BAKALÁŘSKÁ PRÁCE

Vizualizace osvětlení
povrchu vozovky

Tomáš Smlsal

2001

zadání strana 1

zadání strana 2

OBSAH

1. Úvod 5
2. Způsoby osvětlení - předpisy 6
   2.1 Současný stav předpisů 6
   2.2 Vlastnosti světelného kužele 6
   2.3 Dálková světla 6
   2.4 Potkávací světla 8
   2.5 Barvy světel 10
   2.6 Shrnutí 11
3. Programová knihovna OpenGL 12
   3.1 Systémové nároky 12
   3.2 Použití 12
   3.3 Úvod do OpenGL 12
   3.4 Primitiva a příkazy 12
   3.5 Ovládání grafiky 13
   3.6 Model vykonávání příkazů 13
   3.7 Základní operace OpenGL 13
   3.8 Praktické použití v Delphi 14
   3.9 Display List 15
   3.10 Shrnutí 15
4. Model pro reprezentaci 16
5. Programové vybavení 17
   5.1 Trojúhelníkové sítě definující terén 17
6. Zhodnocení výsledků 18
7. Závěr 19
8. Literatura 20
9. Dodatky 22
   A. Ovládání 22
   B. Instalace 25
   C. Programová dokumentace 26

1. Úvod

Podnětem pro tuto práci byla otázka týkající se bezpečnosti nočního silničního provozu, konkrétně oblasti použití dynamických systémů regulace dosahu světlometů [WL00], vylepšených o horizontální nastavování směru světlometů. Pro ověření parametrů těchto systémů bylo také potřeba vyvinout programové vybavení.

Pro nejlepší přiblížení modelu ke skutečnosti bylo nutné zjistit, jaké vlastnosti světelného kuželu jsou pro model podstatné a které je možné zanedbat. Vlastnosti světelného kuželu bylo možné zjistit z existujících platných předpisů a norem a tak lépe porozumět procesu osvětlování vozovky.

Způsob osvětlení povrchu vozovky je nejvíce závislý na tom, v jaké zemi jej provozujeme. Autorovi jsou v současné době známy tři nejrozšířenější systémy světlometů: evropský symetrický (dnes už jen u motocyklů), americký [Fed99] a evropský asymetrický [EHK01] systém (u nás konkrétně s pravostranným provozem). Americký systém se vyznačuje klopením kužele tlumených světel šikmo dolů na stranu od protijedoucích vozidel. Klopení se dosahuje umístěním vlákna tlumeného světla mimo osu paraboly bez použití clonky. Tlumené světlo tohoto systému lépe osvětluje, ale více oslňuje.

Nadále bude uvažován evropský asymetrický systém.

2. Způsoby osvětlení

Pro zhotovení této práce bylo nutné seznámit se se způsoby osvětlování (modelování) povrchu vozovky reflektory automobilu. Je-li zkoumán způsob osvětlení vozovky, je nutné znát co nejlépe vlastnosti světelného kužele vedoucího ze světlometu automobilu. Tyto vlastnosti jsou dány konstrukcí světlometu, pro kterou platí příslušné předpisy a normy, ze kterých bylo nutné vzít v úvahu pouze některá (nejdůležitější) ustanovení mající pro tuto práci zásadní význam.
Dalšími faktory, které ovlivňují vlastnosti světelného kužele, jsou zejména konkrétní klimatické podmínky. Nadále bude uvažováno ideálně čisté prostředí, neboť modelování takových jevů, jako je například lom a odraz světla na dešťových kapkách, není výpočetně zvládnutelné.

2.1 Současný (duben 2001) stav předpisů pro konstrukci světlometů

Podle naší legislativy se osvětlení vozovky řídí vyhláškou [Min95], navazující na vyhlášku 176/1960 Sb., předpisy EHK OSN (Evropské Hospodářské Komise při Organizaci Spojených Národů) [EHK01] a ČSN 47 0004.
Pro tuto práci jsou podstatná pouze ustanovení týkající se předních světlometů v pravostranném provozu, tedy předpis EHK číslo 1, 2, 8 [ECE08] a 20 [ECE20] (viz [EcoDr], [ECE98]).
Nadále budou tyto předpisy souhrnně označovány pojmem *norma*.

2.2 Vlastnosti světelného kužele

Jaké vlastnosti má světelný kužel světlometu, jaké je rozložení svítivosti ve světelném kuželu?

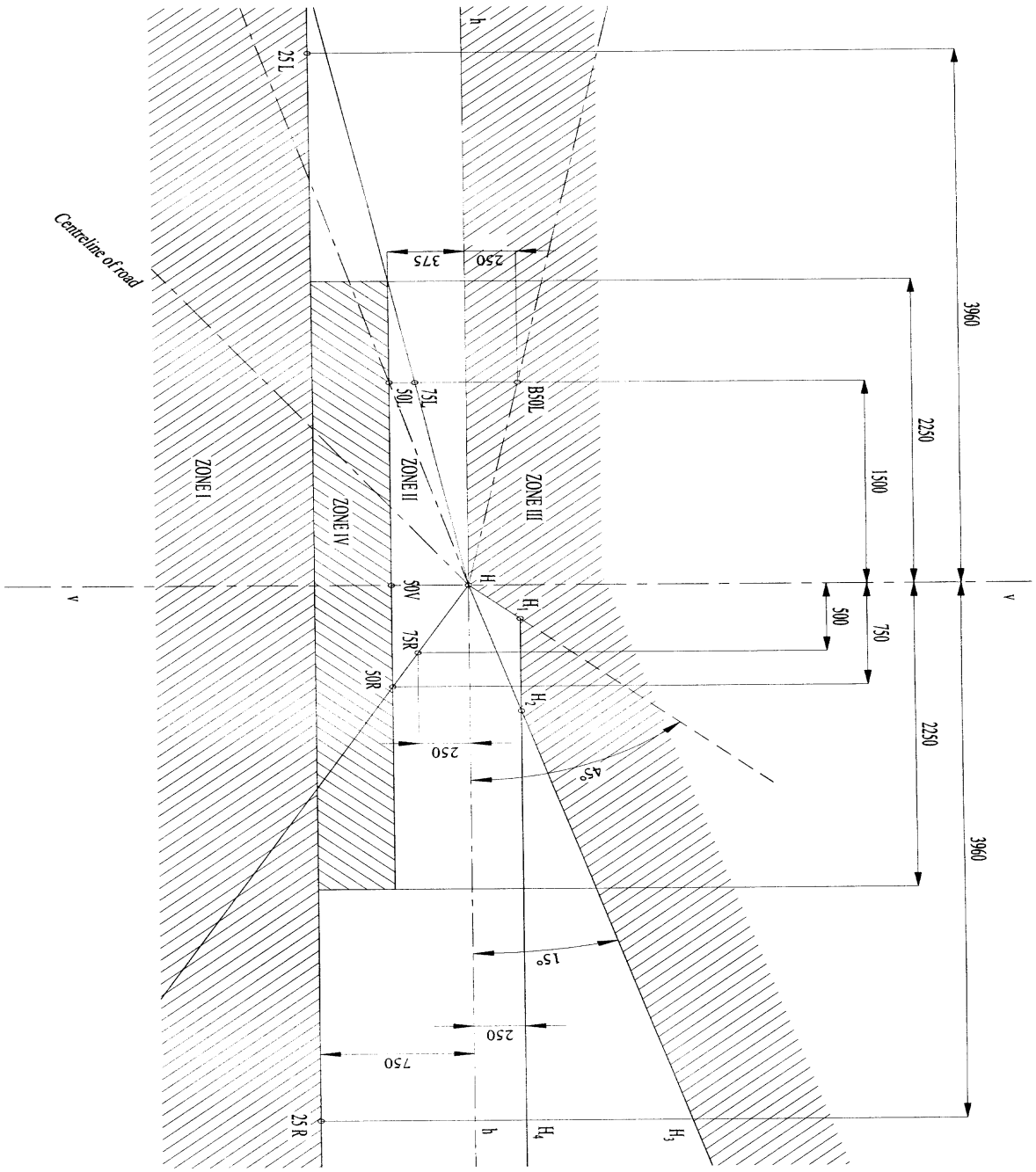
Podle normy jsou rozlišena světla dálková a potkávací, každé se světelným kuželem odpovídajících vlastností. Ty jsou pro kužel definovány v rovině příčného řezu v určité vzdálenosti od počátku kužele. Sledován je především světelný tok, svítivost nebo intenzita osvětlení v závislosti na typu světlometu (žárovkový, halogenový, xenonový, ...). Měření se tedy provádí na zkušební stěně ve vzdálenosti 25 m od světlometu (obr. 1). Zde jsou definovány body, pro které norma předepisuje přípustnou hodnotu osvětlení.

