Primjena linearne algebre u računalnoj grafici

Kokanović, Karlo

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:981257

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstva

PRIMJENA LINEARNE ALGEBRE U RAČUNALNOJ GRAFICI

Završni rad

Karlo Kokanović

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 21.09.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Karlo Kokanović
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R4224, 25.07.2018.
OIB Pristupnika:	61032245505
Mentor:	Doc.dr.sc. Anita Katić
Sumentor:	Izv. prof. dr. sc. Irena Galić
Sumentor iz tvrtke:	
Naslov završnog rada:	Primjena linearne algebre u računalnoj grafici
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak završnog rad:	
Prijedlog ocjene završnog rada:	Dobar (3)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	21.09.2021.
Datum potvrde ocjene od strane Odbora:	28.09.2021.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 11.07.2022.

Ime i prezime studenta:	Karlo Kokanović
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4224, 25.07.2018.
Turnitin podudaranje [%]:	32%

Ovom izjavom izjavljujem da je rad pod nazivom**: Primjena linearne algebre u računalnoj grafici**

izrađen pod vodstvom mentora Doc.dr.sc. Anita Katić

i sumentora Izv. prof. dr. sc. Irena Galić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. Uvod **2**
1.1. Zadatak završnoga rada 2

2. Linearna algebra **3**
2.1. Radijvektori u prostoru 3
2.2. Vektorski prostor 7
2.3. Afin prostor 9
2.4. Homogene koordinate 11

3. Korištene tehnologije **12**
3.1. C++ programski jezik 12
3.2. OpenGL i GLUT 12
3.3. Microsoft Visual Studio 13

4. Osnovne transformacije **14**
4.1. 3D transformacije 14
4.2. Translacija 14
4.3. Skaliranje 15
4.4. Rotacija 16
4.4.1. Rotacija oko osi x 16
4.4.2. Rotacija oko osi y 17
4.4.3. Rotacija oko osi z 17

5. Realizacija programa **18**
5.1. Funkcije 18
5.2. Translatiranje piramide 20
5.3. Skaliranje kugle 23
5.4. Rotacija kocke 25

6. Zaključak **29**

Literatura **30**

Sažetak i ključne riječi **31**

Application of linear algebra in computer graphics **32**

Životopis **33**

1. UVOD

Svakodnevnim razvojem tehnologija sve veća je potreba za vizualizacijom. Ova ideja se pojavljuje još u 19. stoljeću izumom katodnih cijevi. Metode vizualizacije drastično napreduju kroz 20. stoljeće pa sve do danas kada imamo posebnu disciplinu unutar računarstva koja se bavi problematikom predstavljanja, manipuliranja i sinteze podatcima i slikama. Računalna grafika je sveprisutna tehnologija u različitim zaslonima, digitalnoj fotografiji, filmu, video igrama, itd. Temelj za današnju računalnu grafiku čini upravo linearna algebra.

Ovaj rad je podijeljen na šest poglavlja: uvod, linearna algebra, korištene tehnologije, osnovne transformacije, realizacija programa i zaključak.

U drugom poglavlju opisat ćemo sve koncepte iz linearne algebre potrebne za razumijevanje transformacija unutar računalne grafike. Kroz treće poglavlje ćemo ukratko opisati korištenu tehnologiju prilikom razvoja 3D modela. U četvrtome poglavlju ćemo objasniti tri različite transformacije. U petome poglavlju ćemo nacrtati geometrijska tijela te primjeniti na njih ranije opisane transformacije. Prikazat ćemo primjere skaliranja, translacije te rotacije.

1.1. Zadatak završnoga rada

Zadatak rada je opisati i prikazati tri osnovna oblika transformacija unutar računalne grafike. Kako bi mogli definirati transformacije najprije ćemo objasniti pojmove vektora, afinoga prostora te homogenih koordinata. Za vizualizaciju 3D modela i njegovih transformacija koristit ćemo se programskim sučeljem OpenGL, te programskim alatom GLUT.

2. LINEARNA ALGEBRA

U ovom poglavlju definirat ćemo radijvektore, vektorski te afin prostor. Objasniti ćemo i pojam homogenih koordinata.

2.1. Radijvektori u prostoru

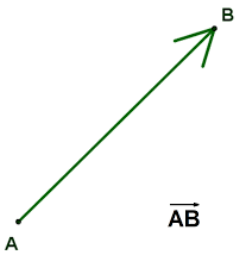
Neka je dan skup točaka u ravnini E^2 ili prostoru E^3, i dvije proizvoljne točke A i B promatrane na toj ravnini ili prostoru. Skup svih točaka na pravcu koji je određen točkama A i B, a koje leže između točaka A i B nazivamo **dužina** i označavamo s $\overline{AB}$. Ako točku A proglasimo početnom, a točku B završnom, takvu dužinu nazivamo **usmjerenom dužinom** i označavamo s $\overrightarrow{AB}$. U literaturi [10] autori definiraju ekvivalentne dužine i pojam vektora na sljedeći način:

Definicija 2.1. *"Kažemo da su dvije usmjerene dužine $\overrightarrow{AB}$ i $\overrightarrow{A'B'}$ ekvivalentne i pišemo $\overrightarrow{AB} \sim \overrightarrow{A'B'}$ ako postoji translacija prostora koja točku A prevodi u A', a točku B u B'."*

Definicija 2.2. *"Vektor $\vec{a}$ je klasa ekvivalencije svih međusobno ekvivalentnih usmjerenih dužina,*

$$\vec{a} = [\overrightarrow{AB}] = \{\overrightarrow{PQ} : \overrightarrow{PQ} \sim \overrightarrow{AB}\}."$$

Napomena 2.1. *Sve usmjerene dužine $\overrightarrow{PQ}$ ekvivalentne usmjerenoj dužini $\overrightarrow{AB}$ ćemo nazvati reprezentant vektora $\vec{a}$.*



Slika 2.1: *Prikaz vektora $\vec{AB}$*

Napomena 2.2. *Uočimo da vektor opisuje smjer, ne položaj.*

Autor u literaturi [10] uvodi sljedeće važne oznake:

Napomena 2.3. *Ako nam norma vektora $\vec{v}$ predstavlja duljinu bilo kojeg reprezentanta, u oznaci $||\vec{v}||$ ćemo uvesti sljedeće oznake:*

- *"Nulvektor $\vec{0}$ je klasa ekvivalencije svih usmjerenih dužina koje imaju istu početnu i završnu točku pri čemu je duljina vektora $||\vec{0}|| = 0$."*

- *"Jedinični vektor zvat ćemo svaki vektor $\vec{v}$ za koga je duljina $||\vec{v}|| = 1$."*

- *"Suprotni vektor vektora $\vec{v}$ je klasa ekvivalencije svih usmjerenih dužina koje imaju suprotnu orijentaciju od orijentacije usmjerenih dužina vektora $\vec{v}$ i označavamo ga s $-\vec{v}$."*

Pojam vektora možemo razmatrati geometrijski i fizikalno. Geometrijska interpretacija bi značila da proizvoljan vektor $\overrightarrow{AB}$ promatramo kao translaciju točke A u točku B. Fizikalna interpretacija znači da proizvoljan vektor $\overrightarrow{AB}$ zamišljamo kao silu kojoj je hvatište u točki A, smjer djelovanja je prema točki B, a intenzitet joj je predočen udaljenošću od A do B.

Sada znamo da je duljina (modul) vektora $\overrightarrow{AB}$ udaljenost točaka A, B (tj. duljina dužine $\overline{AB}$) u oznaci $|\overrightarrow{AB}|$. Smjer vektora $\overrightarrow{AB}$ je smjer koji određuje pravac na kojima su točke A i B. Orijentacija vektora $\overrightarrow{AB}$ je od točke A do točke B.

Autori u literaturi [10] definiraju radijvektor na sljedeći način: "Ako izaberemo jednu fiksnu točku O u prostoru E^3, onda svakoj točki $P \in E^3$ pripada jedinstvena usmjerena dužina $\overrightarrow{OP}$, koju zovemo **radijvektor** ili **vektor položaja**. Skup svih ovakvih radijvektora označit ćemo s

$$V^3(O) = \{\overrightarrow{OP} : P \in E^3\}."$$

Napomena 2.4. *Svaki vektor $\vec{v}$ jednoznačno je određen svojim modulom, smjerom i orijentacijom.*

Za potrebe ovog rada promatrat ćemo vektore u prostoru V^3, stoga je vrlo važno napomenuti (kako autor u literaturi [1] i napominje):

Primjedba 2.1. *"Ukoliko odaberemo i fiksiramo točku $O \in E^3$ onda svaki vektor $\vec{a} \in V^3$ možemo reprezentirati s početkom u točki O. Vektor $\vec{a} = \overrightarrow{[OA]}$ možemo poistovjetiti s njegovim predstavnikom $\overrightarrow{OA}$ i tada $V^3(O)$ pokazuje kako prostor V^3 izgleda "lokalno"."*

Napomena 2.5. *V^3 je realan vektorski prostor.*

Napomena 2.6. *Svaki vektor ima jedinstvenu reprezentaciju pomoću koordinata.*

Dakle, vektor ćemo promatrati kao usmjerenu dužinu - najčešće će predstavljati gradijent, tj. pokazatelj koji će govoriti za koliko se nešto mijenja. Način na koji ćemo zapisivati vektore ovisi o prostoru u kojem se taj vektor opisuje, te broj komponenti ovisi o dimenziji prostora. Ideja je vektore zapisivati u jednostupčanoj matrici ili kao uređene n-torke , npr. vektor $\vec{v} \in V^3$ reprezentiran je izrazom

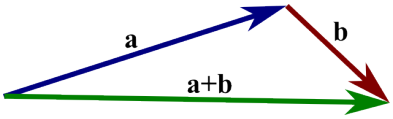
$$\vec{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \text{ ili } \vec{v} = (x, y, z),$$

gdje su komponente vektora $x, y, z \in \mathbb{R}$.

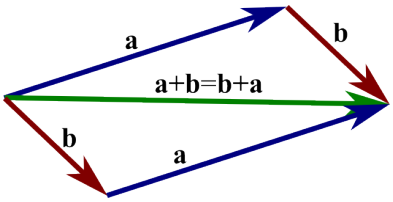
Nadalje, zbrajanje vektora je binarna operacija, tj. funkcija dviju varijabli $+ : V^3 \times V^3 \to V^3$. Za dva vektora $\vec{a}, \vec{b} \in V^3$ definiramo novi vektor $\vec{c} := \vec{a} + \vec{b}$ pravilom trokuta ili pravilom paralelograma.

