

ASCII ray tracing

autori : Ján Kubiš
 Peter Javorský
PGR, 3.12.2008

1. Zadanie :

Úlohou bolo vytvoriť aplikáciu, ktorá bude v reálnom čase zobrazovať naprogramovanú scénu pomocou ray tracingu v ASCII znakoch.

2. Návrh tvorby :

Aplikácia by mala ísť spustiť pod ľubovoľným shellom. Pre lepšiu prácu so znakmi vo vybranných shelloch budeme používať knižnicu ncurses. Scéna bude zložená z guľových telies. Animácia bude tvorená kamerou, ktorá bude krúžiť okolo scény. Modelové scény budú dve Snehuliak a Kyvadlo. Pri spustení aplikácie si užívateľ vyberie danú scénu. Pri zobrazovaní objektov v scéne bude použité difúzne osvetlenie.

3. Definícia použitého ray tracingu :

Pre definíciu ray tracingu si uvedieme najprv čo je zobrazovanie v reálnom prostredí. Vnímanie prostredia človekom je vlastnosť oka reagovať na daný svetelný lúč. Tento svetelný lúč ma fyzikálne vlastnosti, ktoré sa menia vplyvom okolia. Ray tracing je metóda simulovať daný svetelný lúč a jeho vlastnosti vzhľadom na interakciu s prostredím. Keď si predstavíme zdroj daných svetelných lúčov je zrejmé nemožné simulovať všetky lúče zo zdroja. Spolu s ďalšími parametrami vnímania prostredia v reálnom svete je daná problematika neriešiteľná. Preto danú reálnu scénu si vyjadríme modelom, v ktorom si definujeme dané parametre na určitom rozsahu. Rozsah vyjadruje použiteľné zdroje pre dosiahnutie požadovaného výsledku. Jednou základnou vlastnosťou zjednodušiť výpočet je zamerať sa len na lúče, ktoré vníma oko pozorovateľa. To znamená, že môžeme vysielat' lúče od pozorovateľa čo nám daný výpočet zjednoduší a urýchli.

1. Příklad:

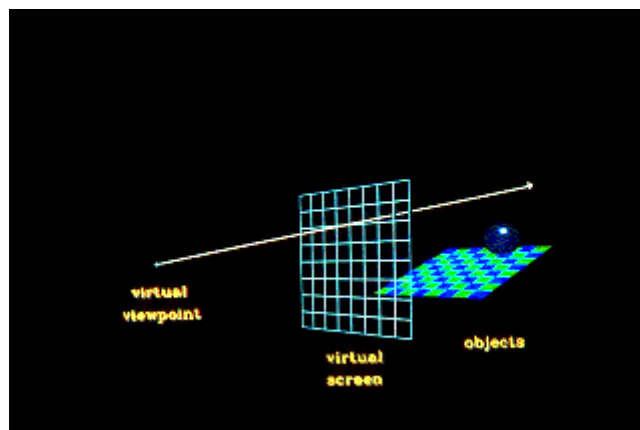
Majme scénu s krabicou, v ktorej je vo vnútri svetelný zdroj. Keby sme brali v úvahu fyzikálne vlastnosti daného lúča a chceli ich simulovať musíme počítať aj lúče, ktoré k pozorovateľovi nikdy nedorazia. Preto počítame vlastnosti pre lúče vychádzajúce z oka pozorovateľa. Pokiaľ lúč dorazí po určitom počte odrazov do zdroja svetla, môžeme uplatniť vlastnosti modelu naň a tak interpretovať svetelný lúč v našom modele pozorovateľovi.

4. Matematické vyjadrenie raz tracingu

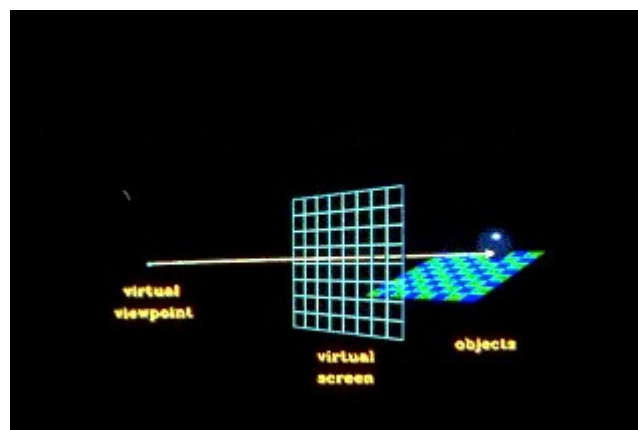
Najčastejšia zobrazovacia technológia používaná v počítačovej technológii je rastorové zobrazenie. Preto náš 3D model scény je prepočítavaný pomocou raz tracingu do rastorového zobrazenia. Takže lúče z pozorovateľa sú smerované na jednotlivé pixely v danom rastru.

