlivých částí cirrocumulů do jednolitého útvaru nebo vypadáváním ledových krystalů z cirrocumulu. Dále může vzniknout také ztenčením altostratu nebo rozšířením kovádliny cumulonimbu.

---

<sup>1</sup> Halové efekty jsou optické úkazy, objevující se na obloze kolem Slunce a Měsíce. Vznikají odrazem a lomem světla na šestibokých ledových krystalech v oblačnosti vysokého patra nebo při velmi nízké teplotě i níže. Nejběžnějším typem je bělavý nebo duhový kruh kolem Slunce o poloměru 22°. Oblaky složené z vodních kapiček dávají za vznik podobným duhovým a perleťovým efektům – korónám, irizacím a gloriolám.

Od altostratu se krom menší tloušťky a halových efektů liší pomalostí pohybů a změn.



Obr. 2.1: Cirrus, cirrocumulus, cirrostratus [Skr06].

### ***Alto cumulus (Ac, vysoká kupa)***

Menší nebo větší skupiny bílých nebo šedých oblaků mající vlastní stíny. Skládají se z malých částí podoby vln, oblázků apod., které mohou být samostatné nebo spolu souviset. Často mají částečně vláknitý vzhled. Jednotlivé části mají obvykle velikost v řádech stovek metrů. Alto cumuly jsou bývají rozmístěny ve dvou či více vrstvách. Vyskytují se někdy také v podobně značně protáhlých více méně hladkých čoček s ostře ohraničenými okraji, v tomto případě mají velmi výrazný vlastní stín. Průsvitnost alto cumulu je značně proměnlivá – někdy přes něj může být Slunce vidět, jindy jsou tak husté, že Slunce plně zakrývají. V případě neprůsvitného oblaku lze na spodní vrstvě pozorovat výrazný reliéf. Alto cumulus se skládá z vodních kapiček, při velmi nízké teplotě také částečně z ledových krystalů, které mohou dávat vzniku halových jevů.

Alto cumulus vzniká ve střední výšce na okraji stoupající vzduchové vrstvy, při turbulenci nebo konvekci<sup>2</sup>. Také může vzniknout zvětšením rozměrů částí cirrocumulu nebo rozdělením vrstvy stratocumulu. Vzniká též shlukováním mas altostratu nebo nimbostratu, případně také rozšířením cumulu nebo cumulonimbu. Čočkový alto cumulus vznikne při vynuceném výstupu vlhkého vzduchu typicky v horských oblastech.

### ***Altostratus (As, vysoká sloha)***

Šedavá nebo modravá plocha či vrstva s nerozeznatelnou, vláknitou nebo žebrovinou strukturou. Altostratus se skládá z několika těsně nad sebou umístěných vrstev, je horizontálně velmi rozsáhlý (až stovky kilometrů) a vertikálně mohutný (až tisíce metrů). Někdy na něm lze pozorovat vlny a široké rovnoběžné pruhy. Altostratus částečně nebo úplně zakrývá oblohu, jen v jeho některých částech je dostatečně tenký, aby bylo možné pozorovat rozmazané obrysy Slunce. Na rozdíl od cirrostratu nevytváří halové efekty. Altostratus je světlejší, než nimbostratus, ovšem při pozorování proti slunci o něco tmavší, než stratus. Oproti těmto druhům má také obvykle patrnou strukturu na základně.

---

<sup>2</sup> Konvekce je termín pro vertikální, ať už vzestupný nebo sestupný pohyb vzduchové hmoty vlivem jejich rozdílné hustoty, dané teplotou vzduchu.

Tento typ oblaků vzniká při pomalém zdvihu velké masy vzduchu do velkých výšek. Dále vzniká zesílením závoje cirrostratu, zeslabením vrstvy nimbostratu nebo rozšířením střední nebo horní části cumulonimbu. Někdy může vzniknout z ledových krystalků, které vypadávají z altocumulů.

Z altostratu mohou vypadávat srážky, pokud dosahují země, jde o trvalé dešťové, sněhové nebo kroupové srážky. Z odpařovaných srážek během jejich pádu se pod základnou altostratu mohou vytvořit malá roztrhaná oblaka, která se při postupném zvětšování mohou spojit v téměř souvislou vrstvu.



Obr. 2.2: *Altocumulus a dva snímky oblaků typu altostratus [Skr06][Atl06].*

### ***Nimbostratus (Ns, dešťová sloha)***

Šedá, často tmavá oblačná vrstva s nestrukturovanou matnou základnou. Ta je způsobena vypadáváním dešťových nebo sněhových srážek, které většinou dosahují země. Vrstva je horizontálně velmi rozsáhlá a tak tlustá, že přes ní nelze odhadnout pozici Slunce. Pod vrstvou Nimbostratu se často vyskytují nízké roztrhané oblaky. Spodní stranu nimbostratu často alespoň částečně zahalují roztrhané oblaky, které rychle mění svůj tvar. Pokud základna dosahuje země, je nimbostratus někdy považován za frontální mlhu.

Nimbostratus se skládá jak z vodních kapiček a větších dešťových kapek, tak i z ledových krystalů a sněhových vloček. Na rozdíl od stratu je hustější a vypadávají z něj srážky v podobně deště, sněhu nebo ledových zrn, kdežto stratus dává srážky ve formě mrholení a sněhových zrněk a i to jen výjimečně.

### ***Stratocumulus (Sc, slohová kupa)***

Šedé nebo bělavé, případně obojí barvy, menší nebo větší skupiny nebo vrstvy oblaků. Téměř vždy mají tmavá místa. Tento plošný oblak se skládá z pospojovaných, ale vzájemně dobře rozlišitelných částí, připomínající valouny nebo dlaždice, až na výjimky nemá vláknitý vzhled. Velikost jednotlivých částí je několik stovek metrů. Stratocumulus lze popsat podobně jako altocumulus s tím, že stratocumulus se vyskytuje v nižších výškách.

Stratocumulus je složen převážně z vodních kapiček. Průsvitnost je velmi variabilní. Oblak může být dostatečně tenký, že lze určit polohu Slunce, nebo až tak hustý, že Slunce plně zakrývá. Spodní strana takového stratocumulu pak často tvoří zřetelný reliéf.

Ze stratocumulu někdy vypadávají lokální dešťové nebo sněhové srážky slabé intenzity.

Oblak může vzniknout z altocumulu zvětšením jeho několika částí. Vzniká také v čiré atmosféře pod altostratem nebo nimbostratem. Někdy vznikne ze stratu jeho zvrážením, případně vlivem konvekce. Rozeznávacím znakem je fakt, že stratocumulus prakticky nikdy netvoří vlákna a vždy je možné rozlišit jeho jednotlivé části.

### ***Stratus (St, sloha)***

Stratus je obvykle velmi hustá, šedá oblačná vrstva s jednotvárnou mlhavou strukturou. Prosvítá-li tenkým stratem Slunce, jsou jeho okraje zřetelně patrné, často je však natolik tlustý, že Slunce nalézt nejde a oblak se ze země jeví velmi temný. Základna oblaku je obvykle ostře ohraničená. Někdy se stratus nevyskytuje ve formě souvislé vrstvy, ale jako roztrhané chuchvalce (stratus fractus).

Stratus je složen z malých vodních kapiček, při nízké teplotě z ledových částic. Pokud z něj vypadávají srážky, jsou slabé a v podobě mrholení, ledových jehliček nebo sněhových zrn.

Oblak typu stratus vzniká jako důsledek ochlazení nízkých vrstev ovzduší. Vzniká též ze stratocumulu při jeho poklesu nebo pokud ztratí svou typickou strukturu. Útržkovitý stratus je stádiem při vzniku nebo zániku oblaku nebo může vzniknout vlivem turbulence ze srážek, vypadávajících z altostratu, nimbostratu, cumulonimbu nebo cumulu. V přímořských oblastech stratus vzniká z mlhy od moře, která je vlivem stoupavých proudů na pevnině nebo pohoří vyzvednuta do větší výšky a přechází tak v oblak.



*Obr. 2.3: Nimbostratus, stratocumulus a stratus při pohledu zhora [Skr06].*

### ***Cumulus (Cu, kupa)***

Cumulus je hustý, osamocený oblak s ostře ohraničenými obrysy. Mají výrazný vertikální vývoj v horní části do kup, kupolí až věží. Horní osvětlená část je zářivě bílá a připomíná kvěťák, základna je tmavá až temná a téměř vodorovná. Jednotlivé oddělené cumuly se někdy formují do řad po směru větru.



Cumulus se skládá z vodních kapiček, případně ve značně podchlazených částech z ledových krystalků.

Tento typ oblačnosti vzniká ve spodních vrstvách atmosféry při dostatečně velkém ubývání teploty s výškou, čehož příčinou může být zahřívání zemského povrchu slunečním světlem nebo ohřívání proudícího vzduchu teplým povrchem země. Cumulus může taktéž vzniknout z altocumulu, stratocumulu nebo stratu.

### ***Cumulonimbus (Cb, bouřkový oblak)***

Mohutný oblak se značným vertikálním rozsahem i velkou horizontální velikostí. Pro tento typ oblaků je charakteristická kovadlina v horní části. Ta je běžně ve výšce 6 - 9 km, někdy až 12 km a v tropických oblastech dokonce ještě výše. Kovadlina je hladká, vláknitá nebo žebrovitá a často v nejvyšší části rovná. Temná základna je velmi nízko nad terénem. Spodní a střední část po strukturu podobnou cumulu.

Oblak se skládá z celé škály částic – od vodních kapiček, přes velké dešťové kapky až po sněhové vločky, ledové krystalky a kroupy. Ze základny téměř vždy vypadávají intenzivní dešťové srážky, často i kroupy, to vše doprovázené blesky.

Cumulonimbus vzniká z výškově dostatečně vyvinutého cumulu, označovaného cumulus congestus. V některých případech vzniknou přeměnou altocumulu, stratocumulu nebo nimbostratu.



*Obr. 2.4: Cumulus, silně vyvinutý cumulus congestus, cumulonimbus včetně kovadliny [Skr06].*

## **2.6 Typy mlh**

Mlha je jev, kdy díky přítomnosti vodního aerosolu klesne viditelnost pod 1 km. Pokud je viditelnost 1- 5 km, mluvíme o kouřmu nebo-li tzv. řídké mlze. Zhoršení viditelnosti na 1 – 5 km pevnými částicemi, prachem, pískem, průmyslovými exhalacemi, se nazývá zákal. Mlha je svým charakterem podobná jednotvárné vrstevnaté oblačnosti stratus a nimbostratus. Pokud se základna těchto druhů oblaků přiblíží zemi, jsou zaměňovány právě s mlhou. Princip vzniku mlhy je podobný vzniku oblaků, ovšem celý proces ochlazování vzduchu a nasycení probíhá v přízemních výškách. Rozpadá

se následkem ohřátí zemského povrchu slunečním zářením, prouděním vzduchu nebo změnou synoptické situace<sup>3</sup>.

### ***Radiační mlha***

Radiační mlha vzniká při nočním ochlazení tenké vrstvy vzduchu nad chladnoucím zemským povrchem. Podmínkou vzniku je rychlost proudění menší než 3 m/s, v opačném případě by se promíchávalo větší množství vzduchu, k potřebnému ochlazení by nedošlo, vzduch by se nenasytil vodními párami a mlha by nevznikla. Radiační mlha obvykle vzniká ve velmi brzkých ranních hodinách. V létě je velmi tenká a rozpadá se po východu Slunce, na podzim má tloušťku 100-200 metrů, rozpadá se dopoledne nebo se změní ve stratus se základnou ve výšce 100-300 metrů. V zimě může radiační mlha přetrvat celý den.

### ***Advekční mlha***

Advekční mlha vzniká při přesunu teplého vlhkého vzduchu nad studený zemský povrch, jeho ochlazením a dosažením stavu nasycení vodními párami. Při rychlejším proudění se zvedá a přechází ve stratus. Advekční mlhy jsou horizontálně rozsáhlé, hrubé od 20 do několika stovek metrů. Jsou dlouhodobějšího trvání, vzniknout mohou v průběhu celého dne, ale obvykle na podzim a v zimě, a to často přesunem vzduchu z moře nad pevninu. Z advekční mlhy obvykle mrholí.

### ***Předfrontální mlha***

Vzniká z vlhkosti, odpařované z teplých vodních kapek. Objevují se 100-200 km před hranicí teplé fronty.

### ***Zafrontální mlha***

Bývá způsobena přesunem teplého a vlhkého vzduchu nad studený povrch za teplou frontou. Je horizontálně rozsáhlá a spojená s frontální oblačností.

### ***Smog***

Smog, nebo-li městská mlha, je oblast s vyšší koncentrací kouře a škodlivých látek v ovzduší, které krom vlastního vlivu na viditelnost také pomáhají kondenzaci vodních par na viditelnou mlhu.

---

<sup>3</sup> Synoptická situace je okamžitá povětrnostní situace nad daným územím.

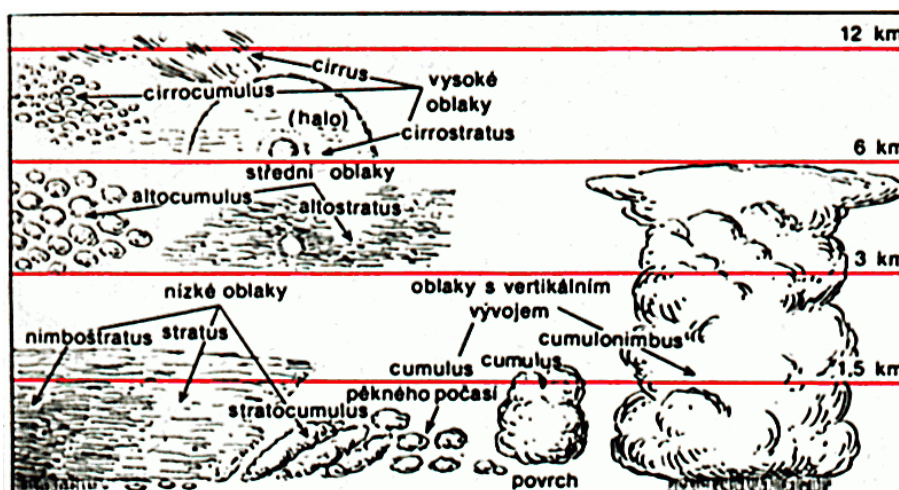
## 2.7 Vertikální rozložení oblačnosti

Jednotlivé druhy oblaků se vždy nacházejí v charakteristickém rozsahu nadmořských výšek. Rozlišujeme oblaky nízkého, středního a vysokého patra, přičemž výškové intervaly se liší v závislosti na zeměpisné šířce.

patro	polární oblast	mírná oblast	tropická oblast
vysoké	3 – 8 km	5 – 13 km	6 – 18 km
střední	2 – 4 km	2 – 7 km	2 – 8 km
nízké	< 2 km	< 2 km	< 2 km

Tab. 2.3: Rozdílné rozložení výškových pater v závislosti na zeměpisné šířce.

Pod vysoké patro spadají oblaky typu cirrus, cirrocumulus a cirrostratus. Oblaky středního patra jsou altocumuly, pro nízké patro to jsou oblaky typu stratocumulus, stratus, samozřejmě také mlha. Ostatní typy, tedy altostratus, nimbostratus, cumulus a cumulonimbus zasahují do více pater.



Obr. 2.5: Běžný výskyt jednotlivých základních druhů oblaků [Ast87].

Stupeň pokrytí oblohy oblačností, nebo-li množství oblačnosti, je důležitým meteorologickým prvkem. Určuje se z pravidla odhadem. V letecké meteorologii se vyjadřuje v osminách – od 0/8 pro jasno až po 8/8 pro zataženo. Často není možné množství oblačnosti odhadnout, příčinou je mlha, silné srážky a podobně.

## 3 Návrh

### 3.1 Požadavky v kontextu cílové aplikace

Před návrhem a realizací celého systému je nutné si ujasnit požadavky na výsledný produkt. V několika případech to dovolí zanedbat některé sice fyzikálně korektní, ale výpočetně zbytečně náročné postupy.

Prostředí, do kterého je úkolem dodat volumetrickou oblačnost, je profesionálním armádním simulátorem především letecké techniky. V bodech uvedu některé důležité charakteristiky simulátoru, především jeho zobrazovacího systému, a jejich vliv na navrhovaný systém zobrazování oblačnosti.

1. Celý systém je rozdělen na dvě části – simulační stroj a obrazový generátor<sup>4</sup>, jejich vzájemná komunikace probíhá síťovým protokolem. Hardwarové vybavení obou lze výkonově zařadit do dnešní vyšší třídy (rok 2007). Díky distribuce výpočtů lze při zobrazování na velkou plochu použít 8 – 32 obrazových generátorů.
2. Zobrazování je postaveno na multiplatformním toolkit OpenSceneGraph<sup>5</sup>, napsaném v jazyce C++ a využívající grafické API OpenGL<sup>6</sup>.
3. Obrazový výstup je zvláště u leteckých simulátorů promítán na kulovou plochu, zorný úhel systému kamer je typicky 200°, někdy i plných 360°. Tento fakt zavrhuje jeden přístup k zobrazování oblaku jako shluku texturovaných ploch – viz dále.
4. Předchozí bod implikuje fakt, že vizualizace musí běžet ve vysokém rozlišení, řádově několik megapixelů.
5. Zobrazování je zcela plynulé. Počet snímků za sekundu je 60. Doba pro zobrazení jednoho snímku je tedy 16.66 ms.
6. Kvalita vizualizace zdaleka nedosahuje dnešních standardů počítačových her, jsou zde jiné priority – viz předchozí dva body.
7. Model počasí je v simulátoru poměrně jednoduchý.
8. Doba běhu jediného spuštění simulace je dle předpokladu v řádech desítek minut. Toto je krátký časový interval, během kterého nemusí docházet k zásadním změnám na obloze. Komplexní simulace vývoje počasí nebude nutná.

S přihlédnutím k uvedeným bodům potom plynou následující požadavky na výsledek. (1) Oblačnost nemusí být nutně fotorealistická, měla by zapadat do kontextu úrovně zpracování zbývajících prvků scény. Oblačnost je jen část výsledné scény a tak by měla pokud možno co

---

<sup>4</sup> Image Generator (IG)

<sup>5</sup> <http://www.openscenegraph.com>

<sup>6</sup> <http://www.opengl.org>

nejméně zatěžovat obrazový generátor. Jak dál ukáží, existují různé metody, jak vytvářet a vykreslovat oblaka. Vzájemně se liší především kvalitou výstupu, výpočetní náročností a univerzalitou použití. Již teď lze říct, že metody založené například na fyzikální simulaci šíření světla oblakem nebudou vhodné. Jakékoliv zanedbání realistických fyzikálních procesů, vedoucí k vyšší rychlosti zobrazování, bude vítané. Doba zobrazování oblačnosti je v krajním případě kolem 10 ms, lépe však 5 ms nebo méně. (2) Při současném stavu modelu počasí v simulátoru nebude na jeho základě možné realisticky modelovat děje, vedoucí ke vzniku oblačnosti. Umístování oblaků do scény lze tedy nyní značně zjednodušit. (3) V profesionálním simulátoru nemusí být zobrazení oblaků jediným efektem, kterým se projevují. Průlet silně podchlazeným vodním aerosolem vede ke vzniku námrazy na trupu letounu, vodní kapičky snižují detekční vzdálenosti některých senzorů a podobně. Velmi zjednodušená implementace alespoň některých těchto jevů by pro simulátor mohla být prospěšná.

## 3.2 Modularita

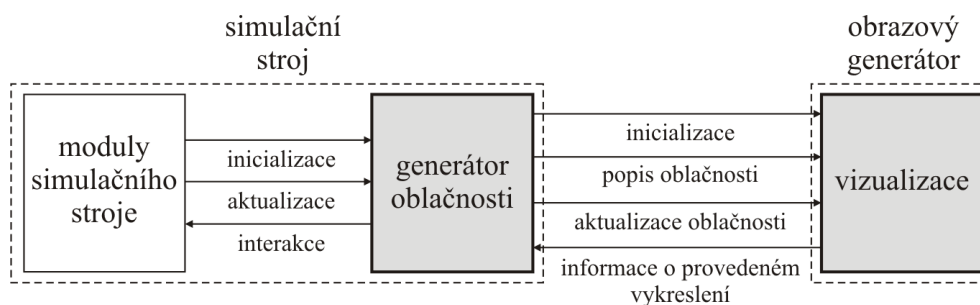
Prvotní návrh systému zobrazování oblačnosti počítal s rozdělením do samostatných dvou částí - modul pro generování oblačnosti a modul pro vizualizaci oblačnosti. Modul generování oblačnosti by byl začleněn do stroje provádějícího simulaci. S ostatními moduly simulačního stroje by obousměrně komunikovat minimálně třemi způsoby:

- inicializace – inicializace přednastaveným scénářem, předání meteorologických map, nastavení úrovně detailů.
- aktualizace – vynucená změna parametrů generátoru po čas průběhu simulace.
- interakce – zpětný přenos informací o simulované oblačnosti ostatním modulům simulačního stroje, viz kapitola „Interakce se simulátorem“.

Modul pro vizualizaci by primárně komunikoval pouze s modulem generování oblačnosti a to obousměrně minimálně čtyřmi způsoby:

- inicializace – nastavení úrovně detailů, alokace prostředků, příprava textur.
- popis oblačnosti – předání absolutního stavu na obloze, tedy pozice, typ, barvy, nasvětlení a další parametry oblaků.
- aktualizace oblačnosti – předání změn parametrů a tedy aktualizace stavu oblačnosti a také předání parametrů, které změní nastavení modulu, tedy např. změna úrovně detailů.
- informace o provedeném vykreslení – sada statistických dat o průběhu a výsledku vykreslování, podle které může být v generátoru oblačnosti změněna úroveň detailů.

