

Mladi za napredek Maribora 2013

30. srečanje

Opisna geometrija

Raziskovalno področje: *matematika*

Raziskovalna naloga

05.11.2013

10.11.2013

15.11.2013

Maribor, februar 2013

Mladi za napredek Maribora 2013

30. srečanje

Opisna geometrija

Raziskovalno področje: *matematika*

Raziskovalna naloga

Maribor, februar 2013

Povzetek

V času razcveta tridimenzionalne računalniške tehnologije sem v raziskovalni nalogi preučil različne tipe projekcij, ki so podlaga za nastanek opisne geometrije. Tako sem s pomočjo literature opisal in med seboj primerjal lastnosti perspektivne in različnih vzporednih projekcij (ortogonalne, aksonometričnih ter poševnih). Glavni del moje naloge, ki sem ga tudi uspešno izvedel, je bila sestava računalniškega algoritma (programa), ki dane objekte v koordinatnem prostoru prikaže na računalniškem zaslonu (projicira v želeni projekciji). Pri tem sem se opiral na pridobljeno znanje iz teoretičnega dela naloge. Algoritem, ki sem ga napisal, ni najkrajši in najhitrejši možni, kar tudi ni bil moj namen, temveč temelji na matematični osnovi geometrijskega projiciranja (uporaba projicirnih žarkov). Poleg algoritma za projiciranje sem izpeljal tudi formule za navigacijo po prostoru, ki omogočajo pogled na opazovane objekte iz poljubne smeri.

Najlepša hvala mentorici, ki me vseskozi navdušuje nad matematiko.

Kazalo

1	UVOD	9
1.1	Cilja naloge	9
1.2	Predpostavki	9
1.3	Metodologija	9
2	TEORETIČNI DEL	10
2.1	Projiciranje in tipi projekcij	10
2.1.1	Perspektivna projekcija	11
2.1.2	Vzporedne projekcije	13
2.2	Primerjava projekcij	17
2.3	Uporaba projekcij	18
2.3.1	Tehnično risanje	18
2.3.2	Računalniška grafika	19
2.3.3	Slikarstvo	21
2.3.4	Sence	24
3	PROGRAMSKI DEL	24
3.1	Teoretična podlaga	24
3.2	Postopek projiciranja	33
3.2.1	Razbitje na daljice/točke	35
3.2.2	Projiciranje točk	38
3.2.3	Pretvorba v zaslonske koordinate	41
3.2.4	Izris	43
3.3	Parametri	44
3.4	Navigacija	44
4	KONČNI IZDELEK	46
4.1	Prikazno okno in nastavitve	46
4.1.1	Objekti	47
4.1.2	Nastavitve projekcije	47
4.1.3	Nastavitve kamere	48
4.1.4	Nastavitve prikaza	48
4.1.5	Drugi podatki	48
4.2	Projekcije	51
4.2.1	Kocka	51
4.2.2	Kvader	51
4.2.3	Krožnice	51
4.2.4	Funkcije	51
4.2.5	Slabost vzporednih projekcij	51

4.3	Pomanjkljivost programa	52
4.4	Slike projekcij	52
5	ZAKLJUČEK	65
6	DRUŽBENA ODGOVORNOST	65
7	VIRI IN LITERATURA	66

Slike

1	Projiciranje	10
2	Delitev geometrijskih projekcij	11
3	Objekt med sečiščem in projekcijsko ravnino	12
4	Projekcijska ravnina med sečiščem in objektom	12
5	Eno- in dvobežiščna perspektiva	13
6	Ortogonalna projekcija	14
7	Mongeova projekcija	14
8	Vzporedna poševna projekcija	16
9	Slabost vzporednih projekcij	17
10	Primer tehniške risbe v izometrični projekciji	18
11	Primer tehniške risbe v dimetrični projekciji	19
12	Primer risbe hiše v različnih projekcijah	19
13	Primer igre v pravokotni projekciji	20
14	Primer igre v poševni projekciji	20
15	Primer igre v izometrični projekciji	21
16	Primer igre v perspektivni projekciji	21
17	Raphael: <i>Atenska šola</i>	22
18	Hans Vredeman de Vries: <i>Perspektiva</i>	22
19	William Hogarth: <i>Satira na lažno perspektivo</i>	23
20	Penrosove stopnice	23
21	Vrtenje okoli točke, prva slika	28
22	Vrtenje okoli točke, druga slika	28
23	Sferne koordinate točke	29
24	Koti med koordinatnimi osmi	31
25	Poševna projekcija enotske kocke	31
26	Vektor projiciranih žarkov pri poševni projekciji	32
27	Diagram poteka programa	34
28	Diagram podprograma <i>projiciranje</i>	35
29	Kvader	36
30	Krožnica	37
31	Risanje krožnice	38
32	Diagram projiciranja točk	39
33	Normala projekcijske ravnine	40
34	Diagram pretvorbe v zaslonske koordinate	41
35	Rotacija ravnine	42
36	Vrtenje kamere	45
37	Premik kamere	46
38	Okno programa	47
39	Dodajanje novega objekta	49

40	Nastavitve projekcije	50
41	Nastavitve kamere	50
42	Nastavitve prikaza	50
43	Prikaz kotov med osmi in drugi podatki	50
44	Kocka v perspektivni projekciji	52
45	Kocka v perspektivni projekciji (pogled od spredaj)	53
46	Kocka v trimetrični projekciji	53
47	Kocka v dimetrični projekciji	54
48	Kocka v izometrični projekciji	54
49	Kocka v pravokotni projekciji (naris)	55
50	Kocka v kavalirski projekciji ($\alpha = 30^\circ$)	55
51	Kocka v kabinetni projekciji ($\alpha = 30^\circ$)	56
52	Kvader v perspektivni projekciji	56
53	Kvadri v perspektivni projekciji ($f = 50$)	57
54	Kvadri v perspektivni projekciji ($f = 100$)	57
55	Krožnice v izometrični projekciji	58
56	Krožnice v perspektivni projekciji	58
57	Grafi funkcij v pravokotni projekciji (naris)	59
58	Grafi funkcij v izometrični projekciji	59
59	Grafi kvadratnih funkcij v izometrični projekciji	60
60	Grafi kvadratnih funkcij v pravokotni projekciji (naris)	60
61	Grafi sinusnih funkcij v izometrični projekciji	61
62	Grafi sinusnih funkcij v pravokotni projekciji (naris)	61
63	Pomanjkljivost programa pri perspektivni projekciji	62
64	Krožnice v izometrični projekciji	62
65	Krožnice v pravokotni projekciji (tloris)	63
66	Krožnice v perspektivni projekciji	63
67	Krožnice v trimetrični projekciji	64

Tabele

1	Primerjava tipov projekcij	17
---	--------------------------------------	----

1 UVOD

Matematične projekcije nam sledijo na vsakem koraku. Z razcvetom tehnologije se je potreba po matematičnih projekcijah še povečala, saj brez njenih načel ne bi bilo ne računalniških igrice niti vse popularnejših računalniško generiranih tridimenzionalnih filmov.

Opisna geometrija je seveda obstajala že veliko let pred prihodom računalnikov. Njeno preučevanje temelji na različnih projekcijah, ki se pojavljajo vse od antike, kjer so se že ukvarjali z arhitekturo, načrtovanjem in geometrijskimi telesi nasploh (povzeto po SAJOVIC, 1975, 3).

1.1 Cilja naloge

V raziskovalni nalogi želim preučiti različne tipe projekcij in jih med seboj primerjati. S pomočjo znanja o projiciranju želim nato napisati matematični algoritem (program), ki na računalniški zaslon projicira geometrijske objekte v želeni projekciji. Pri tem me ne zanima algoritem, ki je najhitrejši in opravi projiciranje z najmanjšim številom operacij, temveč želim algoritem, ki temelji na matematičnih definicijah projiciranja.

1.2 Predpostavki

Pred začetkom raziskovanja sem si postavil dve teoretični predvidevanji:

- obstaja več tipov projekcij, ki se med seboj razlikujejo;
- vsaka projekcija ima svoje prednosti in slabosti.

1.3 Metodologija

Moje delo je potekalo tako, da sem najprej zbral literaturo na temo opisne geometrije, jo preučil in iz nje sestavil teoretični del raziskovalne naloge. Nato sem se lotil sestavljanja algoritma za projiciranje, pri čemer sem se opiral na v teoretičnem delu pridobljeno znanje o projekcijah in znanje o programiranju.

2 TEORETIČNI DEL

Evklidsko geometrijo lahko razdelimo na naslednje panoge:

- ravninska geometrija ali **planimetrija**;
- prostorska geometrija ali **stereometrija**;
- **trigonometrija**.

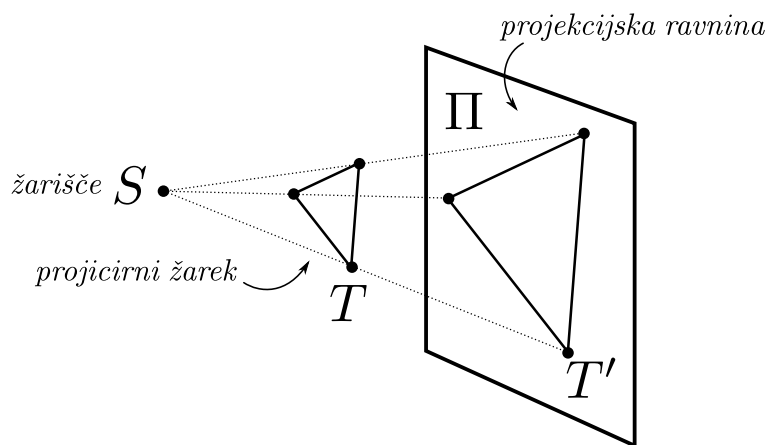
Ravninska geometrija se ukvarja z liki v ravnini, medtem ko prostorska geometrija proučuje like in telesa v prostoru. Ker teles v prostoru za razliko od likov ne moremo poljubno narisati oz. prikazati na ravnini, se je pojavila veda **opisna geometrija** (ali *veda o projiciranju*), ki se ukvarja z upodabljanjem prostorskih geometrijskih objektov na ravnine. Pri tem želimo objekt prikazati tako, da lahko iz risbe določimo tudi dejanske lastnosti prvotnega objekta (velikost, lega, oblika ...).

Upodabljanje objektov na ravnine imenujemo **projiciranje**.

2.1 Projiciranje in tipi projekcij

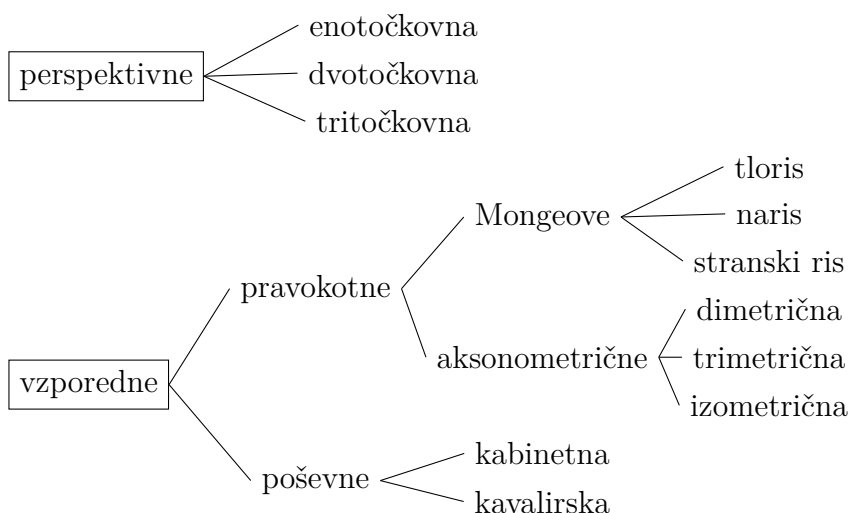
Projiciranje poteka tako, da izberemo **projekcijsko ravnino** Π , na katero bomo projicirali objekte, in **projicirno središče** (ali izvor žarkov ali žarišče ali sečišče) S , iz katerega izhajajo vsi **projicirni žarki**.

Točko T izvornega objekta projiciramo v točko T' , ki jo imenujemo **projekcija točke T** . Množico vseh projekcij točk imenujemo **projekcija objekta**.