2.3 Dálková světla

Dálkové světlo má mít světelný kužel upraven tak, aby svítil účinně do dálky, a přitom musí dávat dostatek rozptýleného světla do stran, aby byl dostatečně osvětlen okraj silnice a předměty vedle ní (dopravní zbačky apod.). Světlo má být rozloženo rovnoměrně, bez tmavých míst i bez místních přesvětlení, aby nebyl zbytečně unavován zrak řidiče vozidla i řidiče protijedoucího vozidla v určité vzdálenosti.

Pro dálkové světlo je tedy méně omezení než pro potkávací, protože se předpokládá jeho použití v případech, kdy nikoho příliš neosluní a proto není dále potřeba jeho světelný kužel nějak zvlášť speciálně tvarovat.
Průsečík H přímek *hh* a *vv* musí být uvnitř 80-ti % isoluxy maximálního osvětlení *Emax*. Hodnota *Emax* nesmí být menší než 32 luxů pro světlomety typu A nebo 48 luxů pro typ B a nesmí být větší než 240 luxů za žádných okolností. Navíc u světlometu s kombinovaným

Obr. 1


Centreline of road
25 L
h
375
250
3960
2250
1500
B50L
75L
50L
ZONE I
ZONE II
ZONE III
ZONE IV
v
50V
H
H1
500
750
75R
50R
H2
250
45°
15°
250
750
25 R
h
H4
H3

potkávacím a dálkovým světlem nesmí být hodnota větší než šestnáctinásobek naměřeného osvětlení v bodě 75R.
Dále, jsou vymezeny úseky do stran od bodu H (a nahoru i dolů), v nichž musí být osvětlení větší, než je mezní hodnota, kterou se zajišťuje, aby dálkové světlo mělo dostatečnou svítivost nejen v úzkém kuželu, ale i s potřebným rozšířením ve vodorovném směru.
Aby se zabránilo oslňování na příliš velkou vzdálenost, je stanovena horní mez 225 000 cd pro součet maximálních svítivostí všech dálkových světel vozidla. Více v [Kub85].

2.4 Potkávací světla

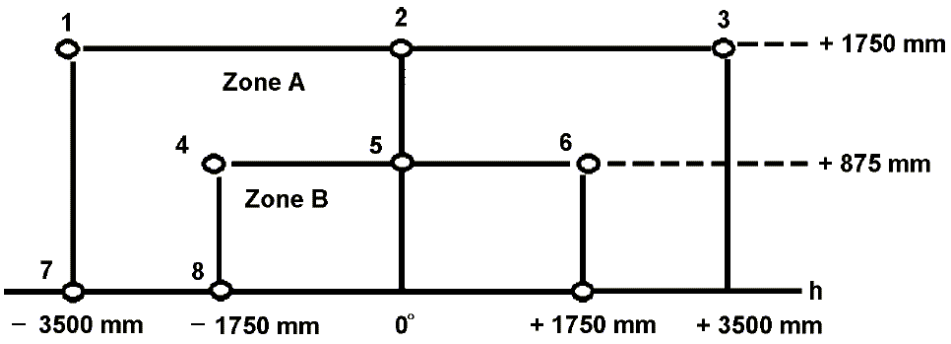
Potkávací, neboli tlumené světlo, je kompromisem mezi požadavkem, aby řidič dobře viděl, a stejným požadavkem protijedoucího řidiče. Pro zmenšení vzájemného oslňování setkávajících se vozidel je světelný kužel tlumeného světla skloněn k vozovce a upraven tak, aby do prostoru, v němž se pohybují oči protijedoucího řidiče, šlo jen omezené množství světla, a to i za cenu zmenšení dosahu světla před vlastním vozidlem. Evropský asymetrický systém se vyznačuje klopením světelného kužele potkávacích světel směrem dolů a výrazným potlačením paprsků v horní polovině kuželu. Zdroj světla je předsunut v ose světlometu před ohnisko parabolického zrcadla a paprsky z něj vycházející ke spodní polovině zrcadla jsou odstíněny clonou. Rozhraní na straně k protijedoucímu vozidlu je vodorovné a na straně odvrácené zdvižené pod úhlem 15° vzhůru, aby se na větší vzdálenost osvětlily překážky na okraji vozovky.

Fotometrické vlastnosti dálkových i tlumených světel jsou stanoveny rozložením osvětlení na zkušební stěně s rozložením kontrolních bodů a polí podle obr. 1. Světlomet předpokládáme umístěný nad rovnou 6 m širokou vozovkou s dvěma pruhy po 3 m, a to tak, že jeho optická osa je rovnoběžná s osou vozovky a je ve výšce 0,75 m nad prostředkem pravého pruhu. Měřicí stěna ve vzdálenosti 25 m je kolmá k optické ose světlometu (je kolmá i k ose vozovky). Dále je na stěně naznačen perspektivní pohled z místa světlometu na vozovku tak, jako by měřicí stěna byla průhledná.
Bod H je úběžný, je zároveň průsečíkem optické osy světlometu s měřicí stěnou a je průsečíkem vodorovné (horizontální) osy hh a svislé (vertikální) osy vv.
Bod B50L je průmět oka protijedoucího řidiče na měřicí stěnu. Předpokládá se, že se pohybuje nad prostředkem levé poloviny vozovky ve výšce 1,25 m a že je právě ve vzdálenosti 50 m před světlometem. Pto tento bod je stanovena velmi nízká mez osvětlení, která nesmí být překročena. Bod 75 R je průmět bodu ležícího na pravé krajnici vozovky ve vzdálenosti 75 m před světlometem. Osvětlení v tomto bodě při asymetrickém tlumeném světle nesmí klesnout pod stanovenou mez. Zajišťuje se tím dostatečné osvětlení v této vzdálenosti při povrchu vozovky a viditelnost bližších překážek. Body 25L (25R) slouží ke kontrole osvětlení do stran u potkávacích světel.

Osvětlení zkušební stěny potkávacím světlem musí vyhovovat těmto požadavkům (příklad pro halogenový světlomet):

Bod na zkušební stěně	Požadované osvětlení [lux]	
Pravostranný provoz	Světlomet typu A	Světlomet typu B
Bod B 50 L	≤ 0.4	≤ 0.4
Bod 75 R	≥ 6	≥ 12
Bod 75 L	≤ 12	≤ 12
Bod 50 L	≤ 15	≤ 15
Bod 50 R	≥ 6	≥ 12
Bod 50 V	-	≥ 6
Bod 25 L	≥ 1.5	≥ 2
Bod 25 R	≥ 1.5	≥ 2
Každý bod v zóně III	≤ 0.7	≤ 0.7
Každý bod v zóně IV	≥ 2	≥ 3
Každý bod v zóně I	≤ 20	≤ 2E */
*/ E je konkrétní naměřená hodnota v bodě 50R		

Obr. 2.