Volbu způsobu vykreslení by prováděly oba moduly. Definitivní rozdělení kompetencí pro rozhodování o způsobu vykreslení bude určeno až při implementaci.



Obr. 3.1: Schématické znázornění modulárního návrhu.

Modul generování oblačnosti lze navrhnout v několika úrovních komplexity.

- téměř statický přístup
- jednoduchý scénářový přístup
- primitivní simulace
- realistická simulace

Při **téměř statickém přístupu** by modul při inicializaci získal předdefinovaný stav oblohy a sadu maximálně několika parametrů, které mohou v průběhu času tento stav mírně změnit (např. vítr). Stav oblohy může být po inicializaci modulu několika základními parametry také vygenerován náhodně.

**Jednoduchý scénářový přístup** by modul generování oblačnosti inicializoval předdefinovaným stavem oblohy a scénářem jejího vývoje. Scénář může obsahovat přesný popis vývoje každého oblaku, tedy čas vzniku, zániku, vývoj tvaru nebo trajektorii, pokud pro daný oblak není vhodné použít globální vítr. Manuálně by bylo možné nadefinovat oblasti s jinými meteorologickými podmínkami, jako mohou být stoupavé proudy na návětrné straně horského hřebenu, turbulence mezi horskými hřebeny, termální stoupavé proudy a podobně.

**Primitivní simulace** by modul inicializovala částečně předdefinovaným stavem oblohy včetně informace o meteorologických parametrech v dané oblasti, tedy teploty vzduchu, vlhkosti, případně mapy úrovně výparu vody ze zemského povrchu a vodních ploch, tepelné mapy povrchu a podobně. Za pomoci dalších několika pomocných parametrů je poté vygenerována oblačnost. Pomocné parametry jsou takové parametry, aby výpočet oblačnosti nemusel být závislý jen na nadefinovaných meteorologických podmínkách, a aby mohl být do jisté míry kontrolován a přizpůsobován dle uživatelské potřeby.

**Realistická simulace** je založena na primitivní simulaci a přidává výpočet vývoje směru proudění vzduchu a s ním i vodní páry a oblačnosti s přihlédnutím na model terénu. Výsledný vývoj počasí by poté působil velmi realisticky, odpadla by nutnost manuálně definovat oblasti s jinými lokálními meteorologickými podmínkami. Editor povětrnostní situace by v tomto případě dokázal pracovat přímo se synoptickými mapami a šla by tak nasimulovat velmi reálná povětrnostní situace. Realistická simulace je ovšem implementačně náročná, poměrně náročný je i její výpočet a

v předpokládaném uplatnění simulátoru by neměla využití. Použití by bylo opodstatněné snad jen při simulacích plachtění.

S přihlédnutím na současnou úroveň modelu počasí vyvstává otázka, zda modulární přístup přinese užitek. Proti modularitě mluví zbytečné komplikace při implementaci a nutnost přenášet data síťovým protokolem. Schůdnějším řešením je zachovat jistou separaci činností spadající pod první i druhý modul, ale vše implementovat jako celek a umístit na obrazový generátor.

## 3.3 Interakce se simulátorem

V komplexním simulovaném prostředí není generování a vizualizace jediným možným výstupem při tvorbě oblačnosti. Atmosférická voda v plynném, kapalném i pevném skupenství interaguje s ostatními objekty (technika, postavy, terén) ve scéně:

- mechanicky
- elektromagneticky

V prvním případě voda v různém skupenství nebo přetrvávající důsledky její přítomnosti působí na objekty kontaktně a omezuje je v činnosti, snižuje jejich pohyblivost nebo je poškozuje. V případě druhém dochází k zeslabení elektromagnetického vlnění různé vlnové délky, což má vliv na činnost senzorů, naváděcích systémů a podobně.

### 3.3.1 Srážky

Přítomnost dešťových, sněhových nebo ledových srážek v simulátoru nelze zanedbat. Krom přímého vlivu na techniku a postavy – například poškození povrchu letounu kroupami, dochází ke změnám vlastností povrchu terénu. Zásadními oblastmi, kde se srážky projeví na terénu jsou:

- zpevněné pozemní komunikace – mokrá vozovka nebo jiná zpevněná plocha, vznik náledí, zasněžení.
- nezpevněný terén – promáčení terénu a tím způsobené zhoršení průjezdnosti techniky.

Výstupní formát, určující množství srážek dopadnuvších na terén, může být jednoduchý rastr o velikosti buněk v řádech desítek až stovek metrů. Zpracování těchto dat a nastavení promáčení, ale i postupné vysoušení terénu, by už případně provedly jiné (existující) moduly.

### 3.3.2 Námraza

Vznik námrazy na trupu, křídle nebo stabilizátorech je jedna z nejnebezpečnějších situací, která může za letu nastat. Jakákoliv vrstva ledu na křídle dramaticky zhoršuje jeho vlastnosti – o desítky procent klesá vztlak, naopak výrazně roste aerodynamický odpor a zvyšuje se pádová rychlost letounu. Podstata vzniku námrazy je kondenzace a desublimace vodní páry vlivem ochlazení vodních par v blízkosti prochlazeného povrchu letounu a namrzání vodních kapiček na silně podchlazeném

povrchu. Nebezpečí výskytu tohoto problému roste s klesající teplotou, rostoucí relativní vlhkostí vzduchu a především přítomností vody v kapalném nebo pevném skupenství (let v oblacích, mlze, za deště či sněžení). K námraze může dojít i ve velmi teplém počasí při klesání letounu z velké výšky, kdy se povrch ještě nestačil prohřát a začínají se na něm formovat ledové krystalky.

V případě letounů s pístovými motory bez přímého vstřikování je velkým nebezpečím vznik námrazy v karburátoru a vstupním potrubí. Může k němu dojít i za bezoblačného počasí, při teplotách i +30°C a s relativní vlhkostí vzduchu alespoň 30%. Vlivem snížení tlaku vzduchu a odpařováním paliva dochází ke značnému poklesu teploty, kondenzaci (desublimaci) a namrzání (ucpávání) vzduchovodů. Problém je řešen zahříváním karburátoru.

Neméně nebezpečný je i sníh, napadáný na povrch letounu při vyčkávání na zemi. I jeho zanedbatelná vrstva může mít katastrofální dopad. Navíc nelze spoléhat, že by se při vyšší rychlosti proudění okolního vzduchu z povrchu „odfoukl“. Díky existenci tzv. mezní vrstvy<sup>7</sup> sníh stále zůstává na povrchu. Stejně se chová jakákoliv jiná nečistota v pevném skupenství na povrchu letounu.

Shrňme si faktory, které mají vliv na vznik námrazy:

- skutečná teplota okolního vzduchu, v letecké terminologii označovaná jako SAT<sup>8</sup>
- vlhkost vzduchu a přítomnost kapalných a pevných vodních částic
- postupné prochlazení materiálu ve větší výšce
- zahřívání povrchu letounu vlivem aerodynamického ohřevu<sup>9</sup>
- činnost odledovacího a protinámrazového zařízení
- usazování sněhu při stání na zemi
- rozprašování vody při kontaktu podvozku se vzletovou a přistávací dráhou

Nabízí se v zásadě dvě možnosti, jak informovat simulační jádro o nebezpečí vzniku námrazy. Při prvním způsobu by modul generování počasí při inicializaci ze simulátoru získal informace o náchylnosti jednotlivých částí letounu na vznik námrazy a modul by poté po čas letu zpět do simulátoru předával informace o stavu námrazy na těchto částech. Vhodnější je však druhý způsob, kdy modul generování počasí simulátoru předává pouze několik základních veličin, které poté simulátor již sám zpracuje a vyhodnotí, a rozhodne, zda námraza vznikne či nikoliv. Tato možnost

---

<sup>7</sup> Mezní vrstva je tenká vrstva vzduchu vznikající na povrchu těles při pohybu ve vzduchu, obecně tekutině. Na povrchu tělesa je rychlost proudu v důsledku vazkosti vzduchu nulová, čím více je proudnice vzdálena od trupu, tím více se rychlost blíží skutečné rychlosti okolního proudu, až u jisté vzdálenosti je možné říct, že rychlost proudění není povrchem tělesa ovlivněna, proud se pak nazývá volný. Mezní vrstva je tedy vrstva, ve které je rychlost proudu nižší než je rychlost vzájemného pohybu tělesa a vzduchu.

<sup>8</sup> „Static Air Temperature“ je teplota, která by byla naměřena, pokud by byl vzájemný pohyb měřicí sondy a okolního vzduchu nulový.

<sup>9</sup> Počínaje letovou rychlostí 700 km/h již námraza vlivem aerodynamického ohřevu nevzniká. Z tohoto důvodu se v námrazových podmínkách létá vysokou rychlostí.



podstatně sníží nároky na mezimodulovou komunikaci a zvýší univerzálnost modulu generování počasí. Ve zcela nejjednodušším řešení by se předávala pouze jediná hodnota – faktor nebezpečí vzniku námrazy. S přihlédnutím k výše uvedeným bodům jsou vstupní hodnoty pro výpočet faktoru nebezpečí vzniku námrazy teplota, vlhkost, množství kapalných a pevných vodních částic, lépe řečeno pozice v oblaku a jeho hustota.

### 3.3.3 Stínění elektromagnetického vlnění

Pod pojmem „stínění elektromagnetického vlnění“ lze chápat vliv přítomnosti oblačnosti na senzory přehledových, vyhledávacích, naváděcích a komunikačních systémů, ale také senzory (oči) umělého intelektu řízených postav/prostředků.

Úroveň útlumu elektromagnetického vlnění závisí na těchto faktorech:

- velikosti, množství a skupenství kondenzovaných částic
- vzdálenosti, kterou paprsek urazí uvnitř oblaku
- vlnová délka

První dva body se odvíjí od velikosti, typu a stádia vývoje oblaku. V následujících odstavcích uvedu vliv na konkrétní systémy či frekvenční spektra.

#### *Viditelné spektrum*

U viditelného spektra<sup>10</sup> je úroveň absorpce záření oblačností všeobecně zřejmá. Pouze je na místě dodat, že kratší vlnové délky (odstíny modré) jsou vodními kapkami absorbovány poněkud méně a proto se obzvláště hrubý oblak může po průchodu slunečního světla jevit lehce namodralý.

#### *Infračervené snímací prvky*

Infračervené záření<sup>11</sup> je oblačností absorbováno téměř tak dobře, jako viditelné světlo. Přesto infračervené snímací systémy umožňují vykonávat průzkum či navádění i při lehkém zadýmení, slabé mlze a za deště.

#### *Laserové paprsky*

Také paprsky generované laserem<sup>12</sup> jsou oblačností absorbovány a rozptýleny. Tato skutečnost dala za vznik generátorům maskovacího kouře na bojové technice, které znemožní navedení laserem naváděných střel.

---

<sup>10</sup> Vlnová délka viditelného spektra je 380 – 750 nm (frekvence 400 - 780 THz).

<sup>11</sup> Infračervené záření má vlnovou délku 0.75 – 1000 μm (frekvence 300 GHz - 400 THz)

<sup>12</sup> Vlnová délka laserového záření je v závislosti na použité technologii v rozsahu 490 – 11000 nm (frekvence 27.3 - 612 THz).

## ***Rádiová komunikace a navigační systémy***

Radiokomunikace a práce navigačních systémů na principu elektromagnetického záření je oblačností ovlivněna jen minimálně<sup>13</sup>. K jistému rušení a šumu může dojít jen při intenzivních srážkách. Při implementaci do simulátoru lze vliv oblačnosti na rádiovou komunikaci zanedbat.

## ***Radiolokátor (mikrovlnné záření<sup>14</sup>)***

V jistých případech oblačnost představuje překážku pro mikrovlnné vlnění, zejména pokud obsahuje větší vodní kapky a kroupy. U starších, méně výkonných radiolokátorů mohou být oblaka detekována jako falešné cíle. Míra útlumu signálu je závislá také na vlnové délce záření – některá pásma se využívají u meteorologických radarů, viz další kapitola. Přes tento zřejmý útlum jsou moderní vyhledávací, zaměřovací a střelecké radary natolik výkonné, že dokáží zachytit odraz od cíle i přes velmi hrubou oblačnost. Každopádně ke snížení detekční vzdálenosti radiolokátoru vlivem oblačnosti dochází.

## ***Meteorologický radar***

Hustá oblačnost a s ní spojené intenzivní srážky a turbulence představují pro letecký provoz značné nebezpečí – ať už povrchové poškození trupu, antén, snímačů, zbraní a motorů ledovými kroupami, mokrá přistávací dráha nebo namáhání konstrukce letounu vlivem skokově se měnícímu proudění vzduchu. Zvláštní pozornost je v tomto případě věnována bouřkovým oblakům typu cumulonimbus, které navíc hrozí poškozením elektroniky důsledkem elektrických výbojů.

Na současných, především dopravních a transportních letounech jsou proto instalovány tzv. meteorologické radary<sup>15</sup>. Ty vysílají směrem k oblačnosti mikrovlnné impulsy v pásmu X<sup>16</sup> (dříve v pásmu C<sup>17</sup>), část impulsů je meteorologickými srážkami odražena zpět k anténě, zachycena a zpracována přijímačem radaru. Intenzita odrazu se odvíjí od množství a skupenství kondenzovaných částic. Vzdálenost je určena ze znalosti rozdílu času vyslání a času přijetí impulsu. Směr pohybu částic lze v některých případech určit díky Dopplerově efektu.

Meteorologický radar dokáže detekovat:

- déšť
- kroupy obklopené vlhkým vzduchem

---

<sup>13</sup> Radiokomunikační a navigační systémy pracují v pásmu LF, MF, HF, VHF nebo UHF, při vlnových délkách 10000 až 0.1 m (frekvence 30 kHz – 3 GHz).

<sup>14</sup> Mikrovlnné záření je elektromagnetické vlnění o vlnové délce 1 – 100 mm (frekvence 3 – 300 GHz).

<sup>15</sup> Anglické označení Weather Radar nebo též WX Radar, zkráceně WXR.

<sup>16</sup> Pásmo X mikrovlnného spektra má dle IEEE vlnovou délku 24 – 42.9 mm (frekvence 7 – 12.5 GHz), využívá se při mapování terénu a sledování počasí.

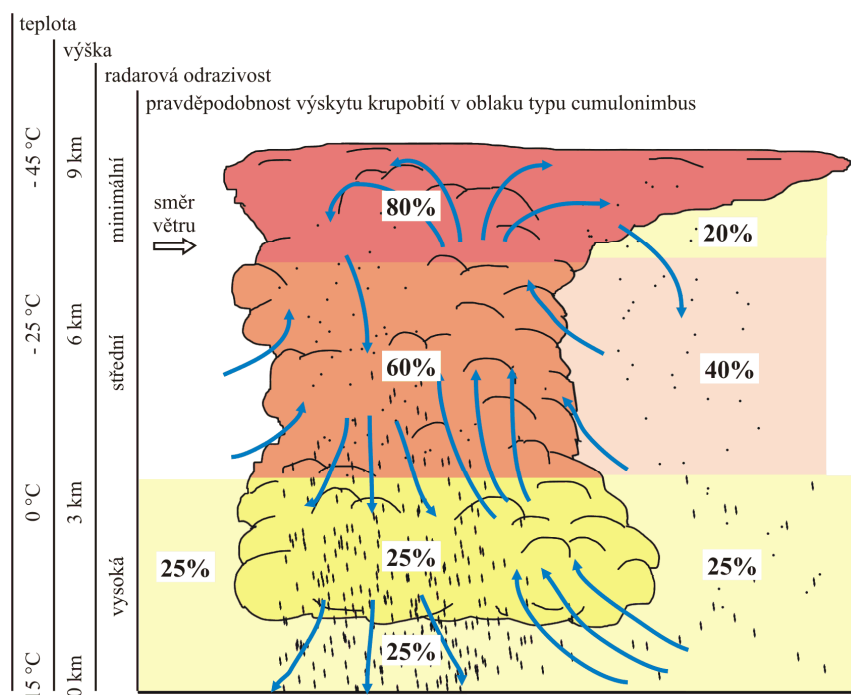
<sup>17</sup> Pásmo C mikrovlnného spektra má dle IEEE vlnovou délku 37.5 – 75 mm (frekvence 4 – 8 GHz).

- vlhkou turbulenci
- ledové krystalky, suché kroupy, suché sněhové vločky, nicméně intenzita jejich odrazu je přibližně pětina ve srovnání s dešťovými kapkami

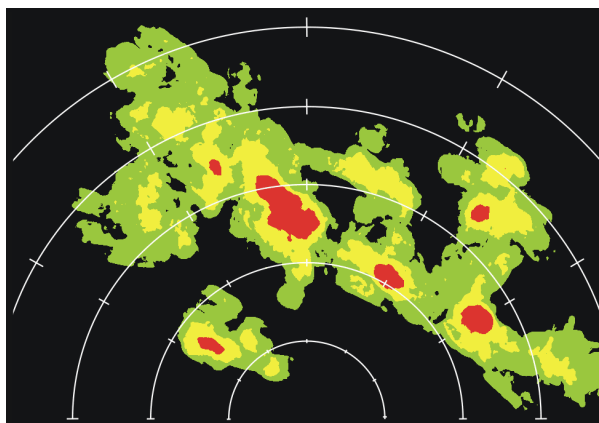
Naopak nedokáže detekovat:

- oblaka, mlhu, vítr – příliš malé nebo žádné vodní kapky
- stříh větru bez vlhkosti
- turbulence v „čistém vzduchu“ bez srážek
- písečnou bouři – mnohé pevné částice jsou pro meteorologický radar prakticky neviditelné
- blesky

Je třeba mít na paměti, že dosah meteorologických radarů je i přes 500 km, modul generování počasí by měl být schopen poskytnout informaci o oblačnosti i na takto velkou vzdálenost. Komplexnější implementace meteorologického radaru by dále měla brát v potaz atmosférickou inverzi, ale také hejna hmyzu, ptáků a další podobné úkazy, které velmi ovlivňují radarový odraz.



Obr. 3.2: Nebezpečnost a intenzita radarového odrazu jednotlivých částí oblaku typu cumulonimbus.



*Obr. 3.3: Typické zobrazení oblačnosti na obrazovce v kabině letounu nebo na pozemním stanovišti.*

Stínění jakéhokoliv elektromagnetického záření lze realizovat nejméně třemi způsoby. Ten první využívá výkonu GPU. Vhodným nastavením projekce a průhlednosti oblaků lze scénu vykreslit do rastru tak, jak ji „vidí“ senzor. Takto lze získat detailní, přesnou a kompletní informaci o útlumu v libovolném místě zorného pole senzoru. Cenou je v tomto případě odebrání výpočetního času GPU. Větším problémem se jeví, alespoň u některých vizualizačních technologiích, nutnost přepočítat geometrii elementů oblaků při změně pozice kamery. Ke značnému zrychlení by vedl podobný přístup s tím rozdílem, že by se vykreslovaly pouze obalové kvádry částí oblaků. Poslední řešení nevyužívá GPU, ale CPU. Míra útlumu by se získala z délky průseku paprsku mezi senzorem a potenciálním cílem s objemem obalových kvádrů oblaků.

### 3.4 Směr další práce

Při řešení projektu se nebudu zabývat tvorbou grafického generátoru počasí. Rozmíst'ování oblaků a definice jejich chování bude řešena voláním příslušných funkcí. Grafické uživatelské rozhraní editoru počasí nebo alespoň popis oblačnosti a meteorologické situace (třeba i ve formě konfiguračního textového souboru se snadno pochopitelnou sémantikou) by se implementovalo až při budoucím rozšiřování projektu. Výpočet stínění infračerveného, laserového a mikrovlnného záření, aplikace jednoduchého scénářového přístupu pro modul generování počasí, případně částečné použití některých vlastností z popsaného přístupu „primitivní simulace“ by mohly být také až budoucím rozšířením tohoto projektu. Ještě pokročilejší systém by mohl simulovat i meteorologický radar, stínění rádiové komunikace lze při současném stavu zanedbat úplně. V posledních větách zmiňované vlastnosti jsem uváděl jen pro snazší návrh případné vylepšené verze systému generování a zobrazování oblačnosti.

## 4 Vizualizace oblačnosti

Realistické zobrazení oblačnosti není jednoduchým úkolem. Vývojáři takovýchto systémů narážejí na řadu problémů či limitací, plynoucích z použité technologie, grafického engine, hardware a grafického rozhraní (API). Musíme si uvědomit, že úkolem je vytvořit vizualizační systém pracující v reálném čase. Tato podmínka dovoluje zahrnout řadu tzv. univerzitních řešení. Jako jejich typického zástupce lze uvést fluidní simulaci a zobrazení oblaku jako shluku velkého množství částic. Metoda nachází uplatnění v simulátorech pro předpověď vývoje a následků počasí a částečně také ve filmovém průmyslu, jedna iterace výpočtu (nebo jeden snímek filmu) je však otázkou desítek hodin procesorového času. Pro náš problém hledáme řešení, které doplní stávající projekci oblaků na vzdálenou kupoli, je možné se kolem a uvnitř takto vytvořených oblaků volně pohybovat a je kompromisem mezi vizuální kvalitou, kvalitou simulace vývoje a nároky na výpočetní výkon. Předpokládaný strop doby vykreslení oblačnosti pro jediný snímek je mezi 5 – 10 ms.

Proces tvorby a vizualizace oblaku lze rozdělit do následujících kroků, přičemž druhý a třetí krok může také proběhnout v obráceném pořadí nebo více méně společně:

- modelování základního tvaru
- modelování detailů
- nasvětlení a stínování

V prvním kroku je určen tvar oblaku s detaily v řádech stovek metrů až kilometrů. Již na této úrovni lze od sebe rozlišit několik kategorií oblaků – kupovité oblaky od vrstevných a tenkých hladinových. V případě kupovité oblačnosti lze velmi věrně modelovat tvary výškově vyvinutých typů oblaků, především cumulonimbus a cumulus (cumulus congestus). V závislosti na volbě konkrétního přístupu se v tomto kroku oblaka modelují například jako implicitní elipsoidy nebo skupiny k pozorovateli směřujících desek.