Slika 1: Projiciranje

Obstaja več različnih načinov projiciranja (projekcij), ki jih uporabljamo glede na naše potrebe. Ravninske geometrijske projekcije (torej take, kjer objekte projiciramo na ravnino) v glavnem delimo glede na lastnost projicirnih žarkov, in sicer:



Slika 2: Delitev geometrijskih projekcij (povzeto in prilagojeno po PREBIL, 2002, 10)

2.1.1 Perspektivna projekcija

Perspektivna ali *centralna* projekcija je taka projekcija, pri kateri vsi projicirni žarki izhajajo iz **končno oddaljene točke** S (žarišča). Perspektivno projekcijo imenujemo tudi **perspektiva**.

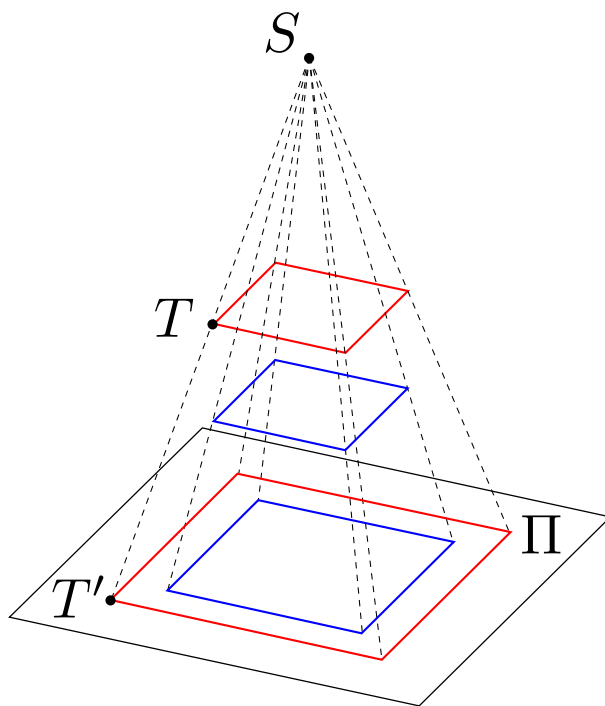
Sečišče lahko postavimo na dve različni mesti, tako da je zaporedje, gledano po projicirnem žarku:

1. sečišče, objekt, projicirna ravnina *ali*
2. sečišče, projicirna ravnina, objekt.

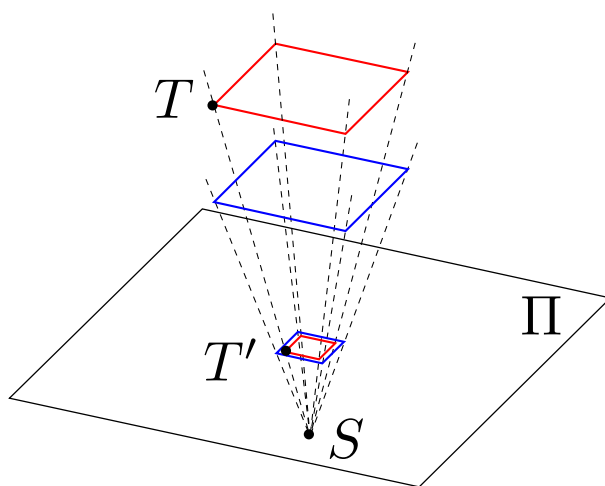
Drugi način je pogostejši, saj so v tem primeru objekti, ki so bližje ravnini, videti večji, kar lahko opazimo tudi v naravi. Sečišče nikakor ne sme ležati na projekcijski ravnini ($S \notin \Pi$), saj se v tem primeru vse točke preslikajo v neskončno oddaljeno točko.

Bežišče imenujemo točko, v kateri se po projiciranju sekajo premice, ki so med seboj vzporedne (bežišče torej leži na projekcijski ravnini). Slika lahko ima eno, dve ali tri bežišča, kar je odvisno od postavitve projekcijske ravnine.

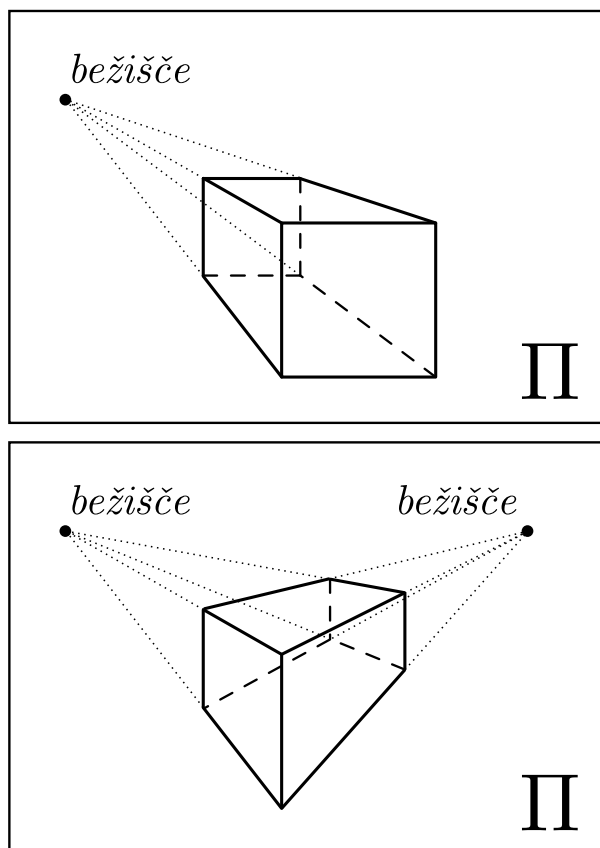
Na sliki 5 lahko vidimo primer projiciranja kvadra tako, da ima slika eno bežišče, in tako, da ima slika dve bežišči.



Slika 3: Objekt med sečiščem in projekcijsko ravnino



Slika 4: Projekcijska ravnina med sečiščem in objektom



Slika 5: Eno- in dvobežiščna perspektiva

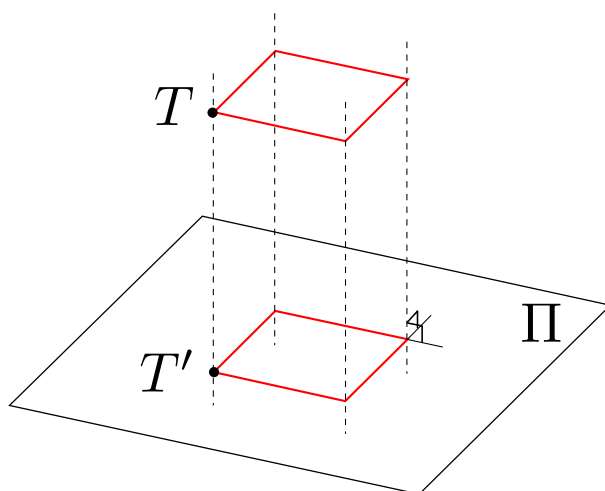
2.1.2 Vzporedne projekcije

Vzporedna ali *paralelna* projekcija je taka projekcija, pri kateri projekcni žarki izhajajo iz **neskončno oddaljene točke**. Tako točko imenujemo tudi *neprava* ali *fiktivna* točka, projekcni žarki pa so med seboj vzporedni.

Vzporedne projekcije delimo na pravokotne in poševne.

Vzporedne pravokotne projekcije Pravokotna ali *ortogonalna* projekcija je taka vzporedna projekcija, pri kateri so projekcni žarki **pravokotni** na projekcijsko ravnino Π .

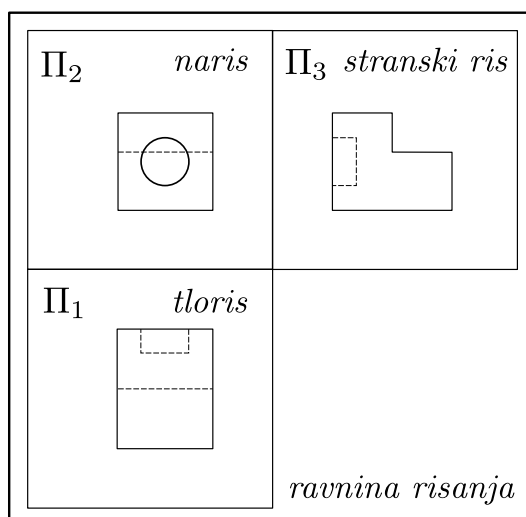
Mongeova projekcijska metoda Pri Mongeovem projiciranju uporabljamo več projekcijskih ravnin, ki so med seboj pravokotne in običajno tudi vzporedne z ravninami, ki jih tvorijo koordinatne osi (ravnine xy , xz , yz). Te ravnine in projekcije nanj imenujemo:



Slika 6: Ortogonalna projekcija

- Π_1 tlorisna ravnina \rightarrow tloris,
- Π_2 narisna ravnina \rightarrow naris,
- Π_3 tretja projekcijska ravnina (neobvezna) \rightarrow stranski ris.

Ravnini Π_1 in Π_2 zvrnemo na tlorisno ravnino, tako da dobimo eno končno ravnino risanja.



Slika 7: Mongeova projekcija

Mongeova projekcija je imenovana po francoskem matematiku Gaspardu Mongeu (1746–1818), ki je soustanovitelj opisne geometrije (SAJOVIC, 1975, 63).

Aksonometrične projekcije Za razliko od Mongeovega projiciranja se pri aksonometrični projekciji uporablja zgolj ena projicirna ravnina, vendar z ustrezno postavitvijo ravnine poskrbimo, da je hkrati vidnih več ploskev objekta (poskrbimo, da ravnina ni vzporedna s katero izmed osnovnih ploskev objekta).

Poznamo tri različne tipe aksonometrije:

- izometrična,
- dimetrična in
- trimetrična,

ki se razlikujejo po kotih med projekcijami koordinatnih osi. Pri izometrični projekciji so vsi trije koti med osmi med seboj enaki in veliki 120° , pri dimetrični projekciji sta med seboj enaka dva kota in pri trimetrični so vsi med seboj različni. Različen tip aksonometrične projekcije dobimo z ustrezno postavitvijo projekcijske ravnine:

- izometrična:
projekcijska ravnina seka koordinatne osi pri istih vrednostih,
- dimetrična:
projekcijska ravnina seka dve koordinatni osi pri istih vrednostih, tretjo pri poljubni,
- trimetrična:
postavitev projekcijske ravnine je poljubna.

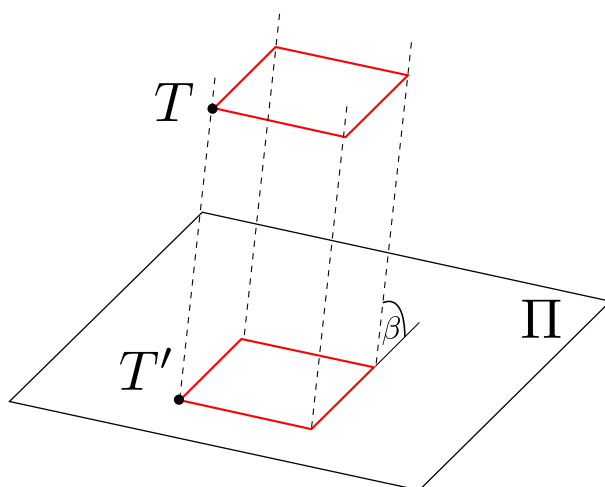
Vzporedne poševne projekcije Poševna projekcija je taka vzporedna projekcija, pri kateri padajo projicirni žarki poševno na projekcijsko ravnino Π (s projekcijsko ravnino oklepajo poljuben kot β).

Nekatere poševne projekcije imajo tudi svoje ime; naj izpostavim kabinetno in kavalirsko poševno projekcijo.

- **Kabinetna projekcija**

$$\alpha = 30^\circ \text{ ali } 45^\circ$$

$$k = 0,5$$



Slika 8: Vzporedna poševna projekcija

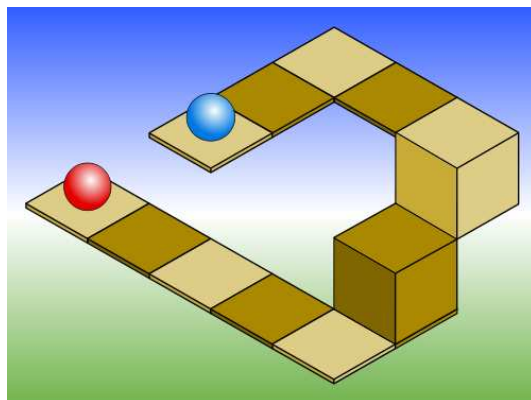
- **Kavalirska projekcija**

$$\alpha = 30^\circ \text{ ali } 45^\circ$$

$$k = 1$$

α predstavlja *zatezni kot*, k pa koeficient skrajšave. Pojma sta opisana v odstavku *Izračun projicirnega žarka* na strani 40.