To znači, ako su

$$\vec{v_1} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \vec{v_2} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}$$



Slika 2.2: *Zbrajanje vektora pravilom trokuta*



Slika 2.3: *Zbrajanje vektora pravilom paralelograma*

tada vrijedi

$$\vec{v_1} + \vec{v_2} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \end{pmatrix} = \begin{pmatrix} x_2 + x_1 \\ y_2 + y_1 \\ z_1 + z_1 \end{pmatrix} = \vec{v_2} + \vec{v_1}. \tag{2.1}$$

Napomena 2.7. *Rezultat zbrajanja vektora je ponovno vektor.*

Množenje vektora sa skalarom je funkcija dviju varijabli $\cdot : \mathbb{R} \times V^3 \to V^3$. Za realni broj $\lambda \in \mathbb{R}$ i vektor $\vec{a} \in V^3$ definiramo novi vektor $\vec{b} := \lambda \cdot \vec{a}$.

Dakle, ako je dan skalar $\lambda \in \mathbb{R}$ i vektor

$$\vec{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

tada vrijedi

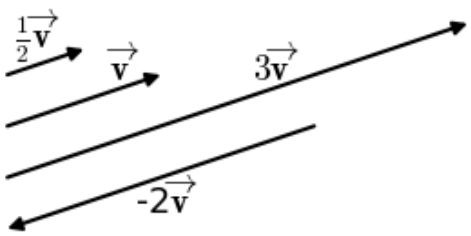
$$\lambda \cdot \vec{v} = \begin{pmatrix} \lambda \cdot x \\ \lambda \cdot y \\ \lambda \cdot z \end{pmatrix}.$$

Očito je tada

$$-\vec{v} = \begin{pmatrix} -x \\ -y \\ -z \end{pmatrix}.$$

Primjedba 2.2. *Oduzimanje vektora svodi se na zbrajanje sa suprotnim vektorom.*

Primjer 2.1. *Neka je dan vektor $\vec{v}$. Množenje vektora $\vec{v}$ skalarima $\lambda_1 = \frac{1}{2}$,$\lambda_2 = -2$,$\lambda_3 = 3$.*



Slika 2.4: *Množenje vektora $\vec{v}$ skalarima*

Za primjene vektora u računalnoj grafici vektor će biti razapet između dvije točke - početne i završne. Primjer reprezentanta vektora $\overrightarrow{AB}$ dan je na Slici 1. Takav vektor, razapet između točaka $A(x_1, y_2)$ i $B(x_2, y_2)$, računa se prema relaciji

$$\vec{v} = (x_2 - x_1)\vec{i} - (y_2 - y_1)\vec{j}. \tag{2.2}$$

Pri tome se točke promatraju kao radijvektori, tj. $i - tu$ komponentu rezultata ćemo dobiti tako da oduzmemo $i - te$ koordinate točaka.

Neka su $\vec{a}, \vec{b}, \vec{c} \in V$ vektori i $\alpha, \beta \in \mathbb{R}$ skalari. Tada vrijede svojstva zbrajanja vektora i množenja skalarom:

1. $\vec{a} + (\vec{b} + \vec{c}) = (\vec{a} + \vec{b}) + \vec{c}$, [asocijativnost zbrajanja]
2. $\vec{0} + \vec{a} = \vec{a} + \vec{0} = \vec{a}$, [postojanje neutralnog elementa $\vec{0}$ za zbrajanje]
3. $\vec{a} + (-\vec{a}) = -\vec{a} + \vec{a} = \vec{0}$, [postojanje inverznog elementa $-\vec{a}$ za zbrajanje]
4. $\vec{a} + \vec{b} = \vec{b} + \vec{a}$, [komutativnost zbrajanja]
5. $\alpha(\beta\vec{a}) = (\alpha\beta)\vec{a}$, [kvaziasocijativnost]
6. $(\alpha + \beta)\vec{a} = \alpha\vec{a} + \beta\vec{a}$, [distributivnost s obzirom na skalarni faktor]
7. $\alpha(\vec{a} + \vec{b}) = \alpha\vec{a} + \alpha\vec{b}$, [distributivnost s obzirom na vektorski faktor]
8. $1 \cdot \vec{a} = \vec{a}$. [postojanje jediničnog elementa za množenje]

U nastavku ćemo reći nešto o elementarnim operacijama s vektorima. Formula za duljinu vektora $\vec{a} = (a_1, a_2, a_3)$ gdje je $(\vec{i}, \vec{j}, \vec{k})$ ortonormirana baza dana je izrazom:

$$\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2}. \tag{2.3}$$

Primjer 2.2. *Neka je $\vec{a} = 3\vec{i} - 6\vec{j} + 2\vec{k}$. Duljina vektora $\vec{a}$ dobije se koristeći formulu:*

$$|\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2} = \sqrt{3^2 + (-6)^2 + 2^2} = \sqrt{49} = 7.$$

Skalarni produkt dvaju 3-dimenzionalnih vektora $\vec{a} = (a_1, a_2, a_3)$ i $\vec{b} = (b_1, b_2, b_3)$ gdje je $(\vec{i}, \vec{j}, \vec{k})$ ortonormirana baza definiran je kao suma umnožaka parova odgovarajućih koordinata (odnosno komponenata) obaju vektora. Dakle:

$$\vec{a} \cdot \vec{b} = (a_1, a_2, a_3) \cdot (b_1, b_2, b_3) = (a_1 \cdot b_1, a_2 \cdot b_2, a_3 \cdot b_3). \tag{2.4}$$

Navedimo dva svojstva skalarnog produkta:

- Dva vektora su međusobno ortogonalna (tj. okomita) ako i samo ako im je skalarni produkt jednak nuli, tj.
$$\vec{a} \cdot \vec{b} = 0 \iff \vec{a} \perp \vec{b}.$$
- Skalarni produkt dvaju vektora $\vec{a}$ i $\vec{b}$ računa se pomoću kosinusa kuta koji oni zatvaraju, tj.
$$cos(\angle(\vec{a}, \vec{b})) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||}.$$

Primjer 2.3. *Neka je* $\vec{a} = (2, 4, 8)$ *vektor u* V^3. *Vektor* $\vec{b}$ *okomit vektoru* $\vec{a}$ *dobijemo rješavajući jednadžbu*

$$(2, 4, 8) \cdot (b_1, b_2, b_3) = 0.$$

Slijedi $2b_1 + 4b_2 + 8b_3 = 0$. *Uočimo da takvih vektora ima beskonačno mnogo. Naime, ako fiksiramo vektor* $\vec{a}$ *u ishodište koordinatnog sustava tada će taj vektor biti okomit na ravninu koja prolazi ishodištem. Dakle, svi vektori koji se nalaze na toj ravnini bili bi rješenje (1). Dakle, dvije koordinate vektora* $\vec{b}$ *možemo izabrati proizvoljno tako da dodamo dvije jednadžbe linearno nezavisne s jednadžbom (1). Npr.*

- *za* $b_1 = 1$ *dobije se* $2 \cdot 1 + 4b_2 + 8b_3 = 0$.
- *Za* $b_2 = 2$ *dobije se* $4 \cdot 2 + 8b_3 = -2$.

Tada je $b_3 = \frac{-10}{8} = \frac{-5}{4}$. *Zaista* $(2, 4, 8) \cdot (1, 2, \frac{-5}{4}) = 0$.

2.2. Vektorski prostor

U literaturi [1] autor definira pojam vektorskoga prostora:

Definicija 2.3. *"Neka je* V *neprazan skup na kojem su zadane binarna operacija zbrajanja* $+ : V \times V \to V$ *i operacija množenja skalarima iz polja* $\mathbb{F}$, $\cdot : \mathbb{F} \times V \to V$. *Kažemo da je uređena trojka* $(V, +, \cdot)$ *vektorski prostor nad poljem* $\mathbb{F}$ *ako vrijedi:*

1. $a + (b + c) = (a + b) + c$, *za sve* $a, b, c \in V$;
2. *postoji* $0 \in V$ *sa svojstvom* $0 + a = a + 0 = a$, *za sve* $a \in V$;
3. *za svaki* $a \in V$ *postoji* $-a \in V$ *takav da je* $a + (-a) = -a + a = 0$;
4. $a + b = b + a$, *za sve* $a, b \in V$;
5. $\alpha(\beta a) = (\alpha\beta)a$, *za sve* $\alpha, \beta \in \mathbb{F}$ *i za sve* $a \in V$;
6. $(\alpha + \beta)a = \alpha a + \beta a$, *za sve* $\alpha, \beta \in \mathbb{F}$ *i za sve* $a \in V$;
7. $\alpha(a + b) = \alpha a + \alpha b$, *za sve* $\alpha \in \mathbb{F}$ *i za sve* $a, b \in V$;
8. $1 \cdot a = a$, *za sve* $a \in V$*."*

Također, autor iz literature [1] napominje pojam skalara:

Napomena 2.8. *"Iz zahtjeva da na skupu V bude definirano i množenje vektora elementima polja $\mathbb{F}$ proizlazi fraza da je V vektorski prostor nad $\mathbb{F}$. Primijetimo da $\mathbb{F}$ može biti bilo koje polje. Elemente polja $\mathbb{F}$ zovemo skalari."*

Primjedba 2.3. *$V^2(O), V^3(O), V^3$ su primjeri vektorskih prostora.*

Definicija 2.4. *Neka je V vektorski prostor, te $\vec{v_1}, \vec{v_2}, ...\vec{v_n}$ vektori. Baza B je skup vektora $B = \{\vec{v_1}, \vec{v_2}, ...\vec{v_n}\} \subseteq V$ ako vrijedi:*

1. *vektori $\vec{v_1}, \vec{v_2}, ...\vec{v_n}$ su linearno nezavisni;*

2. *svaki drugi vektor $v \in V$ se može zapisati kao linearna kombinacija vektora $\vec{v_1}, \vec{v_2}, ...\vec{v_n}$.*

Definicija 2.5. *Baza B je ortonormirana ako je ortogonalna ($\vec{e_i} \perp \vec{e_j}$, za svaki $i \neq j$) i ako vrijedi $||\vec{e_i}|| = 1$, za svaki $i, ..., n$.*

Definicija 2.6. *Neka je V vektorski prostor. Dimenzija prostora V ($dimV$) maksimalan je mogući broj linearno-nezavisnih vektora u tome prostoru.*

U literaturi [11] autorica postavlja teorem dimenzije prostora V:

Teorem 2.1. *"Neka je V vektorski prostor. Svake dvije baze vektorskog prostora V imaju isti broj elemenata. Taj broj je jednak dimenziji prostora V. "*

Uz pojam dimenzije, autor u literaturi [1] definira pojam komplanarnosti vektora:

Definicija 2.7. *"Neka je $\{\vec{v_1}, \vec{v_2}, \dots \vec{v_n}\}, n \geq 2$, konačan skup radijvektora u $V^3(O)$, te neka je $\vec{v_i} = \overrightarrow{OT_i}, i = 1, \dots, n$. Kažemo da su radijvektori $\vec{v_1}, \vec{v_2}, \dots, \vec{v_n}$ komplanarni ako točke $O, T_1, T_2, \dots, T_n$ leže u istoj ravnini. U suprotnom kažemo da su radijvektori $\vec{v_1}, \vec{v_2}, \dots, \vec{v_n}$ nekomplanarni."*

Svaka su dva radijvektora komplanarna. Svaka tri radijvektora mogu i ne moraju biti komplanarni. Autor u literaturi [1] postavlja teorem 2.2. i definira bazu prostora $V^3(O)$:

Teorem 2.2. *"Neka su $\vec{a}, \vec{b}, \vec{c} \in V^3(O)$ nekomplanarni. Za svaki radijvektor $\vec{v} \in V^3(O)$ postoje jedinstveni skalari α, β, γ takvi da vrijedi $\vec{v} = \alpha\vec{a} + \beta\vec{b} + \gamma\vec{c}$."*

Definicija 2.8. *"Svaki skup $\{\vec{a}, \vec{b}, \vec{c}\}$ od tri nekomplanarna radijvektora naziva se baza prostora $V^3(O)$."*

Uzmemo li proizvoljno odabranu bazu prostora, tada ćemo svaki radijvektor iz $V^3(O)$ prikazati kao linearnu kombinaciju elemenata baze. Svojstvo iz teorema 2.2. se odnosi isključivo na tročlane skupove nekomplanarnih radijvektora. $V^3(O)$ je pojednostavljenje prostora V^3 pa sve navedeno vrijedi i na V^3. Odluka da prethodne definicije i teorem iskažemo u terminima $V^3(O)$ bila je motivirana željom da razmatranja budu jednostavnija, tj. rasterećena tehničkih detalja (npr. operacija s klasama ekvivalencije).

U literaturi [10] autor definira pojam baze vektorskog prostora V^3 kako slijedi:

Definicija 2.9. *"Uređena trojka $(\vec{e_1}, \vec{e_2}, \vec{e_3})$ linearno nezavisnih vektora iz V^3 zove se baza vektorskog prostora V^3."*

Neka je $\vec{a} \in V^3$, a $(\vec{e_1}, \vec{e_2}, \vec{e_3})$ baza u V^3. Vektor $\vec{a}$ onda zapisujemo kao

$$\vec{a} = a_1\vec{e_1} + a_2\vec{e_2} + a_3\vec{e_3}.$$

a_1, a_2, a_3 nazivamo komponente vektora $\vec{a}$ u bazi $(\vec{e_1}, \vec{e_2}, \vec{e_3})$." Iz toga nam slijede pravila za zbrajanje vektora i množenje vektora sa skalarom ako su oni zadani sa svojim koordinatama:

$$\vec{a} + \vec{b} = (a_1 + b_1)\vec{e_1} + (a_2 + b_2)\vec{e_2} + (a_3 + b_3)\vec{e_3} \tag{2.5}$$

$$\lambda a = (\lambda a_1)\vec{e_1} + (\lambda a_2)\vec{e_2} + (\lambda a_3)\vec{e_3}. \tag{2.6}$$

U koordinatnom sustavu promatramo sljedeće tri točke :

$$E_1 := (1, 0, 0), E_2 := (0, 1, 0), E_3 := (0, 0, 1).$$

Te točke određuju tri jedinična vektora:

$$\vec{i} = \overrightarrow{OE_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \vec{j} = \overrightarrow{OE_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \vec{k} = \overrightarrow{OE_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Standardna baza za V^3 je onda

$$\{\vec{e_1}, \vec{e_2}, \vec{e_3}\} = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

2.3. Afin prostor

U literaturi [9] autori definiraju pojam Afinoga prostora kako slijedi:

Definicija 2.10. *"Neka je $\mathcal{A}$ neprazan skup, V vektorski prostor nad poljem $\mathbb{F}$, te neka je definirano preslikavanje, $v : \mathcal{A} \times \mathcal{A} \to V$. Uređena trojka $(\mathcal{A}, V, v)$ naziva se afin prostor nad V , a elementi skupa $\mathcal{A}$ točke tog prostora ako vrijedi:*

1. *za svaki $A \in \mathcal{A}$ i svaki $x \in V$ postoji jednoznačno određena točka $B \in \mathcal{A}$ takva da je $v(A, B) = x$;*

2. *za sve $A, B, C \in \mathcal{A}$ vrijedi $v(A, B) + v(B, C) = v(A, C)$ (Charlesova jednakost). [3]*

Ako je $(\mathcal{A}, V, v)$ afini prostor, tada se vektorski prostor V naziva pripadni vektorski prostor ili smjer afinog prostora $\mathcal{A}$."

Napomena 2.9. *Elementi afinog prostora su točke.*

"Preslikavanje $v : \mathcal{A} \times \mathcal{A} \to V$ svakom uređenom paru točaka (A, B) pridružuje vektor $v(A, B)$." napominju autori iz literature [9]. Vektori se pak zbrajaju prema pravilu 2) iz definicije afinog prostora i takvo pravilo nazivamo pravilom trokuta. Autori u literaturi [9] povlače poveznicu između vektorskoga i afinoga prostora kako slijedi u definiciji:

Definicija 2.11. *"Afini prostor je realan ako je pridruženi vektorski prostor realan.*

Afini prostor je konačno-dimenzionalan ako je pridruženi vektorski prostor konačno-dimenzionalan.

Dimenzija afinog prostora jednaka je dimenziji pripadnog vektorskog prostora."

Bilo koji vektorski prostor ujedno možemo tumačiti kao afin prostor nad samim sobom. Tada je dovoljno da definiramo skup točaka jednak skupu V, dok ćemo preslikavanje $v : V \times V \to V$ definirati sa

$$v(x, y) = y - x, x, y \in V.$$

U tome slučaju za svaki $x \in V$ te svaki $a \in V$ postoji takav $y \in V$ da vrijedi $v(x, y) = a$. Možemo zaključiti da je $y = x + a$ jer je $v(x, y) = y - x = (x + a) - x = a$.

Napomena 2.10. *Zbrajanjem točke $a \in \mathcal{A}$ i vektora $v \in V$ je ponovno točka, tj. za $a \in \mathcal{A}$ i $v \in V$ vrijedi $a + v = b \in \mathcal{A}$.*

Udaljenost između točaka računa se kao duljina vektora njihove razlike, tj. $||v|| = a - b$, gdje su $a, b \in \mathcal{A}$ i $v \in V$.

Autori u literaturi [9] definiraju pojam afinog koordinatnog sustava:

Definicija 2.12. *"Neka je $\mathcal{A}_n$ afini prostor, $O \in \mathcal{A}_n$ bilo koja točka, te neka je $B = \{e_1, e_2, \ldots, e_n\}$ bilo koja baza tom afinom prostoru pridruženog vektorskog prostora V_n nad poljem F. Uređena $(n + 1)$-torka*

$$(O; e_1, e_2, \ldots, e_n)$$

naziva se afini koordinatni sustav."

Točka O je ishodište, a $e_1, e_2, \ldots, e_n$ su koordinatni vektori tog afinog koordinatnog sustava. Točke $E_1, E_2, \ldots, E_n$ za koje vrijedi $\overrightarrow{OE_i} = e_i, i = 1, 2, \ldots, n$, nazivaju se jedinične točke. Točke $O, E_1, E_2, \ldots, E_n$ u potpunosti određuju koordinatni sustav, a osnovno je svojstvo da je skup točaka $\{O, E_1, E_2, \ldots, E_n\}$ linearno nezavisan. Neka je $\mathcal{A}$ afini prostor dimenzije 3, O bilo koja točka iz tog afinog prostora, te neka su $\vec{e1}, \vec{e2}, \vec{e3}$ koordinatni vektori tog afinog koordinatnog sustava. Znači $\{\vec{e1}, \vec{e2}, \vec{e3}\}$ je bilo koja baza tom afinom prostoru pridruženog vektorskog prostora. Uređena četvorka $\{o, \vec{e1}, \vec{e2}, \vec{e3}\}$ naziva se afini koordinatni sustav.

Napomena 2.11. *Smjer u točki $a \in \mathcal{A}$ dobije se iz izraza $a - O \in V$.*

Nadalje, neka je u $\mathcal{A}^n$ dan skup koji sadrži $n + 1$ točku $a_0, a_1, \ldots, a_n$. Tada $(n + 1)$ točka tvori bazu prostora $\mathcal{A}^n$, ako niti jedna od njih ne može biti prikazana kao linearna kombinacija preostalih. Autor u literaturi [9] je definirao pojam afine ljuske:

Definicija 2.13. *"Afina kombinacija točaka $a_0, a_1, \ldots, a_n \in S \subseteq \mathbb{R}^n$ je svaka točka oblika*

$$a = \sum_{i=0}^{n} \alpha_i a_i = a_0 + \sum_{i=1}^{n} \alpha_i (a_i - a_0) = O + \sum_{i=1}^{n} \alpha_i e_i,$$

gdje su $\alpha_i \in \mathbb{R}, i = 0, \ldots, n$, *takvi da je* $\sum_{i=0}^{n} \alpha_i = 1$. *Skup*

$$aff(S) = \{\alpha_0 a_0 + \cdots + \alpha_n a_n : a_1, \ldots, a_n \in S, \alpha_1, \ldots, \alpha_n \in \mathbb{R}, \sum_{i=1}^{n} \alpha_i = 1\},$$

naziva se afina ljuska skupa S.”

Zapravo je afina ljuska skupa S najmanji afini skup koji sadrži S, tj. ako je D afin skup koji zadovoljava $S \subseteq D$, tada je $aff(S) \subseteq D$. Pod afinim skupom mislimo na vektorski prostor. Zapravo, afina ljuska su dvije različite točke kroz koje prolazi pravac. Drugim riječima, to je skup od 3 točke u ravnini koje nisu kolinearne.

2.4. Homogene koordinate

Translacija se promatra kao zbrajanje, a skaliranje i rotacija kao množenje. Ideja je biti u mogućnosti promatrati sve tri transformacije na dosljedan način, tako da se mogu lako kombinirati. Korištenje homogenih koordinata omogućuje promatranje sve tri transformacije kao množenje. Za točke izražene u homogenim koordinatama, sve tri transformacije (translacija, skaliranje i rotacija) se mogu prikazati kao množenja matrica i vektora. Također, uvođenjem homogenih koordinata izbjegavamo problem dijeljenja s nulom.