Druhý krok na vytvořený tvar aplikuje procedurální šum či texturu. Tímto oblak získá detailnější strukturu – malé kupy, vlákna, vrásnění ad. Současně lze regulovat průhlednost jednotlivých částí a tím oblaku ubrat na hustotě.

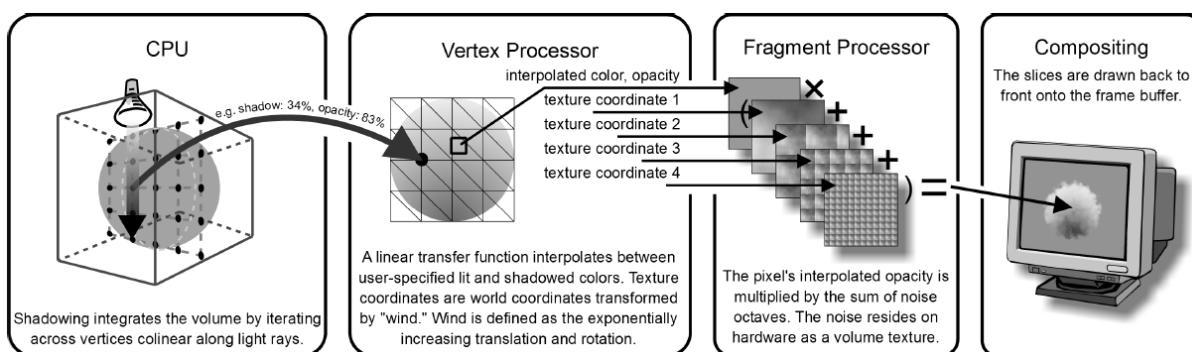
Teprve až správné nasvětlení a stínování dodá oblaku realistický vzhled. Realistické modely šíření světla propočítávají odraz, absorpci a rozptyl paprsků světla uvnitř oblaku. Takto kvalitní model zobrazuje i stínění oblaku na sebe sama (self-shadowing), což dramaticky zlepšuje vizuální dojem. Aproximace těchto efektů při zachování dostatečně dobrého vizuálního dojmu lze dnes svěřit grafickému akcelérátoru.

V následujících kapitolách krátce popíši některé existující metody.

## 4.1 Implicitní tělesa

Jedna z metod založených na implicitních tělesech byla popsána týmem J. Schpoka z Purdue University a University of Utah v roce 2003 [Schp03]. Jejím základem jsou uživatelem definované pozice a rozměry implicitních těles. Výsledný implicitní objekt je vytvořen výpočtem intenzity pomocí Wyvillovy kubické funkce. Volumetrický rendering<sup>18</sup> tohoto objektu lze nyní provést jeho mnohonásobným nařezáním rovinou paralelní nebo kolmou ke světelnému vektoru. Výběr orientace roviny je proveden tak, aby odchylka normály řezu od vektoru pohledu byla menší. Animace na úrovni implicitních těles lze provést více způsoby – klíčovanou animací jednotlivých těles nebo přiřazením jednotlivých těles k elementům částicového systému. Detaily na nižší úrovni se vytvářejí kombinováním různých šumových funkcí (Perlinův šum apod.) o rozdílné frekvenci, neboli velikosti náhodných útvarů. Aby nemusel výpočet šumu při každé iteraci výpočtu nebo při každém zobrazovaném snímku zatěžovat procesor, jsou textury šumu předgenerovány do paměti grafického hardware. Jednoduchá animace pohybu šumových textur různých frekvencí velmi realisticky napodobuje lokální procesy, probíhající v oblaku – např. vznik a zánik jednotlivých věžiček u oblaku typu cumulus, strhávání a zánik okrajových částí oblaku větrem apod. Rozložení činností na výpočetní jednotky je následující:

- CPU – vytvoření implicitního tělesa, generování řezů, výpočet stínu.
- GPU Vertex Shader – interpolace barev, výpočet texturovacích souřadnic.
- GPU Fragment Shader – výpočet průhlednosti, skládání šumových funkcí různých frekvencí.



Obr. 4.1: Osvětlovací model, navržený týmem J. Schpoka. Převzato z [Schp03].

<sup>18</sup> Rendering, nebo-li česky vykreslování či stínování. Jde o proces převodu geometrické reprezentace scény do obrazu.



*Obr. 4.2: Ukázka oblačnosti vygenerované z jediného implicitního tělesa aplikací několika šumových funkcí o různých frekvencích. Převzato z [Schp03].*

Jako demonstraci této metody zveřejnili autoři ukázkovou aplikaci, ve které lze oblaka interaktivně vytvářet a parametricky měnit jejich vzhled. Při testování aplikace si lze všimnout minimálně dvou situací, kdy dochází k výraznému poklesu výkonu. Už při modelování osamocené oblaku o velikosti řádově stovek metrů je nutné nastavit počet řezů poměrně vysoko. V opačném případě jsou jednotlivé řezy velice patrné a rušivé. Ovšem s rostoucím počtem řezů nepřímou úměrou klesá zobrazovací výkon. Pro rozsáhlejší oblačnost o velikosti řádově kilometrů se stovkami řezů lze očekávat pokles hluboce pod únosnou mez. Druhou situací, kdy dochází k výkonovému propadu, je průlet skrze oblačnost. Navíc při jakémkoliv, v zájmu rychlého zobrazování rozumném počtu řezů vždy existuje nějaká malá vzdálenost, ze které lze od sebe jednotlivé řezy rozlišit a výsledek tak působí rušivě. Výskyt tohoto nedostatku by částečně ošetřilo zvýšení průhlednosti řezů blízkých ke kameře. Dalším problémem je skoková změna stínování oblaku při některých změnách úhlu pohledu, přesněji změna orientace řezů z paralelních na kolmé k vektoru přicházejícího světla a naopak. Přes právě popsané problémy působí grafický výstup této metody poměrně velmi realisticky.

Podrobnější popis tohoto algoritmu včetně demonstrační aplikace a ukázkových obrázků a videí je ve zdroji [Schp03] a na URL: <http://cobweb.ecn.purdue.edu/purpl/projects/fastclouds/>.

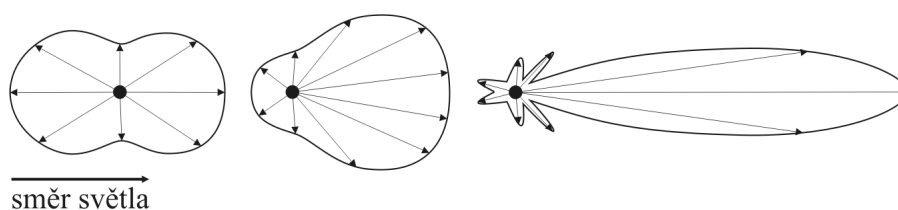
## 4.2 Harrisův přístup

Mark J. Harris se ve své práci [Har01] a [Har02] zabýval fyzikálně podloženým výpočtem osvětlení oblaků. Jeho technologie využívá pro reprezentaci a modelování oblaku částicového systému, kde má každá kulová částice svou polohu, poloměr, barvu a hustotu ve svém středu. Klesání hustoty k od středu k okraji částice popisuje Gaussova distribuce. Částice lze do prostoru umístit náhodně nebo jejich rozložení svěřit grafickému designérovi. Kombinací různých velikostí a hustot částic lze dobře aproximovat skutečná oblaka. Jednotlivé částice jsou zobrazovány technologií splatting, kdy je každá vykreslena jako obrazově orientovaný polygon.

Fyzikální výpočet pracuje s následujícími termíny.

**Absorpce** je jev, kdy je část světelné energie při dopadu na částici oblaku přeměněna na jinou energii, například tepelnou. Úroveň ztráty světelné energie absorpcí je určena koeficientem absorpce.

**Rozptyl** je zpětné vyzařování části absorbované energie ve formě světla. Úroveň ztráty světelné energie rozptylem je určena koeficientem rozptylu. Rozptyl je silně závislý na velikosti částic, na kterých k němu dochází. Rozptyl na částicích menších, než je vlnová délka světla (molekuly vzduchu), popisuje Rayleighova teorie rozptylu. V tomto případě také velmi záleží na vlnové délce. Typickým důsledkem Rayleighovy teorie je modré zbarvení oblohy, i když na vzduchový obal dopadá ze Slunce bílé světlo. U větších částic je třeba použít složitější Mieovy teorie rozptylu. Zde již vlnová délka nehraje takovou roli, rozptyl všech vlnových délek je téměř stejný a proto se oblaka, složená z větších částic, jeví jako téměř bílá. Veškeré tyto jevy lze matematicky popsat, viz [Pon03].



Obr. 4.3: Fázové funkce získané Rayleighovým (vlevo) a Mieovým rozptylem pro středně velkou (uprostřed) a velkou částici (vpravo).

Modely rozptýleného osvětlení simulují absorpci a emisi světla oblakem, ale také rozptyl světla uvnitř oblaku. Popisovány jsou dvě úrovně simulace rozptylu – jednonásobný a vícenásobný rozptyl. Při jednonásobném rozptylem nás zajímá předávání světelné energie mezi částicemi pouze v jediném směru, obvykle ve směru k pozorovateli. Vícenásobný dopředný rozptyl uvažuje všesměrový rozptyl v několika řádech. Tento model je fyzikálně přesnější, výpočetně však značně náročnější. Různé komplexní modely ve svých pracích popisovali např. J. Blinn (1982) a J. Kajiya (1984). T.Nishita (1996) ukázal, že směry vyzařování lze ve vícenásobném rozptylem redukovat a také, že řád výpočtu<sup>19</sup> lze zhora omezit na 4.



Obr. 4.4: Harrisova technologie.

<sup>19</sup> Počet částic, přes které je simulován každý tok postupně rozptylované světelné energie.



Harris přistoupil k dalšímu zjednodušení a použil vícenásobný, ale pouze dopředný rozptyl, tedy rozptyl světla pouze na částice dále po směru světla. Tato fáze výpočtu neprobíhá v reálném čase, je spuštěna jen při inicializaci scény a je během ní vypočítáno osvětlení jednotlivých elementů oblaku. Při samotném vykreslování přichází na slovo jednoduchý rozptyl směrem ke kameře. Světlo opouštějící částici je rovno světlu, které na ni dopadá, není absorbováno (vypočteno v prvním kroku) plus světlo, které částice rozptyluje. Množství rozptýleného světla je znovu vypočítáno ze znalosti fázové funkce částice. Přesný popis algoritmu a potřebné vzorce neuvádím, výsledek mé práce nebude obsahovat fyzikálně přesný popis šíření světla. Podrobnější informace o implementaci Harrisova a jím inspirovaných algoritmů viz [Har01], [Pon03].

Harrisův přístup k renderování oblaků je použitelný v real-time aplikacích, kde je třeba zachovat možnost oblakem proletět. Rychlost zobrazování je po přidání impostorů (viz dále) dobrá i při velkém počtu oblaků. Nicméně je zde několik nedostatků.

- Při změně směru osvětlení se musí veškeré stínování znovu přepočítat. Tento krok nelze provést v reálném čase, ale výpočet je možné rozložit do více snímků a mezivýsledky ukládat do pomocné textury. Texturu je navíc poté možno použít jako světelnou mapu, promítanou na terén a vytvářející efekt stínu.
- Doba přípravy osvětlení většího množství oblaků může být nepříjemně dlouhá. Je to daň za jejich poměrně realistický vzhled.
- Díky použití splattingu s jednoduchou Gaussovou texturou oblaka postrádají ostřejší obrysy.
- Oblačnost, vykreslená Harrisovou metodou, je velmi statická. Vznik, růst a zánik oblaků by musel být doprovázen výpočtem osvětlení.

## 4.3 Shluky texturovaných ploch

Poměrně známá metoda tvorby věrohodně vypadajících oblaků je pomocí skupin několika jednotek až desítek otexturovaných desek. Velmi dobrým příkladem, kde se tuto technologii podařilo úspěšně aplikovat v praxi, je počítačová hra Flight Simulator 2004 společnosti Microsoft [Wan03]. Každý oblak je ve fázi modelování základního tvaru vytvořen pomocí obalových kvádrů ve 3D modelovacím nástroji. Ve fázi modelování detailů jsou kvádry vyplněny texturovanými plochami, jejichž normály směřují k pozorovateli<sup>20</sup>. K dispozici je sada vygenerovaných šumových nebo lépe předpřipravených textur – od neurčitých bez větších detailů pro slohovitá oblaka až po detailně prokreslené textury pro kupovitou oblačnost.

---

<sup>20</sup> Tato rotace normály a tedy i celé plochy se anglicky nazývá billboarding a v běžné praxi se především koncem 90. let hojně využívala v počítačových hrách pro zobrazování postav.

Technologie shluků texturovaných ploch dává v mnoha případech velice realistický výstup, jak je možné vidět na ukázce níže, výpočetní náročnost je velmi přijatelná, ale trpí několika nedostatky. Zobrazení tenké oblačnosti, zejména řasovitých oblaků nebo přízemní mlhy je touto metodou prakticky nemožné – pro zachování malé tloušťky, ale přitom jednotnosti v hustotě by bylo zapotřebí zobrazit značné množství texturovaných desek se samozřejmým dopadem na rychlost vykreslování. Například v případě hry Flight Simulator 2004 je toto jednoduše řešeno velkou texturovanou plochou umístěnou v dané výšce vodorovně s terénem.

Desky lze rotovat čelem k bodu kamery nebo kolmo k vektoru pohledu. Oba přístupy mají své výhody i nevýhody.

Rotace k bodu kamery nevyžaduje žádný přepočít geometrie oblaku, pokud se kamera nepohybuje, ale pouze rozhlídí. Normály elementů oblaku totiž vždy směřují ke středu kamery, vektor pohledu do výpočtu vůbec nevstupuje. S tímto má souvislost pro cílovou aplikaci další velmi podstatná vlastnost, a to, že lze spojit obraz z více kamer umístěných v jednom místě a vytvořit tak panoramatický obraz bez jakýchkoliv nenávazností na okrajích jednotlivých obrazů. Nedostatkem je rychlá rotace desky při blízkosti pohybující se kamery. Dalšími nevýhodami jsou nepřírozená rotace desek pod a nad kamerou, křížení desek atd.

Při rotaci s normálou rovnoběžnou s vektorem pohledu nedochází ke křížení desek, zásadním nedostatkem je však nemožnost vytvořit věrohodně vypadající panoramatický pohled se zorným úhlem nad cca 135°, úplně vyloučeno je spojování obrazů z více kamer.

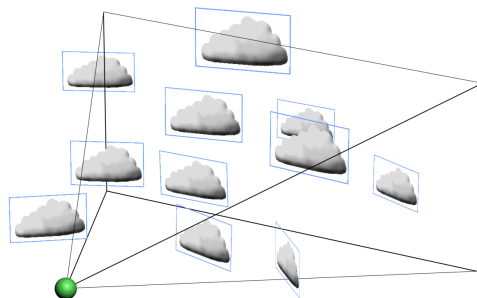


Obr. 4.5: Volumetrická oblaka složená z mnoha texturovaných ploch (Flight Simulator 2004).

## 4.4 Impostor

Použití tzv. impostorů při zobrazování většího množství oblaků vede ke zcela zásadnímu nárůstu rychlosti zobrazování. Impostor je dvourozměrná náhražka volumetrického oblaku, jejíž normála

vždy směřuje k pozorovateli. Impostor vzniká vyrenderováním trojrozměrného oblaku do textury s kanálem průhlednosti. Je zřejmé, že zobrazení takovéto jediné textury je mnohonásobně rychlejší, než zobrazení skutečně volumetrického oblaku. Je v kompetenci mechanismu pro rozhodování o úrovni detailů v jaké vzdálenosti začnou být trojrozměrné oblaky nahrazovány impostory, při jaké změně osvětlení, pozice a úhlu pozorování bude vyžádáno nové generování textury. Ve scéně s velkým množstvím oblaků lze použít každé vygenerované textury na velké množství jednotlivých oblaků, mechanismus výběru úrovně detailů také řeší kolik oblaků může mít totožnou texturu, aniž by si uživatel všiml jejich opakování.



Obr. 4.6: Rotace impostorů směrem ke kameře.

Postup tvorby impostorů je založen na vyrenderování oblaku do framebuffer, kopie takto získaných dat do textury a vytvoření polygonu o velikosti daného oblaku z pohledu pozorovatele. Ač je použit framebuffer, díky double buffering není nijak ovlivněn skutečný výstup na obrazovku.

V prvním kroku (1) je projekce, perspektivní nebo pro zjednodušení ortografická, nastavena tak, že pozice kamery je zachována a cíl pohledu je nastaven na střed oblaku. Zorný úhel, resp. horní, spodní, levá a pravá ořezová rovina u ortografické projekce (frustum planes), je nastaven tak, aby těstě obaloval oblak. Dalším krokem (2) je nastavení velikosti (rozlišení) framebuffer. [Har01] uvádí vztah pro výpočet ideálního rozlišení následovně.

$$\text{rozlišení}_{\text{textura}} = \text{rozlišení}_{\text{obraz}} \cdot \frac{\text{velikostObjektu}}{\text{vzdálenostObjektu}} \quad (4.1)$$

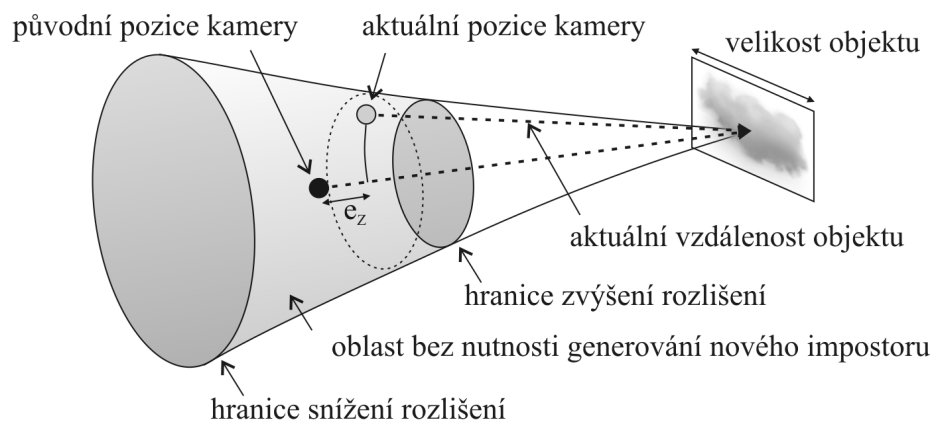
Vztah platí, pokud vzdálenost objektu výrazně převyšuje jeho velikost, není ovšem nikterak zohledněn zorný úhel kamery, ten je předpokládán 1 rad (57.3°). Tento nedostatek napравuje následující vztah.

$$\text{rozlišení}_{\text{textura}} = \frac{\text{rozlišení}_{\text{obraz}}}{\text{zornýÚhel}[\text{rad}]} \cdot \frac{\text{velikostObjektu}}{\text{vzdálenostObjektu}} \quad (4.2)$$

Jak se dále ukáže, přesnější druhý vztah v některých případech nepřináší užitek, spíše naopak. Oblaka ve své struktuře na rozdíl od trojrozměrných geometrických objektů postrádají vyšší frekvence. Důsledkem toho lze textury pro impostory generovat až ve 4x menším rozlišení, než doporučují výše uvedené vztahy. Jen pro připomenutí uvádím, že grafická API, tedy i OpenGL, vyžadují rozlišení mocnin dvou.

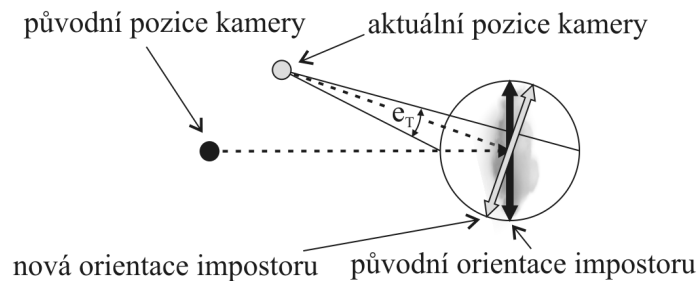
V předposledním kroku (3) se ujistíme, že je smazaný framebuffer a vykreslíme do něj daný oblak. Pokud je správně nastavena projekce, bude oblak dobře zaplňovat vykreslovací plochu. Posledním krokem (4) je vytvoření geometrie impostoru - obdélníkového polygonu s normálou směřující ke kameře a s připravenou texturou. Impostor je poté běžným způsobem vykreslen zobrazovacím řetězcem výsledné scény. V kroku 3 je třeba pamatovat, že je třeba vypočítat i alfa kanál textury, což trochu prodlužuje celý proces. Nejprve je oblak vykreslen pouze do RGB složek fragment bufferu s míchací funkcí nastavenou parametry `GL_SRC_ALPHA` a `GL_ONE_MINUS_SRC_ALPHA`. Poté je povolen zápis pouze do alfa složky fragment bufferu a oblak je vykreslen s nastavením míchací funkce `GL_ONE` a `GL_ONE_MINUS_SRC_ALPHA`.

Je třeba stanovit ve kterých případech dojde k novému generování impostoru. Zavádí se chyba přiblížení, chyba posuvu a doplňkové chyby. Pokud některá z těchto chyb překročí stanovený práh, dojde k novému generování impostoru. Překročení prahu chyby přiblížení (zoom error) nastává, pokud podle vzorce 4.1 dvojnásobně klesne resp. stoupne požadované rozlišení textury oproti současnému. Pokud je rozměr oblaku konstantní, dochází k tomu při pohybu kamery k nebo od oblaku. Pokud by jsme uvažovali vzorec 4.2, byla by chyba přiblížení ovlivněna i změnou zorného úhlu kamery. Textury impostorů by si sice zachovaly vysokou ostrost, ale docházelo by k jejich nárazovému generování. V běžné praxi se zorný úhel při výpočtu chyby nezohledňuje.