Dále hodnoty osvětlení v zónách A a B (obr. 2) musí vyhovovat těmto rovnicím (opět příklad pro halogenové světlomety):

$$BOD1 + BOD2 + BOD3 \geq 0.3 \text{ lux},$$

$$BOD4 + BOD5 + BOD6 \geq 0.6 \text{ lux},$$

$$0.7 \text{ lux} \geq \text{BOD7} \geq 0.1 \text{ lux},$$

$$0.7 \text{ lux} \geq \text{BOD8} \geq 0.2 \text{ lux}.$$

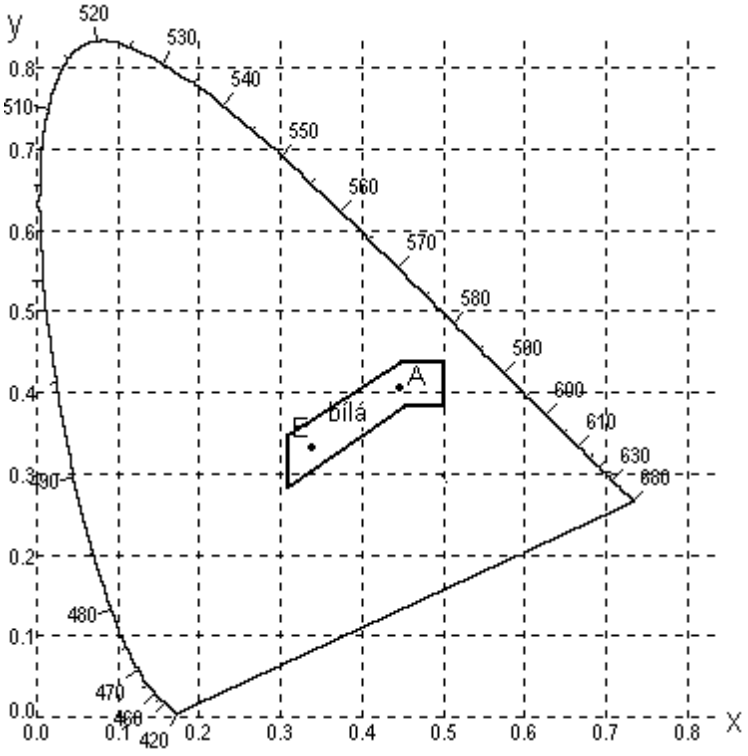
2.5 Barvy světel

Pro světlomety motorového vozidla se používá bílá barva. Rozsah odstínů barvy je mezinárodně normalizován a je určen polem trichromatických souřadnic v kolorimetrickém (viz [WWW3]) trojúhelníku CIE (Commission International de l´Eclairage podle [Ska92]; International Commission on Illumination). Kolorimetrická měrná soustava CIE určuje barevný odstín poměrem tří základních vjemů (měrná červená, zelená a modrá), pro něž volí měřítko z podmínky

$$x + y + z = 1,$$

kde poměrné hodnoty x, y, z jsou souřadnice určité barvy. Nezávisle proměnné jsou jen dvě a stačí pro určení barevného odstínu, takže je lze zobrazit v rovinném diagramu. V kolorimetrickém trojúhelníku na obr. 3 je zakresleno pole bílé barvy předepsané pro světlomet motorového vozidla.

Obr. 3.



Oblá křivka v diagramu je čára čistých spektrálních barev a spojnice jejích konců představuje čáru čistých nespektrálních barev (purpurů), což je směs monochromatických barev z konce spektra v různém poměru. Bod A udává souřadnice normálního bílého světla s teplotou barvy 2854 K, které je předepsáno pro měření barev světel motorových vozidel, bod E je poloha bílého světla s teplotou barvy 5370 K, které se nejvíce přibližuje ideálnímu izoenergetickému bílému světlu. Bod E je těžiště trojúhelníka a znamená nebarevný vjem sytosti 0 (čára spektrálních barev a purpurů představuje barevné vjemy s největší sytostí 1). Více v [Sml01].

Použitelná bílá barva má tedy tyto hranice:

hranice směrem k ...	modré	x	≥	0,310,		
	žluté	x	≤	0,500,		
	zelené	y	≤	0,150	+	0,640x
	.....	y	≤	0,440,		
	purpurům	y	≥	0,050	+	0,750x,
	červené	y	≥	0,382.		

2.6 Shrnutí

Lze konstatovat, že po bližším seznámení se situací v oblasti modelování povrchu vozovky světlomety vozidla bude možné navrhnout odpovídající model.

3. Knihovna OpenGL

V dnešní době je počítačová 3D-grafika dostupná i na osobních počítačích PC. OpenGL (Open Graphics Library), původně vyvinutá firmou Silicon Graphics Incorporated (SGI) pro jejich grafické pracovní stanice, umožňuje aplikacím vytvářet vysoce kvalitní barevné obrázky nezávisle na grafickém uživatelském rozhraní (graphics user interface - GUI), operačním systému a hardware. Nadále bude popisována implementace pro operační systém Windows NT verze 4.0, protože v tomto prostředí bylo také vytvořeno programové vybavení.

3.1 Systémové nároky

Většina programů využívajících funkcí OpenGL by měla mít k dispozici minimálně procesor 486, avšak doporučuje se alespoň Pentium 90 MHz s 32 MB RAM a grafický adaptér s barevnou hloubkou 16/24 bitů.

3.2 Použití

Pokud má být používána knihovna OpenGL, je potřeba zajistit, aby existovaly soubory *opengl32.dll* a *glu32.dll* v adresáři dynamicky linkovaných knihoven operačního systému - například podadresář SYSTEM.

3.3 Úvod do OpenGL

Jako programové rozhraní pro grafický hardware, hlavním účelem OpenGL je vykreslit dvou- a tří-rozměrné objekty do frame-bufferu. Tyto objekty jsou popsány jako sekvence vrcholů (které definují geometrické objekty) nebo pixelů (definujících obrázky). OpenGL postupně provádí několik akcí s těmito daty, aby je převedla na pixely vytvářející výsledný požadovaný obraz ve frame bufferu.

3.4 Primitiva a příkazy

OpenGL vykresluje primitiva (body, úsečky nebo n-úhelníky) v závislosti na některém z vybraných módů. Módy mohou být řízeny nezávisle na sobě. To znamená, že nastavení (aktivace) jednoho módu neovlivňuje, zdali jsou další módy nastaveny (přestože se více módů může vzájemně ovlivňovat v rámci určení výsledného obrazu ve frame bufferu). K určení primitiv, nastavení módů a provedení dalších grafických operací jsou pouze zadávány příkazy v podobě volání funkcí.
Primitiva jsou definována skupinou jednoho nebo více vrcholů. Vrchol definuje bod, koncový bod úsečky nebo vrchol n-úhelníku, kde se stýkají dvě hrany. Data (sestávající se ze souřadnic vrcholu, barev, normál, souřadnic textur apod.) jsou sdružena s vrcholem a každý vrchol s jemu přidruženými daty je zpracován nezávisle, avšak stejným způsobem. Jedinou výjimkou z tohoto pravidla je případ, kdy skupina vrcholů musí být oříznuta, protože část primitiva *padne* mimo určený region. V tomto případě mohou být data vrcholu modifikována a mohou být vytvořeny nové vrcholy. Způsob ořezávání záleží na tom, které primitivum skupina vrcholů reprezentuje.
Příkazy jsou vždy vykonávány v pořadí, v jakém jsou přijímány, přestože se může vyskytnout neurčitá prodleva před tím, než vyvolají nějakou činnost. Znamená to, že každé primitivum je kompletně vykresleno před vykonáním dalšího následujícího příkazu. Potom tedy i příkazy dotazu na stav vrací data, která jsou konzistentní s úplným vykonáním všech předchozích příkazů přijatých knihovnou OpenGL.