1. Výpočet lúča z pozorovateľa a jeho prienik s objektom gule

Máme polohu pozorovateľa P_p a vzdialenosť rastru od pozorovateľa R_p a jeho veľkosť v pixeloch H_d , V_d . Ako dva body pre výpočet priamky lúča môžeme zobrať polohu pozorovateľa a polohu daného pixelu v rastru. Rovnica priamky je $P_{ciel.} = P_{pozor.} + t * P_{pixlu}$ takto definovaná priamka sa vloží do vzorca pre guľu $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$. Priamka definovaná pre jednotlivé súradnice $X = X_0 + X_d * t$ $Y = Y_0 + Y_d * t$ $Z = Z_0 + Z_d * t$ a môžeme za jednotlivé súradnice x, y, z dosadiť do vzorca pre guľu, z ktorej dostaneme po úprave kvadratickú rovnicu $A * t^2 + B * t + C = 0$ z ktorej vypočítame korene $x_{1,2}$. V našom prípade hľadáme kladný diskriminant čo značí kolíziu s telesom. Pokiaľ je diskriminant záporný, kolízia nenastala.

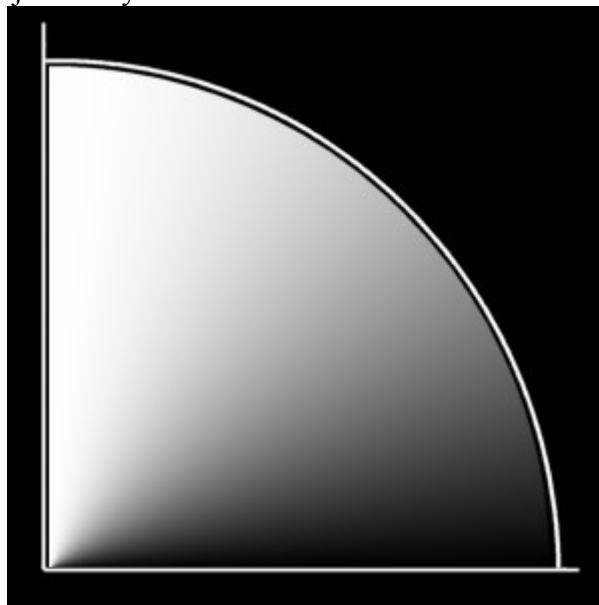


a.) diskriminant je záporný, kolízia nenastala



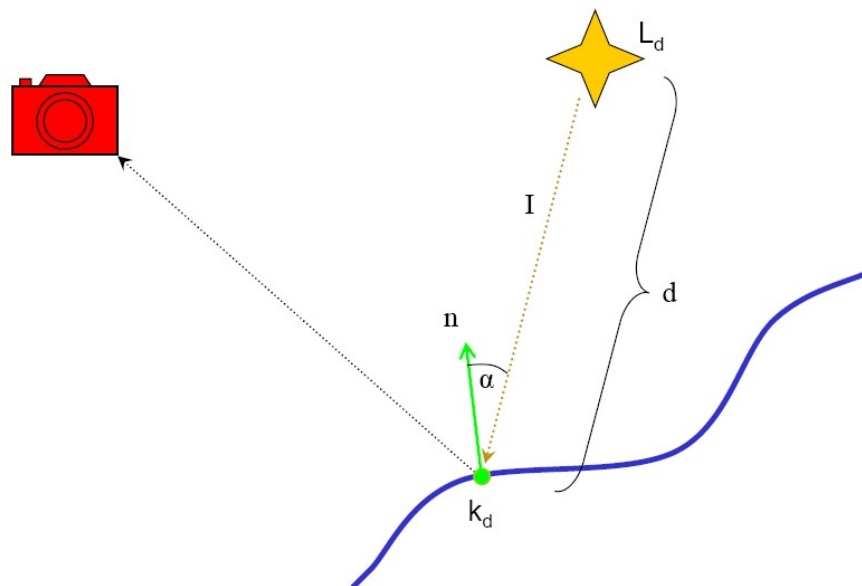
b.) diskriminant je kladný kolízia nastala