Obr. 4.7: Chyba přiblížení ( $e_z$ )

Chyba posuvu (translation error) je velikost úhlu, pod kterým je z aktuální pozice kamery viděna úsečka, která je rovnoběžná s vektorem mezi objektem a původní pozicí kamery, prochází středem objektu a její délka je rovna velikosti objektu. Překročí-li tato odchylka nastavený práh, dojde k novému generování. Prahová hodnota přibližně  $0.15^\circ$  je považována za hranici, při které už lidské oko prakticky nezaznamená skokovou změnu impostoru. Praxe ukázala, že lze jít až na práh o hodnotě  $1^\circ$ , při vyšších hodnotách jsou již náhlé změny textury patrné.



Obr. 4.8: Chyba posunu ( $e_T$ ).

Mezi doplňkové chyby lze zařadit všechny zbývající faktory, které ovlivňují vzhled oblaku. Může to být změna vektoru od středu oblaku ke zdroji světla, změna barvy osvětlení, změna hustoty oblaku a podobně. Mnohé z těchto parametrů jsou v průběhu celého běhu programu konstantní, proto je možné vyžádat nové generování impostoru explicitně, společně se změnou jednoho z těchto parametrů, bez nutnosti sledovat míru chyby a porovnávat ji s prahem.

## 4.5 Další technologie

V bodech uvedu některé další technologie, které se používaly a používají v počítačových hrách, ať už je to částečně volumetrické zobrazování či dvourozměrná náhražka. Některé byly zmíněny již v předchozích kapitolách.

**Kopule** – není možné jí proletět, vždy je zobrazena v konstantní vzdálenosti od kamery. Dnes se od této technologie v leteckých simulátorech ustupuje, či už se prakticky pro zobrazení oblačnosti nepoužívá. Přesto opodstatnění mít může v prostorově omezeném prostředí, řekněme o velikosti 20 na 20 km pro zobrazení oblačnosti ve vzdálenosti například 40 km. Samozřejmě je kopule vhodná pro zobrazování objektů ve velké dálce – typicky Slunce, Měsíc, hvězdy a také barva atmosféry.

**Plochá vodorovná textura** může být vhodná ke zobrazení oblačnosti vysokého patra a to zejména pokud se nepředpokládá přiblížení kamery na bezprostřední vzdálenost k této oblačnosti. Rozsáhlý systém, například oblaků typu cirrus, pak reprezentuje jediná texturovaná plocha, vzhled je věrohodný a rychlost zobrazování scény se prakticky nezmění.

Snížení viditelnosti pomocí testování **Z bufferu** a následného přičtení barvy mlhy do bufferu barev nebo případně analogická funkce grafického API v kombinaci s jakoukoliv předchozí technologií může simulovat snížení viditelnosti při průletu oblakem nebo mlhou. Toho je často využito například v těsné blízkosti plochých vodorovných textur nebo u přízemní mlhy. Takováto mlha mnohdy není vizuálně ideální, je jednoduší bez struktury, pokud jí kamera neprolétá skrz, pak není vidět, ale v některých případech svůj účel plní a to spolehlivě a s nízkými výpočetními nároky – při průletu skrz snižuje viditelnost.

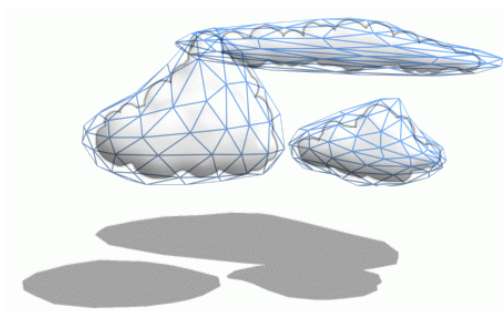
Existují další univerzitní i prakticky aplikovaná řešení zobrazování třírozměrných i dvourozměrných oblaků, o některých z nich se zmiňuji v kapitole „Existující implementace“. Uvedl

jsem zde jen ty neznámější a pro řešený úkol přicházející v úvahu. Při řešení vlastního systému se budu inspirovat právě zmíněnými technologiemi.

## 4.6 Stín

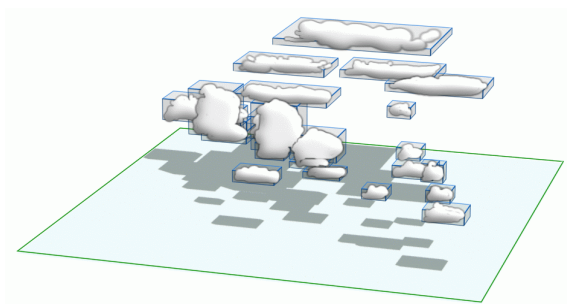
Samotné zobrazení stínu na povrchu země není příliš komplikované. Problém však nastává, když si uvědomíme, že stín by se měl projevit i na objektech, jako jsou budovy, vegetace, pozemní i letecká technika – každý objekt by musel znát úroveň osvětlení resp. stínění na své pozici a podle toho ztlumit své osvětlení. Jak se stín zachová v případě protnutí oblaku s terénem či jakýmkoliv jiným objektem? Oblačnost nedává stíny s ostrým okrajem, zejména oblačnost ve středním a vyšším patře. Jak dosáhnout rozmazání okrajů, při zachování výpočetních nároků v rozumných mezích? Tyto otázky budou zodpovězeny až při případné implementaci stínu.

Způsobů jak generovat stín je více, uvedu zde tři. V prvním případě má každý oblak své polygonální obalové těleso. Stín je pak vytvořen geometricky, poměrně dobře kopíruje okraje oblaku, jeho hranice jsou však ostré. Navíc při umístění několika oblaků nad sebe by docházelo k několikanásobnému překrytí jednotlivých stínů a výsledek by působil velmi uměle. Dalším úskalím tohoto způsobu je uložení obalových těles v paměti a především jejich aktualizace vlivem vývoje oblačnosti. Geometrické transformace vrcholů by mohly nadměrně vytěžovat procesor a proto toto řešení považuji za nepříliš vhodné.

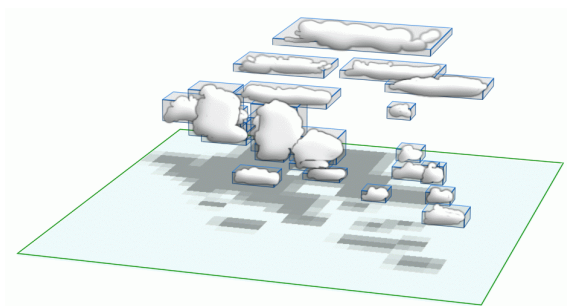


*Obr. 4.9: Stín vytvořený z polygonálního obalového tělesa.*

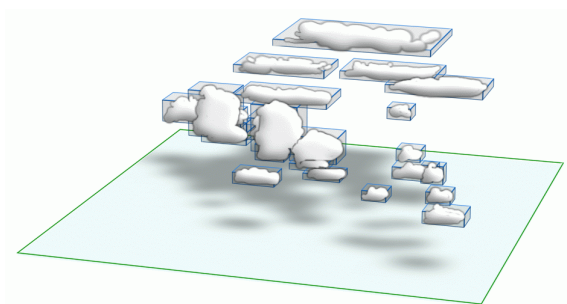
Ve druhém případě je každý oblak umístěn v jednom či ve více obalových kvádrech. Geometrickým výpočtem jsou určeny oblasti zastínění, ty jsou navzorkovány s intervalem desítek až stovek metrů do rastru. Rastr je pomocí multitexturingu a se zapnutým filtrováním aplikován na terén. Takto dojde ke ztmavení oblastí pod oblačností, pro různé typy a výšky oblaků lze regulovat úroveň zastínění, navíc okraje stínů jsou rozmazané. V tomto případě je však náročnost na výpočet také poměrně vysoká, vysoký by byl také objem přenesených textur po sběrnicích počítače a grafické karty.



*Obr. 4.10: Geometrický stín obalových kvádrů.*



*Obr. 4.11: Stín oblaků navzorkovaný v nižším rozlišení.*



*Obr. 4.12: Aplikace stínu navzorkovaného v nižším rozlišení a se zapnutým filtrováním na terén.*

Jako poslední uvádím v praxi úspěšně používanou metodu. Její princip je podobný, jako v předchozích případech, ale na rozdíl od nich využívá výkonu GPU, její implementace je díky grafickému API podstatně usnadněna a výsledné kvality stínu jsou také vyšší. Postup tvorby stínu lze velice krátce shrnout následovně: (1) nastavení ortografické projekce tak, aby zobrazovací frustum pojalo buď veškerou oblačnost či jen jednotlivá oblaka, (2) nastavení velikosti zobrazení na velikost požadované stínové mapy, (3) nastavení pozice kamery na pozici slunce a cíle na pozici středu oblačnosti, (4) vykreslení oblaků do framebuffer, (5) kopie dat z framebuffer do textury, (6) za pomoci multitexturingu aplikujeme získanou texturu na model terénu. Takto jsme vytvořili realistický stín, přesně odpovídající hustotě různých částí oblaků. Výpočet můžeme urychlit vypnutím míchání průhlednosti pro elementy oblaků, použitím pouze obalových kvádrů pro generování oblaků a podobně.

## 4.7 Optimalizace

Vizualizace oblačnosti je výpočetně náročný proces a jakákoliv optimalizace je velmi žádaná. Několik optimalizačních metod již bylo zmíněny v předchozích kapitolách, krátce zde uvedu ještě několik dalších. První je umístění textur oblaků do méně souborů, nejlépe do jediného – odpadá potom zpomalení, způsobené přepínáním textur ve stavovém stroji grafického rozhraní (OpenGL). K urychlení také vede přesunutí grafických, případně i jiných výpočtů z CPU do GPU, tedy do vertex a fragment shaderu. Obvodová realizace grafického čipu totiž vykazuje, například u maticových operací, podstatně vyšší výkon, než jakého je možné dosáhnout na centrálním procesoru daného počítače. Oblačnost není striktně deterministický jev a mnoho jejích parametrů, ať už dodávaných do nebo získávaných ze systému generování a zobrazování oblačnosti, může být zaokrouhleno, tedy uloženo na méně bitech. Dojde tak k poklesu vyžití mezimodulové komunikace, ke snížení nároků na operační paměť a při dobré implementaci i k nárůstu výpočetního výkonu apod.

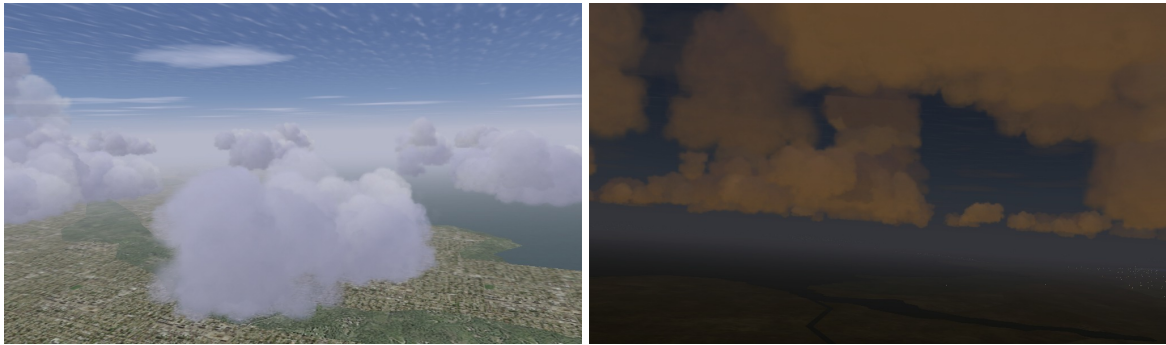
## 4.8 Existující implementace

V této kapitole uvedu příklady několika více či méně úspěšných komerčních, či do kategorie GNU General Public Licence spadajících implementací volumetrické oblačnosti. Srovnám základní vlastnosti, přednosti, zápory a některé implementační detaily.

### *FlightGear*

FlightGear je nekomerčním, ve formě zdrojových kódů dostupný leteckým simulátorem. Oblačnost má v současné verzi 0.9.10. implementovánu ve dvou úrovních. Prvně jde o základní oblaka, zobrazené formou hladivých textur různé struktury nebo hustoty. Druhá úroveň jsou experimentální volumetrická oblaka ve formě shluků texturovaných desek, rotujících čelem ke kameře. Výsledný celek sice trpí několika nedostatky, přesto v kontextu na dnešní dobu podprůměrně zpracovaného okolí jde o přijatelné a dostatečné řešení. Nedostatkem je například minimalizace texturované desky, pokud je vektor pohledu blízký svislé ose. Dalším nedostatkem je zanedbané testování viditelnosti a zakrývání oblaků obou technologií. Hladinová oblaka jsou v jistých případech vykreslena až po volumetrických.





Obr. 4.13: Volumetrická kupovitá oblačnost v leteckém simulátoru FlightGear.

### ***Sundog Software – SilverLining***

Tento produkt je C++ knihovnou pro vykreslování 3D oblaků a oblohy v reálném čase primárně ve hrách a vizualizacích na platformě Windows. Knihovna pracuje jak v grafickém rozhraní OpenGL, tak i Direct3D.

Zvolenou technologií zobrazování kupovitých oblaků jsou shluky texturovaných desek, jejichž normály jsou vždy rovnoběžné s vektorem pohledu. Desky tedy nerotují čelem ke kameře. Při pohybu v těsné blízkosti oblaku odpadá problém skokové rotace desky při malém posunu kamery, naopak je ale znemožněno použití kamery s velkým zorným úhlem. Dále je patrné protnutí texturovaných desek při jejich průletu. Konečně při pohledu o přibližně 135° nad nebo pod horizont dojde k postupné minimalizaci velikosti a převrácení velikosti veškerých desek.

Hladinová oblačnost, oblaka typu cirrus, je jako v drtivé většině jiných implementací reprezentována texturovanou deskou v dané výšce.

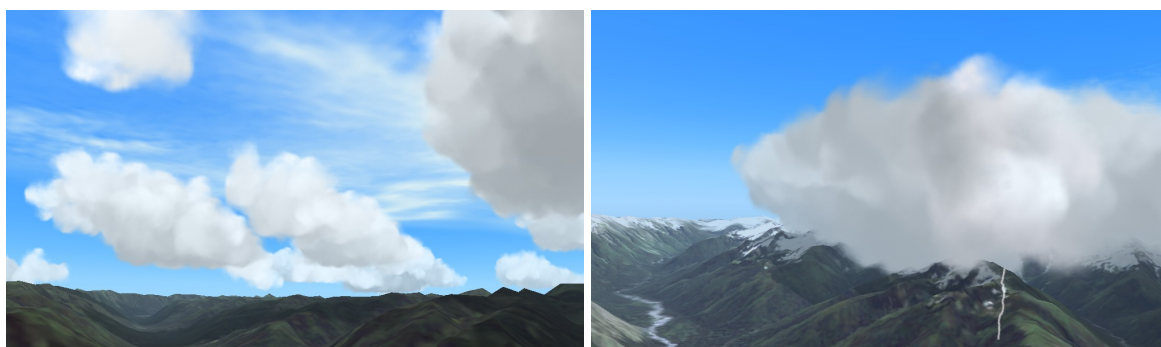
Implementace oblaků typu stratus není dokonalá. Základna a horní hranice oblaku je reprezentována jednoduchými texturovanými deskami. Efekt objemu při průletu je nahrazen hardwarově podporovanou mlhou. Mlha začíná několik desítek až stovek metrů vně prostoru, vymezeného dvěma deskami, v prostoru mezi deskami dosahuje své maximální hustoty. Toto řešení je sice velmi rychlé a v mnoha případech dostačující, přesto u tohoto konkrétního produktu trpí několika nedostatky:

- Při vstupu do oblaku, v okamžiku zobrazení hardwarově podporované mlhy, přestane být viditelná kupovitá oblačnost.
- Při vstupu do oblaku, v okamžiku zobrazení hardwarově podporované mlhy, dojde ke skokové změně barvy vzdálených objektů. Toto se týká především kupole, na které je vykreslena obloha.
- Po překonání hraniční oblasti oblaku, kdy postupně narůstá hustota mlhy až na svou maximální hodnotu, je patrné přibližování horní či spodní texturované desky. Především její protnutí působí rušivým dojmem.

Protrhaná vrstva stratu je simulována vykreslením odpovídající, místy průhledné, místy neprůhledné textury na obě desky. Úroveň hustoty hardwarové mlhy je potom regulována v závislosti na místě, ve kterém se kamera vyskytuje – v oblaku, či v místě protrhané oblačnosti. V případě protrhaného stratu je zde navíc problém neviditelnosti celé vrstvy při pohledu ze strany, ve vertikální poloze mezi oběma deskami a v horizontální poloze v místě protržení oblaku.

Krom vykreslování oblačnosti tento produkt také zajišťuje korektní osvětlení oblaků i ostatních objektů scény v závislosti na zeměpisné poloze a času, vykresluje pozadí oblohy s reálnou pozicí Slunce, Měsíce, planet a hvězd. Užitečnou vlastností je zobrazení stínů kupovité oblačnosti, samozřejmě v závislosti na poloze Slunce.

Více o produktu SilverLining na URL: <http://www.sundog-soft.com>.



*Obr. 4.14: Oblaka typu cumulus a cumulonimbus v podání produktu SilverLining.*

### ***Microsoft – Flight Simulator 2004 (2003)***

Oblačnost v leteckém simulátoru Flight Simulator 2004 lze považovat za jednu z nejlepších komerčních implementací pro platformu her osobních počítačů. Není sice na dnešní špici vizuální dokonalosti, ale vyniká univerzálností a jednoduchostí.

Veškerá kupovitá oblačnosti je reprezentována shluky texturovaných desek, rotujících vždy směrem ke kameře. Také vrstevnatá oblačnost používá stejnou technologii. To je poměrně zajímavé, protože naprostá většina jiných produktů tento problém řeší dvěma horizontálními hraničními deskami a hardwarově podporovanou mlhou. Takový systém sice klade značně vyšší výpočetní nároky, ale má jednu zcela zásadní výhodu – protrhaná vrstevná oblačnost má stejnou úroveň realismu, jako oblačnost kupovitá, což například u produktu SilverLining zdaleka neplatilo.

Hladinová oblačnost je řešena klasickým způsobem formou horizontální desky bez efektu hardwarové mlhy.

Oblaka nebo jejich uskupení nejsou do herního světa umisťovány samostatně, ale vždy jako bloky o rozloze 16x16 km. Řídící mechanismus při jejich umisťování vybírá ze seznamu předpřipravených bloků pro daný typ oblaků a danou hustotu. Popis oblaků v bloku je manuálně vytvořen grafikem v programu 3D Studio Max nebo gmax. Na jedné straně je tento přístup velmi

vítaný, protože dává možnost velmi realisticky navrhnout tvary oblaků a jejich konfiguraci v kontextu větší oblasti. Na druhé straně je ale znemožněna jakákoliv realistická simulace jejich vzniku, vývoje a zániku. Procesy na vyšší úrovni, tedy vznik, zánik, posun podle větru, probíhají vždy v rámci celého bloku.

Celý takto popsaný systém by byl pochopitelně značně náročný na přípravu texturovaných desek, zejména jejich rotace, a samozřejmě následné vykreslení. Proto nastupuje řada optimalizačních metod. Především je to možnost explicitně nastavit kvalitu zobrazení – vzdálenost vykreslování 3D oblačnosti a poměr mezi volumetrickými 3D oblaky a jejich velmi jednoduchými 2D náhradami. Oblaka ve vzdálenosti větší, než je nastavena pro 3D vykreslování, jsou promítnuta na šestiboký válec, pohybující se společně s kamerou. K update impostorů dochází při změně polohy kamery o 15% poloměru válce v horizontální rovině nebo o 2% ve vertikálním směru. Další odlehčení nároků je v odkládaném přepočtu orientace a stínování desek.

Více o produktu Flight Simulator na URL: <http://www.fsinsider.com>

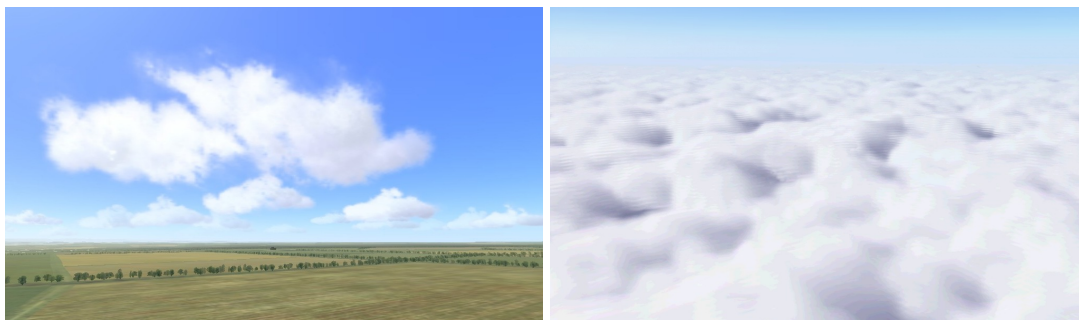


*Obr. 4.15: V popředí prvního obrázku jsou kupovitá oblaka vykreslena volumetricky, oblaka na obzoru jsou vykreslena na šestiúhelník – impostor. Druhý obrázek ukazuje téměř souvislou vrstvu oblaků. Oba snímky pořízeny ze hry Flight Simulator.*

### ***Eagle Dynamics - Lock On: Modern Air Combat (2003)***

Volumetrická oblaka jsou patrně zobrazena technologií splatting. Textura jednotlivých ploch má konstantní bílou barvu, její alfa kanál je vytvořen šumovou funkcí, čímž je oblaku dodána detailnější struktura. Alfa kanál je navíc animovaný, oblaka tedy působí „živě“, nestaticky. Na tomto počítačovém simulátoru stojí za zmínku také realizace souvislé vrstvy stratu. Běžný přístup se dvěma horizontálními deskami a hardwarovou mlhou byl na horní hranici stratu vizuálně vylepšen. Horní hranice stratu je tvořena přibližně dvaceti horizontálními řezy, vzájemně odstupňovanými o několik desítek metrů. Každá vyšší vrstva má texturu s alfa kanál s řidším zastoupení neprůhledných pixelů. V kombinaci s korektním nasvětlením je dosaženo při pohledu z větší výšky realistického plastického vzhledu. Více o této technologii v [Die00]. Problém protrhané vrstevnaté oblačnosti není ani v této aplikaci řešen.