Slabost vzporednih projekcij Pri vzporednih projekcijah se velikost objektov ne spreminja z njihovo oddaljenostjo od projekcijske ravnine. Pri samem projiciranju se to pokaže tako, da lahko projekcijsko ravnino poljubno vzporedno premikamo, slika pa bo ostala enaka. Slabost tega lahko vidimo na sliki 9; modra žogica se nahaja dve enoti višje kot rdeča, čeprav je videti, kot da se nahaja na isti višini.



Slika 9: Slabost vzporednih projekcij (*Vir: Višina, 2013*)

2.2 Primerjava projekcij

V tabeli 1 sem povzel bistvene značilnosti nekaterih tipov projekcij.

Tabela 1: Primerjava tipov projekcij

	perspektivna	vzporedne	
		izometrična	poševna
<i>projicirni žarki:</i>	šop premic (sekajo se v končno oddaljeni točki S)	snop premic	
<i>vzporedni premici se preslikata v:</i>	vzporedni premici , če sta vzporedni s projekcijsko ravnino, sicer v nevzporedni	vzporedni premici	
<i>prednosti:</i>	resničen izgled; kot v naravi	ohranjanje dolžin daljic, ki so vzporedne koordinatnim osem	enostavno risanje
<i>slabosti:</i>	popačenje dolžin	možnost zamenjave višine in oddaljenosti	veliko popačenje ob neprimerni postavitvi projekcijske ravnine

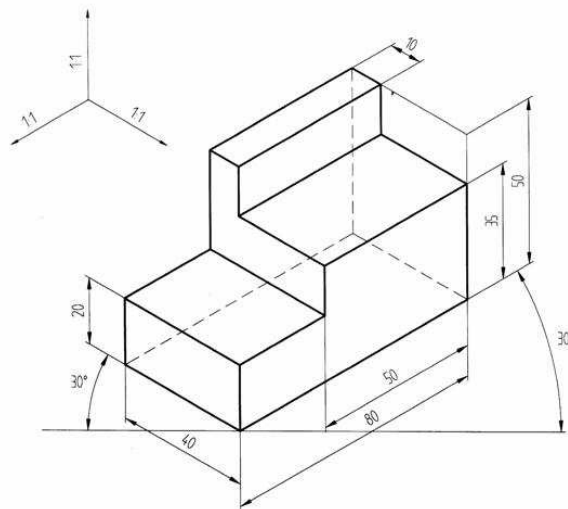
2.3 Uporaba projekcij

Prikazovanje geometrijskih objektov v prostoru se uporablja na mnogih področjih.

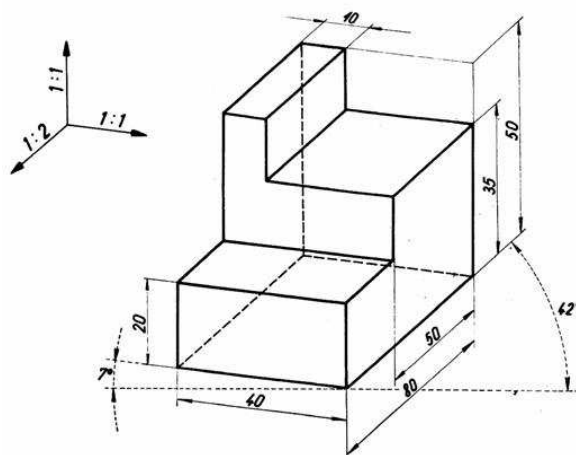
2.3.1 Tehnično risanje

Gradbeni in strojni inženirji ter ostali poklici, ki vključujejo načrtovanje in oblikovanje, potrebujejo pri svojem delu znanje projiciranja, saj jim omogoča risanje in prikazovanje želenih objektov (strojev, tehničnih izdelkov, proizvodov ...). Tehnično risanje je razvilo veliko svojih pravil in standardov, ki jih morajo upoštevati pri izdelavi risb in načrtov (debelina črt, označevanje, kotiranje, izbira merila ...).

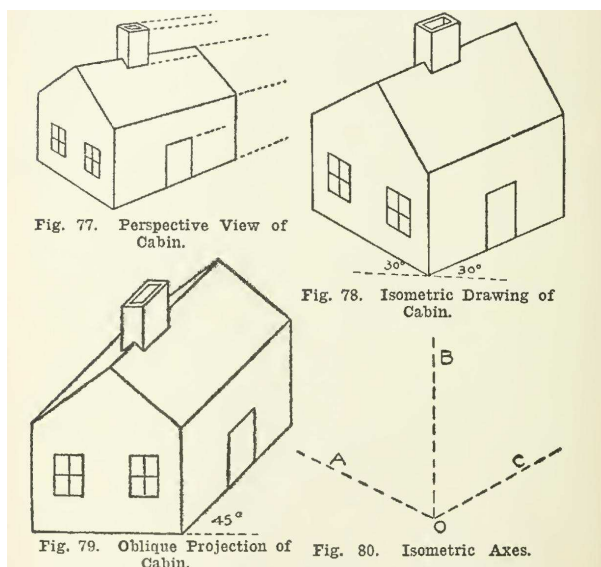
Na slikah 10 in 11 lahko vidimo isti objekt, prikazan v izometrični in dime- trični projekciji. Slika 12 prikazuje skico hiše v treh različnih projekcijah; zgoraj levo: perspektivna projekcija, zgoraj desno: izometrična projekcija in spodaj levo: poševna projekcija ($\alpha = 45^\circ$ in $k = 1$).



Slika 10: Primer tehniške risbe v izometrični projekciji (Vir: *Izometrična, 2013*)



Slika 11: Primer tehniške risbe v dimetrični projekciji (Vir: *Dimetrična*, 2013)



Slika 12: Primer risbe hiše v različnih projekcijah (Vir: *Hiša*, 2013)

2.3.2 Računalniška grafika

Računalniška grafika obsega različna področja, ki uporabljajo projiciranje; to so računalniške igre, računalniško tridimenzionalno modeliranje, računalniško generirani filmi ... Računalniške igre so sprva uporabljale pravokotno projekcijo (tloris ali naris), nakar so začele uporabljati dimetrično, saj osnovne

enote računalniškega zaslona (slikovne pike), ki tvorijo kvadratno mrežo, ne omogočajo tvorbe kota 120° , ki je potreben pri izometrični projekciji. Z izboljšanjem učinkovitosti računalnikov in iznajdbo novih algoritmov so se začele pojavljati igre v izometrični in kasneje tudi perspektivni projekciji, ki danes povsem prevladuje.



Slika 13: Primer igre v pravokotni projekciji (Vir: Mario, 2013)



Slika 14: Primer igre v poševni projekciji (Vir: Paperboy, 2013)



Slika 15: Primer igre v izometrični projekciji (*Vir: Sims, 2013*)



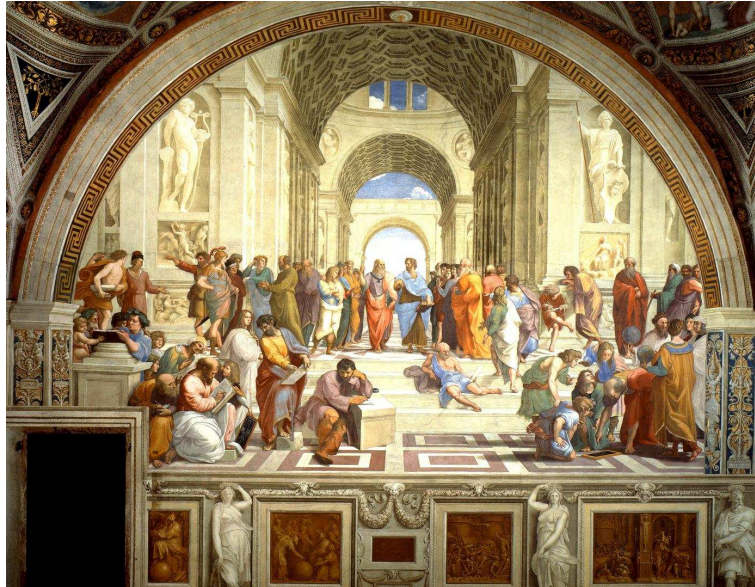
Slika 16: Primer igre v perspektivni projekciji (*Vir: Portal, 2013*)

2.3.3 Slikarstvo

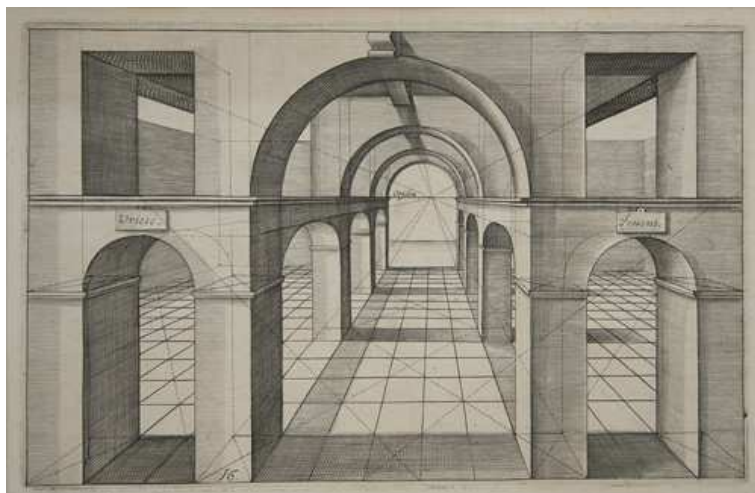
Na Raphaelovi freski *Atenska šola* lahko vidimo perspektivno projekcijo z enim bežiščem, ki se nahaja približno v sredini slike (med glavama Platona in Aristotela).

William Hogarth, ki je tudi sicer naredil veliko grafik na temo perspektive, je leta 1754 izdelal grafiko z naslovom *Satire on False Perspective* (Satira na lažno perspektivo), na kateri lahko vidimo polno napak in neskladnosti, ki onemogočajo, da bi takšen prizor videli v naravi. Takšnim objektom pravimo *nemogoči objekti*, saj jih ni mogoče tudi dejansko izdelati. Na to temo obstaja veliko optičnih prevar, ki prikazujejo nemogoče objekte.

Ena izmed slabosti vzporednih projekcij je možnost zamenjave globine z višino. To lahko vidimo na sliki 20, kjer so prikazane Penrosove stopnice.



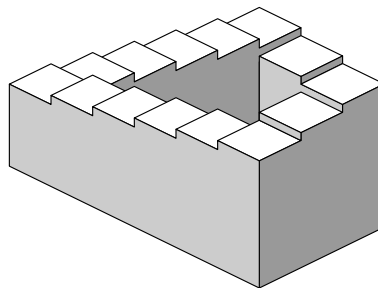
Slika 17: Raphael: *Atenska šola* (Vir: Raphael, 2013)



Slika 18: Hans Vredeman de Vries: *Perspektiva* (Vir: Vries, 2013)



Slika 19: William Hogarth: *Satira na lažno perspektivo* (Vir: Hogarth, 2013)



Slika 20: Penrosove stopnice (Vir: Penrose, 2013)

2.3.4 Sence

Zelo lep, vendar pogosto prezrt primer projiciranja so tudi sence v naravi. Če kot svetilo vzamemo Sonce, potem so sončni žarki naši projicirni žarki. Ker vsi žarki izhajajo iz končno oddaljenega žarišča (Sonca), gre tukaj za perspektivno projekcijo.

3 PROGRAMSKI DEL

Kot primer praktične uporabe opisne geometrije sem se odločil izdelati aplikacijo, ki izrisuje osnovna geometrijska telesa v različnih projekcijah. Program je napisan v razvojnem okolju **Visual Basic Studio 2010 Express** v programskem jeziku Visual Basic z ogrodjem *.NET*.