Ako je neka od koordinata jednaka beskonačno, to znači da se može dobiti dijeljenjem s nulom. Da bi se ovo izbjeglo, potrebno je sve koordinate napisati u obliku razlomka. Ako je točka u radnom prostoru zadana kao:

$$T_P = (T_{P_1}, T_{P_2}, \ldots, T_{P_n})$$

tada ta točka u homogenom prostoru ima zapis:

$$T_{Ph} = (T_{Ph_1}, T_{Ph_2}, \ldots, T_{Ph_n})$$

pri čemu je definirana veza:

$$T_{P_i} = \frac{T_{Ph_i}}{h}$$

odnosno

$$T_{Ph_i} = T_{P_i} \cdot h.$$

Dodajemo treću koordinatu

$$(x, y) \longrightarrow (x, y, w).$$

(x, y, w) i (x', y', w') reprezentiraju istu točku ako i samo ako za $t \neq 0$

$$(x', y', w') = (t \cdot x, t \cdot y, t \cdot w)$$

ili

$$(x, y, w) = (\frac{1}{t} \cdot x', \frac{1}{t} \cdot y', \frac{1}{t} \cdot w')$$

Slijedi definiranje kartezijvih koordinata homogenih točaka i pravca u 3D prostoru autorice u literaturi [3]: ”Ako je $w \neq 0$ tada (x, y, w) reprezentira istu točku kao i $(\frac{x}{w}, \frac{y}{w}, 1)$, gdje $\frac{x}{w}$ i $\frac{y}{w}$ nazivamo kartezijevim koordinatama homogene točke. One točke za koje je $w = 0$ kažemo da su točke u beskonačnosti. Sve trojke (tx, ty, tw), $t \neq 0$ koje reprezentiraju istu točku, čine pravac u 3D prostoru.”

3. KORIŠTENE TEHNOLOGIJE

U ovome poglavlju kratko ćemo opisati potrebne alate za izradu modela, te primjene transformacija na njih.

3.1. C++ programski jezik

C++ je objektno-orijentiran programski jezik široke namjene. Nastao je kao nadogradnja originalnom jeziku C, pa tako i nasljeđuje sintaksu C-a. Danas se koristi prilikom razvoja raznih operacijskih sustava, video igara, grafičkih sučelja i mnogih drugih. Programerima omogućava visoku razinu kontrole nad resursima sustava i memorijom. Glavne značajke su polimorfizam, apstrakcija, enkapsulacija i nasljeđivanje. Razumijevanje ovih koncepata predstavlja temelj razvoja programske podrške, te omogućuje prebacivanje između sličnih objektno-orijentiranih jezika. Ispod možete vidjeti primjer jednostavne funkcije koja održava omjer širine i visine okvira za prikaz.

```
void reshapeScene(GLint width, GLint height)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    int w = height * ASPECT;
    int left = (width - w) / 2;
    glViewport(left, 0, w, height);
    glutPostRedisplay();
}
```

Programski kôd 3.1 : *Funkcija za održavanja omjera širine i visine okvira za prikaz*

3.2. OpenGL i GLUT

OpenGL je grafičko okruženje korišteno prilikom razvoja 2D i 3D aplikacija. Najčešće tu govorimo o industriji video igara, virtualne stvarnosti, arhitekture, modeliranja i sl. Biblioteke OpenGL-a su napisane u C programskome jeziku i dopuštaju izvedbe u mnogim drugim jezicima. Također je dostupan na velikoj većini operacijskih sustava.

Nekolicina biblioteka je osmišljena s idejom izrade prozora koji će podržavati OpenGL, prva je bila GLUT. Ugrađene funkcije unutar GLUT biblioteke uključuju definiciju prozora, kontrolu prozora te nadzor unosa s tipkovnice i miša. Upravo te funkcije nam daju rutine za crtanje brojnih geometrijskih likova i tijela, iz kojih dalje možemo izvoditi složenije oblike. Sve GLUT funkcije započinju prefiksom glut.

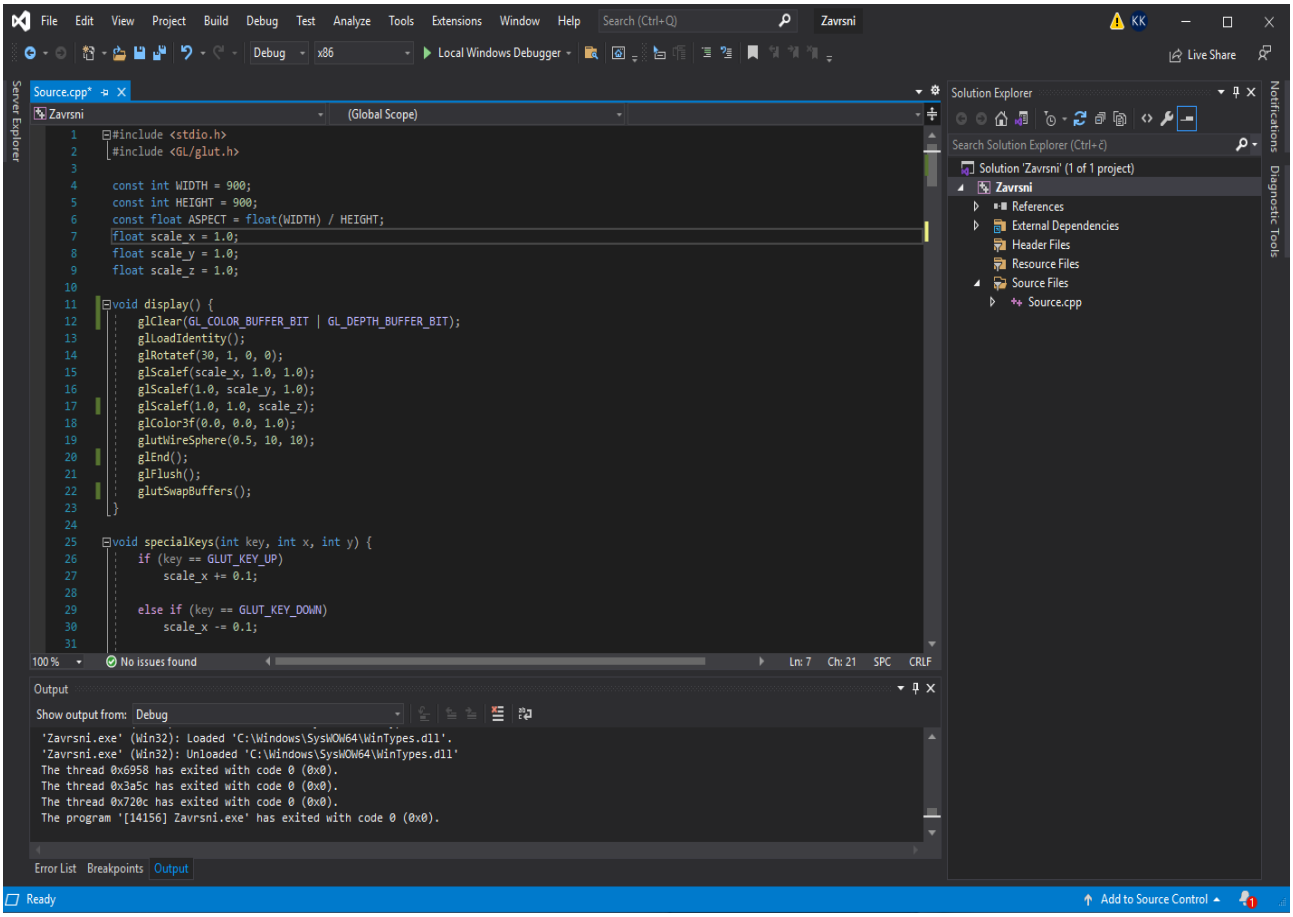
```
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(WIDTH, HEIGHT);
    glutCreateWindow("Kugla");
    glutDisplayFunc(display);
```

```
5     glutReshapeFunc(reshapeScene);
6     glutSpecialFunc(specialKeys);
```

Programski kôd 3.2 : *Neke ugrađene GLUT funkcije*

3.3. Microsoft Visual Studio

Microsoft Visual Studio je integrirano razvojno okruženje (IDE) korišteno za prilikom razvoja aplikacija, web stranica, web aplikacija te različitih usluga. Podržava 36 različitih programskih jezika i omogućuje korisniku i debuggeru (program za ispravak pogrešaka) gotovo sve programske jezike, pod uvjetom da postoji usluga specifična za jezik. Instalacijom dodatnih paketa i proširenja si omogućujemo olakšan razvoj (formatiranje, organizacija i reorganizacija kôda, sortiranje i mnoge druge mogućnosti).



Slika 3.1: *Korisničko sučelje Microsoft Visual Studia*

4. OSNOVNE TRANSFORMACIJE

U ovome odlomku ćemo objasniti korištene geometrijske transformacije najprije na 2D primjerima, pa proširiti na 3D kako bi mogli ih kasnije primijeniti na zadana geometrijska tijela.

4.1. 3D transformacije

Prikazivanje 3D transformacije je pomoću 4x4 matrice u homogenim koordinatama:

$$(x, y, z) \longrightarrow (x, y, z, w)$$

Napomena 4.1. *Nije dozvoljeno (0,0,0,0).*

Napomena 4.2. *U računalnoj grafici je $w = 1$.*

Svaka se točka u 3D prostoru može prikazati pravcem kroz ishodište 4D prostora. Redoslijed djelovanja transformacija važan je za konačan rezultat, osim ako se ne radi o uzastopnom djelovanju iste transformacije kada je redoslijed nebitan.