Více o produktu Lock On: Modern Air Combat na URL: <http://www.lo-mac.com>.



Obr. 4.16: Oblaka typu cumulus a horní hranice souvislé oblačnosti v simulátoru Lock On.

### ***Maddox Games: IL-2 Sturmovik (2001)***

Model oblačnosti je v tomto simulátoru poměrně značně jednoúčelový – dokáže věrohodně zobrazit pouze středně četnou kupovitou oblačnost. Volumetrické oblaky jsou obdobné, jako ve hře Lock On. Souvislá oblačnost zde neexistuje, použita je pouze hardwarová mlha.

Více o produktu IL-2 Sturmovik na URL: <http://www.pacific-fighters.com>.



Obr. 4.17: Jediná volumetrická oblačnosti v simulátoru IL-2 Sturmovik.

## **4.9 Směr další práce**

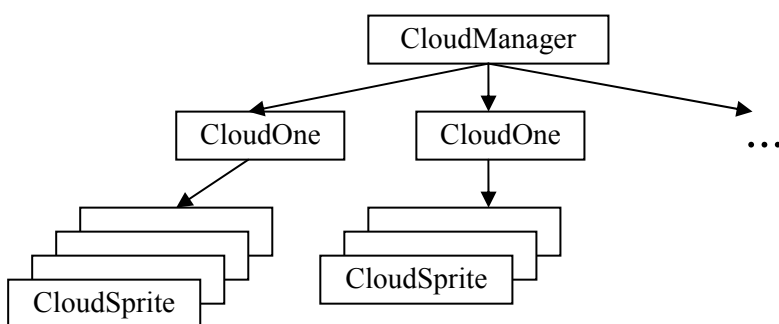
Při implementaci vizualizace oblaků bych se nejprve zaměřil na testování vhodnosti různých technologií, především však na metodu shluků texturovaných ploch. Ve výsledku je důležité získat věrohodný vzhled oblačnosti, ale samozřejmě také rozumné nároky na výpočetní výkon. Co se typů oblaků týče, zaměřil bych se nejprve na oblaky typu cumulus. Pokud se podaří najít vhodnou technologii k jejich zobrazení, potom už bude adaptace na mnohé další typy oblaků poměrně snadná. Druhým typem budou oblaka formovaná do hladiny (cirrus) nebo vrstvy (stratus). Podaří-li se najít i pro ně uspokojivé řešení, bude možné kombinací těchto dvou technologií zobrazit jakýkoliv typ oblaku. V další fázi bych se zaměřil na optimalizace, vyřešení některých problémů s viditelností a zkvalitnění vizuální stránky. Zobrazování stínu bych zatím zanedbal.

## 5 Implementace

Tato kapitola popisuje již konkrétní řešení zadaného problému. Na základě testování různých existujících systémů vybírám jejich vlastnosti a přístupy k řešení dílčích problémů a snažím se je kombinovat tak, aby výsledný produkt splňoval stanovené požadavky. Celý navržený systém vytvářím jako samostatnou OpenGL aplikaci. Jako nosnou technologii jsem vybral shluky texturovaných desek. Zejména řešení navržené Niniane Wang [Wan03] splňuje mnohé z požadavků na nový systém. Není postaveno na přesných fyzikálních výpočtech šíření světla nebo vývoje tvaru oblaku, zaměřuje se spíše na tvůrčí stránku – snadný design různých variant oblaků.

### 5.1 Třídní model

System pracuje se třemi základními třídami – CloudManager pro správu na nejvyšší úrovni, CloudOne pro reprezentaci jednoho volumetrického oblaku a CloudSprite pro reprezentaci texturovaných desek.



Obr. 5.1: Schéma základního třídního modelu.

#### *CloudManager*

Tato třída zajišťuje inicializaci a řízení běhu systému, z pohledu uživatele tvoří abstrakční vrstvu nad samotnou reprezentací oblaků. Třída CloudManager v sobě uchovává:

- informace o osvětlení – směr ke Slunci a Měsíci, barvu slunečního a měsíčního svitu a barvu rozptýleného osvětlení od oblohy
- vektor a rychlost větru
- seznam objektů třídy CloudOne
- seznam stejných objektů třídy CloudOne, tentokrát tříděný od nejvzdálenějšího k nejbližšímu oblaku
- seznam datových struktur s popisem časově závislého chování každého oblaku
- seznam modelů oblaků, načtených z datových souborů

- simulační čas
- nastavení úrovně detailů – dohlednosti
- identifikační číslo textury pro volumetrická oblaka
- pomocné proměnné pro měření času běhu jednotlivých úseků programu

Výhodou je, že parametry popisující život a pohyb oblaku jsou separovány od jeho geometrického modelu a veškerá simulace počasí probíhá ve třídě CloudManager. Takto lze simulaci dále rozvíjet bez nutnosti zasahovat do třídy CloudOne.

Programátor se systémem komunikuje rozhraním, definovaným veřejnými metodami třídy CloudManager. Při inicializaci, ale i po čas simulace jsou k dispozici tyto možnosti:

- nastavit identifikační číslo textury
- zvolit úroveň detailů
- přidat oblak určitého typu a nastavit jeho pozici, relativní velikost a velikost a hustotu jeho elementů
- nastavit časově závislé chování oblaku
- nastavit vektor ke Slunci a Měsíci

Poněkud jiné kategorie jsou následující možnosti:

- vynucení přepočtu geometrie a stínování oblačnosti např. po změně vektorů osvětlení a pozice kamery
- příprava geometrie a stínování před vykreslením
- vykreslení oblačnosti do fragment bufferu

Konečně jsou k dispozici pomocné metody pro získávání informací ze systému:

- dotaz, zda se kamera nachází uvnitř oblaku
- získání parametrů systému – např. počet oblaků, počet texturovaných desek apod.

## ***CloudOne***

Každá instance této třídy reprezentuje jeden volumetrický oblak, uchovávají v sobě tyto informace:

- souřadnice středu základny oblaku v globálním souřadném systému
- souřadnice středu oblaku v globálním souřadném systému pro stínovací algoritmus
- velikost obalového kvádru
- nastavený vývojový stupeň a vývojový stupeň, platný při předchozího přepočtu oblaku
- hodnota, podle které se provádí řazení oblaků od nejbližšího k nejbližšímu
- seznam objektů třídy CloudSprite
- seznam kvádrů, ze kterých byl oblak vygenerován (uchováno pro výpočty interakce)
- vertikální gradient barvy pomocné složky rozptýleného světla
- identifikační číslo textury oblaku

- pole vrcholů<sup>21</sup> pro vygenerovanou geometrii oblaku
- pozice kamery, platná při posledním přepočtu geometrie a stínování oblaku
- režim zobrazení – zda je zobrazen volumetrický oblak, impostor, či je oblak skryt
- sada dat impostoru – pozice, velikost, identifikační číslo textury, pole vrcholů pro vygenerovanou geometrii impostoru a další

Metody lze ve zkratce rozdělit na ty, používané k vytvoření oblaku, změnu stavových veličin oblaku, přepočítání jeho geometrie a osvětlení, generování impostoru a zobrazení.

### ***CloudSprite***

Každá instance této třídy reprezentuje jednotlivé elementy volumetrických oblaků – texturované desky. Uchovává v sobě:

- geometrické informace: pozice, rozměr, normála, rotace
- hodnota, podle které se provádí řazení elementů od nejvzdálenějšího k nejbližšímu
- index do předdefinovaného pole s definicí texturovacích souřadnic
- relativní, nemetrickou vzdálenost od středu základny oblaku (viz kapitola „Simulace“)
- několik proměnných s informacemi o průhlednosti, lišící se procesem, kterým byly vypočítány – např. při snižování průhlednosti desky v blízkosti kamery
- příznak zmrazení normály a relativní pozice kamery vzhledem ke středu desky v okamžiku, kdy došlo ke zmrazení (s touto informací se v projektu nepracuje, nicméně bude patrně použita v budoucnu při predikci normály plochy po průletu v její blízkosti, viz kapitola „Geometrická reprezentace oblaku“)

## **5.2 Simulace**

Pokud není potřeba přesné fyzikální chování a doba běhu programu je nejvýše v řádech desítek minut, lze zatím použít velice zjednodušený model vývoje oblaků. Zjednodušený až na takovou úroveň, kdy do výpočtu vstupuje pouze časově invariantní vektor větru. Aktuální poloha oblaku je získána následovně:

$$poloha_{aktuální} = poloha_{počáteční} + (čas_{aktuální} - čas_{vzniku}) \cdot vítr \quad (5.1)$$

Chování oblaku, co se změny vývojového stupně týče, je definováno pěti parametry – počátek a konec fáze růstu, počátek a konec fáze zániku, maximální hodnota vývojového stupně. Popsaným

---

<sup>21</sup> Pole vrcholů, nebo-li vertex array, je pole obsahující geometrická a materiálová data trojrozměrného objektu. Vykreslení objektu neprobíhá voláním množství příkazů pro nastavení souřadnic vrcholu, jeho barvy a podobně, ale je voláno jen několik málo funkcí, které zajistí vykreslení velmi komplexní geometrie. Na rozdíl od obdobné technologie tzv. display listů, které sice dokáží lépe využít paměti grafické karty, je u pole vrcholů výhoda v možnosti průběžně libovolně měnit data vrcholů.

nastavením lze oblak instruovat, aby vznikl na jisté pozici, postupně rostl a nechal se unášet větrem a konečně pomalu zaniknul.

Různý vývojový stupeň oblaku je vizuálně aproximován postupným a plynulým zobrazováním oblaků od středu základny až do plné velikosti. Vzdálenost jednotlivých desek od středu základny je vypočítána při generování oblaku, vzdálenost přitom není metrická (počítána v kartézském souřadném systému scény), ale relativní v rámci rozměrů obalového kvádrů oblaku. Přidáním náhodnosti do vzdálenosti je dosaženo jistého nedeterminismu, který je vidět v nejlevějším políčku následujícího obrázku – oblak tak neroste přesně od středu, ale náhodně v oblasti kolem středu.



*Obr. 5.2: Vzhled oblaku během různých vývojových stupňů.*

S přidáním souvislé slohovité oblačnosti bylo nutné přidat ještě jeden způsob růstu oblaku. Slohovitá oblačnost se do scény umísťuje po dlaždicích o rozměru přibližně 5 x 5 kilometrů. Růst od středu by zvýraznil jejich pravidelné uspořádání, proto byla přidána možnost růstu formou snižování průhlednosti – oblak se objevuje ve své plné velikosti.

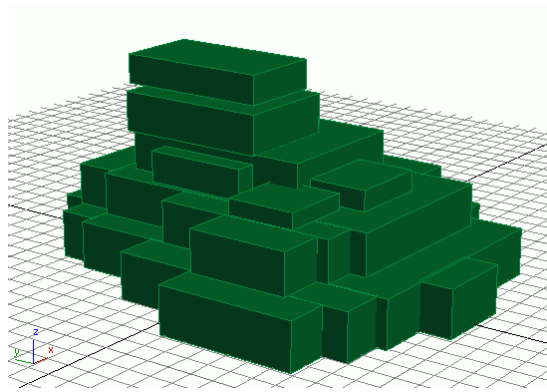
Třídní model je navrhnout tak, aby šlo současný primitivní přístup k simulaci v budoucnu snáze samostatně rozvíjet.

## 5.3 Modelování základního tvaru

V předchozích kapitolách bylo stanoveno, že realistická simulace vývoje počasí bude zanedbána, tedy i reálný růst a zánik oblaků nebude dokonale korespondovat se skutečností. Tato skutečnost nám dává možnost zcela zavrhnout přesné simulační techniky, jako jsou například buňkové automaty nebo fluidní dynamika, a použít tvůrčí přístup pro vymodelování základního tvaru oblaku.

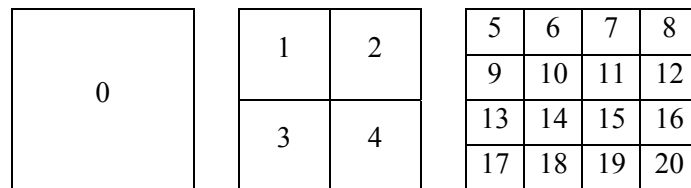
Jako nástroj pro vytvoření základního tvaru oblaku jsem se rozhodl použít prostředí produktu 3D Studio Max a jeho volně šiřitelné varianty gmax. Důvodem použití právě těchto programů je jejich rozšířenost, oblíbenost a především podpora skriptovacího jazyka MaxScript, ve kterém je naprogramován modul pro export modelů oblaků.





Obr. 5.3: Definice tvaru oblaku v 3D Studiu Max.

Definice základního tvaru je ve zmíněných programech provedena umístěním kvádrů do prostoru. Každý kvádr představuje část oblaku, která se později vyplní jednotlivými elementy (texturovanými deskami). Pokud bude kvádrům manuálně přiřazeno Material ID<sup>22</sup> a při exportu bude vybrána patřičná možnost, bude toto číslo určovat část textury, která bude na desky příslušející tomuto kvádru vykreslena.



Obr. 5.4: Číslování částí textury.

Formát souboru s popisem exportovaných kvádrů (soubor s příponou clm) je textový. Řádky uvozené středníkem jsou brány jako poznámka. Řádky uvozené znakem procenta popisují barevný gradient. Za procentem je pět hodnot s plovoucí desetinnou čárkou, kde první je relativní výška v oblaku, pro kterou bude barva platit, a zbývající popisují barevné složky RGB a alfa. Každý další řádek popisuje jeden kvádr. Parametry jsou separovány mezerami, pokud není uvedeno jinak, jsou to hodnoty s plovoucí řádovou čárkou: souřadnice X středu základny, souřadnice Y<sup>23</sup> středu základny, souřadnice Z středu základny, velikost v ose X, velikost v ose Y, velikost v ose Z, číslo části textury (integer). Případné další parametry určí faktory pro různé typy interakce se simulátorem – faktor vzniku námrazy a další možné. Při současné implementaci nejsou tyto další parametry brány v potaz.

<sup>22</sup> Přiřazení Material ID patří mezi standardní funkce 3D Studia Max a gmax.

<sup>23</sup> Souřadnice Y v OpenGL standardně vyjadřuje vertikální osu.

## 5.4 Textury

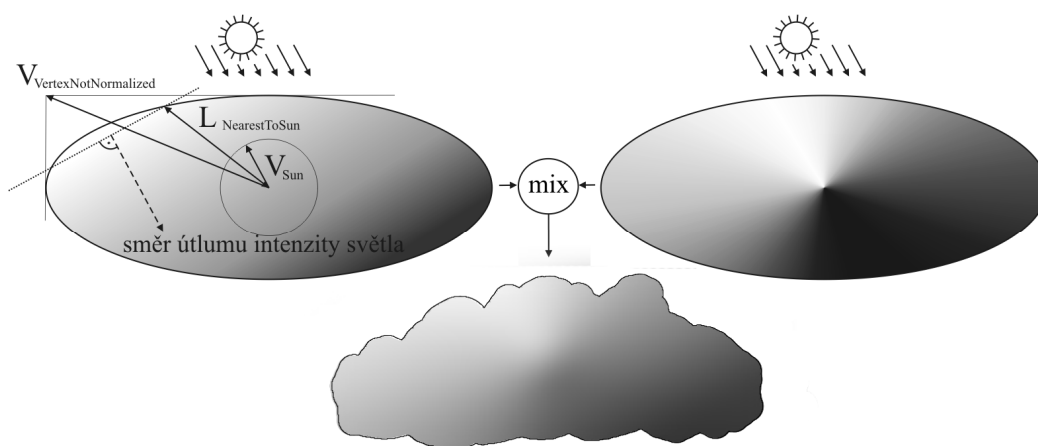
Veškeré textury jsou předem připravené. Jejich generováním pomocí Perlinova šumu bychom přišli o možnost mít plnou kontrolu nad jejich vzhledem. Navíc by se jen s obtížemi podařilo dosáhnout jejich věrohodnosti. V kapitole „Optimalizace“ jsem zmínil použití jediné textury pro vykreslení volumetrických oblaků. V souladu s tím jsem pro volumetrická oblaka připravil RGBA texturu o celkem 16 polích, kde každé představuje různě detailní a hustou strukturu elementů oblaku. Textury pro hladinovou a případně zjednodušenou vrstvou obláčnost jsou samostatné a v několika variantách pro různé subverze a stupně pokrytí oblohy.

## 5.5 Stínování

Nasvětlení elementů oblaku je prováděno nastavením barvy vrcholů příslušného polygonu. Barva vzniká spojením tří složek – světlo od Slunce, světlo od Měsíce a rozptýlené světlo z oblohy a země. Nejdůležitější složka, která se na výsledné barvě podílí, je sluneční světlo, následováno slabším rozptýleným světlem a konečně poměrně nevýrazným, ale přesto důležitým měsíčním světlem. Vztah pro výpočet barvy je následující s tím, že konstanty před všemi třemi složkami jsou zvoleny přibližně tak, aby výsledek působil věrohodně.

$$C = 0.8 \cdot I_{Sun} \cdot C_{Sun} + 0.2 \cdot I_{Moon} \cdot C_{Moon} + 0.5 \cdot C_{Ambient} \quad (5.2)$$

Výpočet intenzity slunečního svitu  $I_{Sun}$  jsem složil ze dvou částí. První aproximuje postupné zeslabení intenzity procházejícího světla uvnitř oblaku směrem od nasvětlené k zastíněné části. Oblak by však působil velmi ploše, proto druhá část simuluje osvětlení za pomoci výpočtu skalárního součinu normalizovaného vektoru od středu oblaku k vrcholu ( $V_{Vertex}$ ) a normalizovaného vektoru od středu oblaku ke Slunci ( $V_{Sun}$ ) a přidává tak osvětlení na „dramatičnosti“.



Obr. 5.5: Výpočet intenzity slunečního svitu.

Intenzita slunečního světla se vypočítá podle následujících vztahů:

$$I_{Sun} = mix \cdot \left( e^{-L_{Ray}/L_{Falloff}} \right) + (1 - mix) \cdot \left( \frac{(V_{Vertex} \bullet V_{Sun})}{2} + 0.5 \right) \quad (5.3)$$

$$L_{Ray} = \max\left(- (V_{VertexNotNormalized} \bullet V_{Sun}) + L_{NearestToSun}, 0\right) \quad (5.4)$$

$$L_{NearestToSun} = |V_{CloudHalfSize} \cdot V_{Sun}| \quad (5.5)$$

Kde  $V_{VertexNotNormalized}$  je nenormalizovaný vektor od středu oblaku k vrcholu. Hodnota  $L_{Ray}$  vyjadřuje velmi hrubou aproximaci délky průchodu slunečního paprsku oblakem až k vrcholu. Konstanta  $L_{Falloff}$  určuje rychlost exponenciálního útlumu intenzity slunečního svitu při průchodu oblakem. Její hodnota je vzdálenost, na které klesne intenzita na  $1/e$ , tedy na přibližně 36.8%.  $L_{NearestToSun}$  je vzdálenost mezi středem oblaku a bodem nejbližším Slunci. Na obrázku je vidět, že to ve skutečnosti není nejbližší bod, ale tato aproximace je vzhledem k rychlosti výpočtu dostatečná. Vzdálenost  $L_{NearestToSun}$  je získána jako velikost součinu nenormalizovaného vektoru od středu oblaku k nejbližšímu bodu obalového kvádru a normalizovaného vektoru ke Slunci. Podíl obou složek na výsledné intenzitě určuje parametr  $mix$ , jehož hodnota je pro dobrý vizuální výsledek nastavena v intervalu přibližně  $\langle 0.4, 0.7 \rangle$ .

Osvětlení Měsícem není příliš významné, spíše se jedná o doplňkový efekt, proto jsem intenzitu měsíčního svitu  $I_{Moon}$  omezil pouze na skalární součin normalizovaného vektoru od středu oblaku k vrcholu s vektorem od středu oblaku ke Měsíci ( $V_{Moon}$ ).

$$I_{Moon} = \left( \frac{(V_{Vertex} \bullet V_{Moon})}{2} + 0.5 \right) \quad (5.6)$$

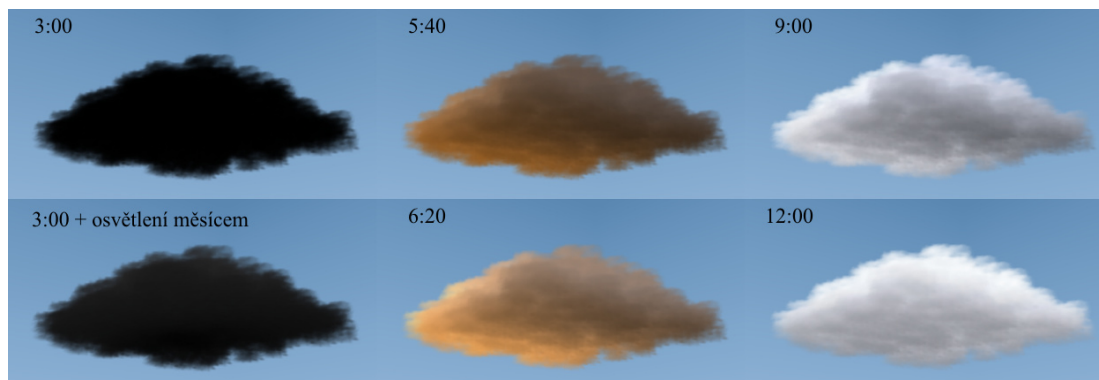
Barva slunečního ( $C_{Sun}$ ) i měsíčního ( $C_{Moon}$ ) světla je funkcí, jejímž jediným parametrem je vertikální složka normalizovaného vektoru k jednomu či druhému zdroji světla. Vertikální složka o hodnotě 1.0 odpovídá osvětlení, kdy je Slunce, resp. Měsíc přímo nad scénou a dopadající světlo je téměř bílé. Hodnota 0.0 platí pro zdroj světla na horizontu, kdy jsou díky vlastnostem atmosféry utlumeny kratší vlnové délky a světlo se jeví jako oranžové. Pokud zdroj světla klesá pod horizont, klesá i intenzita oranžové barvy, do zcela černé přechází přibližně  $11^\circ$  pod horizontem, což je ekvivalentní přibližně 45 minutám po západu slunce. Barvy jsou voleny v souladu s následující tabulkou.

