DOKUMENTACIJA LABORATORIJSKIH VJEŽBI

KOLEGIJ: INTERAKTIVNA RAČUNALNA GRAFIKA

Filip Lukas

Zagreb, lipanj 2020

Sadržaj

Vježba 1 – Osnovne matematičke operacije u računalnoj grafici

Prva laboratorijska vježba odnosi se na rad s matricama i vektorima. Implementirana je u tri podzadatka od kojih prvi podzadatak opisuje operacije nad vektorima i matricama koji su zadani u programskom kodu te ne traži nikakvu interakciju s korisnikom.

Operacije nad vektorima: suma vektora, skalarni umnožak , vektorski umnožak, normiranje vektora te određivanje suprotnog vektora.

Operacije nad matricama: suma matrica, transponirana matrica, inverz matrice.

U drugom podzadatku se od korisnika traži unos tri jednadžbe s tri nepoznanice tako da se slijedno unose brojevi od kojih svaka četiri broja odgovaraju jednoj jednadžbi (npr. 1 2 3 4 ≡ 1x + 2y + 3z = 4). Svaki svaki četvrti broj se sprema u matricu B (3, 1), a ostali brojevi u matricu A (3, 3). Rješenje jednadžbi jest matrica C = $A^{-1} \times B$.

Primjer:

- Unos: 1, 1, 1, 6, -1, -2, 1, -2, 2, 1, 3, 13
- Jednadžbe:
    - 1x + 1y + 1z = 6
    - -1x + -2y + 1z = -2
    - 2x + 1y + 3z = 13
- Preslikavanje u matrice: $A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ 2 & 1 & 3 \end{bmatrix}$, $B = \begin{bmatrix} 6 \\ -2 \\ 13 \end{bmatrix}$
- $C = A^{-1} \times B$, $C = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$
- Rješenje jednadžbi: x = 1, y = 2, z = 3

Treći podzadatak ispisuje baricentrične koordinate unesene točke T s obzirom na zadani trokut. Baricentrične koordinate su korisne jer se pomoću njih lako odredi odnos točke i trokuta. Ako su sve koordinate između 0 i 1 točka je u trokutu, u protivnom točka je izvan trokuta ili leži na trokutu. Program od korisnika traži unos vrhova trokuta te proizvoljne točke T u 3D prostoru (unos x, y, i z koordinata). Potprogram je sličan drugom potprogramu, u ovom slučaju matricu A čine koordinate vrhova trokuta, a matricu B koordinate točke T. Matrica C, odnosno rješenje jednadžbi u prethodnom podzadatku je ekvivalentno baricentričnim koordinatama točke T u odnosu na zadani trokut.

- Vrhovi trokuta: A (A_x, A_y, A_z), B (B_x, B_y, B_z), C (C_x, C_y, C_z)
- Točka T (T_x, T_y, T_z)
- Preslikavanje u matrice: $A = \begin{bmatrix} \mathrm{Ax\ Bx\ Cx} \\ \mathrm{Ay\ By\ Cy} \\ \mathrm{Az\ Bz\ Cz} \end{bmatrix}$, $B = \begin{bmatrix} \mathrm{Tx} \\ \mathrm{Ty} \\ \mathrm{Tz} \end{bmatrix}$
- Baricentrične koordinate: $C = A^{-1} \times B$, $C = \begin{bmatrix} \mathrm{t1} \\ \mathrm{t2} \\ \mathrm{t3} \end{bmatrix}$

Vježba 2 - Pravac, linija

U drugoj laboratorijskoj vježbi implementiran je Bresenhamov algoritam crtanja linije čija se točnost uspoređuje s linijom nacrtanom pomoću ugrađene LINE naredbe koja je za 20 piksela povišena od linije nacrtane pomoću Bresenhamovog algoritma. Od korisnika se traži da pomoću miša zadaje početne i krajnje točke linija. Za iscrtavanje linija na zaslonu se koristi zastavica 'draw'. Početna vrijednost zastavice 'draw' je 'False'. Vrijednost zastavice se postavi na vrijednost 'True' kada korisnik zada krajnju točku linije. Tada se na zaslonu prikažu obje linije. Vrijednost zastavice se postavi na vrijednost 'False' kada korisnik zada početnu točku linije. Tada se izbrišu prijašnje linije sa zaslona. Korisnik može zadavati proizvoljan broj linija.

Bresenhamov algoritam je definiran kroz 8 potprograma od kojih svaki pokriva slučaj za određeni kut nagiba zadane linije (0 – 45, 45 – 90, 90 – 135, 135 – 180, 180 – 225, 225 – 270, 270 – 315, 315 – 360).

