Test

Adamofus

December 21, 2022

Contents

Chapter 1

Projekce scény

Projekcí scény je v této práci myšleno perspektivní vidění. Paprsek jdoucí od promítaného bodu do oka pozorovatele se promítá na určenou rovinu v prostoru, tz. tvoří průnik s touto rovinou. Výsledný bod je přenesený na 2D soustavu souřadnou v této rovině.

1.1 Stínění objektů v 3D

1.2 Zoom

Chapter 2

Rasterizace

Rasterizace je proces převodu vektorově definované grafiky do tzv. rastru, tedy mřížky skládající se z bodů (pixelů). Takový rastr je základem obrazového výstupu na digitálních zařízení. V následujících kapitolách jsou představeny algoritmy rasterizace 2 objektů: úsečky a kruhu. Samostatná kapitola je pak věnovaná rasterizaci Beziérových křivek.

2.1 Úsečka

Převedení vstupu
Úsečka je část přímky definovaná dvěma body: b_1 a b_2. Směrnice úsečky je $a = dy/dx$. Pro následující algoritmy platí, že vektor $\vec{b} = b_2 - b_1$ má kladnou souřadnice, protože se algoritmus posouvá o jeden dílek doprava od $b_1 x$ do $b_2 x$. Počítá se s vstupem $a \subset < 0; 1 >$, takže úsečka svírá s osou x úhel $0 - 45$ stupňů.

Tyto podmínky umožňují zvolit efektivní algoritmy, ale zároveň vyžadují transformaci souřadnic.

DDA
DDA využívá zaokrouhlované hodnoty $n * a$ pro výpočet y_{new}, je to prostý přístup. Současně ale zbytečně používá funkci zaokrouhlování či přetypování, kterou lze pro optimalizaci nahradit podmínkou s použitím proměnné, protože jsou pouze 2 možnosti pro y_{next}: $y_{next} = y$, nebo $y_{next} = y + 1$.

Proměnnou vyjadřuje $error(n) = ((n * a) \mod 1) - 1/2$, kde n je krok iterace, takže $error(0) = -1/2$. Pokud platí zmíněná podmínka $error(n) >= 0$, platí současně $y_{next} = y + 1$ a $error(n + 1) = error(n) + a - 1$, jinak platí $error(n + 1) = error(n) + a$.

Bresenhamův algoritmus
Další optimalizací je zbavení se desetinné čárky (tzv. float point number).

Představme si $error(n) = 2 * dx * error(n)$. Jsou 2 možnosti:
$error(n+1) = error(n) + 2 * dy$ $error(n+1) = error(n) + 2 * dy - 2 * dx$
Nerovnice podmínky se nemění, protože je jako bychom ji vynásobili $2 * dx$: $error(n+1) >= 0 * 2 * dx$.

2.2 Kruh

2.3 Ostatní

2.4 Bezierova křivka

2.5 Vyplňování

Řádkovací metoda
Inverzní vyplňování
Flood fill

Chapter 3

Analýza

3.1 Detekce kolize

3.2 Oblasti průniku dvou objektů a průřez

Sutherland–Hodgman algoritmus
Weiler–Atherton clipping algoritmus

3.3 Obsah polygonu

https://mathworld.wolfram.com/PolygonArea.html

3.4 Je A v B?

Chapter 4

Zrcadlení přes střed a osu

Chapter 5

Rotace

Otáčet bod vůči středu soustavy souřadné je jako nanášet ho na otočenou soustavu souřadnou, tedy násobit vektory udávájící osy x, y, atd. takové soustavy. Tyto vektory mají délku 1. Lze zapsat takto:

$$p_n = \begin{pmatrix} X_1 & Y_1 \\ X_2 & Y_2 \end{pmatrix} \vec{p}$$

5.1 2D

Otočené souřadnice můžeme vyjádřit takto:

$X_1 = \cos(-\alpha) = \cos(\alpha)$
$X_2 = \sin(2\pi - \alpha) = -\sin(\alpha)$
$Y_1 = \cos(\pi/2 - \alpha) = \sin(\alpha)$
$Y_2 = \sin(\pi/2 - \alpha) = \cos(-\alpha) = \cos(\alpha)$

$$\begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

5.2 3D

Rotace bodu vůči ose Z ve směru hodinových ručiček:

$$\begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Rotace bodu vůči zvolené ose:

Máme vstup osu o a úhel α.

Mějme 2 soustavy souřadné A a B popsané jednotkovými vektory. Přitom A je naše výchozí, na které vykreslujeme:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matice B má tvar:

$$\begin{pmatrix} a_x & b_x & o_x \\ a_y & b_y & o_y \\ a_z & b_z & o_z \end{pmatrix}$$

3. řádek B tvoří souřadnice osy o, takže 3. souřadnice p_b je právě vzhledem k o.

Čili je z následující rovnice zaručeno, že získáme přesně takový bod p_b, kde jeho 3. souřadnice je vzhledem k o. To potřebujeme, protože jsme zvolili matici rotace podle osy Z (respektive 3. osy...), kterou chceme bod násobit. Takže tato souřadnice bodu v B po rotaci bude stejná.

Z rovnice $p_a = p_b B$ vyjádříme tedy p_b vynasobením B^{-1}: $p_b = p_a B^{-1}$

Zbývající vektory a a b v B musí být jednotkové a vzájemně kolmé. takový a dostaneme třeba ignorováním z a přehozením souřadnic následovně: $\vec{a} = \{-y, x, 0\}$, kde x a y jsou souřadnice o. b je už jen vektorovým součinem a a b.

To je tedy převod relativních souřadnic ze soustavy A do soustavy B.

Rotace probíhá následovně:

Vstup: osa otáčení o, úhel α

1 převedeme souřadnice z A do B

2 zrotujeme (například přes matici 1.1, tj. vůči 3. ose)

3 vykreslujeme v A, takže převedeme z B do A

5.3 Posunutí středu a osy otáčení

Pro posunutý střed v 2D platí, že vektor $\vec{XS}$, kde S je střed a X bod, má fakticky souřadnice bodu $\vec{p}$. K výpočtu stačí potom přičíst S.

Stejně u posunuté osy v 3D, je takový vektor vlastně bod p_a V praxi, grafickém editoru, můžeme osu určit typicky dvěma body. A ten jeden z nich (třeba pro intuici první určený) je totiž tento střed S. Nakonec stačí opět přičíst S.

Chapter 6

Generování tvarů

Chapter 7

Fraktály