Teplota barvy	RGB rozsah barev	situace
2 000 – 3 000 K	255 137 18 – 255 180 107	sluneční svit při východu a západu
5 000 – 5 400 K	255 228 206 – 255 235 220	sluneční svit v poledne
5 500 – 6 500 K	255 237 223 – 255 247 248	společný denní svit Slunce a oblohy
5 500 – 6 500 K	255 237 223 – 255 247 248	sluneční svit skrz oblačnost nebo mlhu
6 000 – 7 500 K	255 243 239 – 235 236 255	sluneční svit přes zataženou oblohu
8 000 – 10 000 K	227 233 255 – 204 219 255	obloha s lehkou oblačností

Tab. 5.1: Přibližné barvy slunečního svitu a svitu oblohy. Zdroj [www.3drender.com](http://www.3drender.com)

Rozptýlená barva závisí na poloze Slunce. Je světle modrá pro pozici Slunce vysoko nad obzorem, tmavší při pozici na horizontu a přechází v černou asi 17° pod horizontem. Barva je dodatečně modulována manuálně nadefinovaným vertikálním barevným přechodem směrem od základny k vrcholu oblaku. Takto je designérovi dána možnost například výrazněji ztmavit oblohou neosvětlovanou základnu oblaku.

$$C_{Ambient} = C_{Sky} \cdot C_{Vertical} \quad (5.7)$$



Obr. 5.6: Osvětlení oblaku v závislosti na vektoru ke Slunci a Měsíci. Vyznačený čas platí pro rovník, kdy Slunce vychází v 6:00.

## 5.6 Geometrická reprezentace oblaku

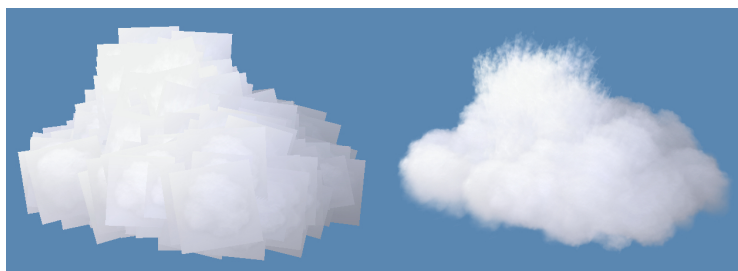
Pro geometrické vyjádření oblaku jsem zvolili technologii texturovaných desek, které vždy směřují čelem (normálou) ke kameře. Pro dosažení relativně vysoké rychlosti komunikace s grafickou kartou a rychlosti zobrazování jsem pro uložení výsledného zobrazitelného modelu použil pole vrcholů, a to vždy pro každý oblak jedno. Velikost tohoto pole se dynamicky mění s tím, kolik má být právě zobrazeno desek.

Pole vrcholů je generováno<sup>24</sup> z dat třídy CloudOne, jen pokud systém vizualizace oblačnosti rozhodne, že je k tomu má dojít. V kapitole „Impostory“ jsem popsal chybu posuvu (otáčení kamery kolem oblaku) a chybu přiblížení. Stejný mechanismus je použit v této implementaci a to nejen pro impostory, ale také přímo pro trojrozměrná oblaka. Navíc byla přidána chyba vývojového stupně. Nové generování je tak vyvoláno, až rozdíl vývojového stupně, pro který bylo platné předchozí generování, a aktuálního vývojového stupně dosáhne nastavené hodnoty. Pro zachování plynulosti animace růstu musí být tento rozdíl velmi malý (postupný růst oblaku je rozdělen na stovky diskretních kroků). Za předpokladu, že bude růst oblaku trvat až několik minut, budou intervaly mezi

<sup>24</sup> Generováním je myšlen výpočet osvětlení, výpočet pozic vrcholů a v případě potřeby také rendering a výpočet geometrie impostoru.

takto vyvolanými generováními až jednotky sekund. Přepočít geometrie po změně osvětlení musí být vynucen manuálně. Předpokladem je, že směr osvětlení bude po čas celé simulace konstantní a nebo bude docházet jen k jeho velmi pomalým změnám. Kontrola, zda došlo ke změně polohy Slunce či Měsíce by znamenala další výpočty, v tomto případě téměř zbytečné.

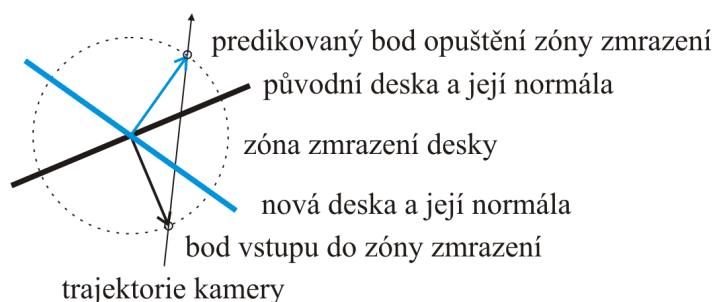
Oblak unášený větrem není před každým vykreslením generován znovu, ale jsou v poli vrcholů pouze připočteny polohové diference a ke generování dojde až po překročení popsanych chyb. Takto je dosaženo zcela plynulého pohybu, přitom s minimálními výpočetními nároky.



*Obr. 5.7: Oblak složený z desítek texturovaných desek.*

V předchozích kapitolách jsem zmínil nedostatky technologie desek, otáčejících se k bodu kamery. Největším úskalím je jejich rychlá rotace při blízkosti kamery. Problém je řešen zmrazením orientace desky, klesne-li vzdálenost od kamery pod nastavenou mez. Tou mezí je ideálně přibližně třetina až polovina velikosti desky. Pokud se kamera k desce dále přibližuje, stává se deska postupně průhlednou. V okamžiku protnutí desky trajektorií kamery dojde k otočení normály desky, po větším vzdálení začíná klesat průhlednost a v jisté vzdálenosti kamery od desky je deska znovu plně viditelná. Toto řešení není zdaleka ideální a při dalším rozvíjení projektu bude patrně nutné hledat lepší přístup. Jsou k tomu dva důvody. (1) Zejména při průletu okraje oblaku nastává často situace, kdy se při pohledu vpřed zdá, že kamera prolétává hustým oblakem (téměř kolmo na desky), ale při pohledu do stran (desky viděny téměř z boku) se oblak ztrácí a je jasně vidět pozadí. Primitivním řešením by mohlo být zamlžení obrazu, pokud se kamera nachází v blízkosti jakékoliv desky. Úkaz desek viděných téměř z boku by sice zůstal, ale došlo by k požadovanému snížení viditelnosti ve všech směrech. (2) Druhým důvodem neideálnosti takto jednoduchého zmrazování desek je náhlá rotace desky v okamžiku, kdy kamera opouští prostor zmrazení. V ideálním případě by kamera proletěla středem desky, normála desky se invertuje a kamera opouští prostor zmrazení ve směru stejném, jako je normála desky – k jakékoli skokové rotaci by nedošlo. Běžně však kamera neprolétá přímo středem desky, normála po inverzi tedy nesměruje do bodu, ve kterém kamera zónu zmrazení opouští a při pohledu vzad je v jistém okamžiku vidět náhlé otočení desky. Podle vektoru pohybu kamery lze predikovat bod opuštění zóny zmrazení a normálu desky po jejím protnutí kamerou nasměrovat do tohoto bodu. Toto by daný problém z velké části vyřešilo (za předpokladu, že se kamera pohybuje po přímce), nicméně to přináší další komplikace – potenciálně i vytvoření nové

desky a příkaz původní desce, aby za nějaký časový okamžik plynule zmizela. V současné implementaci jsem použil jednoduchou inverzi normály.



Obr. 5.8: Predikace bodu opuštění zóny zmrazení a vytvoření desky s novou normálou.

Dalším nedostatkem technologie je rychlá rotace desek blízko vertikály kamery. Tento jev vzniká při rychlém pohybu pod/nad oblakem – tedy téměř výhradně u leteckých simulátorů. Za předpokladu, že let probíhá většinou téměř horizontálně a pilotův pohled směřuje obvykle vpřed či do stran, nemusí být rotující desky příliš vidět, zvláště pokud projekční plocha není plně kulová a část nad pilotovou hlavou není zobrazována. Proto jsem se rozhodl tento problém nijak neřešit.

Nicméně problém rychlé rotace desek blízko vertikály kamery je nutno zohlednit při rozhodování o potřebě přepočtu stínování a geometrie. Aby animace rotace desek byla plynulá, je úhlová chyba násobena koeficientem, který roste s tím, jak se kamera blíží pod/nad oblak. Takto uměle zvýšená úhlová chyba zajistí častější přepočet geometrie a tudíž plynulejší animaci desek.

## 5.7 Vykreslovací řetězec scény

Korektní vykreslení volumetrických oblaků vyžaduje dodržení přesného postupu při zobrazování scény a nastavování stavového stroje OpenGL. Posloupnost úkonů lze obecně popsat následovně:

1. Jsou vykresleny veškeré objekty scény, které tvoří oblačnost.
2. V OpenGL je zapnut test hloubky (`GL_DEPTH_TEST`) a testovací funkce je nastavena tak, aby test prošel, pokud hloubka nově zpracovávaného fragmentu je menší, než ta, uložená ve hloubkovém bufferu (`GL_LESS`). Zapnutím testu hloubky je zajištěno, že oblaky za již vykreslenými objekty se nezobrazí.
3. V OpenGL je vypnut zápis do hloubkového bufferu. Takto se zamezí vzniku artefaktů, pokud jsou dvě texturované desky příliš blízko u sebe a jedna ve fragment bufferu přemazávala druhou.
4. V OpenGL je zapnuto míchání (`GL_BLEND`) a příslušná funkce nastavena standardně, jak je zvykem při vykreslování více průhledných polygonů za sebou (`GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA`).
5. Volumetrické oblaky se seřadí od nejvzdálenějšího k nejbližšímu ke kameře.

6. Jednotlivé elementy (texturované desky) se v rámci jednoho oblaku setřídí od nejvzdálenějšího k nejbližšímu ke kameře.
7. Elementy celé oblačnosti se vykreslí v pořadí, připraveném v předchozích krocích, tedy od nejvzdálenějšího k nejbližšímu.
8. Zpětně se povolí zápis do hloubkového bufferu.

Posloupnost se mírně změní po zavedení impostorů. Ty musí být vytvořeny před vykreslením objektů scény, protože se při jejich tvorbě využívá fragment buffer. Posloupnost úkonů je potom přibližně následující: (1) výpočet geometrie oblaků včetně řazení, (2) tvorba impostorů, (3) vykreslení objektů scény, (4) vykreslení oblačnosti.

Problém nastává, pokud některý objekt vykreslený v prvním kroku (objekt scény) je poloprůhledný, ať už nastavením barvy nebo na základě textury s alfa kanálem. Takový objekt byl totiž zapsán do hloubkového bufferu a oblaka za ním se nevykreslují. Takto je problematické také vykreslení kouře. Při implementaci do cílové aplikace bude nutné brát ohled na tyto skutečnosti a vykreslovací řetězec celé scény i s oblačností patřičně upravit.

## 5.8 Rychlost výpočtů a zobrazování

Testování bylo provedeno na hardwarovém vybavení CPU Intel Core Duo T2300E 1.66GHz, RAM 1024MB DDR2 667MHz, VGA GeForce Go7300 128MB (frekvence jádra 350MHz, frekvence paměti 700MHz). Operační systém byl Windows XP SP2, ovladač grafické karty ForceWare 92.91. Pokud není uvedeno jinak, bylo rozlišení při testování 640 x 480 pixelů. Rychlost přípravy a zobrazení je závislá na množství oblaků, nastavených detailech a vzájemné pozici kamery a jednotlivých oblaků. Popíšu zde stručně výkon při různě nastavené oblačnosti. Při všech je kamera přibližně ve středu celé scény, při statických testech je kamera mírně nad oblačností a sleduje ji pod takovým úhlem, aby 70% obrazu zakrývala oblaka a zbývajících 30% bylo pozadí nad horizontem. Pokud se kamera pohybuje, pak je to průlet nad, pod a skrz oblačnost rychlostí 720km/h. Celková dohlednost je 100 km, vzdálenost přepnutí mezi 3D oblaky a impostory je přibližně 60 km. Je-li nastaven vítr, je jeho rychlost 10 m/s.

Testování jsem provedl při dvou konfiguracích oblačnosti. První obsahuje souvislý stratus o pokrytí 6/8. Stratus je tvořen 100 oblaky a celkem 30 000 texturovanými deskami. Díky nastaveným detailům je 50 oblaků trojrozměrných (15 000 desek), 50 oblaků jsou impostory. Druhá konfigurace obsahuje rozsáhlý, ale řídký systém cumulů o pokrytí oblohy přibližně 2/8. Je tvořen 1 000 oblaky s celkem 9 700 texturovanými deskami, z čehož 710 oblaků je trojrozměrných (6 900 desek), zbývajících 290 jsou impostory.

konfigurace oblačnosti	-	stratus				cumulus			
kamera	-	pevná		průletová		pevná		průletová	
vítr	-	ne	ano	ne	ano	ne	ano	ne	ano
snímky za sekundu	1/s	22.6	21.9	51.0	38.6	31.5	32.7	23.4	22.2
využití času oblaky <sup>25</sup>	%	96.7	95.2	80.8	75.3	97.1	96.8	94.8	95.4
čas oblaků na snímek <sup>26</sup>	ms	43.2	43.7	15.8	19.5	30.8	29.6	40.4	42.9
přepoččet po změně	ms	0.0	0.0	2.0	2.6	0.0	0.0	11.2	12.4
řazení oblaků	ms	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2
přepoččet oblaků	ms	0.0	0.0	1.6	2.0	0.0	0.0	3.6	5.0
příprava	ms	2.1	6.6	2.2	6.1	1.2	2.9	1.5	2.7
simulace	ms	1.1	5.2	1.0	5.0	0.5	1.8	0.5	1.7
řazení oblaků	ms	0.0	0.0	0.0	0.0	0.1	0.2	0.1	0.2
přepoččet oblaků	ms	1.0	1.3	1.2	1.1	0.6	0.9	0.8	0.9
zobrazení	ms	41.1	37.2	11.5	10.8	29.6	26.7	27.8	27.5
3D	ms	33.4	32.9	10.4	10.1	16.3	15.4	16.4	15.4
impostor	ms	7.3	3.8	0.8	0.3	9.6	7.7	7.7	8.5

Tab. 5.2: Srovnání rychlosti výpočtů a zobrazení celého systému a poté hlavních metod.

Tabulka ukazuje potenciálně úzká místa systému. Zejména metoda „simulace“ konzumuje u čtvrtého měření až 25% času. Metoda krom samotné simulace provádí také připočítání offsetu k souřadnicím vrcholů v polích vrcholů u všech oblaků, což je operace s časem přímo úměrným počtu vrcholů. Připočítání offsetu se v simulaci provádí po každé změně polohy oblaku vlivem větru a děje se před vykreslením každého snímku. Velikost mezisnímkového offsetu patrně nikdy nebude vyšší, než několik metrů. Při běžných podmínkách s větrem o rychlosti 10 m/s a snímkovací frekvencí 60Hz je mezisnímkový offset dokonce jen v řádech nejvýše desítek centimetrů. Zde je při budoucím zdokonalování projektu místo pro značně přínosnou a přitom velice jednoduchou optimalizaci – připočítání offsetu je možné podmínit tak, aby se provedlo pouze tehdy, pokud tím dojde k posuvu oblaku na obrazovce minimálně o jeden pixel.

$$práhPosunu = \frac{zornýÚhel[rad]}{rozlišení} \cdot vzdálenostOblaku \quad (5.7)$$

Pokud je offset menší, než pomocí předchozího vzorce vypočítaná vzdálenost, není třeba offset připočítávat do pole vrcholů. V praxi toto ošetření povede k přibližně 50 – 200 násobnému snížení počtu zásahů metody „simulace“ do polí vrcholů.

<sup>25</sup> Procentuální množství času běhu aplikace, které bylo využito metodami systému oblačnosti. Zbývající čas je využit např. k vykreslování ostatních objektů scény, textu apod.

<sup>26</sup> Střední doba pro přípravu a vykreslení oblačnosti na jednom snímku.

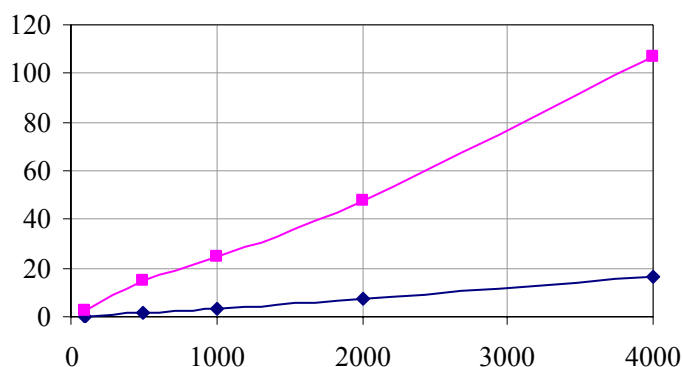


Řazení oblaků se i při pohybu kamery ukázalo jako zcela nezatěžující operace. Při tisíci oblacích ani v nejhorším případě nepřesahuje 2% času zpracování oblačnosti.

Podívejme se, jak doba výpočtu a zobrazení závisí na počtu oblaků. Toto je testováno na scéně s cumuly, bez větru a s kamerou letící v konstantní výšce, konstantní rychlostí 720 km/h. Ukáže se, že doba přípravy i zobrazení rostou s počtem oblaků téměř lineárně.

počet oblaků	-	100	500	1000	2000	4000
snímky za sekundu	1/s	323.0	59.3	35.1	17.3	7.7
využití času oblaky	%	93.7	96.6	96.1	96.5	96.8
čas oblaků na snímek	ms	2.9	16.3	27.4	55.7	125.2
přepočít po změně	ms	0.00	0.03	0.11	0.39	2.00
řazení oblaků	ms	0.00	0.00	0.00	0.00	0.00
přepočít oblaků	ms	0.00	0.03	0.11	0.39	2.00
příprava	ms	0.15	1.26	2.90	7.52	16.15
simulace	ms	0.02	0.13	0.47	1.34	3.00
řazení oblaků	ms	0.01	0.03	0.11	0.33	1.00
přepočít oblaků	ms	0.11	1.08	2.33	5.85	12.14
zobrazení	ms	2.75	15.01	24.41	47.73	107.05
3D	ms	0.37	8.39	14.11	27.85	61.97
impostor	ms	2.02	4.81	6.60	12.53	30.17

Tab. 5.3: Srovnání rychlosti výpočtů a zobrazení celého systému a poté hlavních metod.

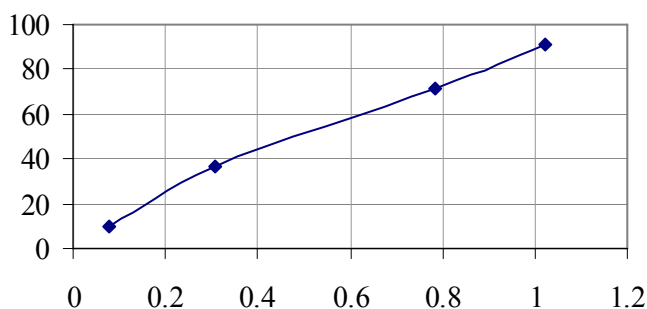


Obr. 5.9: Závislost doby přípravy [ms] (dolní křivka) a doby zobrazování [ms] (horní křivka) na počtu oblaků ve scéně.

Nyní ještě vliv rozlišení na rychlost zobrazení. Při testu je použita scéna se stratem, se statickou kamerou a aktivním větrem. Ani jeden test nebyl proveden v herním módu (fullscreen). Doba přípravy oblaků je u všech testů stejná 6.4 - 6.5 ms. Jak se ukáže, doba zobrazování roste s rozlišením téměř lineárně.

rozlišení	-	320x240	640x480	1024x768	1280x800
rozlišení	MPx	0.0768	0.307	0.786	1.024
snímky za sekundu	1/s	55.1	22.4	12.5	10.0
využití času oblaky	%	91.5	96.1	97.2	97.4
čas oblaků na snímek	ms	16.6	42.9	77.6	97.3
čas pro zobrazení	ms	10.1	36.5	71.2	90.9

Tab. 5.4: Srovnání rychlosti zobrazení při různém rozlišení.



Obr. 5.10: Závislost doby zobrazování [ms] na rozlišení výstupu [MPx].