3.5 Ovládání grafiky přes OpenGL

OpenGL poskytuje jednoduché přímé řízení základních operací nad dvou- a tří-rozměrnou grafikou. Zahrnuje to určení takových parametrů, jako jsou matice transformace, koeficienty rovnic pro osvětlení, metody antialiasingu a operátory pro obnovu pixelu. Avšak neposkytuje prostředky pro popis nebo modelování složitějších geometrických objektů. Potom tedy příkazy zadané v OpenGL spíše říkají, jak konkrétně bude výsledek vytvořen (jaké procedury by měly probíhat), než jako co přesně bude výsledek vypadat. OpenGL je blíže procedurálnímu jazyku než-li popisnému. Aby mohlo být plně pochopeno používání OpenGL, je dobré znát pořadí, ve kterém jsou prováděny jednotlivé operace.

3.6 Model vykonávání příkazů

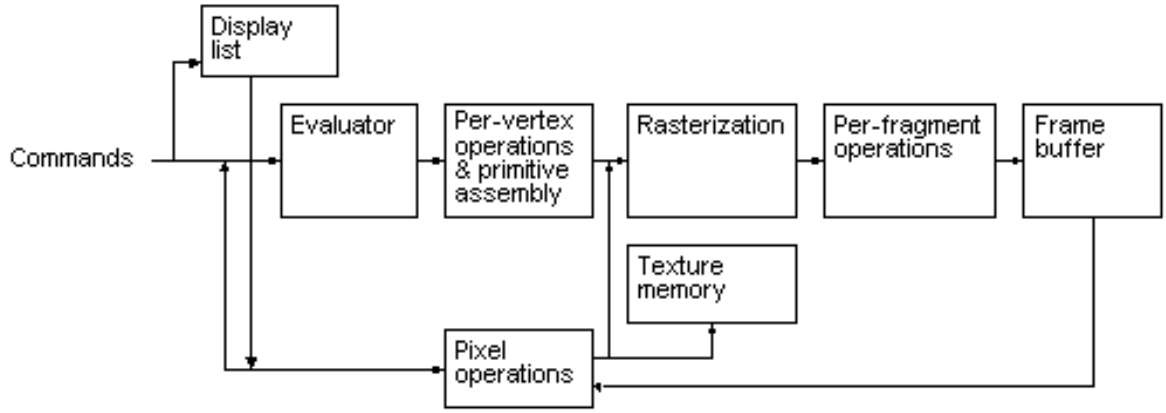
Modelem pro interpretaci příkazů OpenGL je klient/server. Kód aplikace (klient) vysílá příkazy, které jsou interpretovány a zpracovány knihovnou OpenGL (server). Tento server může nebo nemusí operovat (*běžet*) na stejném stroji jako klient. V tomto smyslu je OpenGL síťově transparentní. Server může udržovat několik kontextů OpenGL, z nichž každý je zapouzdřeným stavem OpenGL. Klient se může připojit na kterýkoliv jeden z těchto kontextů. Požadovaný síťový protokol může být implementován rozšířením již existujícího protokolu nebo použitím nezávislého protokolu. Žádné příkazy OpenGL nejsou poskytnuty pro získání vstupu od uživatele.
Celkově vzato, GUI, které alokuje zdroje frame bufferu, na něm řídí vykonávání příkazů OpenGL. Potom tedy uživatelské rozhraní:

určuje, s jakými částmi frame bufferu může OpenGL manipulovat v přiděleném čase,
sděluje OpenGL, jak jsou tyto části strukturovány.

Nyní je tedy zřejmé, že neexistují žádné příkazy OpenGL k nakonfigurování frame bufferu nebo k inicializaci OpenGL. Konfigurace frame bufferu je provedena vně OpenGL ve spojitosti s GUI, inicializace OpenGL se provádí během alokace (zdrojů) okna GUI pro vykreslování knihovnou OpenGL.

3.7 Základní operace OpenGL

Blokový diagram uvedený na obr. 4 velmi zjednodušeně ilustruje, jak OpenGL zpracovává data. Příkaz vstupuje do diagramu z levé strany a postupuje skrze něco, co by mohlo být nazváno zpracovácajícím proudem - processing pipeline. Některé příkazy specifikují geometrické objekty pro vykreslení, jiné určují, jak bude s objektem zacházeno během různých fází zpracování.



Obr. 4.

Fáze zpracování při zjednodušeném pohledu na OpenGL jsou následující:
display list (seznam pro zobrazení) - Nežli bezprostředně zpracovávat všechny příkazy skrze pipeline, je lepší, aby byly některé z nich nashromážděny v display listu pro pozdější zpracování.
evaluator (vyhodnocení) - Fáze vyhodnocování poskytuje účinný nástroj pro aproximaci geometrických křivek a ploch vyhodnocením polynomiálních příkazů ze vstupních hodnot.
per-vertex operations and primitive assembly (operace nad vrcholy a skládání primitiv) - OpenGL zpracovává geometrická primitiva - body, úsečky a n-úhelníky - zadaná všechna jejich vrcholy. Vrcholy jsou transformovány a osvětleny a primitiva jsou oříznuta podle transformace viewport při přípravě na rasterizaci.
rasterization (rasterizace) - Fáze rastrování produkuje řadu adres z frame bufferu se sdruženými údaji používající dvourozměrný popis bodů, úseček a n-úhelníků. Každý fragment takto vytvořený je poskytnut poslední fázi zpracování.
per - fragment operations (operace nad fragmenty) - Jsou posledními operacemi vykonanými nad daty, než jsou uložena jako pixely do frame bufferu. Obsahují podmíněné obnovení frame bufferu založené na příchozích a stávajících z-ových hodnotách (pro z-buffering) a na míchání barev příchozích pixelů s uloženými hodnotami barev, zrovna tak jako maskování a další logické operace nad hodnotami pixelů.

Data mohou být zadána kromě vrcholů i ve formě pixelů. V této formě, například jako popis obrázku pro mapování textury, se vynechává první fáze zpracování popsaná výše. Místo toho jsou tato data zpracována jako pixely, ve fázi operací nad pixely. Potom jsou buďto uložena jako paměť textury (pro použití ve fázi rastrování), nebo rasterizována a následně vložena do frame bufferu právě tak, jako by byla vygenerována z geometrických dat.

3.8 Praktické použití OpenGL v Delphi5

Nejprve připojíme knihovnu uvedením jejího názvu v klauzuli `uses: OpenGL`.

Pak musíme před začátkem práce s touto knihovnou provést nastavení kontextu (určení, kam bude OpenGL vykreslovat svůj výstup) a formátu pixelu. Formát pixelu musí být nastaven i při změně velikosti okna, do kterého je směrován výstup. Vykreslení celé scény pak musíme provádět vždy, když dojde ke:

změně ve scéně,
změně polohy pozorovatele,

odkrytí části okna (například do té doby zakryté jiným oknem), do kterého je směrován výstup OpenGL.

Vlastní vykreslení scény se sestává z nastavení stavu OpenGL, tedy módu, způsobu osvětlení, matic transformace (modelová a projekční) apod., a definování souřadnic vrcholů geometrických primitiv nebo již použití hotových display listů.

Za jednu z nejzdařilejších publikací je pokládána [NDW96], dobrým pomocníkem je též [Mic98]. Dále existuje též zajímavá literatura [Wai96] dostupná i na katedrálním serveru [WWW4].