3.1 Teoretična podlaga

Točka Točko v prostoru podamo s tremi kartezičnimi koordinatami $T(x, y, z)$ ali s tremi sfernimi koordinatami $T(R, \varphi, \vartheta)$. Za pretvarjanje glej formule (3.5).

Premica Premico v prostoru lahko podamo z vektorsko parametrično obliko enačbe premice:

$$\vec{r} = \vec{r}_A + t(\vec{r}_B - \vec{r}_A)$$

pri čemer sta točki A in B fiksni točki, točka T je poljubna točka na premici skozi A in B , \vec{r} je krajevni vektor točke T ($\vec{OT} = \vec{r}$), \vec{r}_A krajevni vektor točke A ($\vec{OA} = \vec{r}_A$), \vec{r}_B pa krajevni vektor točke B ($\vec{OB} = \vec{r}_B$). Enačbo lahko spremenimo tudi v sistem skalarnih parametričnih enačb

$$p : \begin{cases} x = x_A + t(x_B - x_A) \\ y = y_A + t(y_B - y_A) \\ z = z_A + t(z_B - z_A) \end{cases}$$

Z izločitvijo parametra t dobimo klasično obliko enačbe premice

$$x = x_A + t(x_B - x_A)$$

$$x - x_A = t(x_B - x_A)$$

$$\frac{x - x_A}{x_B - x_A} = t \Rightarrow \boxed{\frac{x - x_A}{x_B - x_A} = \frac{y - y_A}{y_B - y_A} = \frac{z - z_A}{z_B - z_A}}$$

Premico, ki je vzporedna vektorju $\vec{v} = (a, b, c)$ in gre skozi točko $A(x_A, y_A, z_A)$, podamo s klasično obliko enačbe premice

$$\frac{x - x_A}{a} = \frac{y - y_A}{b} = \frac{z - z_A}{c}$$

Ravnina Ravnino najlažje podamo s točko $T(x_T, y_T, z_T)$ in normalo ravnine $\vec{n} = (a, b, c)$ ter uporabimo enačbo:

$$a(x - x_T) + b(y - y_T) + c(z - z_T) = 0$$

ki preide v:

$$ax + by + cz - d = 0$$

pri čemer je:

$$d = ax_T + by_T + cz_T$$

Prebodišče premice z ravnino Za izračun prebodišča premice z ravnino uporabimo zapis premice v sistemu parametričnih enačb in zapis ravnine v normalni obliki. Pri tem je premica podana z dvema točkama (A, B) , ravnina pa s točko in normalo:

$$\Pi : ax + by + cz - d = 0$$

$$p : \begin{cases} x = x_A + t(x_B - x_A) \\ y = y_A + t(y_B - y_A) \\ z = z_A + t(z_B - z_A) \end{cases}$$

Vstavimo.

$$a(x_A + t(x_B - x_A)) + b(y_A + t(y_B - y_A)) + c(z_A + t(z_B - z_A)) - d = 0$$

$$ax_A + at(x_B - x_A) + by_A + bt(y_B - y_A) + cz_A + ct(z_B - z_A) - d = 0$$

$$at(x_B - x_A) + bt(y_B - y_A) + ct(z_B - z_A) + ax_A + by_A + cz_A - d = 0$$

$$t(a(x_B - x_A) + b(y_B - y_A) + c(z_B - z_A)) + ax_A + by_A + cz_A - d = 0$$

$$t = -\frac{ax_A + by_A + cz_A - d}{a(x_B - x_A) + b(y_B - y_A) + c(z_B - z_A)}$$

Koordinate prebodišča so torej:

$$\begin{aligned} x' &= x_A - (x_B - x_A) \frac{ax_A + by_A + cz_A - d}{a(x_B - x_A) + b(y_B - y_A) + c(z_B - z_A)} \\ y' &= y_A - (y_B - y_A) \frac{ax_A + by_A + cz_A - d}{a(x_B - x_A) + b(y_B - y_A) + c(z_B - z_A)} \\ z' &= z_A - (z_B - z_A) \frac{ax_A + by_A + cz_A - d}{a(x_B - x_A) + b(y_B - y_A) + c(z_B - z_A)} \end{aligned} \quad (3.1)$$

Če imamo premico podano s smernim vektorjem $\vec{v} = (l, m, n)$ in točko $A(x_A, y_A, z_A)$, izračunamo prebodišče sledeče:

$$\Pi : ax + by + cz - d = 0$$

$$\begin{aligned}
p : \frac{x - x_A}{l} &= \frac{y - y_A}{m} = \frac{z - z_A}{n} = t \\
\frac{x - x_A}{l} &= t \\
\frac{y - y_A}{m} &= t \\
\frac{z - z_A}{n} &= t \\
p : \begin{cases} x = tl + x_A \\ y = tm + y_A \\ z = tn + z_A \end{cases}
\end{aligned}$$

Vstavimo.

$$\begin{aligned}
a(tl + x_A) + b(tm + y_A) + c(tn + z_A) - d &= 0 \\
atl + ax_A + btm + by_A + ctn + cz_A - d &= 0 \\
t(al + bm + cn) + ax_A + by_A + cz_A - d &= 0 \\
t(al + bm + cn) &= -ax_A - by_A - cz_A + d \\
t &= -\frac{ax_A + by_A + cz_A - d}{al + bm + cn}
\end{aligned}$$

Koordinate prebodišča $T'(x', y', z')$ so torej:

$$\begin{aligned}
x' &= x_A - \frac{ax_A + by_A + cz_A - d}{al + bm + cn}l \\
y' &= y_A - \frac{ax_A + by_A + cz_A - d}{al + bm + cn}m \\
z' &= z_A - \frac{ax_A + by_A + cz_A - d}{al + bm + cn}n
\end{aligned} \tag{3.2}$$

Vrtenje točke v prostoru Točko v prostoru lahko zavrtimo okoli katere koli koordinatne osi za kot θ , kar storimo s pomočjo rotacijskih matrik. Vrtenje za kot θ je vrtenje v nasprotni smeri urinega kazalca, gledano iz smeri koordinatne osi proti koordinatnemu izhodišču.

$$\begin{aligned}
R_x(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \\
R_y(\theta) &= \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}
\end{aligned}$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Primer: Točka A , predstavljena s stolpičnim vektorjem, se pri vrtenju za kot α okoli osi x preslika v A' :

$$A = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$A' = R_x(\alpha)A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Koordinate točke A' so torej:

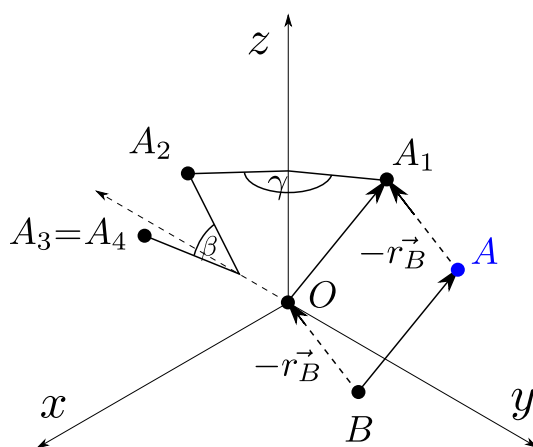
$$\begin{aligned} x' &= x \\ y' &= y \cos \alpha - z \sin \alpha \\ z' &= y \sin \alpha + z \cos \alpha \end{aligned} \quad (3.4)$$

Algoritem 1 (Vrtenje točke okoli točke) Če želimo točko A zavrteti okoli točke B v točko A' ($A \mapsto A'$), to storimo po sledečem postopku (glej pripadajoči sliki 21 in 22):

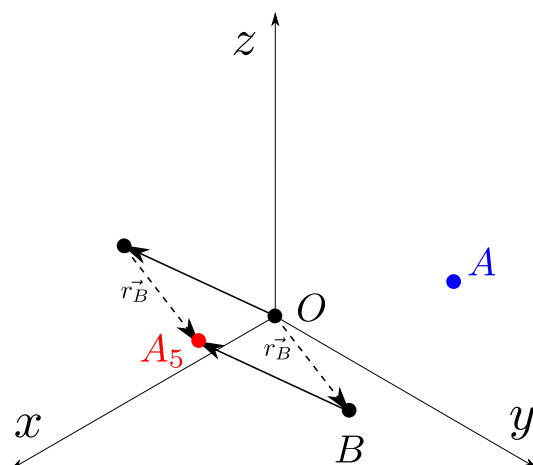
- vektor $\vec{v} = \vec{BA}$ vzporedno premaknemo tako, da začetek vektorja sovpada s koordinatnim izhodiščem, kar pomeni, da ga premaknemo za vektor $\vec{BO} = -\vec{r}_B = (-x_B, y_B, z_B)$

$$\vec{BA} \mapsto \vec{OA}_1$$

- točko A_1 zavrtimo za poljubne kote okoli koordinatnih osi v točko A_4 (na slikah sta opravljene samo vrtenje okoli osi z za kot γ in okoli osi y za kot β , zato $A_3 = A_4$);
- vektor \vec{OA}_4 vzporedno premaknemo za vektor \vec{r}_B v vektor \vec{BA}_5 , s čimer začetek vektorja ponovno sovpada s točko B ;
- točka A_5 predstavlja končni rezultat vrtenja točke A okoli točke B .



Slika 21: Vrtenje okoli točke, prva slika



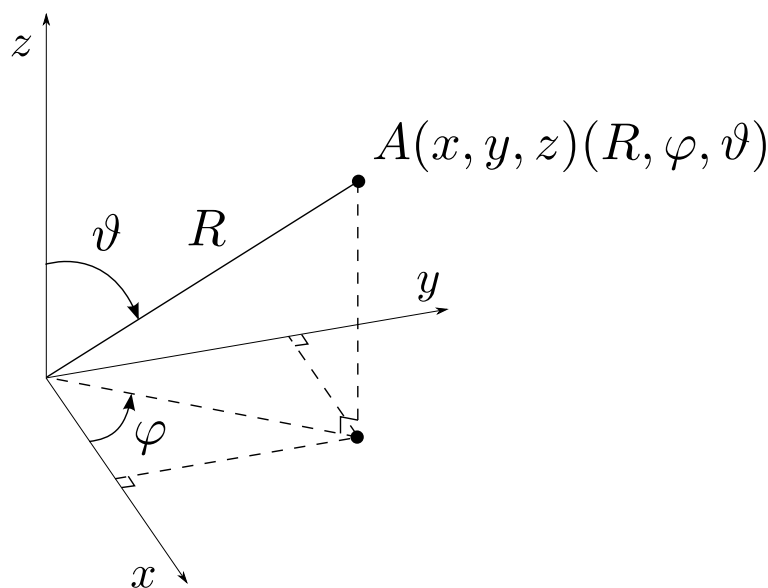
Slika 22: Vrtenje okoli točke, druga slika

Pretvorba sfernih koordinat Položaj vsake točke v prostoru lahko podamo tudi s sfernimi koordinatami. Točka $A(R, \varphi, \vartheta)$ ima tri sferne koordinate: R – oddaljenost od koordinatnega izhodišča, φ – azimut in ϑ – polarni kot.