Konečně se podívejme na nároky na operační paměť a paměť grafické karty. Mějme typickou scénu s 500 oblaky a celkem 5000 texturovanými deskami. Ze znalosti vlastností tříd a velikosti datových typů lze přibližně odhadnout následující:

- velikost objektů tříd CloudManager, CloudOne, CloudSprite, bez polí vrcholů – 250kB
- pole vrcholů – 960kB
- nekomprimovaná textura volumetrických oblaků – 1024kB

V součtu tedy tyto položky v běžné praxi nepřesáhnou hodnotu několika MB. Velikost paměti pro uložení impostorů však může přerůst v nepříjemný problém. Zejména pokud je nastaveno jejich příliš vysoké rozlišení a ve scéně je velký počet oblaků. Velikost textury impostoru je při rozlišení 128 x 128 pixelů 64kB. Typická scéna obsahuje 500 oblaků, což odpovídá 31.25MB paměti na grafické kartě. Při zvýšení rozlišení impostorů na 256 x 256 pixelů a s 2 000 oblaky ve scéně (tato situace může nastat) je už vyžadovaný paměťový prostor pro textury 500MB, což není při současném stavu hardware akceptovatelné. Komprese textur, dynamická změna jejich rozlišení a inteligentní skládání více menších textur do jedné může potřebný prostor redukovat až na jednu desetinu.

## 5.9 Směr další práce

System zobrazování volumetrické oblačnosti je v nyní demonstrován jednoduchou OpenGL aplikací, prioritou další práce na projektu je implementace do prostředí OpenSceneGraph ve spolupráci s průmyslovým zadavatelem. Poté by následovalo řešení problémů, kterými současná aplikace trpí a implementace nových vlastností. Shrnutí současných problémů a návrh jejich řešení následuje společně s výčtem dalších vhodných úprav.

Chování texturovaných desek v blízkosti kamery není ideální, bude vhodné se pokusit o využití predikce bodu, kterým kamera vystoupí ze zóny zmrazení rotace desky. Více viz kapitola „Geometrická reprezentace oblaku“.

Seřazování oblaků od nejvzdálenějšího k nejbližšímu na základě vzdálenosti kamery od středu jejich základny není dostatečně robustní řešení. Dobré výsledky dává, pokud je oblačnost řídká a oblaka mají přibližně stejnou velikost. Pokud však ve scéně existují u sebe blízko rozmístěná oblaka výrazně rozdílné velikosti, může dojít k nekorektnímu řazení a „prosvítání“ menšího oblaku skrz oblak větší. Částečným řešením je změna výpočtu vzdálenosti kamery od oblaku tak, že se nepočítá se středem základny, ale s ke kameře nejbližším bodem, který je ještě součástí oblaku.

Výpočet geometrie oblaků i stínování je plně v kompetencích CPU. S výhodou by pro tyto úkoly šlo využít vertex a fragment shaderů, které umožňují podstatně rychlejší maticové a vektorové operace.

Horní hladinová oblačnost není vykreslována metodami třídy CloudManager, je do scény demonstrační aplikace přidána jako samostatný objekt. To s sebou nese problém nekorektního výpočtu průhlednosti. Pro přidání hladinové oblačnosti do třídy CloudManager bude nutná kompletní revize algoritmu řazení oblaků a jejich elementů tak, aby krom přímé vzdálenosti k oblakům zohledňoval také vertikální vzdálenost vrstev oblačnosti od kamery.

U pohybujících se objektů je při současném řešení patrné protnutí elementárních desek oblaků. Typickou situací může být let ve formaci v oblacích. Jedno z částečných řešení je popsáno v [Har01], kde desky nejsou renderovány přímo do scény, ale do dvou impostorů, kdy jeden je vykreslen mezi kamerou a letounem a druhý je vykreslen až za letounem. Modelem letounu tedy neprochází žádné desky a k viditelným protnutím nedochází. Také tato metoda není pro všechny případy zcela ideální a bude předmětem dalšího zkoumání, jak problém uspokojivě vyřešit. Průlet současnou oblačností navíc nevyvolává dostatečný efekt snížení viditelnosti blízkých objektů – při letech ve formaci jsou letouny i ve vzdálenosti několika stovek metrů téměř plně viditelné, viz obrázek.



*Obr. 5.11: Formační let uvnitř oblaku, letouny jsou ve vzdálenosti 30, 80, 150 a 300 metrů.*

Další oblastí, kterou se bude třeba zabývat je korektní zobrazování průhledných objektů ve scéně s oblačností. Nyní tento problém není ošetřen. Pokud je ve scéně objekt s průhledností a je vykreslen před vykreslením oblačnosti, oblačnost za ním se nezobrazí. Pokud je vykreslován až po oblačnosti, popsáný problém zaniká, naopak se objevuje nový – objekt lze vždy vidět i přes oblak.



*Obr. 5.12: Ukázka problému průhlednosti. Na obrázku vlevo byla řídicí věž vykreslena před vykreslením oblačnosti, na obrázku vpravo byla vykreslena až po vykreslení oblačnosti.*

Komplikace způsobené dalšími průhlednými objekty ve scéně lze řešit rozdělením oblaků do několika částí, kde jejich „děličem“ bude vždy objekt s průhledností. Tyto části jsou společně s průhlednými objekty scény seřazeny od nejvzdálenější k nejbližší a v tomto pořadí vykresleny.

Poměrně jednoduše budou upraveny reakce oblaků na vítr. Nyní se předpokládá statický vítr po čas celé simulace, při jeho náhlé změně dojde ke skokové změně pozice oblaků. Tento fakt vyplývá z použitého vzorce pro výpočet pozice oblaku, viz kapitola „Simulace“. Po jednoduché úpravě bude možné vektor větru během simulace libovolně měnit.

Dalším z mnoha vylepšení by měl být záznam vývoje oblačnosti. Fáze generování využívá náhodných čísel a pro zpětné přehrání, návrat do daného času simulace a analýzu výsledků provedené simulace je třeba tuto náhodnost zaznamenat.

Konečně by do systému měla být časem dodána interakce se simulátorem. Ta musí být připravena v kontextu konkrétního použití hotového simulátoru – např. pokud se bude předpokládat nasazení v tankovém simulátoru, nebude třeba simulovat vznik námrazy.

## 6 Závěr

V tomto dokumentu jsem se pokusil čtenáře seznámit s částí meteorologie týkající se řešeného problému, dále jak oblačnost a jevy s ní spojené dokáží ovlivnit činnost leteckých a vojenských prostředků, nastínil jsem několik přístupů k řešení modulu pro generování počasí a popsal několik přístupů k samotné vizualizaci oblaků. V další kapitole popisuji konkrétné řešení. Vizualizace oblačnosti a s oblačností související efekty, jako je třeba déšť, jsou rozsáhlým tématem. Projekt bude dle zájmu průmyslového zadavatele dále optimalizován a rozvíjen nad rámec této diplomové práce. Proto se v kapitole návrhu i v kapitole implementace z velké části věnuji technikám a vlastnostem, které sice nejsou součástí pro diplomovou práci vyvinutého software, ale budou vytvořeny dodatečně.

# Literatura

- [Ast87] Astapenko, P. D., Kopáček, J.: *Jaké bude počasí*. Praha, Lidové nakladatelství 1987.
- [Skr06] Skřehot, P., aj.: *Atlas oblaků*. Dostupné na URL: <http://mraky.astronomie.cz> (2006).
- [Mik89] Mikulčák, J. aj.: *Matematické, fyzikální a chemické tabulky*. Praha, Státní pedagogické nakladatelství 1989.
- [Air06] Airbus: *Optimum Use Of Weather Radar*. Dostupné na URL: <http://www.wingfiles.com/files/systems/wxradaruseairbus.pdf> (2006).
- [Vas06] Vašíček, J.: *Vysvětlení některých meteorologických pojmů a jevů*. Dostupné na URL: [http://www.chmi.cz/meteo/olm/Let\\_met/Pojmy.htm](http://www.chmi.cz/meteo/olm/Let_met/Pojmy.htm) (2006).
- [Atl06] Mühr, B.: *Der Karlsruher Wolkenatlas*. Dostupné na URL: <http://www.wolkenatlas.de>
- [Wan03] Wang, N.: *Realistic and Fast Cloud Rendering*. Microsoft Corporation 2003. Dostupné na URL: <http://ofb.net/~niniane/clouds-jgt.pdf> (2006).
- [Schp03] Schpok, J., aj.: *A Real-Time Cloud Modeling, Rendering, and Animation system*. 2003. Dostupné na URL: <http://cobweb.ecn.purdue.edu/purpl/papers/scaclouds.pdf> (2006).
- [Kni02] Kniss, J., aj.: *Interactive Translucent Volume Rendering and Procedural Modeling*. 2002. Dostupné na URL: [http://www.cs.utah.edu/~jmk/papers/kniss\\_volumeshading02.pdf](http://www.cs.utah.edu/~jmk/papers/kniss_volumeshading02.pdf) (2006).
- [Kni03] Kniss, J., aj.: *A Model for Volume Lighting and Modeling*. 2003. Dostupné na URL: [http://www.cs.utah.edu/~jmk/papers/kniss\\_tvlg03\\_volshade.pdf](http://www.cs.utah.edu/~jmk/papers/kniss_tvlg03_volshade.pdf) (2006).
- [Har01] Harris, M. J., Lastra, A.: *Real-Time Cloud Rendering*. University of North Carolina 2001. Dostupné na URL: [http://www.markmark.net/PDFs/RTClouds\\_HarrisEG2001.pdf](http://www.markmark.net/PDFs/RTClouds_HarrisEG2001.pdf) (2006).
- [Har02] Harris, M. J.: *Real-Time Cloud Rendering for Games*. Proceedings of Game Developers Conference 2002. Dostupné na URL: [http://www.markmark.net/PDFs/RTCloudsForGames\\_HarrisGDC2002.pdf](http://www.markmark.net/PDFs/RTCloudsForGames_HarrisGDC2002.pdf) (2006).
- [Die00] Dietrich S.: *Elevation Maps*. nVidia Corporation 2000. Dostupné na URL: <http://developer.nvidia.com/attach/6597>
- [Eli00] Elinas, P., Stuerzlinger, W.: *Real-time rendering of 3D clouds*, University of British Columbia, York University. 2000. Dostupné na URL: <http://www.cse.yorku.ca/~wolfgang/papers/clouds.pdf>
- [Sta] Stam, J.: *A General Animation Framework for Gaseous Phenomena*. University of Toronto. Dostupné na URL: <http://www.dgp.toronto.edu/people/stam/suomi/stam/gas.html>
- [Pon03] Poneš, M.: *Modelování a renderování mraků*, ČVUT Praha 2003, Dostupné na URL: [http://netra.felk.cvut.cz/Zope/cgg/publications/diplom/PonesMichal/download\\_folder/](http://netra.felk.cvut.cz/Zope/cgg/publications/diplom/PonesMichal/download_folder/)

# Seznam Příloh

Příloha A. Soubory projektu na datovém nosiči

Příloha B. Ovládání demonstračního programu

Příloha C. Ovládání skriptu pro 3dsmax a gmax

Příloha D. Přehled vytvořených modelů

Příloha E. Použití třídy CloudManager

Příloha F. Nastavení vizualizace formou maker

Příloha G. Ukázky grafického výstupu

## Příloha A. Soubory projektu na datovém nosiči

Následuje seznam nejdůležitějších adresářů a souborů na přiloženém datovém nosiči a popis zdrojových souborů projektu.

cloudsDemo.exe	Zkompilovaná demonstrační aplikace.
*.*	Soubory projektu pro MS Visual Studio 2003.
data/*.clm	Soubory s definicemi modelů několika typů oblaků.
cu*.clm	Oblaka typu cumulus.
st*.clm	Oblaka typu stratus.
cb*.clm	Oblaka typu cumulonimbus.
c*.clm	Další oblaka speciálních tvarů.
doc/	Adresář pro textové dokumenty.
max/	Soubory pro použití v programech 3dsmax a gmax.
cloudExport.ms	Exportní/importní skript pro 3dsmax a gmax. Standardní umístění pro tento soubor je do adresáře Scripts v instalačním adresáři 3dsmax nebo gmax.
*.max	Soubory s ukázkami modelů oblaků, kompatibilní s 3dsmax 8.
src/	Zdrojové kódy projektu.
cloudsDemo.cpp	Demonstrační aplikace.
Clouds.cpp/h	Celý systém volumetrických oblaků.
vector3f.cpp/h	Obsahuje třídu pro reprezentaci a operace nad třísložkovým vektorem.
tga.h	Pomocná třída pro načtení a uložení rastrového formátu TGA. Autor: Kevin Harris, 2005
l3ds.cpp/h	Načítání formátu 3DS. Autor: Lev Povalahev, 2001, <a href="http://www.levp.de/3d/">http://www.levp.de/3d/</a>
textures/	Textury oblačnosti a pomocné textury demonstrační aplikace.
cldvolum.tga	Hlavní a jediná textura volumetrických oblaků.
high*.tga	Několik textur pro vysokou oblačnost.
orto*.tga	Textury terénu.
*.tga	Další pomocné textury pro demonstrační aplikaci.
ukazky/	Ukázkové obrázky a videa



## Příloha B. Ovládání demonstračního programu

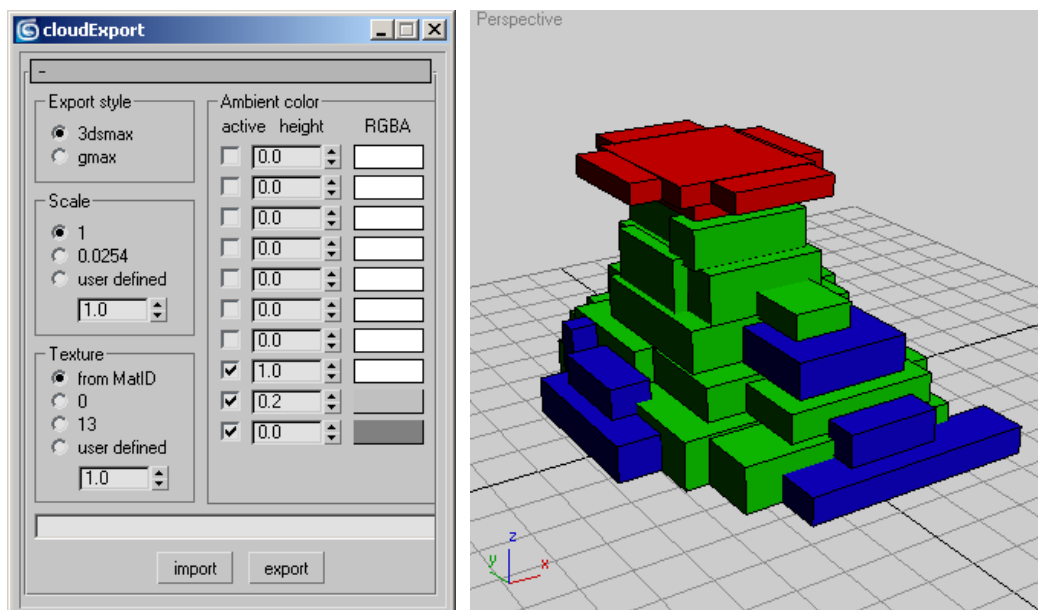
Demonstrační aplikace ukazuje funkčnost systému zobrazování volumetrické oblačnosti. Ovládání umožňuje především změnu pohledu, změnu osvětlení, přidávání, mazání oblaků a změnu jejich vývojového stupně.

tlačítka myši + pohyb myši	rozhlížení, pohyb a rotace kamery
šipky	pohyb kamery
Z + pohyb myši	změna zorného úhlu kamery (zoom)
S + pohyb myši	azimut a elevace Slunce
M + pohyb myši	azimut a elevace Měsíce
L + pohyb myši	regulace vývojového stupně všech oblaků ve scéně
F + pohyb myši	průlet scénou
C	postupné odstraňování oblaků
Y	přidání jednoho oblaku blízko středu scény (brzy zaniká)
X	přidání jednoho oblaku blízko středu scény (nezaniká brzy)
I	přepínání úrovně detailů – zobrazení impostorů od 10 nebo 50 km
B	zapnutí/vypnutí blendingu pro 3D oblaka
T	zobrazení statistik
R	přepínání barev statistik
O	blízko středu scény zobrazí dva kvádry o výšce 500 metrů
J	skryje terén a oblohu
H	skryje horní oblačnost
G	skryje volumetrickou oblačnost
K	zobrazí kokpit
P	zobrazí letadla v různých vzdálenostech před kamerou
E	zobrazí řídicí věž v nadživotní velikosti (ukázka chyb průhlednosti)
V	přepíná hustotu mlhy
D	přepíná povětrnostní situaci
ESCAPE	ukončení programu

## Příloha C. Ovládání skriptu pro 3dsmax a gmax

Exportní-importní skript pro program 3dsmax a gmax komunikuje s uživatelem jedním hlavním oknem. Následuje popis jednotlivých voleb.

Export style	Ovlivňuje způsob, jakým bude export proveden.
3dsmax	Při kliknutí na Export bude uživatel vyzván k zadání cesty a jména pro exportovaný soubor.
gmax	Po kliknutí na Export se zobrazí okno s textovým popisem exportovaného oblaku. Toto je dáno omezením gmax, díky kterému není možné pomocí MaxScript vytvářet soubory. Textový popis je nutné přes schránku zkopírovat do manuálně vytvořeného souboru.
Scale	Měřítko importu a exportu.
1	Implicitně zvolená možnost, platí v případě, že jsou jednotky v 3dsmax/gmax metrické.
0.0254	Další často používaná hodnota, možné použití např. pokud jsou jednotky v 3dsmax/gmax nastaveny na základní jednotky (Generic Units).
user defined	Možnost nastavit libovolné měřítko ve zvláštních případech.
Texture	Volba čísla textury, které se uloží společně s jednotlivými částmi oblaku.
from MatID	Číslo textury jednotlivých částí je rovno jejich Material ID.
0	Textura bude pro celý oblak jednotná.
13	Textura bude pro celý oblak jednotná, textura číslo 13 je vhodná pro kupovitou i slohovitou oblačnost.
user defined	Možnost nastavit vlastní jednotné číslo textury pro celý oblak.
Ambient Color	Nastavení gradientu pomocné rozptýlené barvy.
active	Zaškrtnutí určuje, že tato barva bude uložena do modelu. Aktivní barvy musí být všechny naspoju seznamu.
height	Relativní výška v oblaku, pro kterou bude nastavená barva platit. Hodnota 0.0 označuje základnu, hodnota 1.0 nejvyšší bod oblaku. Je možné a vhodné používat také hodnoty mírně vně interval $\langle 0, 1 \rangle$ , protože texturované desky přesahují vertikální rozsah oblaku.
RGBA	Volba barvy pro danou výšku oblaku a to včetně alfa složky.
Import	Vyzve k otevření souboru s modelem oblaku, načte a zobrazí oblak a nastaví barevný gradient v exportním skriptu.
Export	Exportuje oblak a barevný gradient v souladu s možností nastavenou v Export Style.



*Exportní-importní skript a model bouřkového oblaku s různě nastavenými Material ID.*

### ***Vytvoření, export a import oblaku***

Před samotnou tvorbou oblaku je vhodné nastavit metrické jednotky v 3dsmax/gmax (menu > Customize > Units Setup). Vytvoření modelu má tyto kroky:

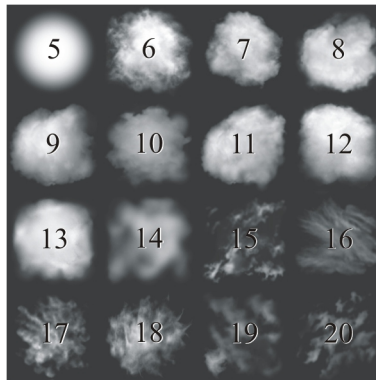
1. Model oblaku je složen z několika kvádrů, typicky z jednoho až několika desítek. Modelujeme tak, aby byl půdorys systému kvádrů přibližně ve středu souřadného systému a aby základna systému měla přibližně nulovou vertikální souřadnici. Jakékoliv rotace jiné, než o 90°, vedou k nepředpokládaným výsledkům.
2. Kvádrům jsou přiřazena Material ID. Ta určí část textury, která se použije pro texturované desky, později umístěné do daného kvádrů. Klíč k výběru textury následuje za tímto návodem.
3. Všechny kvádry mohou být pro větší přehlednost sloučeny do jediného objektu (např. Editable Mesh).

Postup exportu má tyto kroky:

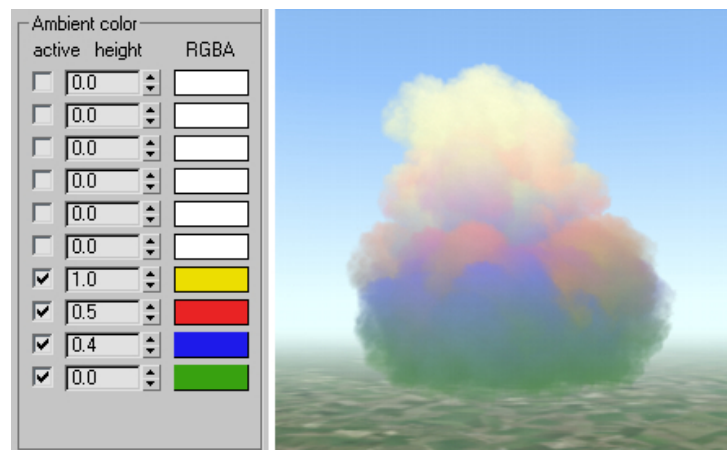
1. Označíme objekt(y) pro export.
2. Zkontrolujeme správné nastavení Export Style, Scale a Texture.
3. Nastavíme barevný přechod vhodný pro tento typ oblaku, tím lze například výrazně ztmavit základnu bouřkového mraku nebo nastavit jinou, než typicky šedou barvu.
4. Tlačítkem Export uložíme model oblaku do souboru.

Import má pouze dva kroky:

1. Zvolíme měřítko, hodnota 1.0 je správná, pokud jsou jednotky v 3dsmax/gmax metrické.
2. Po kliknutí na tlačítko Import vybereme soubor s popisem oblaků.