3.9 Display List

Display list je jedním z charakteristických prvků OpenGL. Jeho největší výhoda spočívá zjednodušeně v tom, že umožňuje spočítat opakující se výpočty pouze jednou, jejich výsledky uložit v jakési kompaktní formě do paměti grafické karty (pokud poskytuje hardwareovou podporu display listů z OpenGL) a v případě potřeby pouze udat odkaz na tyto výsledky grafické kartě. Pokud se například primitiva ve scéně nemění, je vhodné vytvořit pro takovouto scénu display list, čímž při jeho vytváření dojde k jednomu přenosu dat (popisujících scénu) po sběrnici, display list je uložen v paměti grafické karty a k dalšímu přesunu těch samých dat po sběrnici nemusí dojít, stačí pouze vyslat příkaz pro použití display listu.

3.10 Shrnutí

OpenGL je knihovna implementovaná na mnoha platformách a její značná nezávislost na HW je její velkou předností. Bohužel je také nutné konstatovat, že ne na všech platformách je její implementace zdařilá a výsledky nejsou shodné. (Například Windows: programové vybavení poskytovalo nekorektní výsledky na jednom stroji s OS Windows NT 4.0, zatímco na jiných strojích se stejným OS pracovalo bezchybně tak, jak mělo.)

4. Návrh modelu pro reprezentaci vozovky a automobilu

Model musí být schopný umožnit efektivně reprezentovat *automobil* i *vozovku*. Pod pojmem vozovka si lze představit terén zadaný trojúhelníkovou sítí. Automobil s řidičem je pak nad touto sítí v určité poloze. Automobil se skládá z řidiče, na kterém je zajímavá pro potřeby této práce pouze poloha a směr pohledu jeho očí. Dále se skládá z levého a pravého dálkového světlometu, který lze natočit vůči karosérii o úhel Fí v horizontálním směru a o úhel Theta ve směru vertikálním.

Hlavní součásti modelu jsou tedy tyto:
vozovka zadaná trojúhelníkovou sítí neboli terén,
poloha automobilu v prostoru souřadnic trojúhelníkové sítě,
poloha a směr natočení levého světlometu,
poloha a směr natočení pravého světlometu,
charakteristika parametrů světelného kuželu obou světlometů.

Na trojúhelníkovou síť jsou v zájmu implementace kladena jistá omezení. Nesmí se v ní vyskytnout převisy (jakékoliv dva trojúhelníky se společnou plochou průmětu do roviny *xy*). Musí být ve speciálním tvaru, což umožňuje značné urychlení algoritmů pro výpočet normál trojúhelníků a vrcholů sítě, vyhledání trojúhelníku sítě, nad kterým se právě nacházím, s konstantí časovou složitostí apod.

Automobil je tedy popsán polohou P[*x*, *y*, *z*], třemi vektory se souřadnicemi (*x*, *y*, *z*) udávajícími směry vpřed, nahoru a doprava. To umožňuje snadnou implementaci operací posunu a zejména rotace terénu okolo zákoladních os automobilu.
Poloha vozidla v prostoru souřadnic trojúhelníkové sítě reprezentovaná bodem a směrovými vektory má tu výhodu, že pokud ji budeme považovat za počátek souřadného systému karosérie vozidla, směrové vektory se stanou novými bázovými vektory, od kterých se dá odvozovat směr svícení světlometů. Pak se tedy vše odehrává ve vztažné soustavě karoserie a pro definování pozic světlometů a dalšího vybavení vozidla postačí relativní souřadnice.

5. Programové vybavení

Programové vybavení bylo navrženo tak, aby umožnilo ověřit světelné poměry pro dálková světla. Jejich světelný kužel je možné přímo definovat v OpenGL a proto implementace není příliš komplikovaná. U potkávacích světel jsou potřebné další algoritmy navíc pro ořezávání trojúhelníkové sítě v E_3, s čímž souvisí nutnost použití vhodnějších datových struktur (například okřídlená hrana). Důsledkem by samozřejmě byla i větší časová a paměťová složitost. Kapacita paměti by nebyl problém, avšak interaktivita ovládání programu by byla silně ohrožena.

Dokumentace viz dodatky A, B a C.

5.1 Trojúhelníkové sítě definující terén

Trojúhelníkové sítě, se kterými program pracuje, musí být ve speciálním tvaru pro účelné použití v této práci. Veškeré osvětlování je prováděno pouze v rámci OpenGL, což znamená, že tato knihovna pouze spočítá intenzitu osvětlení ve vrcholech sítě a plochy trojúhelníků v síti jsou vybarvovány pomocí interpolace hodnot ve vrcholech. Posvítí-li tedy světlo například do středu velkého trojúhelníku tvořícího síť, zatímco jeho vrcholy zůstanou mimo světelný kužel, tento trojúhelník ve výsledném obrazu zůstane neosvětlen. Pro tyto důvody (navíc rychlost a snadnější implementace) je používána síť, s ekvidistantními vzdálenostmi vrcholů, ve speciálním tvaru tak, jak je popsáno v dokumentaci.

Programové vybavení je připraveno pro další rozšiřování funkčnosti, které by se mělo hlavně týkat implementace algoritmů pro rozbíjení sítě a její ořezávání (pro potkávací světla).

6. Zhodnocení dosažených výsledků

Pro různé jízdní situace byly zjištěny tyto výsledky:

Úroveň osvětlení působila lepším dojmem, pokud byl použit terén skládající se z hustší trojúhelníkové sítě, kdy hranice osvětlené oblasti byly přesnější.
Dále bylo zjištěno, že nastavení směru světlometů ani tak příliš neovlivňuje kvalitu osvětlení povrchu vozovky jako to, v jaké výšce nad vozovkou jsou světlomety umístěny. Dá se předpokládat, že OpenGL přeci jen neposkytuje tak dobré vizualizační vjemy jak by měla, ale vše je dáno mírou zjednodušení a s tímto přístupem se dají obrázky vykreslené OpenGL považovat za velmi reálné.

Nejnáročnější částí algoritmů při generování terénu z hlediska výpočetních nároků je úsek pro výpočet souřadnic z pro všechny body.

Použité algoritmy byly optimalizovány tak, aby byly maximálně efektivní. I při nejhorších podmínkách (z hlediska různých vstupních dat) pracují přibližně stejně rychle.
Použití DisplayListu běh aplikace velice urychlilo. I při vykreslování složitějších scén se interaktivnost aplikace jeví příznivě.

7. Závěr

Byla velice detailně a poměrně přesně prozkoumána oblast bezpečnosti silničního provozu týkající se osvětlování povrchu vozovky, neboli modelování povrchu vozovky kuželem světla ze světlometu, jehož směr svícení lze měnit. Výsledky byly získány z nejnovějších (nejpřesnějších) zdrojů, jaké jsou v současné době k dispozici [EHK01], dokonce byly porovnány i s připravovanými novelami, které měly teprve v době zpracování této práce vstoupit v platnost [EcoDr].

Během zpracovávání první části bylo nutné získat jakýsi přehled v oblasti vnímání a definic barev, což se nakonec vyvinulo ve zpracování velmi rozsáhlé publikace o barvách [Sml01], pojatých očima počítačového grafika. Jsou zde zastoupeny informace ze všech odvětví vědy, jako informatika, biologie, matematika, fyzika a jiných, o barvách z pohledu právě té které disciplíny. Publikace je zpracována na CD.

Z důvodů náročnosti a množství algoritmů bylo zpracováno programové vybavení pouze pro dálková světla, což i tak přineslo autorovi velké množství zkušeností s programovou knihovnou OpenGL, s vývojem grafické aplikace a spoustu podnětů pro další pokračování vývoje programového vybavení v podobě projektu.
Kvalita získaných informací je tak výborná, že vypracování programového vybavení všechny tyto informace využívající je nad rámec možností.

Do budoucna je u této práce možné přidat spoustu dalších rozšíření, aby se daný model opět o kousek blíž posunul k realitě.

8. Literatura a další zdroje

[ECE08] ECE, UN: **Regulation No. 8 - Halogen Filament Lamps.** ECE/TRANS/505, Geneva, 1999.