Sferne koordinate izračunamo iz kartezičnih na sledeč način:

$$R = \sqrt{x^2 + y^2 + z^2}$$

$$\varphi = \begin{cases} \arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right) & \text{za } y \geq 0 \\ 2\pi - \arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right) & \text{za } y < 0 \end{cases}$$



Slika 23: Sferne koordinate točke

$$\vartheta = \arccos\left(\frac{z}{R}\right) = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \quad (3.5)$$

Obratno pretvarjamo:

$$\begin{aligned} x &= R \cos(\varphi) \sin(\vartheta) \\ y &= R \sin(\varphi) \sin(\vartheta) \\ z &= R \cos(\vartheta) \end{aligned} \quad (3.6)$$

Izračun kotov med koordinatnimi osmi Pri projiciranju se koordinatne osi preslikajo v tri med seboj sekajoče se premice na ravnini. Za izračun kotov med osmi uporabimo kosinusni izrek, saj v postopku projiciranja osi pravzaprav predstavimo s tremi daljicami poljubne dolžine in ne premicami. Te daljice imajo skupno začetno točko v koordinatnem izhodišču, vsaka posamezna pa ima končno točko na eni izmed osi. Osi so nato na zaslonu izrisane s tremi daljicami (glej sliko 24), in sicer os x z daljico OO_x , os y z OO_y in os z z OO_z , pri čemer so koordinate posameznih točk $O(x_O, y_O)$, $O_x(x_{O_x}, y_{O_x})$, $O_y(x_{O_y}, y_{O_y})$ in $O_z(x_{O_z}, y_{O_z})$. Pri izračunu kota upoštevamo orientiranost trikotnika, ki jo izračunamo z determinanto:

$$D = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$$

pri čemer so točke $T_1(x_1, y_1)$, $T_2(x_2, y_2)$ in $T_3(x_3, y_3)$ oglišča trikotnika, ki mu določamo orientacijo. Trikotniki, preko katerih izračunamo kote med osmi, so $\triangle O_y O_z O$, $\triangle O_x O_y O$ in $\triangle O_z O_x O$. Če je kamera v prvem oktantu, so vsi trije trikotniki pozitivno usmerjeni (kot na sliki 24); če je kamera v sedmem oktantu, so vsi trikotniki negativno usmerjeni. Posamezne determinante so:

$$\begin{aligned} D_{xy} &= (x_{O_y} - x_{O_x})(y_O - y_{O_x}) - (x_O - x_{O_x})(y_{O_y} - y_{O_x}) \\ D_{yz} &= (x_{O_z} - x_{O_y})(y_O - y_{O_y}) - (x_O - x_{O_y})(y_{O_z} - y_{O_y}) \\ D_{xz} &= (x_{O_y} - x_{O_z})(y_O - y_{O_z}) - (x_O - x_{O_z})(y_{O_y} - y_{O_z}) \end{aligned}$$

$$\begin{aligned} \alpha' = \angle O_y O_z O &= \begin{cases} \arccos\left(\frac{d_z^2 + d_y^2 - d_{yz}^2}{2d_z d_y}\right) & \text{za } D_{yz} \geq 0 \\ 360^\circ - \arccos\left(\frac{d_z^2 + d_y^2 - d_{yz}^2}{2d_z d_y}\right) & \text{za } D_{yz} < 0 \end{cases} \\ \beta' = \angle O_x O_y O &= \begin{cases} \arccos\left(\frac{d_y^2 + d_x^2 - d_{xy}^2}{2d_x d_y}\right) & \text{za } D_{xy} \geq 0 \\ 360^\circ - \arccos\left(\frac{d_y^2 + d_x^2 - d_{xy}^2}{2d_x d_y}\right) & \text{za } D_{xy} < 0 \end{cases} \\ \gamma' = \angle O_z O_x O &= \begin{cases} \arccos\left(\frac{d_x^2 + d_z^2 - d_{xz}^2}{2d_x d_z}\right) & \text{za } D_{xz} \geq 0 \\ 360^\circ - \arccos\left(\frac{d_x^2 + d_z^2 - d_{xz}^2}{2d_x d_z}\right) & \text{za } D_{xz} < 0 \end{cases} \end{aligned}$$

pri čemer je:

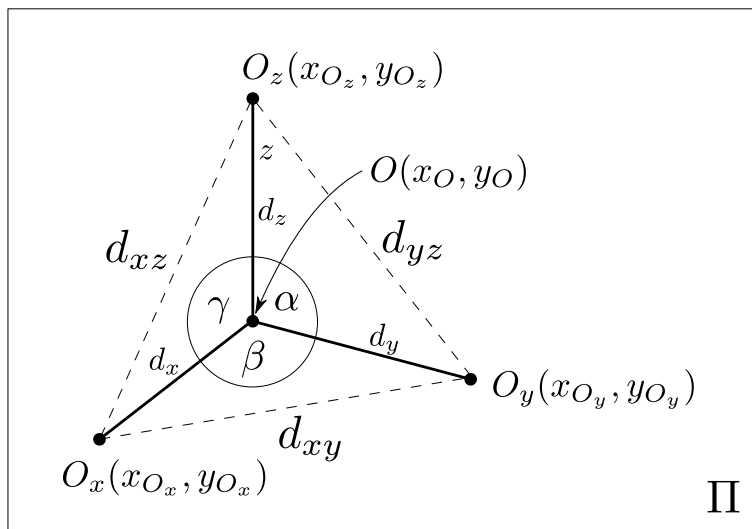
$$\begin{aligned} d_x &= \overline{O_x O} = \sqrt{(x_O - x_{O_x})^2 + (y_O - y_{O_x})^2} \\ d_y &= \overline{O_y O} = \sqrt{(x_O - x_{O_y})^2 + (y_O - y_{O_y})^2} \\ d_z &= \overline{O_z O} = \sqrt{(x_O - x_{O_z})^2 + (y_O - y_{O_z})^2} \\ d_{xy} &= \overline{O_x O_y} = \sqrt{(x_{O_y} - x_{O_x})^2 + (y_{O_y} - y_{O_x})^2} \\ d_{xz} &= \overline{O_x O_z} = \sqrt{(x_{O_z} - x_{O_x})^2 + (y_{O_z} - y_{O_x})^2} \\ d_{yz} &= \overline{O_y O_z} = \sqrt{(x_{O_z} - x_{O_y})^2 + (y_{O_z} - y_{O_y})^2} \end{aligned}$$

Zadnji korak izračuna kotov je preverjanje usmerjenosti vseh treh trikotnikov. Če sta pozitivno usmerjena vsaj dva trikotnika (kamera v I., II., IV. ali V. oktantu), potem velja:

$$\alpha = \alpha' \quad \beta = \beta' \quad \gamma = \gamma'$$

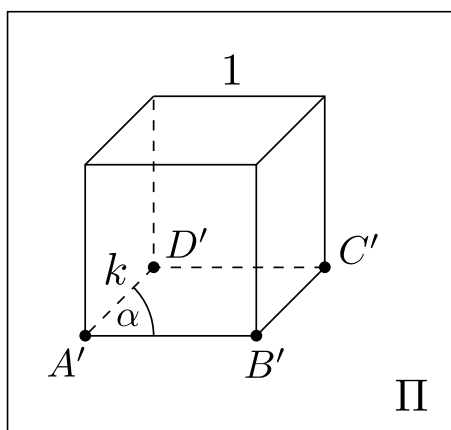
sicer:

$$\alpha = 360^\circ - \alpha' \quad \beta = 360^\circ - \beta' \quad \gamma = 360^\circ - \gamma'$$

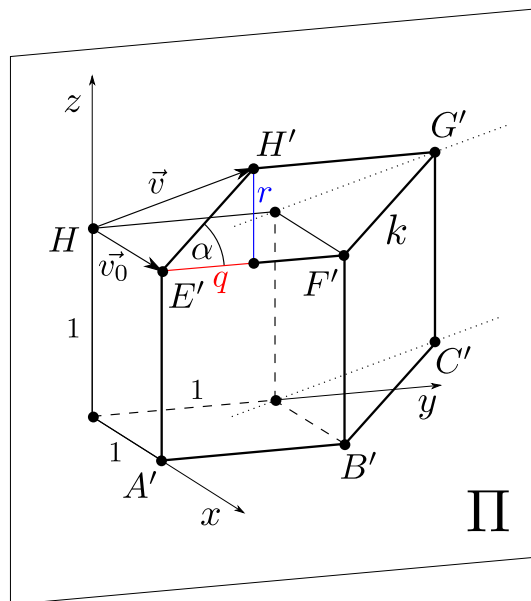


Slika 24: Koti med koordinatnimi osmi

Poševna projekcija Pri poševni projekciji projekcijsko ravnino postavimo tako, da je vzporedna ravnini yz (kameri lahko spreminjamo samo oddaljenost od točke gledanja R_G , ki pa zaradi vzporedne projekcije pravzaprav nima vpliva na končno sliko), saj sicer pride do prevelikega popačenja objekta in projekcija ni uporabna. Kot parameter ne podajamo kota, ki ga projekcijski žarki oklepajo s projekcijsko ravnino, temveč **zatezni kot** α in **koeficient skrajšave** k . To je najlažje ponazoriti s projekcijo enotske kocke, pri čemer je $\alpha = \angle B'A'D'$ in $k = \frac{AD'}{AD}$ (glej sliko 25). Za izračun enačb projekcijskih



Slika 25: Poševna projekcija enotske kocke



Slika 26: Vektor projicirnih žarkov pri poševni projekciji

žarkov (smerni vektor) se osredotočimo na oglišči kocke A in D , ki tvorita daljico $AD = d$; $\overline{AD} = 1$ in ležita na osi x . Naj D sovpada s koordinatnim izhodiščem in naj projekcijska ravnina Π poteka skozi točko A . Pri projiciranju v poševni projekciji se izvedejo preslikave

$$A \mapsto A' = A$$

$$D \mapsto D'$$

$$d \mapsto k$$

Projicirni žarki so premice, vzporedne vektorju $\vec{v} = \overrightarrow{DD'}$. Na sliki 26 lahko vidimo prikaz te projekcije.

Vpeljimo nov vektor $\vec{v}_0 = \overrightarrow{DA}$. Premaknimo konec vektorja za q v smeri osi y in za r v smeri osi z tako, da bo sovpadal z D' . Dobljeni vektor $\overrightarrow{DD'}$ predstavlja vektor \vec{v} . Premika, ki smo jo izvedli, izračunamo z uporabo kotnih funkcij:

$$q = k \cos(\alpha)$$

$$r = k \sin(\alpha)$$

Na sliki 26 je prikazana poševna projekcija enotske kocke. Zaradi preglednosti je slika prilagojena: vse točke niso projicirane in prikazane, vektor \vec{v} poteka od H do H' (in je enak prej omenjenemu vektorju $\overrightarrow{DD'}$), kot α je prikazan kot $\angle F'G'H'$, dolžina k pa je označena na daljici $\overline{F'G'}$.

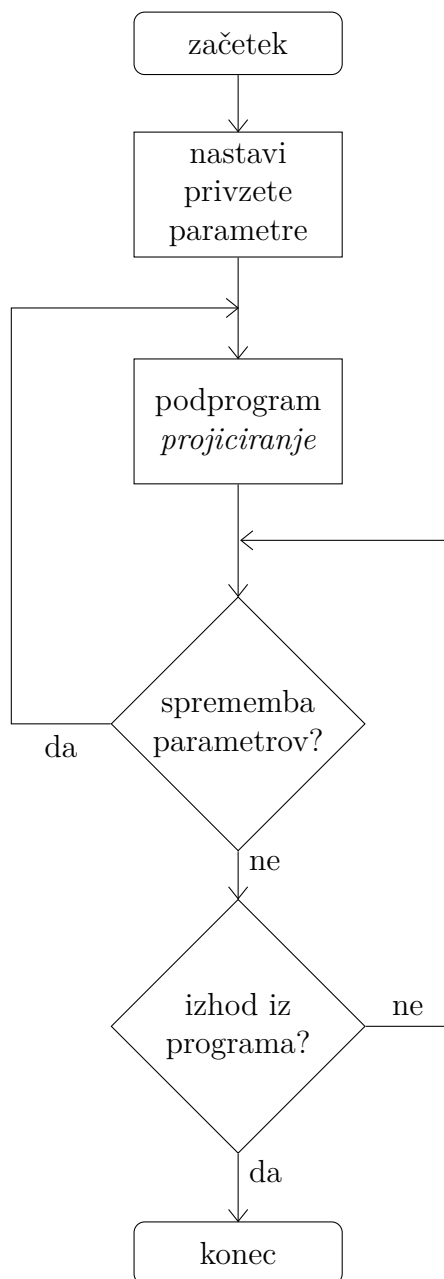
3.2 Postopek projiciranja

Delovanje programa je prikazano s poenostavljenimi diagrami poteka. Zaradi poenostavitve so prikazani samo vhodni podatki, pomembnejši izhodni podatki pa so omenjeni v besedilu.