*Klíč k výběru části textury, první polovina je vhodná především pro kupovitou oblačnost, zbývající textury se hodí spíše k rozpadajícím se částem oblaku. Výběr záleží především na „citu“ designéra.*

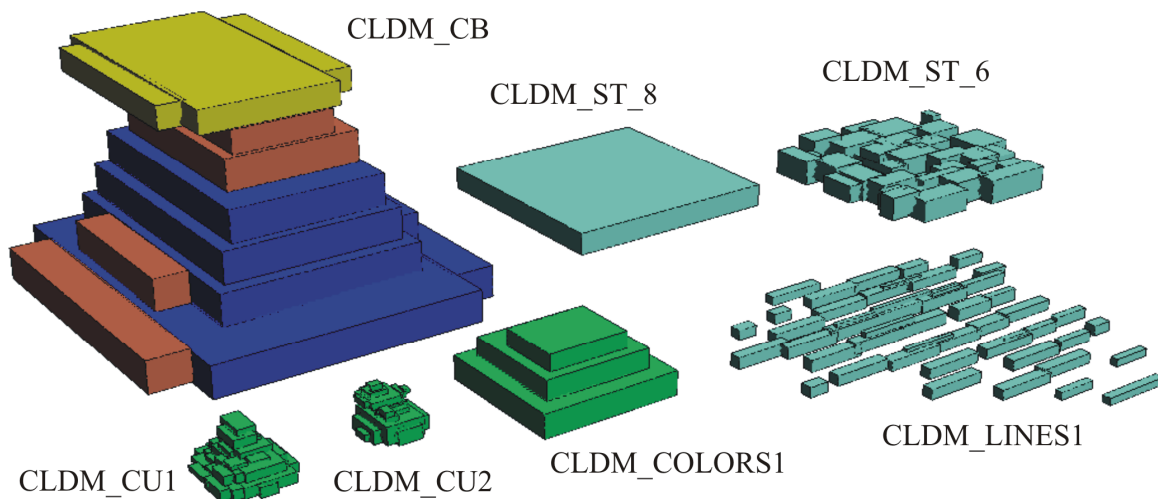


*Gradient pomocné rozptýlené složky v 3dsmax a v demonstrační aplikaci.*

## Příloha D. Přehled vytvořených modelů

Při vkládání oblaků do scény je výběr modelu proveden předdefinovaným makrem. Zde je seznam dostupných maker, odpovídajících souborů a popis.

Makro	Soubor modelu	Rozměr	Popis
CLDM_CU1	data/cu1.clm	1.6 x 2.2 x 0.8 km	cumulus
CLDM_CU2	data/cu2.clm	1.2 x 1.4 x 0.8 km	cumulus
CLDM_ST_6	data/st6_5x5x05.clm	5.0 x 5.0 x 0.6 km	protrhaný stratus
CLDM_ST_8	data/st8_5x5x05.clm	5.0 x 5.0 x 0.5 km	souvislý stratus
CLDM_CB	data/cb1.clm	7.7 x 8.8 x 4.6 km	cumulonimbus
CLDM_LINES1	data/c_lines1.clm	7.6 x 7.3 x 0.4 km	pokusné řady oblaků
CLDM_COLORS1	data/c_colors1.clm	3.0 x 3.0 x 1.3 km	pokusný barevný cumulus



*Modely oblaků, přiřazení čísla textury k barvě je:*

*červená – 6, zelená – 7, tmavě modrá – 12, světle modrá – 13, žlutá – 16.*

## Příloha E. Použití třídy CloudManager

V této příloze jsou uvedeny komentované fragmenty zdrojového kódu demonstrační aplikace, ve kterých jsou volány metody třídy CloudManager (dále jen CM). Předpokládáme, že máme zdrojový kód jednoduché OpenGL aplikace s typickými funkcemi:

- `init` – volána jednou při inicializaci aplikace.
- `shutDown` – volána jednou při ukončování aplikace.
- `windowProc` – funkce pro zpracování zpráv, tedy i vstupů od uživatele.
- `render` – volána aplikací v nekonečné smyčce, funkce provádí vykreslení scény.

### *Vložení hlavičkového souboru a globální proměnné*

Připojíme hlavičkový soubor a deklarujeme proměnné.

```
#include "Clouds.h"
CloudManager* CM;
// CM bude objekt, přes který můžeme přistupovat ke většině voleb
// systému oblačnosti
GLuint cldtex;
// jméno textury, která bude použita pro volumetrickou oblačnost
Vector3f sunVect, moonVect;
// normalizovaný vektor ke Slunci a Měsíci
Vector3f camPos;
// souřadnice pozice kamery
```

### *Inicializační funkce*

V této funkci musíme krom jiného nejprve nastavit vektor ke Slunci, ke Měsíci a polohu kamery (tedy proměnné `sunVect`, `moonVect` a `camPos`). Musíme také vygenerovat jméno pro texturu volumetrických oblaků (proměnná `cldtex`) a pod tímto jménem načíst rastrová data textury. Poté již můžeme inicializovat CM.

```
CM = new CloudManager();
// vytvoření objektu
CM->assignTexId3d((int)cldtex);
// uloží číslo textury do CM
CM->setSunVect(sunVect);
// uloží směr ke Slunci do CM
CM->setMoonVect(moonVect);
// uloží směr k Měsíci do CM
CM->setWind(45,10);
// nastavení větru, směr ze 45°, rychlost 10 m/s
CM->setDetails(40000, 50000, 90000, 100000);
// nastavení úrovně detailů, v tomto případě se bude režim zobrazení
// přepínat takto:
// 3D model -> impostor ve vzdálenosti 50 km od kamery
// impostor -> 3D model ve vzdálenosti 40 km
// impostor -> skrytí oblaku ve vzdálenosti 100 km
// skrytý oblak -> impostor ve vzdálenosti 90 km
```

V této fázi máme připraveny základní parametry CM a můžeme začít s vkládáním oblaků do scény. Ukáží několik možností, jak oblaka vložit s různě detailně zadanými parametry.

Nejjednodušeji lze oblak přidat jediným příkazem s 9 parametry. Takovýto mrak se ve scéně objeví v plném vývojovém stupni ihned při prvním následujícím vykreslování a délka jeho života je 100.000 sekund. Důležitá je návratová hodnota `cldIndex`, která nám oznamuje index, který byl právě přidánému oblaku přiřazen.

```
cldIndex=CM->addCloud(CLDM_CU2, 2000, 1500, 3000, 1.5, 1.0, 1.5,
  CLD_LOAD_RANDOM, 1200);
// vloží oblak, jehož model je definován makrem CLDM_CU3 (cumulus)
// střed základny bude na XZ souřadnici [2000,3000], ve výšce 1500 m
// oblak bude mít horizontálních souřadnicích velikost 150%, výška
// bude odpovídat výšce původního modelu
// parametr CLD_LOAD_RANDOM zajistí náhodnou diskretní rotaci kolem
// svislé osy s krokem 90° a oblak náhodně převrátí v ose X a Z
// střední velikost jednotlivých texturovaných desek bude 1200 metrů
```

Metoda je přetěžovaná, její další varianta má přidány dva parametry, které určí množství texturovaných desek, které se budou do oblaku vkládat.

```
cldIndex=CM->addCloud(CLDM_CU3, 2000, 1500, 3000, 1.5, 1.0, 1.5,
  CLD_LOAD_RANDOM, 1200, 40, 400);
// předposlední parametr udává počet desek, které se vloží do
// jednoho krychlového kilometru objemu oblaku
// poslední parametr řídí následnou redukci desek, pokud je vzájemná
// vzdálenost dvou desek menší, než 400 metrů, je jedna deska
// odstraněna
```

Jednoduše definovaný oblak lze dále blíže specifikovat.

```
CM->setCloudLife(cldIndex, 300, 900, 2400, 3600, 0.7);
// nastavuje životní cyklus oblaku, v tomto případě oblak vzniká začíná
// vznikat ve 300. sekundě simulace, v 900. sekundě dosáhne maximálního
// vývojového stupně 0.7, od 2400. sekundy začíná zanikat, ve 3600.
// sekundě zaniká úplně a je odstraněn ze simulace
CM->cloudSetColor(cldIndex, 0.8, 0.8, 0.8, 1.0);
// nastavuje doplňkovou barvu oblaku, standardně mají všechny složky
// RGBA hodnotu 1.0, v tomto případě bude intenzita ve složkách RGB
// snížena na 80%, což může simulovat zastínění oblaku vyšší oblačnou
// vrstvou
CM->cloudSetColor(cldIndex, (rgbaf)color);
// nastavuje doplňkovou barvu oblaku, parametrem je proměnná typu rgbaf
CM->cloudSetOpacity(cldIndex, 0.3);
// nastavení průhlednosti na 30%, oblak pak bude mít méně výraznou
// strukturu
CM->cloudSetYStretch(cldIndex, 1.8);
// způsobí roztažení oblaku ve svislé ose na 180%
CM->cloudDisable3dShading(cldIndex);
// vypne trojrozměrné stínování, stínování nebude závislé na azimutu
// zdroje světla
CM->cloudEnable3dShading(cldIndex);
// naopak zapne trojrozměrné stínování (je implicitně aktivní)
CM->cloudSetLifeMode(cldIndex, CLD_LIFE_GROW);
// volba režimu vzniku a zániku oblaku, v tomto případě bude růst od
// středu ven, při použití parametru CLD_LIFE_APPEAR se objevuje
// zvyšováním neprůhlednosti
```

Metoda vkládání oblaků je přetížená ještě o jednu variantu, která některé právě zmíněné specifikující metody nahrazuje svými parametry.

```

cldIndex=CM->addCloud(CLDM_CU3, 2000, 1500, 3000, 1.5, 1.0, 1.5,
CLD_LOAD_RANDOM, 1200,
0.8, 0.8, 0.8, 1.0,
0.3, 1.8, false, CLD_LIFE_GROW);
// prvních 9 parametrů má stejný význam, další parametry jsou postupně
// R, G, B, A složka doplňkové barvy, průhlednost,
// následující 4 parametry nastavují doplňkovou barvu, roztažení ve
// vertikální ose, boolovská hodnota povolení trojrozměrného stínování,
// režim vzniku

```

Po rozmístění oblaků do scény konečně vynutíme první přepočítání stínování a geometrie.

```

CM->recalc(camPos, true);
// poslední parametr true zajistí, že přepočítání skutečně proběhne
// a nebude se předem zjišťovat, zda je potřebný či nikoliv

```

### ***Ukončující funkce***

Zde se volá pouze destruktorka CM. Destruktor uvolňuje svou alokovanou paměť, neuvolňuje však texturu volumetrických oblaků (cldtex), to je třeba provést manuálně zavoláním OpenGL funkce `glDeleteTextures`.

```
delete (CM);
```

### ***Funkce zpracování zpráv***

V této můžeme přistupovat k parametrům CM podobně, jako při inicializaci, můžeme také přidávat oblaka, či je mazat. Po provedení změn je vhodné vynutit přepočítání stínování a geometrie.

```

CM->removeCloud(12);
// odstraní oblak s číslem 12 ze scény
CM->recalc(camPos, true);
// vynutí přepočítání oblaků

```

### ***Funkce vykreslení scény***

Před jakýmkoliv nastavením parametrů OpenGL, které je zvykem vídat na začátku těchto funkcí musíme nejprve připravit scénu volumetrických oblaků. Metoda `prepare` rozhodne, zda je třeba přepočítat geometrii jednotlivých oblaků a také případně provede renderování impostorů. Z tohoto důvodu musí být na začátku funkce, protože při renderování impostorů mění nastavení projekce a bufferů. Druhý parametr metody určí, zda bude zapnutý alpha blending či nikoliv.

```
CM->prepare(camPos, true);
```

Nyní připravíme stavový stroj OpenGL pro vykreslování scény, nastavíme projekční matici a umístíme kameru na souřadnice `camPos`. Vykreslíme 3D objekty scény. Až nyní vykreslíme volumetrickou oblačnost, znovu s druhým parametrem, určujícím zapnutý či vypnutý alpha blending.

```
CM->show(camPos, true);
```

Na samém konci vykreslovací funkce přehodíme buffery (double buffering) a naše scéna s 3D objekty a oblačností by se nám měla ukázat na zobrazovacím zařízení.



## Příloha F. Nastavení vizualizace formou maker

Veškerá makra pro nízkoúrovňové nastavení kvality vizualizace jsou definována v souboru Clouds.h.

CLD\_ANGLE\_3DUPDATE 1

Úhlová chyba ve stupních, po jejímž překročení dojde k novému generování oblaku.

CLD\_DISTANCE\_UPDATE 0.98f

Relativní chyba přiblížení, po jejímž překročení dojde k novému generování oblaku. Pokud byl oblak vygenerován pro vzdálenost 1.000 metrů, bude nové generování vyvoláno po přiblížení na 980 metrů nebo po oddálení na 1.020 metrů (1.000 / 0.98).

CLD\_LIFESTATE\_UPDATE 0.005f

Rozdíl životního stupně, po jehož překonání dojde k novému generování oblaku.

CLD\_DEFAULT\_IMPOSTORFAR 30000

Vzdálenost přepnutí z trojrozměrného oblaku na impostor.

CLD\_DEFAULT\_IMPOSTORNEAR 25000

Vzdálenost přepnutí z impostoru na trojrozměrný oblak.

CLD\_DEFAULT\_MAXDISTANCEFAR 90000

Vzdálenost maximální viditelnosti oblaků, tedy vzdálenost, ve které dojde ke skrytí impostoru.

CLD\_DEFAULT\_MAXDISTANCENEAR 85000

Vzdálenost, ve které je skrytý oblak zobrazen jako impostor.

CLD\_NEARTRANSPARENCY\_START 100

CLD\_NEARTRANSPARENCY\_END 350

Mezi těmito vzdálenostmi kamery od roviny texturované desky dochází k postupnému zprůhledňování desky, při vzdálenosti <100 metrů je deska plně průhledná, při vzdálenosti >350 metrů je maximálně viditelná.

CLD\_SHAD\_FALLOFFRANGE 4000

Rychlost exponenciálního útlumu intenzity slunečního svitu při průchodu oblakem. Hodnota je vzdálenost, na které klesne intenzita na 1/e, tedy na přibližně 36.8%.

CLD\_SHAD\_SUNMUL 0.9f

Intenzita slunečního světla v konečném skládání světelných složek ve funkci stínování

CLD\_SHAD\_MOONMUL 0.2f

Intenzita měsíčního světla

CLD\_SHAD\_AMBIENTMUL 0.7f

Intenzita rozptýleného světla

CLD\_SHAD\_SUNLINEAR 0.4f

Podíl lineárního osvětlení ve výpočtu nasvětlení od Slunce a Měsíce, odpovídá parametru mix ve vzorcích v kapitole „Stínování“.

CLD\_IMP\_MAXSIZE 128

Maximální rozlišení impostorů.

CLD\_FREEZE\_RADIUS 0.5f

Relativní vzdálenost vzhledem k velikosti texturované desky, ve které dojde ke zmrazení rotace desky.

CLD\_REMOVE\_AFTER\_DEATH true

Volba, zda odstranit oblak z paměti po jeho smrti.

CLD\_SPRITES\_ROTATION 0.3f

Rozsah náhodné rotace desek kolem své normály v radiánech. V tomto případě je to tedy 0.3 radiánu na jednu i na druhou stranu (+- 17°).

Jsou definována dvě pomocná makra pro optimální výpočet některých parametrů při nahrávání oblaků, parametrem je velikost oblaku.

CLD\_calcDensity(rrr)

Vypočítá vhodnou hustotu desek pro vyplnění do modelu oblaku.

CLD\_calcReduce(rrr)

Vypočítá optimální vzdálenost pro metodu redukce desek v oblaku.

Další série maker jsou modifikátory metody „addCloud“, umožňující rotace a zrcadlení oblaku, názvy jsou samovysvětlující. Jde o tato makra:

CLD\_LOAD\_NONE

CLD\_LOAD\_ROTATELEFT

CLD\_LOAD\_ROTATERIGHT

CLD\_LOAD\_ROTATEBACK

CLD\_LOAD\_FLIPX

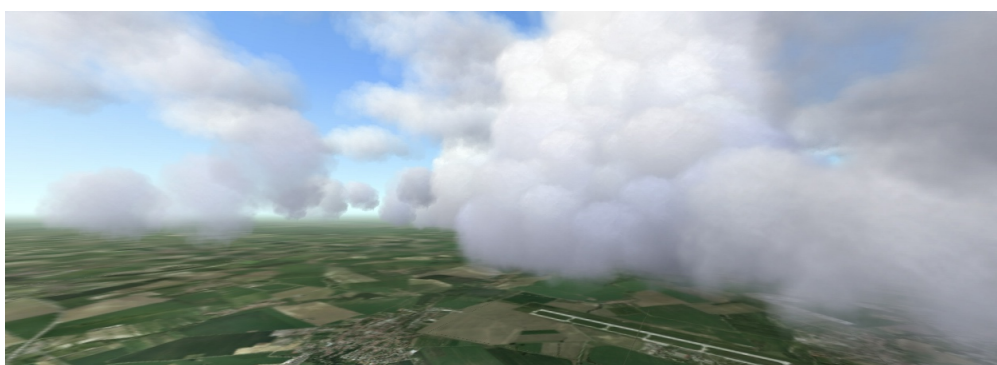
CLD\_LOAD\_FLIPZ

CLD\_LOAD\_RANDOM

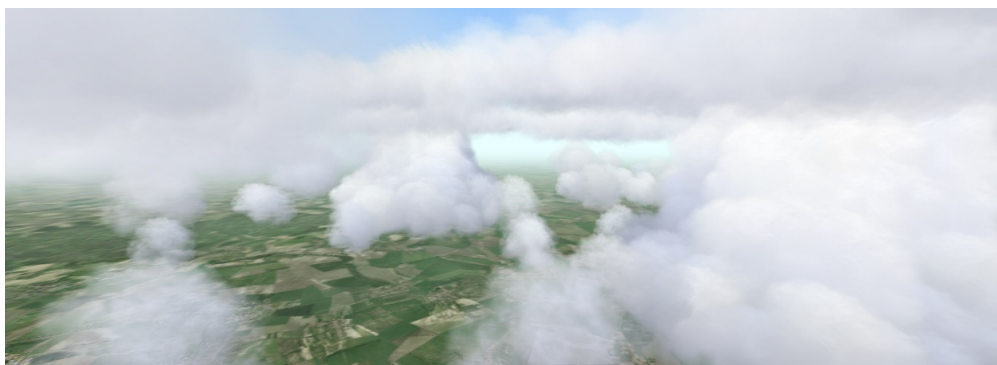
## Příloha G. Ukázky grafického výstupu



*Cumuly ve výšce 1500 metrů, vrstva altocumulů ve výšce 4000 metrů a několik cumulonimbů.*



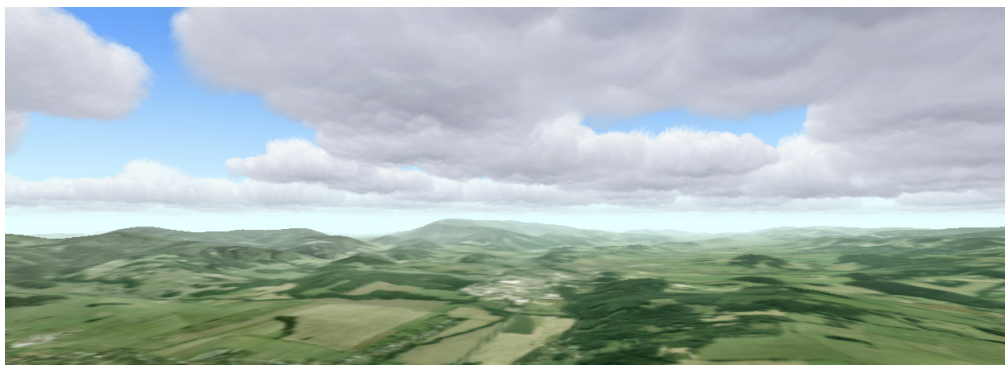
*Stejná povětrnostní situace s detailem cumulonimbu.*



*Stejná povětrnostní situace.*



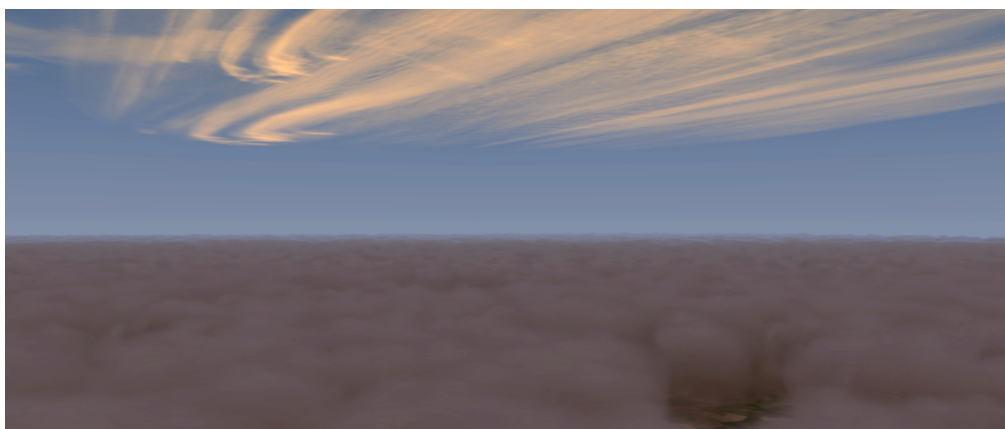
*Rozsáhlá, ale řídká kupovitá oblačnost.*



*Oblačnost s velkým množstvím polygonů.*



*Protrhaný stratus s pokrytím 6/8.*



*Horní oblačnost a stratus s pokrytím 7/8 těsně po západu Slunce.*