4.2. Translacija

Točka $P(x, y)$ se translatira u točku $P'(x', y')$ za dx po x-osi i za dy po y-osi pri čemu vrijedi

$$x' = x + dx, y' = y + dy \tag{4.1}$$

Tada zapisujemo izvornu točku kao $P = \begin{bmatrix} x & y \end{bmatrix}$, a novonastalu kao $P' = \begin{bmatrix} x' & y' \end{bmatrix}$, pri čemu je vektor translacije $T = \begin{bmatrix} dx & dy \end{bmatrix}$. Stoga nam vrijedi da je $P' = P + T$. Translacija pomoću homogenih koordinata dana je izrazom

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{4.2}$$

Znamo da je $P' = T(dx, dy) \cdot P$, gdje je $T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$.
Za $P' = T(dx_1, dx_2) \cdot P$ i $P'' = T(dx_2, dy_2) \cdot P'$ dobivamo sljedeće

$$P'' = T(dx_2, dy_2) \cdot T(dx_1, dy_1) \cdot P.$$

Nadalje, vrijedi

$$\begin{bmatrix} 1 & 0 & dx_2 \\ 0 & 1 & dy_2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & dx_1 \\ 0 & 1 & dy_1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx_1 + dx_2 \\ 0 & 1 & dy_1 + dy_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(dx_2, dy_2) * T(dx_1, dy_1) = T(dx_1 + dx_2, dy_1 + dy_2).$$

Analognim postupkom izvodi se izraz za matricu 3D translacije zadanu homogenim koordinatama

$$T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$T(dx, dy, dz) \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + dx \\ y + dy \\ z + dz \\ 1 \end{bmatrix}. \tag{4.4}$$

Određivanjem inverza dobije se matrica za negativne pomake

$$T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.5}$$

4.3. Skaliranje

Skaliranja se izvodi za promjenu veličine objekta čija se dimenzija može promijeniti(skalirati) u bilo kojem smjeru x, y, z kroz s_x, s_y, s_z faktore skaliranja. Skaliranjem svaku koordinatu množimo odgovarajućim koeficijentom (rastežemo ili stežemo).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}, P' = S \cdot P. \tag{4.6}$$

Prelaskom u homogene koordinate dobijemo:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, P' = S(s_x, s_y) \cdot P,$$

gdje je

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.7}$$

Za $P' = S(s_{x_1}, s_{x_2}) \cdot P$ i $P'' = S(s_{x_2}, s_{y_2}) \cdot P'$ dobivamo sljedeće

$$P'' = S(s_{x_2}, s_{y_2}) \cdot S(s_{x_1}, s_{y_1}) \cdot P,$$

gdje vrijedi

$$\begin{bmatrix} s_{x_2} & 0 & 0 \\ 0 & s_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{x_1} & 0 & 0 \\ 0 & s_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x_1} \cdot s_{x_2} & 0 & 0 \\ 0 & s_{y_1} \cdot s_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S(s_{x_2}, s_{y_2}) \cdot S(s_{x_1}, s_{y_1}) = S(s_{x_1} \cdot s_{x_2}, s_{y_1} \cdot s_{y_2}).$$

Kao i na primjeru translacije, analogno izvodimo matricu skaliranja

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

$$S(s_x, s_y, s_z) \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x \cdot x \\ s_y \cdot y \\ s_z \cdot z \\ 1 \end{bmatrix}. \tag{4.9}$$

Odredimo li inverz, dobit ćemo matricu skaliranja s recipročnim vrijednostima:

$$S(s_x, s_y, s_z) = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 & 0 \\ 0 & 0 & \frac{1}{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.10}$$

4.4. Rotacija

Točku $P(x, y)$ rotiramo za kut θ oko ishodišta. Za razliku od translacije, rotacija se odnosi na kretanje krutoga tijela koje zadržava točku nepokretnom.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \cdot cos(\theta) - y \cdot sin(\theta) \\ x \cdot sin(\theta) + y \cdot cos(\theta) \end{bmatrix}, \tag{4.11}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}, P' = R \cdot P.$$

Prikaz u homogenim koordinatama dan je izrazom

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{4.12}$$

$P' = R(\theta) \cdot P$, gdje je

$$R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Za $P' = R(\theta_1) \cdot PiP'' = R(\theta_2) \cdot P'$ je $P'' = R(\theta_2) \cdot R(\theta_1) \cdot P$ te vrijedi

$$\begin{bmatrix} cos(\theta_2) & -sin(\theta_2) & 0 \\ sin(\theta_2) & cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 \\ sin(\theta_1) & cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cos(\theta_1 + \theta_2) & -sin(\theta_1 + \theta_2) & 0 \\ sin(\theta_1 + \theta_2) & cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(\theta_2) \cdot R(\theta_1) = R(\theta_1 + \theta_2).$$

Ponavljamo postupak kako bi odredili matricu rotacije u 3D prikazu. Tada razlikujemo sljedeće rotacije:

- rotacija oko osi x,
- rotacija oko osi y,
- rotacija oko osi z.

4.4.1 Rotacija oko osi x

Gledajući iz pozitivnog smjera osi u ishodište koordinatnog sustava, rotacija oko osi x rotira točku u $y - z$ ravnini pri čemu x-koordinata točke se ne mijenja. Matrica rotacije oko osi x, za smjer obrnut od kazaljke na satu je

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) & 0 \\ 0 & -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.13}$$

Odnosno za smjer kazaljke na satu odredimo inverznu matricu

$$R'_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.14}$$

4.4.2 Rotacija oko osi y

Gledajući iz pozitivnog smjera osi u ishodište koordinatnog sustava, rotacija oko osi y rotira točku u $x-z$ ravnini pri čemu y-koordinata točke se ne mijenja. Nadalje, matrica rotacije oko osi y za smjer obrnut od kazaljke na satu je

$$R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.15}$$

U smjeru kazaljke na satu odredimo inverznu matricu

$$R'_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.16}$$

4.4.3 Rotacija oko osi z

Gledajući iz pozitivnog smjera osi u ishodište koordinatnog sustava, rotacija oko osi z rotira točku u $x-y$ ravnini pri čemu z-koordinata točke se ne mijenja. Dakle, matrica rotacije oko osi z u smjeru suprotnome od kazaljke na satu je

$$R_z(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 & 0 \\ -sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.17}$$

U smjeru kazaljke na satu odredimo inverznu matricu

$$R'_z(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.18}$$

5. REALIZACIJA PROGRAMA

U ovom poglavlju ćemo izlistati i objasniti sve korištene funkcije prilikom izrade i transformiranja 3D modela. Prije nego krenemo rukovati funkcijama, važno je uključiti biblioteke i definirati veličinu prozora te njegov omjer.

```
#include <stdio.h>
#include <GL/glut.h>

const int WIDTH = 900;
const int HEIGHT = 900;
const float ASPECT = float(WIDTH) / HEIGHT;
```

Programski kôd 5.1 : *Definiranje veličine prozora za prikaz*

5.1. Funkcije

Najprije definiramo funkciju *reshapeScene* koja će primati parametre visine i širine prozora koji se otvara svaki put kada se realizira objekt (funkcija je izlistana u programskome kôdu 1). Želimo održati omjer širine i visine prozora za prikaz.

U *main* funkciji se nalazi najvažniji dio programa, ona ima prioritet prilikom pokretanja programa. U njoj ćemo pozvati ugrađene funkcije te sve ranije definirane funkcije.

```
int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(WIDTH, HEIGHT);
    glutInitWindowPosition((glutGet(GLUT_SCREEN_WIDTH) - WIDTH) / 2, (
   glutGet(GLUT_SCREEN_HEIGHT) - HEIGHT) / 2);
    glutCreateWindow("Kugla");
    glEnable(GL_DEPTH_TEST);
    glutDisplayFunc(display);
    glutReshapeFunc(reshapeScene);
    glutSpecialFunc(specialKeys);
    glutMainLoop();
    return 0;
}
```

Programski kôd 5.2 : *Main funkcija*

- $glutInit(\&\ argc, argv)$ - koristimo za inicijalizaciju GLUT biblioteke.

- *glutInitDisplayMode*($GLUT_DOUBLE \mid GLUT_RGB \mid GLUT_DEPTH$) - postavlja početni način prikaza. GLUT_RGB predstavlja bit masku za odabir prozora načina rada RGB, GLUT_DOUBLE je bit maska za odabir prozora s dvostrukim međuspremnikom, a GLUT_DEPTH bit maska za odabir prozora s međuspremnikom dubine.
- *glutInitWindowSize* - postavljanje veličine prozora. Postavljeno je na 900x900 rezoluciju.
- *glutInitWindowPosition* - postavljanje početnoga položaja prozora. Optimizirano na osnovu svake rezolucije koju korisnik unese.
- *glutCreateWindow* - otvaranje novoga prozora i pridruživanje imena.
- *glEnable(GL _DEPTH _TEST)* - omogućavanje GL mogućnosti. Ako je GL _DEPTH _TEST omogućeno, napravi se usporedbe dubina i ažurira međuspremnik dubine.
- *glutDisplayFunc(display)* - postavlja povratni poziv na zaslonu za trenutni prozor. Parametar funkcije (*display()*) se razlikuje svaki puta za novo geometrijsko tijelo.
- *glutReshapeScene(reshapeScene)* - postavlja promjenu oblika povratnog poziva za trenutni prozor. Parametar koji poprima je ranije opisana funkcija *reshapeScene()* (programski kôd 1).
- *glutSpecialFunc(specialKeys)* - postavlja poseban povratni poziv tipkovnice za trenutni prozor (drugim riječima pritiskom određene tipke tipkovnice pozivamo funkciju *specialKeys()*).
- *glutMainLoop()* - označava ulaz u petlju obrade svih GLUT događaja.

Također definiramo neke ranije korištene funkcije te nove kojima ćemo se koristiti prilikom realizacije modela:

- *glClear*($GL_COLOR_BUFFER_BIT \mid GL_COLOR_DEPTH_BIT$) - očistiti međuspremnike prema unaprijed postavljenim vrijednostima. $GL_COLOR_BUFFER_BIT$ označava međuspremnike koji su trenutno omogućeni za pisanje u boji, dok $GL_COLOR_DEPTH_BIT$ označava međuspremnik dubine.
- *glLoadIdentity()* - zamjenjuje trenutnu matricu jediničnom.
- *glRotatef()* - pomnoži trenutnu matricu s matricom rotacije, a prima *float* vrijednosti brojeva(koordinate vektora translacije i kut).
- *glScalef()* - pomnoži trenutnu matricu matricom skaliranja, a prima *float* vrijednosti brojeva(koordinate vektora translacije i kut).
- *glColor3f()* - postavlja trenutnu boju, a parametri su *float* vrijednosti nijanse crvene, plave i zelene boje.
- *glutWireSphere* - stvoriti prozirnu sferu (poprima vrijednosti radijusa, broj podcjelina oko osi z i broj podcjelina na osi z).
- *glEnd()* - označava kraj iscrtavanja, svaka *glBegin()* funkcija ju mora imati.
- *glFlush()* - prisilno izvršavanje GL naredbi u konačnom vremenu.
- *glutSwapBuffers()* -zamjenjuje međuspremnike trenutnog prozora ako je međuspremnik dvostruk.

- *glBegin(GL_POLYGON)* - definicija vrhova između kojih će biti iscrtana površina. *GL_POLYGON* crta jedan mnogokut kojega definiraju vrhovi od 1 do N. *GL_TRIANGLES* tretira svaku trojku vrhova kao samostalan trokut. Vrhovi $3n-2$, $3n-1$ i $3n$ definiraju trokut n. Nacrta se $\frac{n}{3}$ trokuta.
- *glVertex3f()* - određivanje vrhova, u ovome slučaju tri koordinatne vrijednosti predstavljene *float* vrijednostima.
- *glutPostRedisplay()* - označava trenutni prozor za ponovno prikazivanje.
- *glViewPort()* - postavljanje okvira za prikaz modela.
- *glTranslatef()* - pomnoži trenutnu matricu s matricom translacije, a primljene vrijednosti (koordinate vektora translacije i kut) su tipa *float*.