Glavni elementi (geometrijski elementi in spremenljivke), ki se uporabljajo znotraj programa in so omenjeni kasneje v opisu postopka, so:

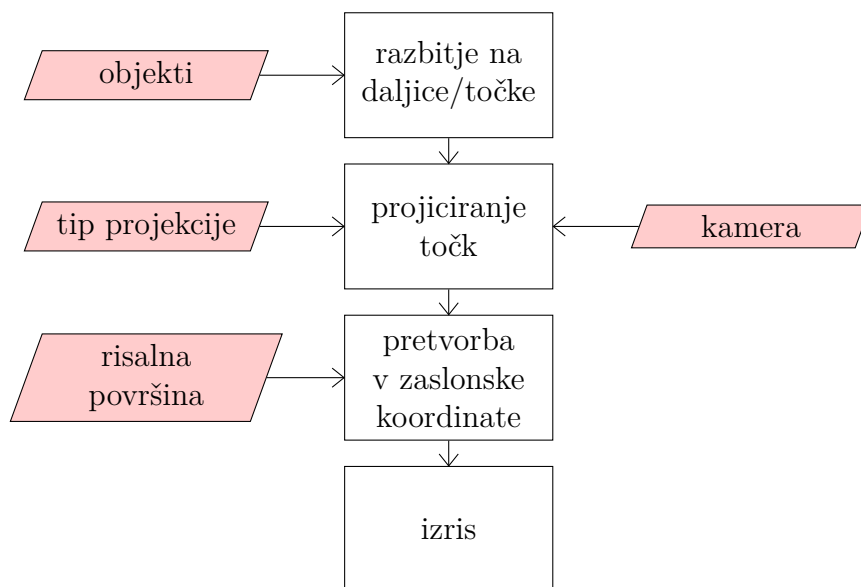
- točke:
 - koordinatno izhodišče $O(x_O, y_O, z_O)$,
 - kamera $K(x_K, y_K, z_K)$,
 - točka gledanja $G(x_G, y_G, z_G)$,
 - žarišče $S(x_S, y_S, z_S)$,
 - točka objekta $T(x, y, z)$,
 - projekcija točke objekta $T'(x', y', z')$,
 - zaslonska točka objekta $T_Z(x_Z, y_Z)$.
- vektorji:
 - normala ravnine $\vec{n} = (a, b, c)$
- premice:
 - projekcinski žarek p
- ravnine:
 - projekcijska ravnina Π
- skalarne vrednosti:
 - goriščna razdalja f ,
 - faktor goriščne razdalje k_f ,
 - dolžina risalne površine w ,
 - širina risalne površine h ,
 - faktor povečave (merilo) f_p .

Elementi, ki so podčrtani, so podani s strani uporabnika in jih lahko dinamično spreminja, ostali so izračunani.



Slika 27: Diagram poteka programa

Na diagramu 27 je orisan glavni potek celotnega programa. Na vseh nadaljnjih diagramih poteka so z rdečimi paralelogrami prikazani vhodni podatki, ki jih določen korak oz. del programa potrebuje, s pravokotniki pa so označeni posamezni deli programa. Zaobljeni pravokotnik predstavlja začetek ali konec programa, romb pa odločitev programa.



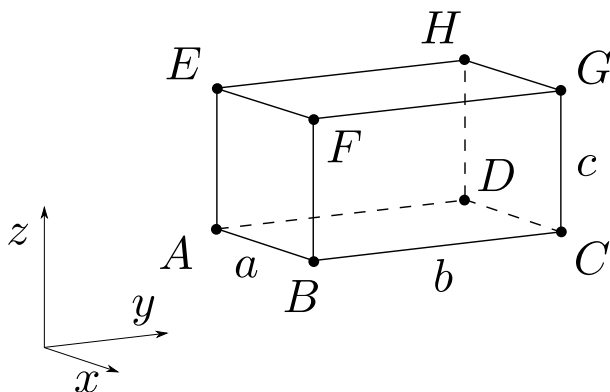
Slika 28: Diagram podprograma *projiciranje*

3.2.1 Razbitje na daljice/točke

V programu bi lahko izrisali katerikoli geometrijski objekt, vendar sem se za potrebe prikazovanja odločil za nekaj osnovnih objektov, in sicer: točka, daljica, trikotnik, krožnica, kvader in ravninska krivulja. Vsak objekt moramo razbiti na daljice, katerih krajišča potem projiciramo. Enako moramo storiti tudi pri neoglatih likih ter gladkih krivuljah (krog; kvadratna funkcija, trigonometrične funkcije ...), kjer zvezno funkcijo pretvorimo v diskretne točke in daljice med njimi (diskretizacija).

Kvader Kvader je podan s točko $A(x_A, y_A, z_A)$ in dolžino robov a, b, c . Realne koordinate posameznih oglišč v prostoru se izračunajo:

$$\begin{aligned} A(x_A, y_A, z_A) \\ B(x_A + a, y_A, z_A) \\ C(x_A + a, y_A + b, z_A) \\ D(x_A, y_A + b, z_A) \\ E(x_A, y_A, z_A + c) \\ F(x_A + a, y_A, z_A + c) \\ G(x_A + a, y_A + b, z_A + c) \\ H(x_A, y_A + b, z_A + c) \end{aligned}$$



Slika 29: Kvader

Iz njih se tvorijo daljice (robovi) $AB, BC, CD, DA, AE, BF, CG, DH, EF, FG, GH$ in HE .

Krožnica Krožnico podamo s polmerom R in njenim središčem $S(p, q, r)$. Zaradi poenostavitve naj bo krožnica vzporedna z eno izmed osnovnih ravnin. Enačbe različnih krožnic so tako:

- vzporedna z ravnino xy (slika 30)

$$(x - p)^2 + (y - q)^2 = R^2$$

$$z = r$$

- vzporedna z ravnino xz

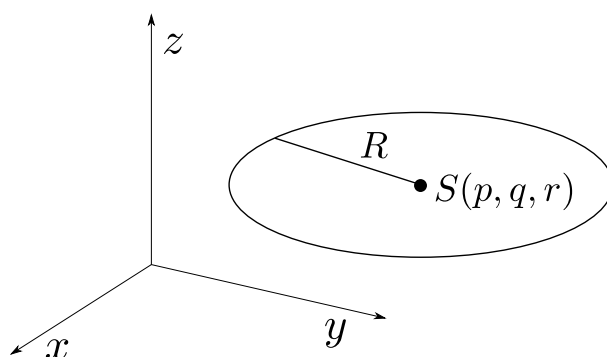
$$(x - p)^2 + (z - r)^2 = R^2$$

$$y = q$$

- vzporedna z ravnino yz

$$(y - q)^2 + (z - r)^2 = R^2$$

$$x = p$$



Slika 30: Krožnica

Krožnico najprej razbijemo na poljubno mnogo (n) točk tako, da s povečevanjem kota α od 0° do 360° za poljubno vrednost zaporedno izračunavamo koordinate (slika 31):

- vzporedna z ravnino xy

$$T_i(R \cos \alpha_i + p, R \sin \alpha_i + q, r)$$

- vzporedna z ravnino xz

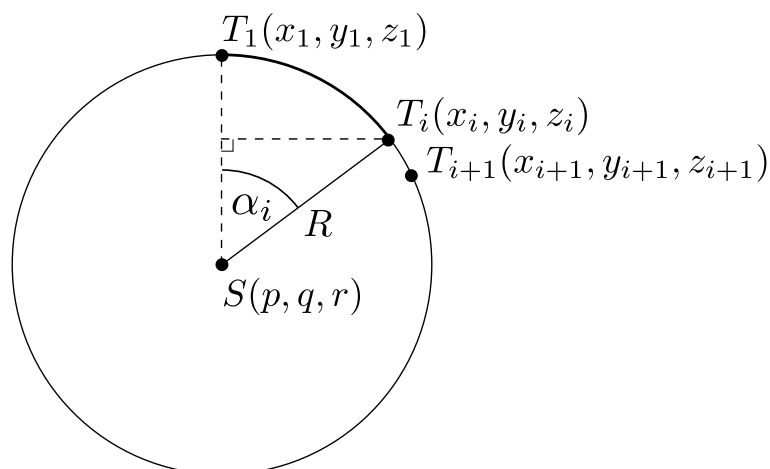
$$T_i(R \cos \alpha_i + p, q, R \sin \alpha_i + r)$$

- vzporedna z ravnino yz

$$T_i(p, R \cos \alpha_i + q, R \sin \alpha_i + r)$$

Dobljene točke nato samo še zaporedno povežemo v daljice:

$$T_1T_2, T_2T_3, T_3T_4, \dots, T_{n-1}T_n, T_nT_1 \quad (3.7)$$



Slika 31: Risanje krožnice

Funkcije Zaradi poenostavitve obravnavamo le funkcije, katerih grafi ležijo vzporedno z ravnino yz ($x = p$). Enačbe grafov podamo z vektorjem premika $\vec{v} = (p, q, r)$ in raztegoma v smeri osi y (s faktorjem n) ter osi z (s faktorjem m) (samo sinusna funkcija). Enačbe grafov funkcij so tako:

- kvadratna

$$z = \left(\frac{y - q}{n}\right)^2 + r$$

- kubična

$$z = \left(\frac{y - q}{n}\right)^3 + r$$

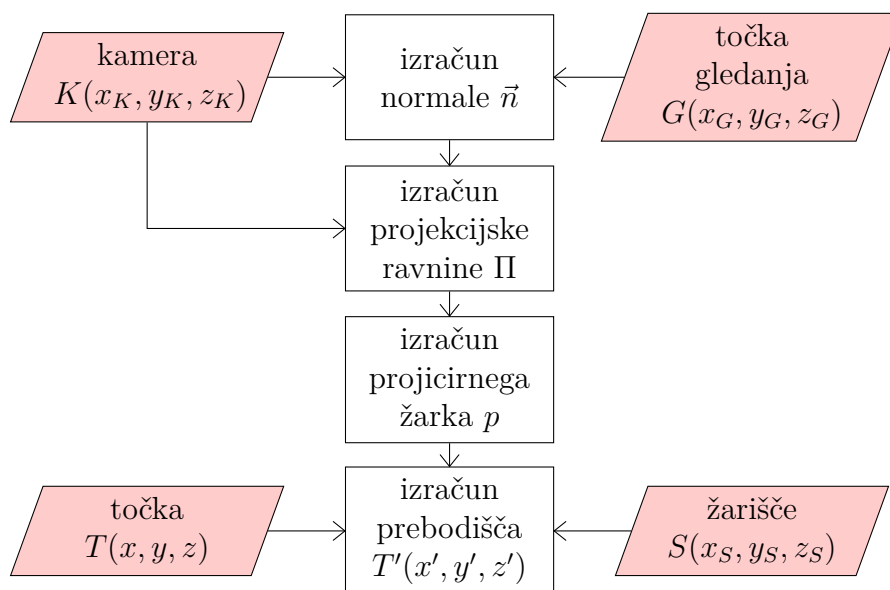
- sinusna

$$z = m \sin\left(\frac{y - q}{n}\right) + r$$

Točke grafa nato izračunamo s pomočjo programske zanke, kjer y povečujemo od poljubne začetne do poljubne končne vrednosti za nek korak, pri tem pa vsakič izračunamo pripadajočo vrednost funkcije (koordinata z). Daljice dobimo kot v formuli 3.7, le da izpustimo zadnjo ($T_n T_1$).

3.2.2 Projiciranje točk

V nadaljevanju bom v opisu postopka uporabljal samo eno točko objekta (T), program pa pravzaprav celoten postopek ponovi za vse točke objektov, ki jih projiciramo. Kjer se bom nanašal na množico vseh točk vseh objektov, bom uporabljal $\mathcal{T} = \{T_1, T_2, T_3 \dots\}$.



Slika 32: Diagram projiciranja točk

Izračun normale Normalo ravnine predstavlja vektor od kamere do točke gledanja (slika 33):

$$\vec{n} = \vec{KG}$$

Vzporedno ga premaknemo do izhodišča (torej za vektor $-\vec{r}_K$).

$$G \mapsto G'(x_G - x_K, y_G - y_K, z_G - z_K)$$

$$\vec{n} = \vec{OG'}$$

Komponentno obliko vektorja, ki jo potrebujemo za nadaljevanje, dobimo iz koordinat točke G' :

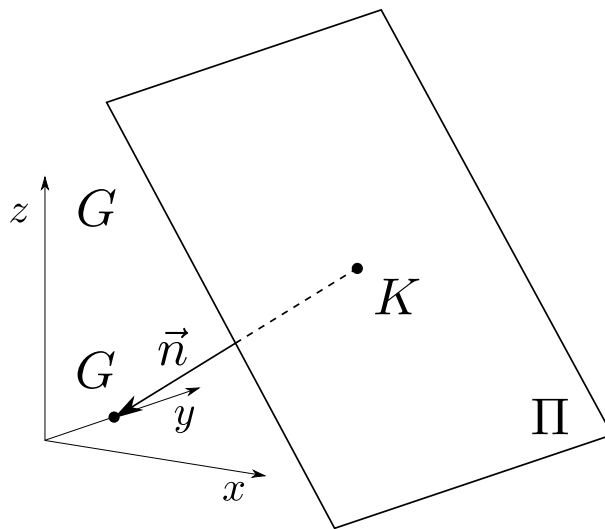
$$\vec{n} = (x'_G, y'_G, z'_G)$$

Izračun projekcijske ravnine

$$\vec{n} = (a, b, c)$$

$$\Pi : ax + by + cz - D = 0$$

$$D = ax_K + by_K + cz_K$$



Slika 33: Normala projekcijske ravnine

Izračun projicirnega žarka Projicirni žarek izračunavamo različno glede na tip želene projekcije.