[ECE20] ECE, UN: **Regulation No. 20 - Headlamps Asymmetrical Passing/Driving Beam.** ECE/TRANS/505, Geneva, 1998.

[ECE98] ECE, UN: **Regulation No. 98 - Gas-Discharge Lamp Units.** ECE/TRANS/505, Geneva, 1998.

[EcoDr] Economic Commission For Europe, UN: **Draft Regulation: Motor Vehicle Headlamps With Filament Lamps.** TRANS/WP.29/773e, Geneva, 2001.

[EHK01] EHK OSN: **Předpisy EHK OSN.** Ženeva, 2001.

[Fed99] Federal Motor Vehicle Safety Standard (MVSS): **USA - MVSS 108 - Lamps, Reflective Devices and Associated Equipment.** NHTSA,Washington, DC, 1999.

[Kub85] Kubín, P.: **Elektrická zařízení osobních automobilů.** SNTL, Praha, 1985.

[Mic98] Microsoft Corp.: **OpenGL Programmer's Reference.** Microsoft Corp., 1998.

[Min95] Ministerstvo dopravy: **Vyhláška č. 102/1995 Sb**. Sbírka zákonů, částka č. 26/1995 Sb.

[NDW96] Neider, J. - Davis, T. - Woo, M.: **OpenGL Programming Guide.** Addison-Wesley, New York, 1996.

[Ska92] Skala, V.: **Algoritmy počítačové grafiky III.** Ediční středisko ZČU, Plzeň, 1992.

[Sml01] Smlsal, T.: **Color CD.** Disk CD, Plzeň, 2001.

[SR94] Šťastný, J. - Remek, B.: **Autoelektrika a autoelektronika.** Nakladatelství T. Malina, Praha, 1994.

[Wai96] Waite group Press: **Open GL Super Bible.** Macmillan Computer Publishing, 1996.

[WL00] Wörner, B. - Lippart, H. J.: **BOSCH: Osvětlovací zařízení.** Robert Bosch odbytová s.r.o., Automobilová diagnostika, Praha 4 Krč, 2000

[WWW1] **Ústav pro výzkum motorových vozidel**
http://www.uvmv.cz/predpasc/index.htm

[WWW2] **Český státní normovací úřad**
http://www.csni.cz/

[WWW3] **Color Science**
http://www.physics.sfasu.edu/astro/color.html

[WWW4] **OpenGL Super Bible**
http://herakles.zcu.cz/local/manuals/
OpenGL_SuperBible/

9. Dodatky

Příloha A - úplná uživatelská dokumentace

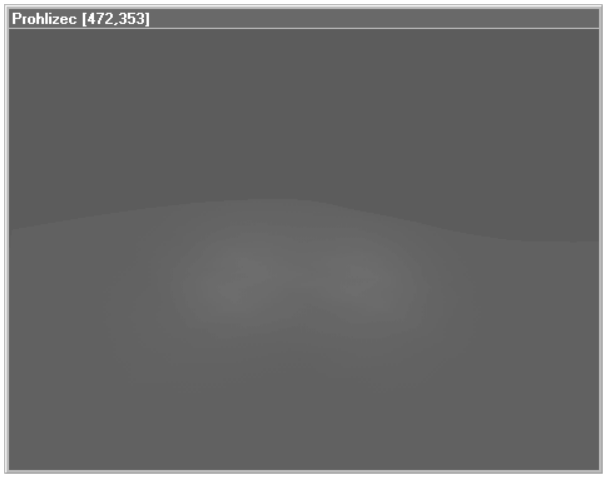
Programové vybavení bylo vyvinuto pro operační systém Windows NT 4.0 a Windows 95. Instalace se provede zkopírováním souboru SVETLA.EXE do adresáře, ze kterého bude spouštěn. Spuštěním tohoto souboru dojde i ke spuštění celého programu.

Minimální systémové nároky jsou shodné s požadavky pro operační systém Windows NT 4.0 a Windows 95.

V programu je použita čeština bez diakritiky, což umožňuje jeho provozování i na nelokalizovaných verzích operačního systému.

Po spuštění programu se objeví následující okno nazvané *Vykreslovani*:

Z tohoto okna se provádí základní obsluha programu. Veškerý grafický výstup je viditelný v tomto okně *Prohlizec*:



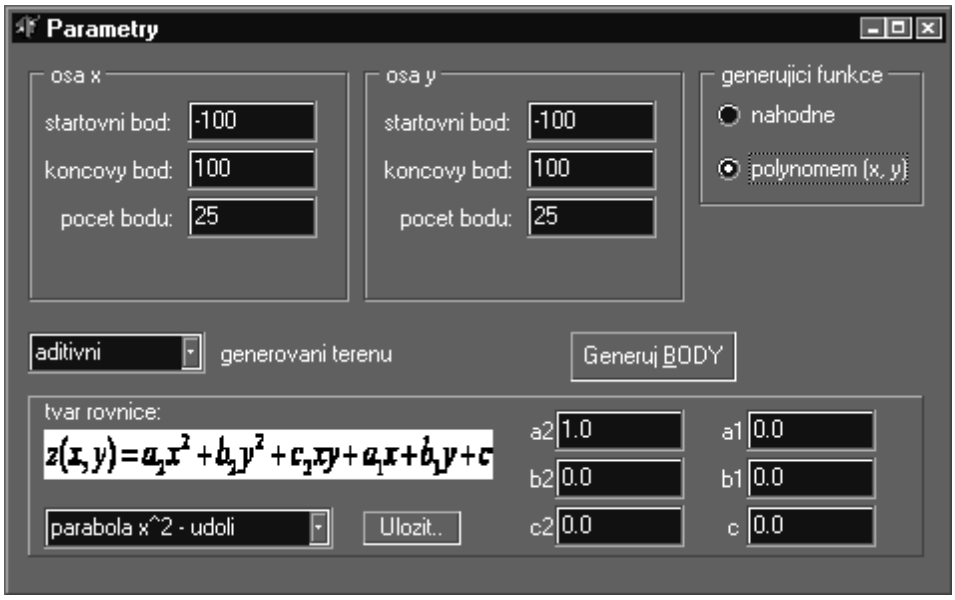
Obsluha se tedy provádí z okna nazvaného *Vykreslovani* a dalších oken podle toho, jaký způsob generování sítě byl použit. V levé části okna jsou dvě vtisknuté plochy, které na sobě shromažďují tlačítka pro ovládání pohybu terénem myší. První skupina tlačítek slouží pro ovládání posunu ve směrech nahoru a dolů (kříž, A - V), doleva a doprava (kříž < - >) a dopředu a dozadu (A - V, rychlejší pak AA - VV). Další skupina tlačítek slouží pro rotace

okolo vlastních hlavních os pozorovatele. Vrchní tlačítka ve tvaru kříže naznačují směr jakoby otočení hlavy ve směru tlačítka, dolní dvě tlačítka otáčejí scénou tak, jako bychom se v automobilu nakláněli do strany.
Napravo od těchto skupin tlačítek je další skupina tlačítek a zaškrtávacích polí. Postupně od shora dolů mají tento význam: panel pro nastavení barvy, tlačítko pro změnu módu vykreslování. Vykreslují se buďto vrcholy, nebo strany trojúhelníků nebo celé trojúhelníky.
Zaškrtávací tlačítko POHYB má ten význam, že po jeho zatržení není nutné opakovaně tisknout tlačítka pro ovládání posunu a rotace, ale tato činnost probíhá automaticky, dokud není aktivováno jiné okno. Zaškrtávací pole označené Z_KONST určuje, budeme-li se nad terénem pohybovat s konstantním odstupem od terénu, nebo libovolně. Tlačítko RESET slouží k nastavení automobilu do počáteční polohy a taktéž nastavení světlometů do počátečního směru. Tato akce je užitečná v případě, že uživatel například ztratil pojem o orientaci v terénu.
V pravé části okna jsou dvě pole pro nastavování směru světel. To se provede stisknutím levého tlačítka myši, tažením zeleného balónku do nové polohy a uvolněním tlačítka myši. V horizontálním směru je nastaven světlomet v rozmezí úhlu přibližně -80° (vlevo) až +80° (vpravo). Jedná se o úhel Fí. Analogicky je nastavován i vertikální sklon světlometu udaný hodnotou Theta.
Na pravém okraji okna jsou dvě zaškrtávací pole. První, DEN, aktivuje celkové osvětlení scény pro případ, že dálková světla nesvítí směrem k povrchu vozovky a uživatel ztratí orientaci, nebo pro komfortnější prohlížení celého terénu. Zatržením volby REF_STRED je zobrazena v počátku souřadného systému terénu referenční krychle, sloužící pro lepší orientaci nebo pro otestování správné funkčnosti knihovny OpenGL.