Slučajevi kada je dx > dy:

- Linija s osi x zatvara sljedeće kuteve:
  - 0 – 45
    - osnovni Bresenhamov algoritam
    - $i \in (0, dx)$, $dx = x_2 - x_1$, dx je pozitivan
    - svaki korak kada je $d > 0$, y++
  - 135 – 180
    - $i \in (0, -dx)$, dx je negativan
    - svaki korak kada je $d > 0$, y++
  - 180 – 225
    - $i \in (0, -dx)$, dx je negativan
    - svaki korak kada je $d > 0$, y--
  - 315 – 0
    - $i \in (0, dx)$, dx je pozitivan
    - svaki korak kada je $d > 0$, y--
- Za svaki x se crta po jedan y

Slučajevi kada je dy > dx:

- Linija s osi x zatvara sljedeće kuteve: 45 – 90, 90 – 135, 225 – 270, 270 – 315
  - 45 – 90
    - $i \in (0, dy)$, $dy = y_2 - y_1$, dy je pozitivan
    - svaki korak kada je $d > 0$, x++
  - 90 – 135
    - $i \in (0, dy)$, dy je pozitivan
    - svaki korak kada je $d > 0$, x--
  - 225 – 270
    - $i \in (0, -dy)$, dy je negativan

        - svaki korak kada je d > 0, x--
    - 270 – 315
        - i ∈ (0, -dy), dy je negativan
        - svaki korak kada je d > 0, x++
- Za svaki y se crta po jedan x

Vježba 3 - Crtanje i popunjavanje konveksnog poligona

U trećoj laboratorijskoj vježbi korisnik pomoću miša zadaje vrhove konveksnog poligona u smjeru kazaljke na satu. Nakon što se poligon iscrta moguće je ispitati odnos proizvoljno odabrane točke i iscrtanog poligona.

U prvom koraku zadatka se od korisnika očekuje da pomoću miša zada proizvoljan broj vrhova željenog poligona te pritisne tepku 'ENTER'. Nakon unosa vrhova poligona program određuje lijeve i desne bridove poligona te pomoću njih provjerava je li poligon ispravno zadan. Ako je poligon ispravno zadan zastavica 'draw' se postavlja na vrijednost 'True' te se poligon iscrta i oboji, u protivnom zastavica 'draw' ostaje 'False' te se od korisnika traži da ponovno unese vrhove poligona sve dok ne budu ispravno uneseni. Poligon mora biti konveksan te njegovi vrhovi (minimalno 3 vrha) moraju biti zadani u smjeru kazaljke na satu.

Kada korisnik ispravno unese vrhove poligona, poligon se oboji tako da se za svaki y ∈ (y_{min}, y_{max}) pronađu sjecišta zadanog y s lijevim bridovima te se odredi sjecište s najvećom x koordinatom (L_{max}) i sva sjecišta zadanog y s desnim bridovima te se odredi sjecište s najmanjom x koordinatom (D_{min}). Područje između L_{max} i D_{min} se oboji.

Drugi korak zadatka je ispitivanje odnosa zadane točke i iscrtanog poligona. Nakon što je poligon ispravno iscrtan korisnik pomoću miša zadaje proizvoljan broj točaka na ekranu za koje želi ispitati jesu li unutar ili izvan poligona. Ako postoji brid poligona da je zadana točka iznad njega, tada se točka nalazi izvan poligona te se ispiše „X is outside polygon.“, inače točka se nalazi u poligonu te se ispiše „X is inside polygon.“.

Vježba 4 - Ravnina, tijelo

U četvrtoj laboratorijskoj vježbi implementirano je iscrtavanje objekta na zaslonu te ispitivanje odnosa zadane točke i iscrtanog objekta. Objekt je zadan u tekstualnoj datoteci koja sadrži popis vrhova objekta s koordinatama te popis poligona s pripadnim vrhovima.

Primjer zapisa vrha: „v 2.00 1.00 1.00“ – koordinate zadanog vrha su (2, 1, 1)

Primjer zapisa poligona: „f 1 3 2“ – redoslijed vrhova poligona dan u smjeru suprotno kazaljke na satu

Prvi korak zadatka je učitavanje datoteke s popisom vrhova i poligona objekta.

Izračuna se centar objekta za svaku os te se objekt translatira u centar prozora. Zatim se izračuna visina i širina objekta te se odredi faktor skaliranja takav da slika objekta bude skalirana na 80% vrijednosti min(visinaProzora, širinaProzora) na sljedeći način:

viewFactor = 0.8

scaleX = viewFactor * widht / objWidth

scaleY = viewFactor * height / objHeight

scale = min(scaleX, scaleY)

gdje su varijablje 'widht 'i 'height' širina i visina ekrana, 'objWidth' i 'objHeight' širina i visina objekta, 'scaleX' i 'scaleY' pomoćni faktori skaliranja od kojih se odabire manja vrijednost kao konačni faktor skaliranja.