5.2. Translatiranje piramide

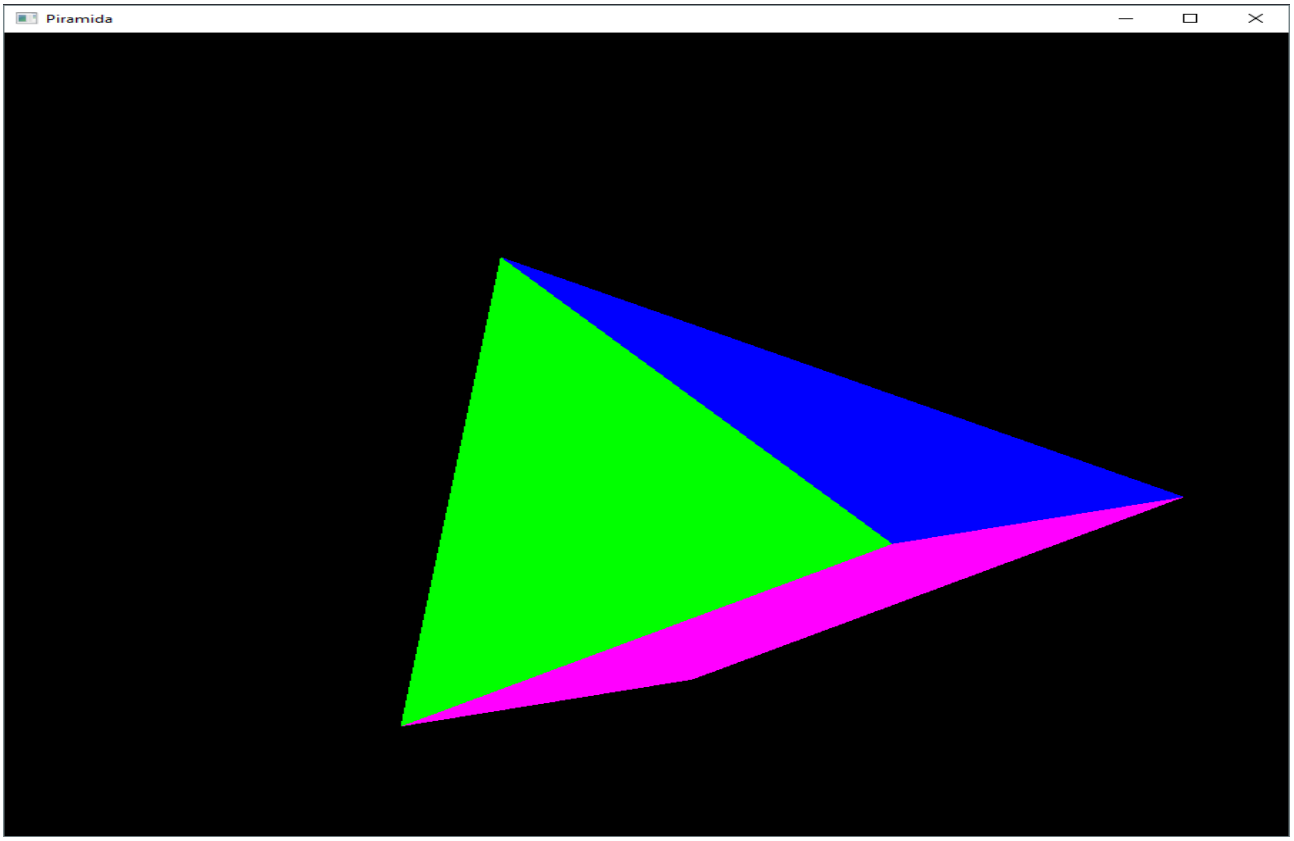
Prilikom crtanja piramide koristit ćemo se ranije opisanim funkcijama. Prije nego funkcija krene s crtanjem, moramo očistiti okvir za prikaz i međuspremnik. Zbog bolje preglednosti, rotiramo piramidu za kut u vrijednosti 40 te pomičemo po y-osi i z-osi. Piramida je geometrijsko tijelo sastavljeno od baze (u ovome slučaju pravokutnik) i stranica (trokuti). Najprije crtamo bazu zadavajući koordinate vrhova te ju bojamo ružičastom bojom. Stranice crtamo (kao i za bazu zadavajući vrhove) iznad baze jednu po jednu. Zbog boljeg uočavanja svaku stranicu bojali smo različitom bojom. Na samom kraju osiguravamo si izvršavanje naredbi te funkciju *display* pozivamo u *main* funkciji.

```
float translate_y = 0;
float translate_x = 0;
float translate_z = 0;
void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glRotatef(40,0,1,1);
    glTranslatef(translate_x,translate_y,translate_z);
    glBegin(GL_POLYGON);
        glColor3f(1.0, 0.0, 1.0);
        glVertex3f(0.5, -0.5, -0.5);
        glVertex3f(0.5, -0.5, 0.5);
        glVertex3f(-0.5, -0.5, 0.5);
        glVertex3f(-0.5, -0.5, -0.5);
    glEnd();
    glBegin(GL_TRIANGLES);
    glColor3f(1.0, 0.0, 0.0);
    glVertex3f(0.0, 0.5, 0.0);
    glVertex3f(-0.5, -0.5, 0.5);
    glVertex3f(0.5, -0.5, 0.5);
```

```
21      glColor3f(0.0, 0.0, 1.0);
22      glVertex3f(0.0, 0.5, 0.0);
23      glVertex3f(0.5, -0.5, 0.5);
24      glVertex3f(0.5, -0.5, -0.5);
25      glColor3f(0.0, 1.0, 0.0);
26      glVertex3f(0.0, 0.5, 0.0);
27      glVertex3f(0.5, -0.5, -0.5);
28      glVertex3f(-0.5, -0.5, -0.5);
29      glColor3f(1.0, 1.0, 0.0);
30      glVertex3f(0.0, 0.5, 0.0);
31      glVertex3f(-0.5, -0.5, -0.5);
32      glVertex3f(-0.5, -0.5, 0.5);
33      glEnd();
34      glFlush();
35      glutSwapBuffers();
36  }
```

Programski kôd 5.3 : *Funkcija crtanja piramide*

Prikaz piramide prije primjene translacije dan je na sljedećoj slici.



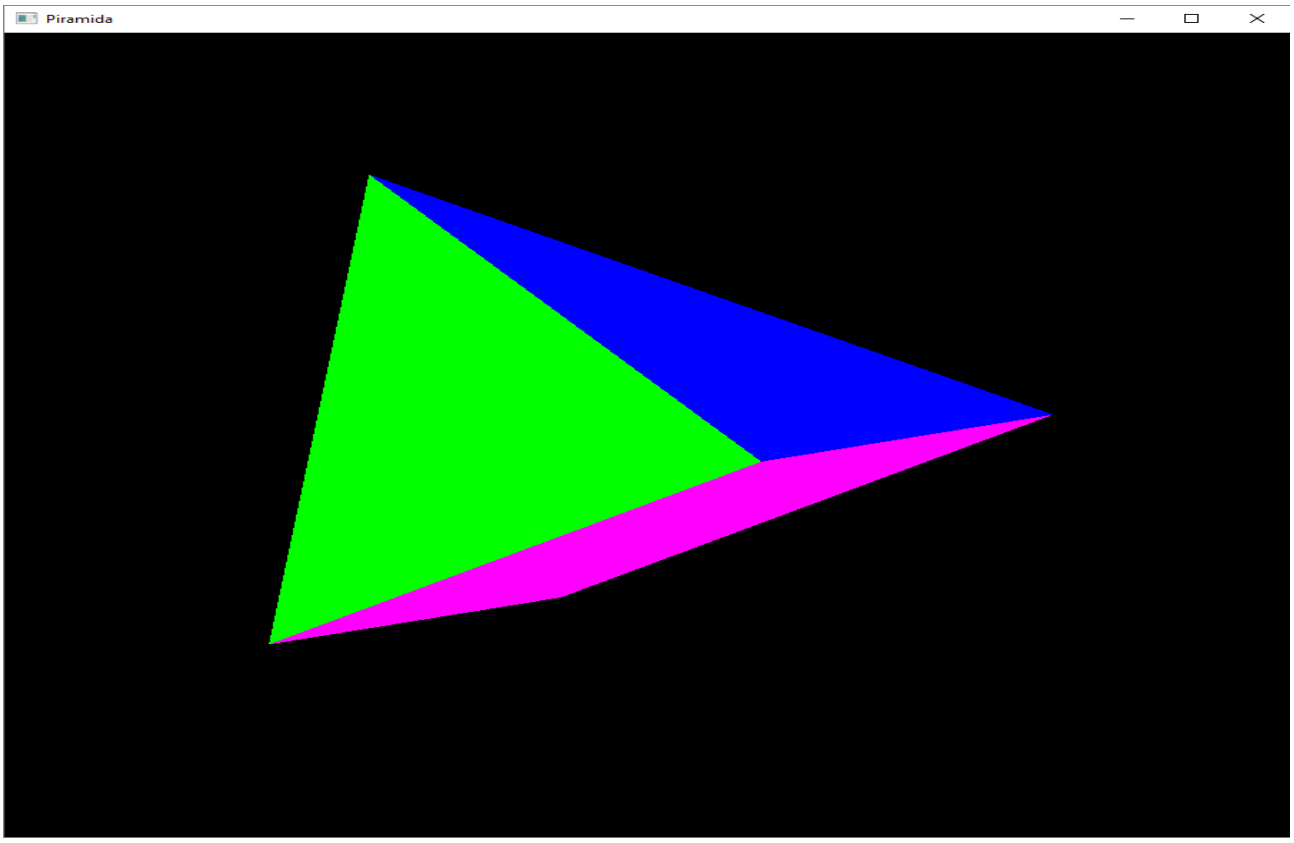
Slika 5.1: *Piramida prije translacije*

Sljedeći korak je napisati funkciju pomoću koje korisnik može translatirati piramidu (programski kôd 6). Pritiskom određenih tipki tipkovnice korisnik mijenja vrijednosti varijabli $(scale_x, scale_y, scale_z)$ te poziva funkciju *glTranslatef()* i translatira nacrtanu piramidu. Strelicama na tipkovnici korisnik može translatirati po x i y osima koordinatnog sustava, a pomoću funkcijskih tipki F1 i F2 po osi z. U ovome primjeru pomičemo objekt za 0.01 svakim pritiskom.

```
void specialKeys(int key, int x, int y) {
    if (key == GLUT_KEY_UP)
        translate_y += 0.01;
    else if (key == GLUT_KEY_DOWN)
        translate_y -= 0.01;
    else if (key == GLUT_KEY_RIGHT)
        translate_x += 0.01;
    else if (key == GLUT_KEY_LEFT)
        translate_x -= 0.01;
    else if (key == GLUT_KEY_F2)
        translate_z += 0.01;
    else if (key == GLUT_KEY_F1)
        translate_z -= 0.01;
    glutPostRedisplay();
}
```

Programski kôd 5.4 : *Funkcija translacije*

Prilikom pokretanja programa, u prozoru možemo sami povećavati ili smanjivati vrijednosti za koje želimo translatirati piramidu. Objekt translatiran za sljedeće parametre funkcije *translatef(-0.12, 0.3, 0.04)* se nalazi ispod.