Perspektivna projekcija:

Projicirni žarek poteka skozi točki T in S (sečišče se izračuna glede na goriščno razdaljo; glej odstavek *Goriščna razdalja* na strani 44).

$$p : \frac{x - x_T}{x_S - x_T} = \frac{y - y_T}{y_S - y_T} = \frac{z - z_T}{z_S - z_T}$$

Vzporedna pravokotna projekcija:

Projicirni žarki so vzporedni normali ravnine \vec{n} .

$$p : \frac{x - x_T}{a} = \frac{y - y_T}{b} = \frac{z - z_T}{c}$$

Vzporedna poševna projekcija:

Projicirni žarki so vzporedni vektorju $\vec{v} = (d, e, f)$, ki ga izračunamo glede na zatezni kot in koeficient skrajšave (glej stran 31).

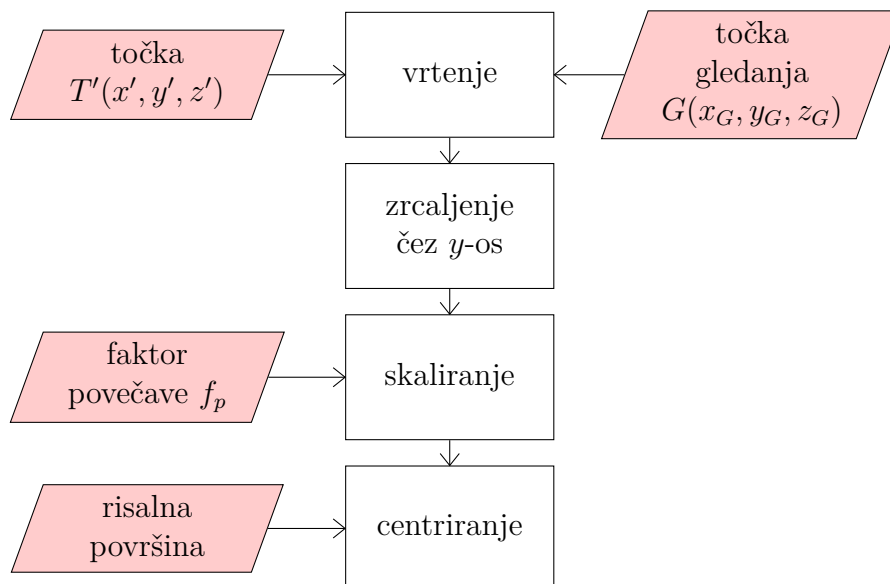
$$p : \frac{x - x_T}{d} = \frac{y - y_T}{e} = \frac{z - z_T}{f}$$

Izračun prebodišča Projekcijo točke T izračunamo tako, da najdemo prebodišče projicirnega žarka s projekcijsko ravnino.

$$p \cap \Pi = \{T'(x', y', z')\}$$

Če projiciramo v perspektivni projekciji, uporabimo formule (3.1), sicer pa formule (3.2).

3.2.3 Pretvorba v zaslonske koordinate



Slika 34: Diagram pretvorbe v zaslonske koordinate

Potek pretvorbe projekcije točke $T'(x', y', z')$ v zaslonsko točko $T_Z(x_Z, y_Z)$ vsebuje več korakov, pri katerih sem vmesne točke, ki jih izračunavamo, označeval z indeksi, in sicer z rimskimi številkami

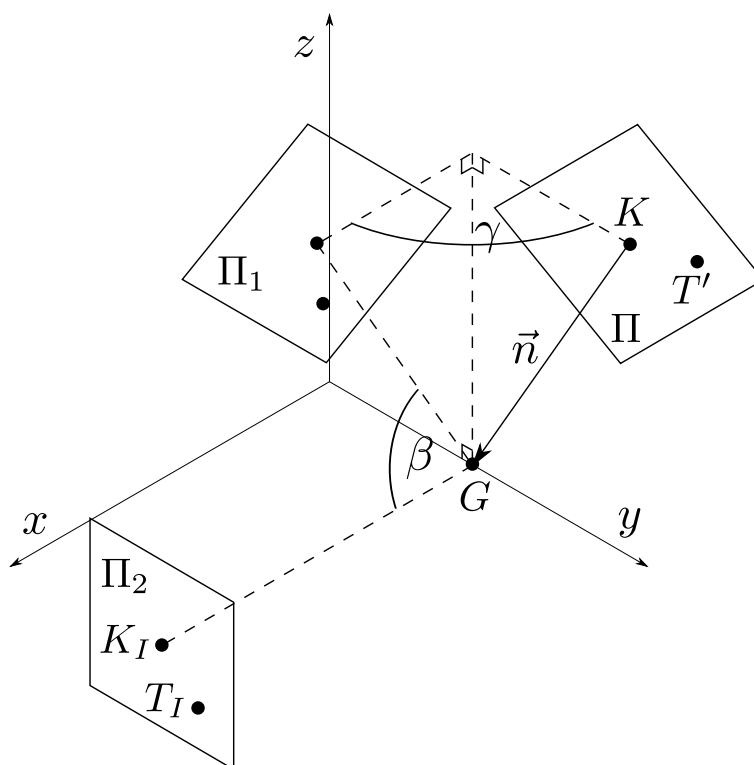
$$T_I(x_I, y_I, z_I), T_{II}(x_{II}, y_{II}), T_{III}(x_{III}, y_{III}), \dots$$

Vrtenje Točko T' zavrtimo okoli točke gledanja G tako, da je ob isti rotaciji točke K vektor \vec{GK} vzporeden z osjo x . Drugače povedano, celotno projekcijsko ravnino zavrtimo okoli točke G tako, da je nova ravnina Π_2 vzporedna z ravnino yz .

Za vrtenje bomo potrebovali dva kota (γ za vrtenje okoli osi z in β za vrtenje okoli osi y), ki ju izračunamo iz prilagojenih sfernih koordinat K glede na G (glej razdelek *Navigacija* na strani 44):

$$\begin{aligned}\gamma &= -\varphi_G \\ \beta &= 90^\circ - \vartheta_G\end{aligned}$$

Točko T' torej zavrtimo z rotacijama $R_z(\gamma)$ in $R_y(\beta)$ (glej algoritem (1)). Naj še enkrat poudarim, da moramo zaradi nekomutativnosti vrtenja v prostoru najprej vrteti okoli osi z in nato okoli osi y .



Slika 35: Rotacija ravnine

Po opravljeni transformaciji ležijo vse točke na ravnini, ki je vzporedna z ravnino yz . Točka T_I , ki je rezultat vrtenja, ima tri koordinate $T_I(x_I, y_I, z_I)$, ker pa za nadaljevanje potrebujemo samo dve koordinati, definiramo novo točko $T_{II}(y_I, z_I)$ tako, da zanemarimo absciso točke T_I , saj so vse abscise vseh projekcij točk po vrtenju zaradi lege na z ravnino yz vzporedni ravnini med seboj enake ($x_{i_{II}} = x_{j_{II}}, \forall T_i(x_{i_{II}}, y_{i_{II}}, z_{i_{II}}), T_j(x_{j_{II}}, y_{j_{II}}, z_{j_{II}}) \in \mathcal{T}_{II}$).

Zrcaljenje Točko na računalniškem zaslonu podamo s koordinatama x in y , vendar je razlika v tem, da y narašča po zaslonu navzdol. Točko T_{II} moramo zato prezrcaliti čez os y .

$$T_{II}(x_{II}, y_{II}) \mapsto T_{III}(x_{II}, -y_{II})$$

Skaliranje Eden od parametrov, ki jih moramo določiti, je tudi merilo (faktor povečave f_p), ki pove dolžino enotskega vektorja na zaslonu (v slikovnih pikah). Na točki T_{III} opravimo središnji razteg (homotetijo) s središčem

v $S(0, 0)$ in koeficientom f_p .

$$T_{III}(x_{III}, y_{III}) \mapsto T_{IV}(x_{III} \cdot f_p, y_{III} \cdot f_p)$$

Centriranje Celotno sliko želimo imeti poravnano tako, da leži točka gledanja G_{IV} na sredini risalne površine. Najprej torej točko G_{IV} premaknemo na sredino risalne površine (w in h sta dimenziji risalne površine; glej seznam točk, uporabljenih med projiciranjem, str. 33):

$$G_{IV}(x_{IV}, y_{IV}) \mapsto G_V \left(\frac{w}{2}, \frac{h}{2} \right)$$

za isti vektor pa premaknemo tudi točko T_{IV}

$$T_{IV}(x_{IV}, y_{IV}) \mapsto T_Z \left(\frac{w}{2} - x_{G_{IV}}, \frac{h}{2} - y_{G_{IV}} \right)$$

3.2.4 Izris

Izrisovanje na zaslon sem opravil s knjižico GDI+, ki je grafična knjižica za uporabo dvodimenzionalne grafike v operacijskih sistemih Microsoft Windows (povzeto po GDI, 2012). Uporabil sem zgolj tri osnovne rutine knjižice, in sicer:

- izris črte,
- izris zapolnjene elipse,
- izpis besedila.

Izris črte Črto določene barve izrišemo tako, da podamo njeno začetno in končno točko. Vse projicirane daljice, like, telesa in funkcije tako izrišemo zgolj z ravnimi črtami, ki pa daje videz gladkosti.

Izris zapolnjene elipse Ta funkcija izriše pobarvano elipso z neko dolžino (velika os) in neko širino (mala os) na določeni točki. Uporabljal sem jo za prikaz točke, pri čemer sem izbral enako dolžino in širino, tako da je nastal krog.

Izpis besedila Funkcijo, ki izpiše željeno besedilo na podani točki, sem uporabljal za označevanje osi in točk.

3.3 Parametri

Goriščna razdalja Goriščna razdalja f je razdalja od sečišča žarkov S do kamere K .

$$\overline{SK} = f$$

$$f = \sqrt{(x_S - x_K)^2 + (y_S - y_K)^2 + (z_S - z_K)^2}$$

Sečišče žarkov izračunamo glede na k in goriščno razdaljo, in sicer tako da množimo krajevni vektor kamere z neko skalarno vrednostjo. To vrednost (k_f) izračunamo sledeče:

$$\sqrt{x_K^2 + y_K^2 + z_K^2} + f = \sqrt{k_f^2 x^2 + k_f^2 y^2 + k_f^2 z^2}$$

po urejanju dobimo

$$k_f = \frac{\sqrt{x_K^2 + y_K^2 + z_K^2} + f}{\sqrt{x_K^2 + y_K^2 + z_K^2}} = 1 + \frac{f}{\sqrt{x_K^2 + y_K^2 + z_K^2}}$$

3.4 Navigacija

Premikanje po prostoru Pri navigaciji se v izračunih uporabljajo prilagojene sferne koordinate točke K , in sicer tako, kot bi za izhodišče vzeli točko G :

- ϑ_G kot med premico, ki je vzporedna osi z ter poteka skozi točko G , in daljico GK ;
- φ_G kot med GK in premico, ki je vzporedna osi x ter poteka skozi točko G ;
- R_G oddaljenost K od G ($R_G = \overline{KG}$).

Kamero premikamo po prostoru s tremi različnimi metodami.