Tako se dobije translatirana i skalirana slika objekta u centru prozora.

Drugi korak je računanje koeficijenata ravnina poligona (originalno zadani poligoni i translatirani poligoni).

Treći korak je korisnikov unos proizvoljne točke za koju želi ispitati odnos s objektom (originalnim i translatiranim).

Nakon unosa točke program ispisuje jednu od sljedećih mogućnosti za originalni te translatirani objekt:

- „Point is outside of the original/translated object.“
- „Point is on the original/translated object.“
- „Point is inside of the original/translated object.“

Vježba 5 - Transformacija pogleda i perspektivna projekcija

U petoj laboratorijskoj vježbi implementirana je transformacija pogleda i perspektivna projekcija nad zadanim objektom.

Prvi korak zadatka je učitavanje objekta (vrhova i poligona) te učitavanje očišta i gledišta iz pripadnih datoteka. U datoteci u kojoj su sadržane koordinate očišta i gledišta prvi red čine koordinate očišta , a drugi red koordinate gledišta.

Nakon učitavanja poziva se potprogram koji računa matricu transformacije pogleda. Matrica transformacije pogleda T je umnožak 5 pomoćnih matrica:

- T_1 – pomak ishodišta koordinatnog sustava u ishodište scene
- T_2 – rotacija za kut α oko z-osi
- T_3 – rotacija za kut β oko y-osi
- T_4 – rotacija za kut 90° oko z-osi
- T_5 – promjena predznaka na x-osi

Slijedi računanje matrice perspektivne projekcije P.

Sljedeći korak je množenje svakog vrha objekta s matricama T i P kako bi se dobila konačna slika koja se preslika na središte ekrana na način opisan u vježbi 4.

Korisnik ima mogućnost mijenjanja položaja gledišta i očišta na sljedeći način.

Mijenjanje koordinata gledišta:

- Pritisak na tipku '6' – G ($g_x + 1$, g_y, g_z)
- Pritisak na tipku '4' – G ($g_x - 1$, g_y, g_z)
- Pritisak na tipku '8' – G (g_x, $g_y + 1$, g_z)
- Pritisak na tipku '2' – G (g_x, $g_y - 1$, g_z)
- Pritisak na tipku '9' – G (g_x, g_y, $g_z + 1$)
- Pritisak na tipku '1' – G (g_x, g_y, $g_z - 1$)

Mijenjanje koordinata očišta:

- Pritisak na lijevu tipku miša – O($o_x + 1$, $o_y + 1$, $o_z + 1$)
- Pritisak na lijevu tipku miša – O($o_x - 1$, $o_y - 1$, $o_z - 1$)

Vježba 6 - Prikaz prostornih krivulja postupkom Beziera

U šestoj laboratorijskoj vježbi implementiran je prikaz aproksimacijske krivulje Beziera. Uz aproksimacijske krivulje postoje i interpolacijske krivulje Beziera. Razlika je ta što aproksimacijska krivulja prolazi samo početnom i krajnjom točkom kontrolnog poligona dok se ostalim točkama približuje, a interpolacijska krivulja prolazi svim točkama kontrolnog poligona.

Korisnik pomoću miša unosi proizvoljan broj vrhova kontrolnog poligona dok ne pritisne tipku 'ENTER'. Ako su unesena manje od 3 vrha program ispisuje sljedeće: „You must enter at least 3 vertex.“ Te se od korisnika očekuje da ponovi unos. Nakon pritiska tipke 'ENTER' izračunavaju se pripadne točke aproksimacijske krivulje Beziera definirane polinomima Bernsteina na sljedeći način:

- $p(t) = \sum_{i=0}^{n} r_i \, b_{i,n}(t)$
- $p(t)$ – točke na krivulji (vektorski zapis $x(t)$, $y(t)$, $z(t)$)
- r_i – vrhovi kontrolnog poligona (točke koje zadajemo)
- $b_{i,\,n}$ – bazne (težinske) funkcije stupnja n

Bazne funkcije definirane su Bernsteinovim polinomima na sljedeći način:

- $b_{i,\,n} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$

Na zaslonu se iscrtava kontrolni poligon s pripadnom aproksimacijskom krivuljom Beziera.

Animacija nije implementirana.

Vježba 7 – Sjenčanje tijela

U sedmoj laboratorijskoj vježbi implementirana su dva načina sjenčanja tijela: konstantno sjenčanje i Gouraudovo sjenčanje.

Program učitava dvije tekstualne datoteke. Prva datoteka sadrži koordinate očišta (1. red), koordinate gledište (2. red) te koordinate izvora svijetla (3. red). Druga datoteka sadrži opis tijela na način opisan u vježbi 4.