Slika 5.2: *Piramida nakon translacije*

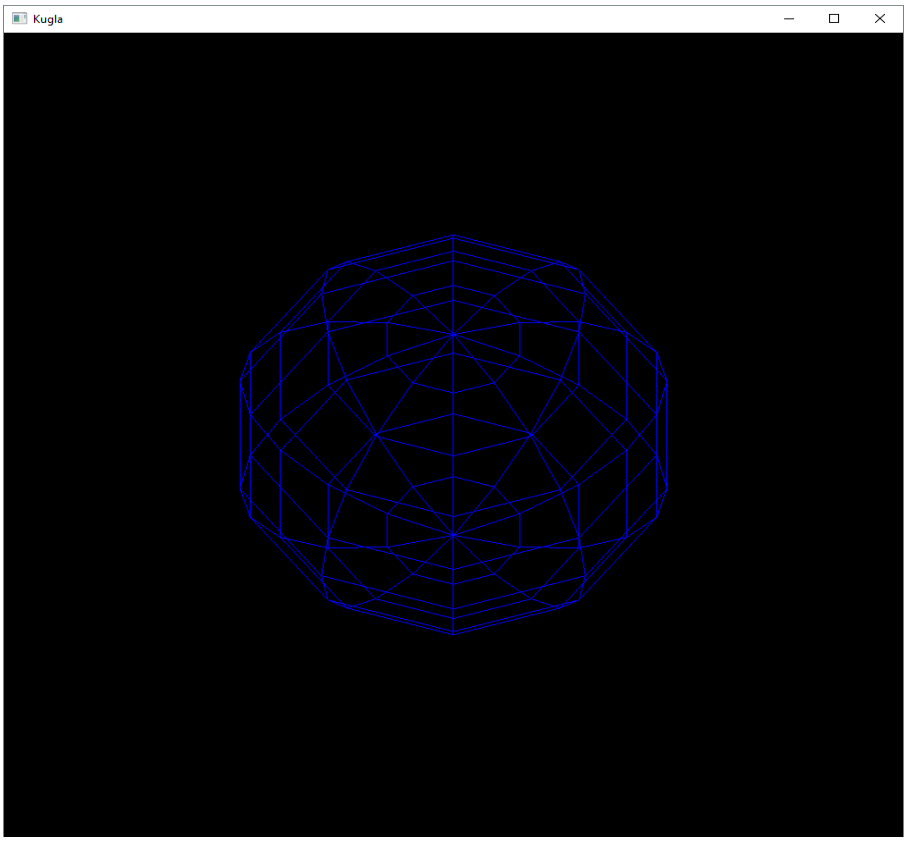
5.3. Skaliranje kugle

Prilikom crtanja kugle potrebno je ponovno definirati funkciju *display()* (programski kôd 7). Objekt ćemo rotirati za kut u vrijednosti 30° kako bismo bolje uočili trodimenzionalnost. Kuglu bojamo plavo i crtamo ju prozirnu zbog preglednosti. Zadajemo nužne parametre (radijus 0.5, te 10 podcjelina oko i na osi z). Unutar *display()* funkcije tri puta se poziva funkcija za skaliranje za svaki parametar skaliranja koji kontrolira korisnik. Naravno, nužno je osigurati izvršenje svih željenih naredbi kako ne bi došlo do smetnji prilikom izvođenja.

```
1  float scale_x = 1.0;
2  float scale_y = 1.0;
3  float scale_z = 1.0;
4  void display() {
5      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
6      glLoadIdentity();
7      glRotatef(30, 1, 0, 0);
8      glScalef(scale_x, 1.0, 1.0);
9      glScalef(1.0, scale_y, 1.0);
10     glScalef(1.0, 1.0, scale_z);
11     glColor3f(0.0, 0.0, 1.0);
12     glutWireSphere(0.5, 10, 10);
13     glEnd();
```

```
14      glFlush();
15      glutSwapBuffers();
16  }
```

Programski kôd 5.5 : *Funkcija crtanja kugle*



Slika 5.3: *Kugla prije skaliranja*

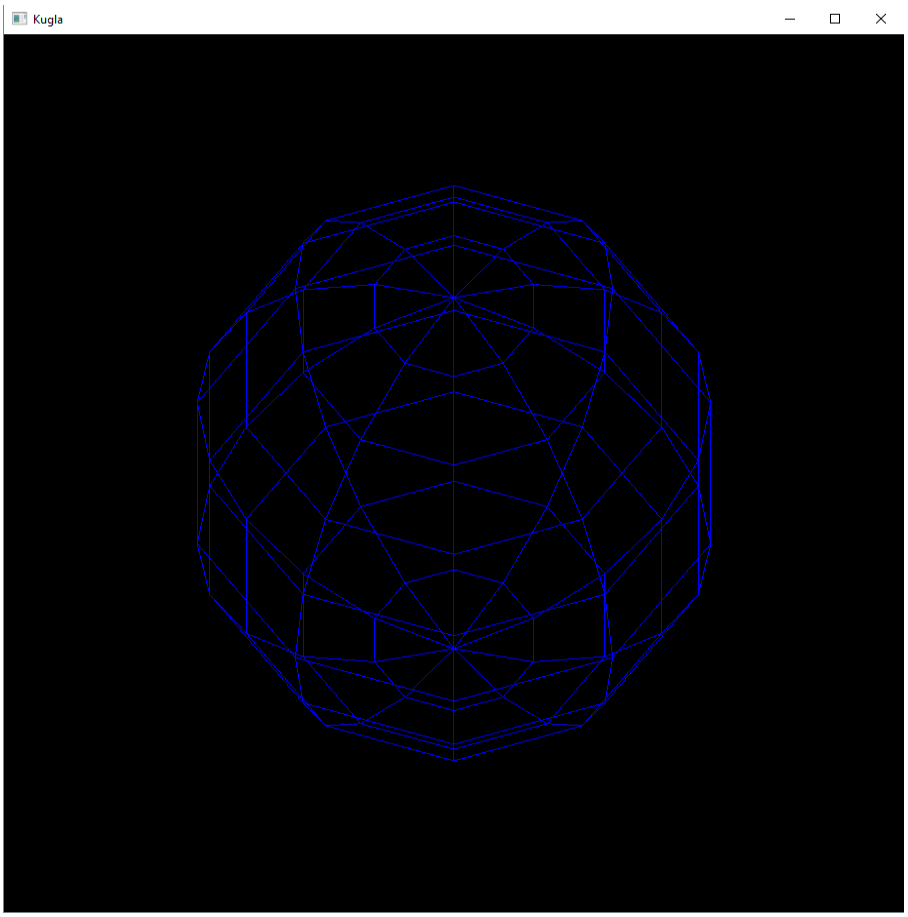
Sljedeći korak je definirati funkciju koja će rukovati promjenom parametara matrice skaliranja. Kao i na primjeru translacije, korisnik proizvoljno može mijenjati vrijednosti za koje želi skalirati kuglu po jednoj od osi koristeći se strelicama na tipkovnici i funkcijskim tipkama F1 i F2.

```
1   void specialKeys(int key, int x, int y) {
2       if (key == GLUT_KEY_UP)
3           scale_x += 0.1;
4       else if (key == GLUT_KEY_DOWN)
5           scale_x -= 0.1;
6       else if (key == GLUT_KEY_RIGHT)
7           scale_y += 0.1;
8       else if (key == GLUT_KEY_LEFT)
9           scale_y -= 0.1;
10      else if (key == GLUT_KEY_F1)
```

```
        scale_z += 0.1;
    else if (key == GLUT_KEY_F2)
        scale_z -= 0.1;
    glutPostRedisplay();
}
```

Programski kôd 5.6 : *Funkcija skaliranja*

Primjer kugle skalirane za vrijednosti $(0.2, 0.2, 0.8)$ dan je na sljedećoj slici (slika 9).



Slika 5.4: *Kugla nakon primjene skaliranja*

5.4. Rotacija kocke

Kocka je geometrijsko tijelo koje se sastoji od šest stranica (šest jednakih kvadrata). Redom crtamo stranicu po stranicu i svaku bojamo u različitu boju. Unutar funkcije *display()*je tri puta pozvana funkcija *glRotate3f()* kojom će korisnik moći rotirati kocku za različite vrijednosti varijabli $(rotate_y, rotate_x, rotate_z)$.

```
float rotate_y = 0;
float rotate_x = 0;
float rotate_z = 0;
```

```
4  void display() {
5      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
6      glLoadIdentity();
7      glRotatef(rotate_x, 1.0, 0.0, 0.0);
8      glRotatef(rotate_y, 0.0, 1.0, 0.0);
9      glRotatef(rotate_z, 0.0, 0.0, 1.0);
10     glBegin(GL_POLYGON);
11     glColor3f(1.0, 1.0, 0.0);
12     glVertex3f(0.5, -0.5, -0.5);
13     glVertex3f(0.5, 0.5, -0.5);
14     glVertex3f(-0.5, 0.5, -0.5);
15     glVertex3f(-0.5, -0.5, -0.5);
16     glEnd();
17     glBegin(GL_POLYGON);
18     glColor3f(1.0, 0.0, 1.0);
19     glVertex3f(0.5, -0.5, 0.5);
20     glVertex3f(0.5, 0.5, 0.5);
21     glVertex3f(-0.5, 0.5, 0.5);
22     glVertex3f(-0.5, -0.5, 0.5);
23     glEnd();
24     glBegin(GL_POLYGON);
25     glColor3f(1.0, 1.0, 1.0);
26     glVertex3f(0.5, -0.5, -0.5);
27     glVertex3f(0.5, 0.5, -0.5);
28     glVertex3f(0.5, 0.5, 0.5);
29     glVertex3f(0.5, -0.5, 0.5);
30     glEnd();
31     glBegin(GL_POLYGON);
32     glColor3f(0.0, 0.0, 1.0);
33     glVertex3f(-0.5, -0.5, 0.5);
34     glVertex3f(-0.5, 0.5, 0.5);
35     glVertex3f(-0.5, 0.5, -0.5);
36     glVertex3f(-0.5, -0.5, -0.5);
37     glEnd();
38     glBegin(GL_POLYGON);
39     glColor3f(1.0, 0.0, 0.0);
40     glVertex3f(0.5, 0.5, 0.5);
41     glVertex3f(0.5, 0.5, -0.5);
42     glVertex3f(-0.5, 0.5, -0.5);
43     glVertex3f(-0.5, 0.5, 0.5);
```

```
44      glEnd();
45      glBegin(GL_POLYGON);
46      glColor3f(0.0, 1.0, 0.0);
47      glVertex3f(0.5, -0.5, -0.5);
48      glVertex3f(0.5, -0.5, 0.5);
49      glVertex3f(-0.5, -0.5, 0.5);
50      glVertex3f(-0.5, -0.5, -0.5);
51      glEnd();
52      glFlush();
53      glutSwapBuffers();
54  }
```