Nyní je potřeba definovat terén. Můžeme ho buďto vygenerovat a později i uložit do souboru, nebo přímo již takto vytvořený terén načíst ze souboru. Načítání a ukládání terénu ze (do) souboru spustíme z menu SOUBOR, generování terénu z menu TEREN.

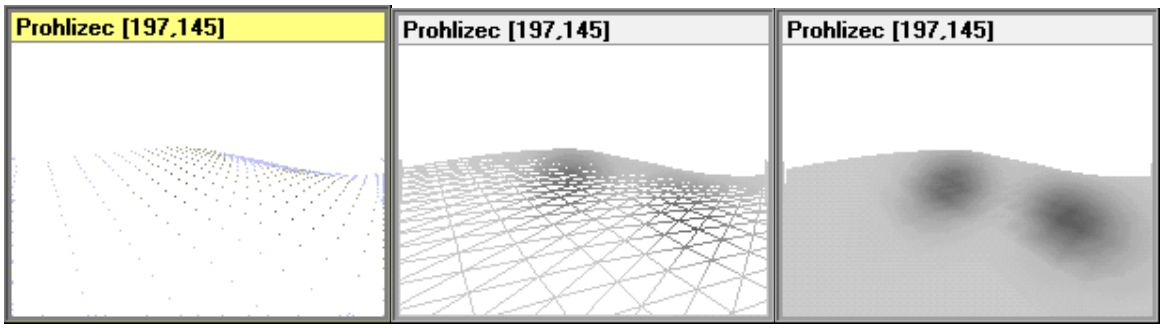
Pokud již terén existuje, je možné se jím pohybovat řízením z hlavního okna, nebo po přepnutí se do okna prohlížeče kurzorovými klávesami.

Postup při generování nebo úpravě stávajícího terénu:
Nejprve zvolíme v menu TEREN položku PARAMETRY. Zobrazí se následující okno:



Odtud nadefinujeme vlastnosti sítě. Způsob, jakým bude vytvořena, a tvar můžeme ovlivnit volbou přepínače *generujici funkce*. Při náhodném generování je volena hodnota z intervalu z-ových hodnot od dolní do horní meze. Parametr vyhlazeni určí, jak moc hladká vygenerovaná síť bude (je dobré volit hodnoty okolo 0,985). Při generování polynomem získáváme paraboloidické tvary podle nastavení koeficientů rovnice. Nastavení je možné pojmenovávat či ukládat, i do souboru pod označením schéma.
Dále je možné nastavit způsob přidávání nových hodnot. *Počáteční* znamená, že celá síť bude vytvořena podle aktuálního předpisu znovu, *aditivní* nebo *subtraktivní* určí, že se síť bude modifikovat pouze tehdy, jeli nová hodnota Z větší (menší) než původní Z daného bodu. Potvrzení akce se provede stiskem tlačítka GENERUJ BODY.
Předchozí proces může být opakován libovolněkrát za sebou, na závěr se stiskne tlačítko GENERUJ SIT. Pak již výslednou síť můžeme přes hlavní menu nechat uložit do souboru nebo si ji prohlédnout v prohlížeči. Pokud zde není nic vidět, doporučuji zkusit trik dvakrát kliknout na tlacitko POSUN + VV, které způsobí naše větší oddálení od scény, zvětšení zorného pole a zobrazení větší části scény.
Poznámka - práce se schématy parametrů je intuitivní (viz panel parametry - rovnice).

Ukázka různých módů zobrazování, jež mají podstatný vliv na rychlost zpracování dat aplikací (barvy obrázků byly invertovány pro lepší názornost):



Zleva doprava: vrcholy, hrany nebo celé trojúhelníky.

V okně *Prohlizec* je možné vidět po každé straně okna v polovině výšky okna zvlášť umístěný trojúhelník. To je referenční značka a měla by být nejvíce osvětlena, pokud je světlomet na té straně okna nasměrován přímo vpřed, podle osy vozidla.

Maximální snahou bylo poskytnout uživatelsky přátelské rozhraní, odolné proti nekorektním vstupům.

Příloha B - úplná dokumentace pro instalaci

Instalace se provede pouze zkopírováním souboru SVETLA.EXE do požadovaného adresáře.

Možnou chybou v případě instalace by mohla být chybějící knihovna OpenGL v systémovém adresáři Windows, tedy *opengl32.dll* nebo *glu32.dll*.

Odstranění chyby: zkopírování chybějících souborů do adresáře knihoven DLL v adresáři Windows.

Příloha C - úplná dokumentace programátora

Byl zvolen programovací jazyk Pascal ve vývojovém prostředí Delphi Standard Edition v. 5.0. Celá aplikace je rozdělena do více než desítky modulů (toto množství je dáno použitím rámců - frame - pro zobrazování částí formulářů na jiné formuláře).

Použité moduly a jejich význam

Používám několik důležitých modulů, kde UNIT1 slouží pro ovládání činnosti aplikace, UNIT2 obsluhuje *prohlizec*, ve kterém jsou vykreslovány všechny výstupy této aplikace pomocí knihovny OpenGL. Každému modulu přísluší vlastní okno ve spuštěné aplikaci, což poskytuje větší komfort pro uživatele. Dalšími moduly jsou například formuláře pro zobrazování a nastavování různých parametrů UNIT9, UNIT6 zajišťující práci s trojúhelníkovou sítí, UNIT12 s funkcemi pro generaci bodů a výpočty sítě a podobně.

Volba implementujících datových typů a datových struktur

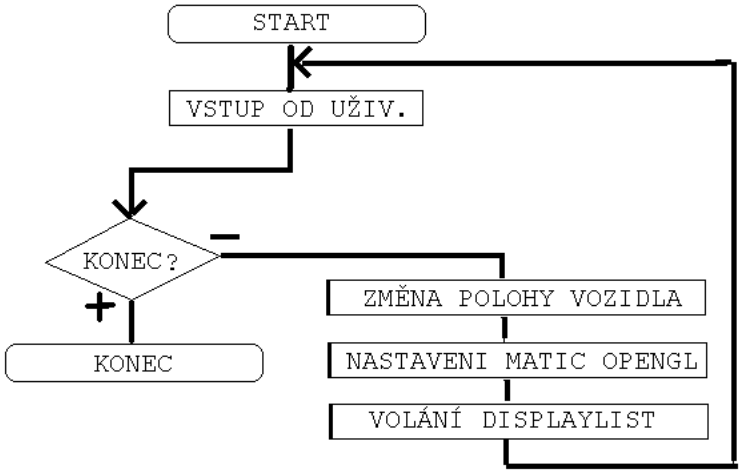
Pro uložení terénu v paměti jsem se rozhodl použít tabulky formátu TRI. Výhodou je jejich úsporné obsazení paměti. Síť je tedy struktura obsahující tabulky souřadnic vrcholů, normál ve vrcholech, trojúhelníků, normál trojúhelníků a několik základních informací o síti. Tabulka je zde reprezentována jednorozměrným polem prvků tabulky, což umožňuje efektivní práci s polem pomocí pointerů.

```
type TSit = record //pro uložení trojúhlové sítě do paměti
              NormalaVrch: ^Real;
              Vrchol: ^Real; //první vrchol X,Y,Z v poli vrcholů
              PocetVrcholu: Integer; // délka pole s vrcholy
              Troj: ^Integer; //1. trojúhel(ník)
              Normala: ^Real; //normála 1. trojúhlu
              PocetTroj: Integer; // počet trojúhlů i jejich normál
              MaterialAmbDif: array [0..3] of GLfloat; //barva materiálu
              XL, YL, ZL, XH, YH, ZH: Real; // extrémy
            end;
```