Nakon što se objekt učita smjesti se u radni prostor [-1, 1] tako da se odredi najveći raspon tijela po koordinati (max(velicina_x, velicina_y, velicina_z)) te se svaki vrh poligona translatira za određen iznos te skalira na sljedeći način:

V ((v_x - srediste_x)*scale, (v_y - srediste_y)* scale, (v_z - srediste_z)* scale)

gdje su v_x, v_y i v_z koordinate zadanog vrha, srediste_x, srediste_y i srediste_z koordinate središta tijela i scale faktor skaliranja dobiven formulom:

$$\text{scale} = \frac{2}{\max\left(\text{velicina}_x, \text{velicina}_y, \text{velicina\_z}\right)}$$

Nakon što su vrhovi tijela smješteni u radni prostor računaju se matrice transformacije pogleda T i perspektivne projekcije P na način opisan u vježbi 5. Vrhovi se množe matricama T i P te se dobije slika u 2D sustavu prikaza koja se translatira i skalira na središte ekrana na način opisan u vježbi 4.

Sljedeći korak je određivanje normala poligona te određivanje prednjih i stražnjih poligona. Za svaki poligon se redom izračunaju njegove normale, odredi težište pomoću kojeg se izračuna vektor od centra trokuta prema očištu A. Ako je kut između normalne poligona i vektora A manji od 90 stupnjeva poligon je prednji te se u matricu provjere zapiše 1, u protivnom poligon je stražnji i u matricu provjere zapiše se 0. Pri prolasku kroz svaki poligon u trenutku crtanja objekta provjerava se je li u matrici provjere zapisan broj 1 te ako je poligon se iscrta na zaslonu u protivnom poligon se preskače.

Kada su normale poligona izračunate potrebno je izračunati ambijentnu i difuznu komponentu poligona. Za svaki poligon odredi se vektor od težišta trokuta prema izvoru svijetla L te se normira. Ako je kosinus kuta između vektora L i normalne poligona pozitivan u matricu difuzne komponente poligona se zapiše $i_d = i_i * k_d *$ izračunati kosinus, u protivnom poligon je stražnji i u matricu se zapiše 0.

Normale vrhova se računaju tako da se za svaki vrh provjeri koji se poligoni dodiruju u njemu te se izračuna srednja vrijednost svih normala dodirnih poligona.

Difuzna komponenta vrha se računa na isti način kao i difuzna komponenta poligona (vektor L je u ovom slučaju vektor od vrha prema izvoru svjetla).

Intenzitet svakog poligona/vrha se određuje kao zbroj difuzne komponente (I_d) te ambijentne komponente ($i_g = i_a * k_a$). Koeficijenti i_i, k_d, i_a, k_a su zadani u programskom kodu.

Pri iscrtavanju objekta korisnik pritiskom na tipku 'ENTER' bira želi li vidjeti konstantno sjenčanje ili Gouraudovo sjenčanje.

Vježba 8 – Fraktali

Osma vježba podijeljena je u 2 programa. U prvom programu na ekranu se iscrtaje Mandelbrotov fraktalni skup, a u drugom programu Julijev fraktalni skup. U zadatku je implementirano preslikavanje točke iz kompleksne ravnine u ravninu prikaza (prevođenje iz sustava (O u v) u sustav (O' x y). U oba zadatka se za svaku točku ravnine prikaza (x_0, y_0) odredi pripadna točka kompleksne ravnine te se za nju ispita konvergencija niza.

Područje kompleksne ravnine unutar kojega iterativno preslikavanje generira konvergentne nizove naziva se Mandelbrotov skup.

Za Julijev skup potrebno je odabrati $c \in C$ (točku kompleksne ravnine), a z_0 je točka kompleksne ravnine za koju ispitujemo konvergenciju niza. Ako se za $c \in C$ odabere točka unutar Mandelbrotovog skupa Julijev skup će biti povezan, a inače nepovezan.

U zadatku u kojem se iscrtaje Mandelbrotov skup prag (*eps*) je postavljen na 100, maksimalan broj iteracija (m) na 16 te područje kompleksne ravnine koja se promatra (u_{min}, u_{max}) = (-1.5, 0.5) i (v_{min}, v_{max}) = (-1, 1).

U zadatku u kojem se iscrtaje Julijev skup prag (*eps*) je postavljen na 100, maksimalan broj iteracija (m) na 16 te područje kompleksne ravnine koja se promatra (u_{min}, u_{max}) = (-1, 1) i (v_{min}, v_{max}) = (-1.2, 1.2). U ovom primjeru učitana je i kompleksna konstanta c (c_{real}, c_{imag}) = (0.32, 0.043).