Pri kolízii si vypočítame kolízny bod na telese, z ktorého ďalej zistíme normálu pre tento bod. Normálu potrebujeme pre výpočet odrazu daného lúča. V našom prípade je uvažované bez odrazov od objektov. Počítame s difúznym svetlom, ktoré počítame pomocou metódy pre difúzne osvetlenie. Pri tomto výpočte potrebujeme uhly, ktoré sú medzi normálou a dopadajúcim lúčom zo zdroja svetla. Ďalší lúč pôjde z bodu priesečníku smerom do zdroja svetla. Pokiaľ je lúč bez kolízie s ďalšími objektami, tak môžeme počítať difúzne osvetlenie. Toto osvetlenie sa vypočíta z hodnôt uhlov lúčov od zdroja svetla k bodu prieniku a danej normály.



c.) difúzne osvetlenie

Na obrázkoch môžeme vidieť intenzitu svetla pre rôzne uhly. Uhol vypočítavame podľa nasledujúceho obrázku, kde sú znázornené uhly.



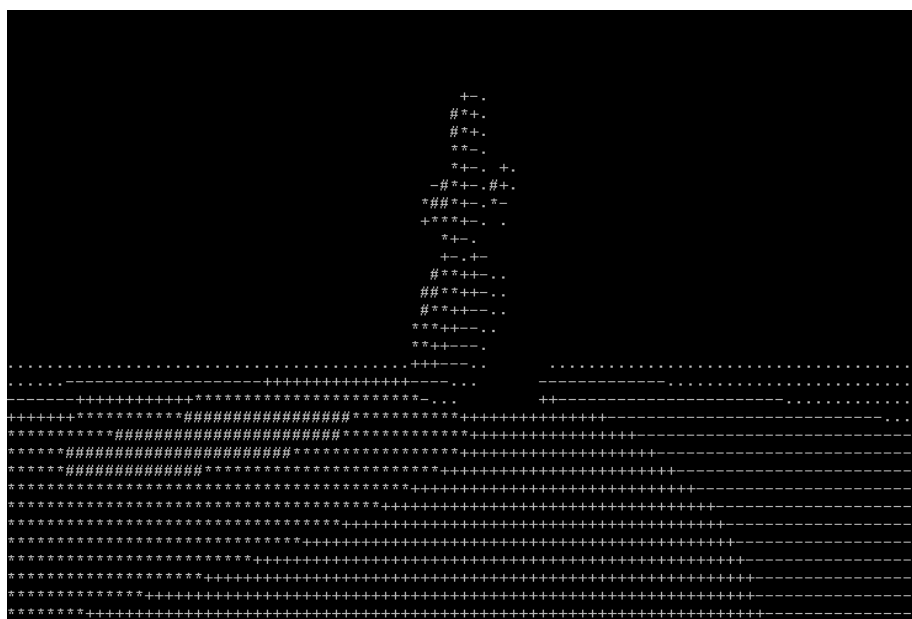
e.) uhol zvierajúci lúč zdroja svetla a normály

Takže v prípade, ktorý znázorňuje obrázok môžeme vybrať intenzitu svetla podľa uhla α a to pre difúzne svetlo. Výsledná hodnota nám udáva konečnú hodnotu intenzity svetla a tá sa namapuje na daný ASCII znak.

5. Aplikácia :

Naša aplikácia vypočítava ray-Tracing a výsledná hodnota je zobrazovaná pomocou ASCII znakov v shellu.

Je snaha, aby naprogramovaná aplikácia fungovala pod rôznymi shellmi. Pre lepšiu prácu so znakovou sadou, ktorú používa shell sme použili knižnicu ncurses. Aplikácia znázorňuje dve scény, snehuliaka a kyvadlo. Ray tracing je vypočítavaný pre difúzni model osvetlenia. To znamená, že určovanie či a aký znak bude vykreslený definuje práve tento osvetľovací model. Dole sú screenshots z danej obrazovky.



f.) screenshot z aplikácie pre snehuliaka



g.) screenshot z aplikácie pre kyvadlo

Aplikácia pracuje len s objektami guľového tvaru, ktoré sú umiestnené na rovine. Aplikácia je naprogramovaná v C a používa už zmienenú knižnicu ncurses. Build danej aplikácie je pomocou príkazu gcc -Wall -o raytracer raytracer.c -lm -lncurses.

6. Záver :

Aplikácia je základom v ray tracingu. Dokáže zobrazovať iba guľové objekty a to len pre difúzny osvetľovací model. Animácia je tvorená krúžením kamery okolo scény. Je zrejmé, že aplikáciu je možné rozšíriť pre výpočet tieňov a doplnenie ďalších svetelných modelov.

7. Použité zdroje

Webové stránky :

- www.codegear.com
- www.wikipedia.org

Materialy z prednášok – slaidy, opora z PGR