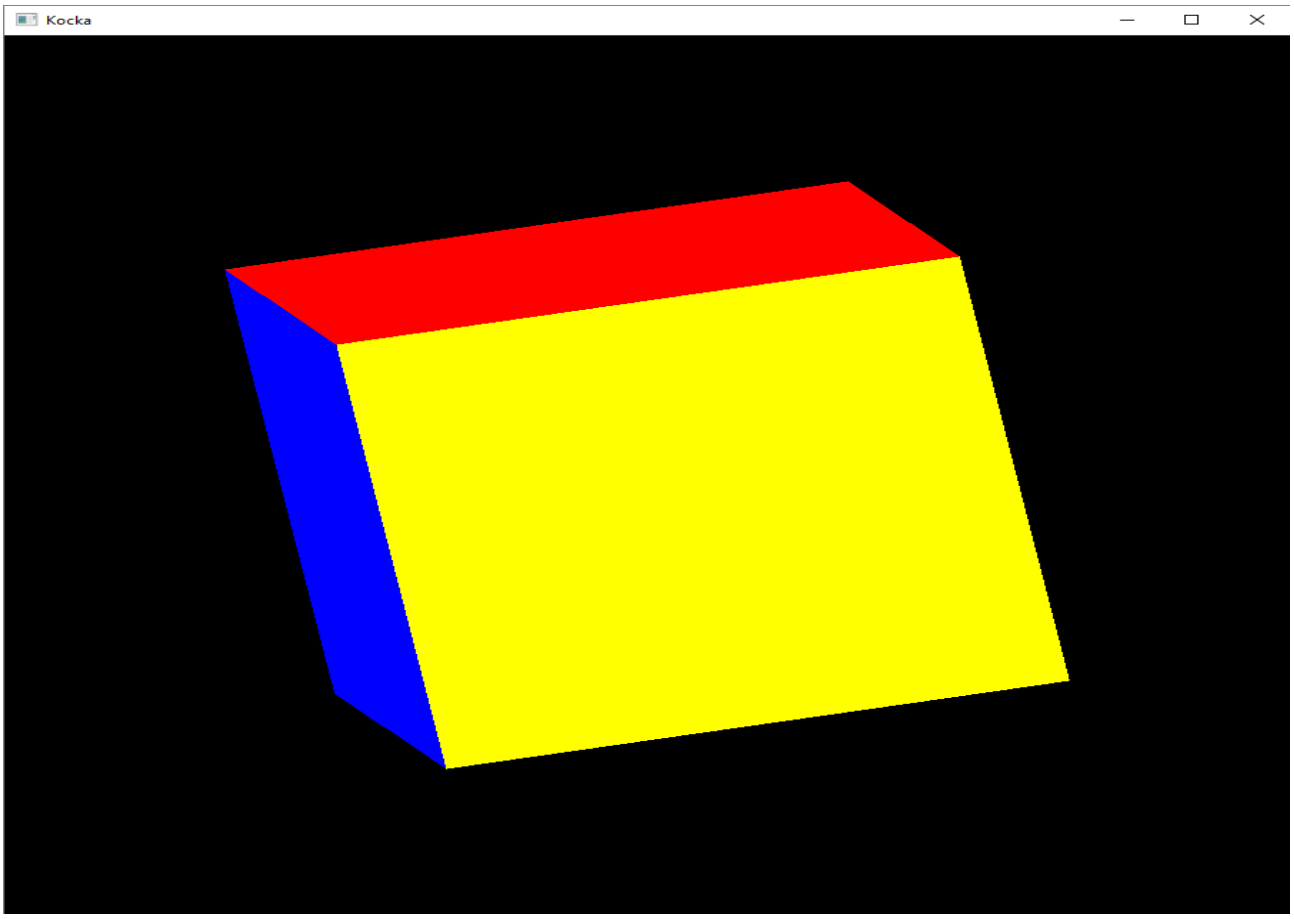
Programski kôd 5.7 : *Funkcija crtanja kocke*

Funkcija *specialKeys()* omogućuje samu rotaciju. Korisnik pritiskom strelica i funkcijskih tipki F1 i F2 mijenja parametre za koje želimo rotirati te ujedno poziva funkciju za rotaciju.

```
1  void specialKeys(int key, int x, int y) {
2      if (key == GLUT_KEY_RIGHT)
3          rotate_y += 5;
4      else if (key == GLUT_KEY_LEFT)
5          rotate_y -= 5;
6      else if (key == GLUT_KEY_UP)
7          rotate_x += 5;
8      else if (key == GLUT_KEY_DOWN)
9          rotate_x -= 5;
10     else if (key == GLUT_KEY_F1)
11         rotate_z += 5;
12     else if (key == GLUT_KEY_F2)
13         rotate_z -= 5;
14     glutPostRedisplay();
15 }
```

Programski kôd 5.8 : *Funkcija rotacije*

Primjer kocke rotirane za vrijednosti $(10, -10, 10)$ dan je na sljedećoj slici (slika 10).



Slika 5.5: *Kocka nakon rotacije*

6. ZAKLJUČAK

U ovom završnom radu pokazana je primjena osnovnih transformacija - translacije, skaliranja i rotacije na tri geometrijska tijela. Kako bi smo mogli razumjeti postupak samih transformacija, nužno je bilo definirati pojmove radijvektora, vektorskoga prostora, afinoga prostora i homogenih koordinata. Prije samoga crtanja definirali smo korištene transformacije. Prilikom definiranja funkcija potrebnih za crtanje i primjenu transformacija koristili smo programski jezik C++, za implementaciju funkcija alatima OpenGL i GLUT te razvojno okruženje Visual Studio. Uočimo da smo za realizaciju zadatka mogli koristiti i brojne druge razvojne alate kojima bi različitim postupkom programiranja i crtanja dobili slične rezultate.

Na ovim primjerima možemo primijetiti široku primjenu 3D razvojnih alata prilikom vizualizacije svakodnevnih pojava. Složeniji primjeri se koriste svakodnevno u različitim granama znanosti (građevinarstvo, arhitektura, računarstvo i brojne druge).

Intuitivna dokumentacija, besplatno preuzimanje i dostupnost materijala za izobrazbu čine OpenGL i GLUT odličnim rješenjem prilikom savladavanja osnovnih koncepata vizualizacije.

Literatura

[1] D. Bakić, *Linearna algebra* , PMF - Matematički odjel, Sveučilište u Zagrebu, 2020.

[2] A. Baumgartner, I. Galić, *Računalna grafika* [Powerpoint slajdovi], https://loomen.carnet.hr/pluginfile.php/319229/mod_resource/content/1/predavanja/01_RG_Transformacije_strucni.pdf

[3] N. Bosner, *Geometrijske transformacije* [Powerpoint slajdovi], https://web.math.pmf.unizg.hr/~nela/rgpredavanja/geometrijske_transformacije

[4] M. Čupić, Ž. Mihajlović, *Interaktivna računalna grafika kroz primjere u OpenGL-u*, 2018.

[5] I. Gusić, *Lekcije iz matematike I*, http://matematika.fkit.hr/novo/matematika%201/predavanja/Lekcije_iz_Matematike1.pdf

[6] D. Ilijević, *3D Model kuće*, završni rad, FERIT, 2019.

[7] A. Kraljević, *Linearna algebra u stvarnom životu*, Odjel za matematiku, Sveučilište u Osijeku, 2020.

[8] P. Nujić, *Konveksni skupovi*, Odjel za matematiku, Sveučilište u Osijeku, 2017.

[9] M. Polonijo, D. Crnković, T. Ban Kirigin, M. Bombardelli, Z. Franušić, R. Sušanj, *Euklidski prostori*, PMF - Matematički odjel, Sveučilište u Zagrebu, 2008.

[10] R. Scitovski, D.Marković, D. Brajković, *Linearna algebra I*, Odjel za matematiku, Sveučilište u Osijeku, 2020.

[11] J. Sedlar, *Vektorski prostori* [Powerpoint slajdovi], http://gradst.unist.hr/Portals/9/docs/katedre/Matematika/PSGG%20AGLA/T05_Vektorski%20prostori.pdf

SAŽETAK

U završnom radu razvijen je program koji iscrtava piramidu, kocku, kuglu te primjenjuje translaciju, rotaciju i skaliranje. Važno je bilo definirati koncept vektora, vektorskoga prostora, afinoga prostora i homogenih koordinata kako bi razumjeli ideju iza primjene transformacija unutar računalne grafike. Za vrijeme izrade modela korišten je programski jezik C++, OpenGL i GLUT biblioteka. OpenGL je korišten za definiranje samih oblika, a GLUT za povezivanje funkcija kojima korisnik direktno određuje vrijednosti transformacija pritiskom tipkovnice. Korisnik ima direktan uvid u same objekte te kako se mijenjaju prilikom primjena transformacija.

KLJUČNE RIJEČI

OpenGL, rotacija, skaliranje, translacija, vektorski prostor

APPLICATION OF LINEAR ALGEBRA IN COMPUTER GRAPHICS

SUMMARY

In this paper, a program was developed that draws a pyramid, a cube, a sphere and applies the transformations of translation, rotation and scaling. It was important to define the concept of vector, vector space, affine space and homogeneous coordinates in order to understand the idea behind the application of transformations within computer graphics. During the creation of models the C++programming language and the OpenGL and GLUT libraries were used. OpenGL was used to define the shapes themselves, and GLUT to connect functions by which the user directly determines the transformation values by pressing the keyboard. The user has a direct insight into the objects themselves, and how they change during the application of transformations.

KEYWORDS

OpenGL, rotation, scaling, translation, vector space

ŽIVOTOPIS

Karlo Kokanović rođen je 2. prosinca 1999. godine u Slavonskome Brodu. Živi i odrasta u Županji te je pohađao Osnovnu školu Mate Lovraka u Županji. Nakon završetka osnovne škole 2014. godine upisuje Gimnaziju Županja, prirodoslovno-matematički smjer. Srednju školu završava 2018. godine te upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, preddiplomski sveučilišni smjer računarstvo.