Pomocný datový typ pro reprezentaci rovnice, která posloužila ke generování terénu. Jedná se o prvek spojového seznamu.

```
PSchema = ^TSchema; // seznam pro schéma parametrů
  TSchema = record
              a2, b2, c2: Real; // koeficienty rovnice
              a1, b1, c: Real;
              dalsi: PSchema; // ukazatel na další schéma
            end;
```

Běh programu je založen na principu přijímání příkazů o změně polohy a parametrů od uživatele a jejich následným zpracováním tak, jak je naznačeno na následujícím schématu:



Procedury, funkce, proměnné: hlavičky tak, jak jsou uvedeny v komentáři

```
// otoceni dvou kolmych jednotkovych vektoru v jejich rovine
// vstup: LX, LY, LZ .. souradnice prvniho vektoru
//        UX, UY, UZ .. druhy vektor
//        Fi .. uhel otoceni ve stupnich
// vystup: pootocene vektory
procedure OtocLU(var LX, LY, LZ, UX, UY, UZ: Real; Fi: Real);


// nacteni seznamu schemat do souboru
procedure NactiSchema;

//ulozeni seznamu schemat do souboru
procedure ZapisSchema;

//vlozeni prvku (schemata parametru) do seznamu
procedure VlozSchema(a2, b2, c2, a1, b1, c: Real);

//aktualizuje parametry
procedure AktualizujParam;


OkoX, OkoY, OkoZ,
VrchX, VrchY, VrchZ,
CeloX, CeloY, CeloZ,
RukaX, RukaY, RukaZ: Real; // definice umisteni pozorovatele
                           // je myslena prava ruka pro urceni co je vpravo


VyskaNadTerenem: Real;


// fce pro vypocet vysky nad trojuhelnikem z troj site
// vstup: souradnice X,Y polohy pozorovatele nad siti
// vraci: Z-ovou souradnici
function VyskaNadT(OkoXP,OkoYP: Real): Real;


// alokovani pameti pro promenne struktury TSit
//  vstup: SIT .. promenna typu TSit; trojuhlova sit
//         V .. pocet vrcholu
//         T .. pocet trojuhelniku a normal
//  vraci: 0 .. vse probehlo bez problemu, pamet pridelena
//         jinak .. bylo malo pameti
function DejMistoNaSit(var Sit: TSit; V, T: Integer): Integer;


// odebrani pameti troj. siti
// vstup: SIT .. promenna typu TSit; trojuhlova sit
procedure OdeberMistoNaSit(var Sit: TSit);
```

```
// vytvoreni site a vypocet normal trojuhlu troj. site
procedure VytvorSit;


// ulozeni site do souboru
procedure UlozSit;


// zjisteni, kam se bude ukladat
procedure KamUlozitSit(var S: String);

// nacteni site ze souboru
procedure NactiSit;

// zjisteni extremnich hodnot
procedure DejExtremySite(var XL, YL, ZL, XH, YH, ZH: Real);

// vytvori novy list podle aktualniho stavu site Sit, lepe je nastavit ListExistuje na false a
// zavolat invalidate
// naplneni hodnot souradnic
procedure GenerujXY;


// !!! nasl 3 proc pouzivaji pro generaci fci definovanou parametry

// pocatecni zpusob; stare hodnoty z=f(x,y) jsou prepisovany
procedure Pocatecni;

// aditivni zpusob; ke starym hodnotam Z jsou nove hodnoty pricitany, pouze
// pokud jsou >0
procedure Aditivni;

// subtraktivni zpusob; ke starym Z jsou nove Z pricitany, pouze
// pokud jsou mensi nez 0
procedure Subtraktivni;


// nasledujici fce pouzivaji nahodne generovani hodnot

// pocatecni zpusob; stare hodnoty z=f(x,y) jsou prepisovany
procedure PocatecniN;

// aditivni zpusob; ke starym hodnotam Z jsou nove hodnoty pricitany, pouze
// pokud jsou >0
procedure AditivniN;

// subtraktivni zpusob; ke starym Z jsou nove Z pricitany, pouze
// pokud jsou mensi nez 0
procedure SubtraktivniN;


// vygenerování nového display listu
procedure NovyDisplayList;
```

Výčet specifických vlastností programu

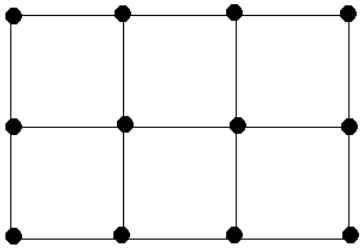
S nastavením směru dálkových světlometů lze experimentovat nad mnoha typy terénů, které si může uživatel vytvořit podle svých představ.

Vykreslování celé scény probíhá v případě, že byla změněna velikost okna, nebo bylo kresleno přes toto okno.

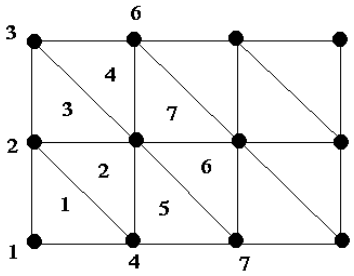
Veškeré datové i programové struktury jsem pečlivě posoudil, zdali jsou skutečně optimální a bylo-li by je možné ještě nějak vylepšit. (Pokud naleznete takový případ, informujte mě o něm prosím.) Veškerá pole jsou jednorozměrná z důvodu rychlého přístupu ukazateli, které jsou počítány většinou jednoduchými součty. Využívám vlastností trojúhelníkové sítě, a proto (pro výpočty) v paměti čtu pouze z-ové souřadnice. To je rychlejší, než kdybych četl i ostatní dvě. Dále je to využití znalosti o pořadí bodů, kdy při výpočtu normály trojúhelníku pro

závěrečné zobrazení pro jednu normálu provedu vědomě pouze 3 přístupy do paměti pro hodnoty souřadnic z a 7 násobení, což považuji za velmi dobrý výsledek (normovaná normála). Vypočítané normály jsou nakonec uloženy v paměti a připraveny pro export nebo zobrazení.

Formát vstupních dat - trojúhelníková síť TRI:
Vstupní data musí vyhovovat formátu .TRI (http://iason.zcu.cz/~lederbuc/local/zpg/TRISpec1.2.html) a navíc podléhají následujícím omezením:



jimi reprezentovaná síť obsahuje ekvidistantně rozložené body ve smyslu souřadnic x, y, které tvoří tuto strukturu (např. při z = 0, pohled na rovinu xy): viz obrázek vpravo;
body jsou uspořádány a jejich pořadi je dáno takto: sloupce zleva doprava a ve sloupci odspodu nahoru.



V této síti jsou trojúhelníky rozmístěny právě takto- viz obrázek vpravo.
Dále jsou čteny normály trojúhelníků, které musí mířit v kladném směru osy z.

Formát výstupních dat: trojúhelníková síť TRI verze 1.2.