1. vrtenje okoli točke gledanja G

Kliknemo miškin gumb in premaknemo kurzor v želeno smer. Pri tem se K zavrti okoli G (glej sliko 36) za:

$$\begin{aligned}\Delta\vartheta_G &= \alpha = -\Delta X \cdot k_p \\ \Delta\varphi_G &= \beta = -\Delta Y \cdot k_p\end{aligned}$$

k_p je prednastavljen koeficient, ki predstavlja razmerje med kotno enoto in slikovno piko na zaslonu. Koordinate nove točke K izračunamo po

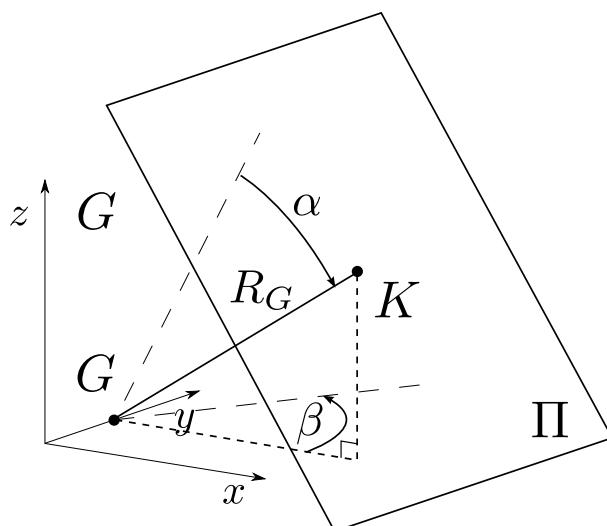
algoritmu za vrtenje točke okoli točke (glej algoritem 1).

Pri izometrični projekciji onemogočimo vrtenje kamere, saj morajo biti koordinate kamere enake ($x_K = y_K = z_K$). Pri dimetrični projekciji pa omejimo vrtenje kamere tako, da je omogočena samo sprememba koordinate φ_G ; s tem dosežemo, da sta abscisa in ordinata kamere ves čas enaki ($x_K = y_K$).

2. približevanje in oddaljevanje (zoom)

Ko zavrtimo miškin kolesček za en premik, se poveča oddaljenost točke K od G za neko vnaprej določeno vrednost (k_z):

$$\Delta R_G = \Delta z_{asuk} \cdot k_z$$



Slika 36: Vrtenje kamere

3. premik kamere

S klikom na desni miškin gumb in premikanjem kurzorja premikamo kamero za vektor premika \vec{v} , ki leži na Π . Hkrati se za isti vektor premakne točka G (glej sliko 37).

$$K_2 = K + \vec{v}$$

$$G_2 = G + \vec{v}$$

Vektor premika izračunamo na sledeč način:

- uvedemo začetni vektor premika $\vec{v}_0 = K'\vec{K}'_2$

$$K'_2(x_{K'}, -\Delta X \cdot f_p, \Delta Y \cdot f_p)$$

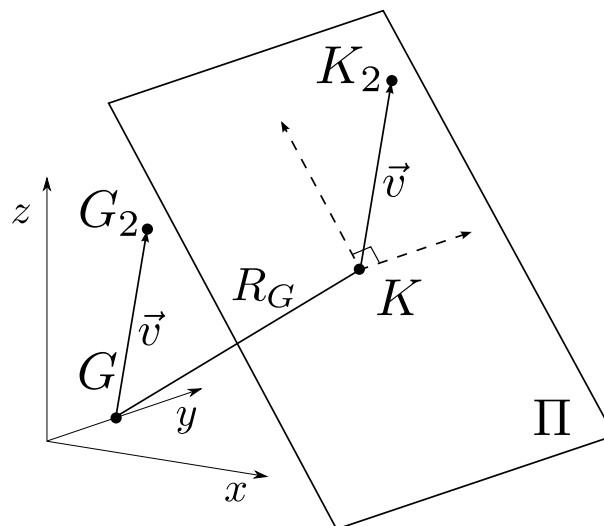
pri čemer je ΔX enak premiku miške v vodoravni smeri in ΔY premiku miške v navpični smeri; $K'(x_{K'}, y_{K'}, z_{K'})$ je projekcija točke K' ;

- krajišči vektorja \vec{v}_0 zavrtimo okoli točke gledanja G (glej formule (1)) za:

$$\gamma = \vartheta - 90^\circ$$

$$\beta = \varphi$$

pri čemer sta ϑ in φ sferni koordinati točke K . To vrtenje je pravzaprav nasprotno vrtenju, ki ga izvedemo po projiciranju (glej odstavek *Vrtenje* na strani 41).



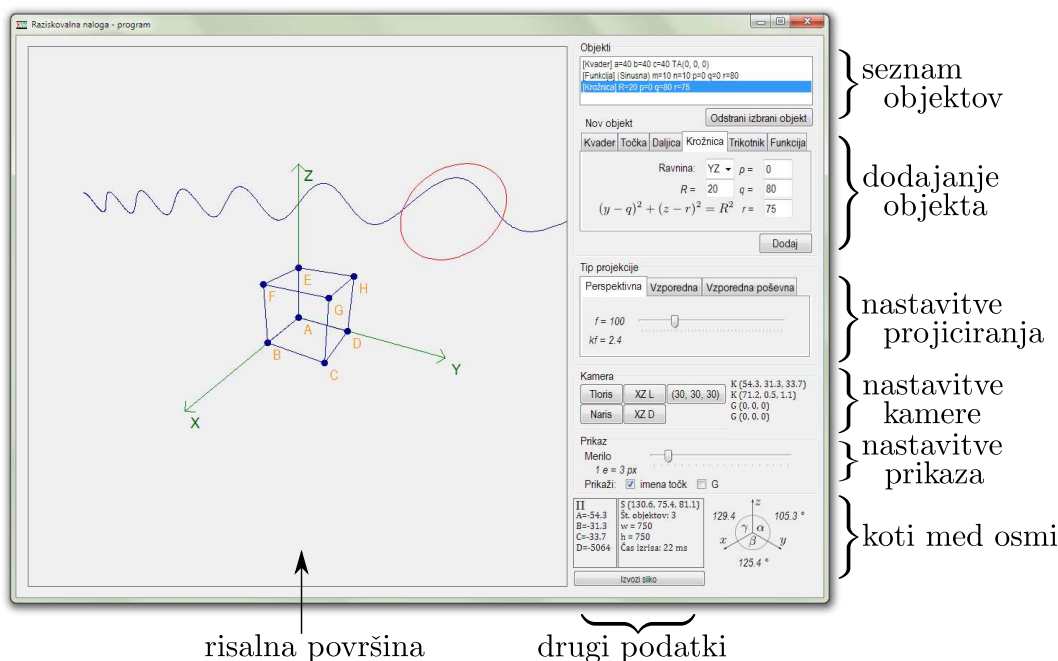
Slika 37: Premik kamere

4 KONČNI IZDELEK

4.1 Prikazno okno in nastavitve

Nastali program je sestavljen iz enega programskega okna, v katerem so hkrati prikazane nastavitve in nastala projicirana slika. V levem delu okna

se nahaja risalna površina, na desni strani pa so nastavitve (glej sliko 38). Velikost okna lahko poljubno spreminjamo, s čimer se spreminja tudi velikost risalne površine. Program ima dodano možnost izvoza slike, ki je prikazana na risalni površini. Te slike so prikazane v podpoglavju *Projekcije*.



Slika 38: Okno programa

4.1.1 Objekti

V seznam objektov, ki jih projiciramo, lahko dodamo šest različnih tipov objektov (kvader, točka, daljica, krožnica, trikotnik in funkcija). Vsakemu objektu lahko nastavimo poljubne lastnosti.

4.1.2 Nastavitve projekcije

Nastavitve projekcije se najprej deli na tri glavne tipe: perspektivna, vzporedna pravokotna in vzporedna poševna. Pri perspektivni projekciji lahko nastavimo poljubno goriščno razdaljo, pri vzporedni pravokotni lahko omejimo premikanje kamere (trimetrična, dimetrična in izometrična projekcija), pri vzporedni poševni projekciji pa lahko nastavimo poljuben zatezni kot in koeficient skrajšave.

4.1.3 Nastavitve kamere

Kamero lahko premikamo z miško (glej podpoglavje *Navigacija*) ali pa izberemo enega izmed prednastavljenih položajev:

- tloris
 $K(0, 0, 30)$
- naris
 $K(30, 0, 0)$
- levi stranski ris
 $K(0, -30, 0)$
- desni stranski ris
 $K(0, 30, 0)$
- privzeti položaj
 $K(30, 30, 30)$

Pri izbiri katerega izmed prednastavljenih položajev se tudi točka gledanja G premakne do izhodišča

$$G(0, 0, 0)$$

Ob nastavitvah položaja kamere se nam izpisujejo kartezične in sferne koordinate točk T in G .

4.1.4 Nastavitve prikaza

Izberemo lahko poljubno merilo, vklopimo ali izklopimo prikaz imena točk ter vklopimo in izklopimo prikaz točke G .

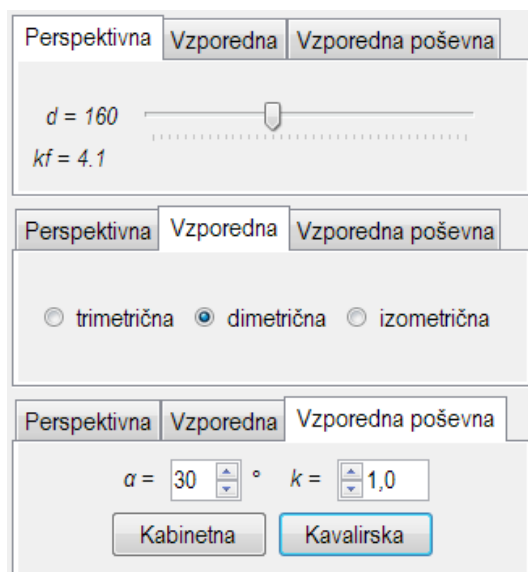
4.1.5 Drugi podatki

V dveh okvirčkih na dnu okna se izpisujejo:

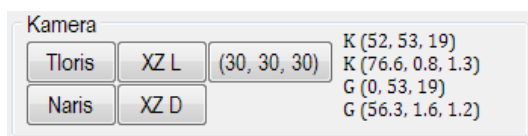
- koeficienti projekcijske ravnine (a , b , c in d);
- sečišče žarkov S (samo pri perspektivni projekciji);
- število objektov, ki jih projiciramo;
- velikost risalne površine in
- čas, ki ga je program potreboval za izračun in prikaz objektov (čas izvajanja podprograma *projiciranje*).

Kvader	Točka	Daljica	Krožnica	Trikotnik	Funkcija	
	$XA =$	<input type="text" value="10"/>	$a =$	<input type="text" value="40"/>		
	$YA =$	<input type="text" value="0"/>	$b =$	<input type="text" value="15"/>		
	$ZA =$	<input type="text" value="-20"/>	$c =$	<input type="text" value="40"/>		
Kvader	Točka	Daljica	Krožnica	Trikotnik	Funkcija	
	Ime točke: <input type="text" value="A"/>		$X =$	<input type="text" value="-50"/>		
			$Y =$	<input type="text" value="35"/>		
			$Z =$	<input type="text" value="30"/>		
Kvader	Točka	Daljica	Krožnica	Trikotnik	Funkcija	
	$XA =$	<input type="text" value="-40"/>	$XB =$	<input type="text" value="50"/>		
	$YA =$	<input type="text" value="-20"/>	$YB =$	<input type="text" value="30"/>		
	$ZA =$	<input type="text" value="0"/>	$ZB =$	<input type="text" value="20"/>		
Kvader	Točka	Daljica	Krožnica	Trikotnik	Funkcija	
	Ravnina: <input type="text" value="XZ"/>	$p =$	<input type="text" value="10"/>			
		$R =$	<input type="text" value="20"/>	$q =$	<input type="text" value="70"/>	
		$(x - p)^2 + (z - r)^2 = R^2$	$r =$	<input type="text" value="20"/>		
Kvader	Točka	Daljica	Krožnica	Trikotnik	Funkcija	
	$XA =$	<input type="text" value="0"/>	$XB =$	<input type="text" value="10"/>	$XC =$	<input type="text" value="-20"/>
	$YA =$	<input type="text" value="-20"/>	$YB =$	<input type="text" value="30"/>	$YC =$	<input type="text" value="-10"/>
	$ZA =$	<input type="text" value="-60"/>	$ZB =$	<input type="text" value="20"/>	$ZC =$	<input type="text" value="25"/>
Kvader	Točka	Daljica	Krožnica	Trikotnik	Funkcija	
	$p =$	<input type="text" value="0"/>	$z = m \sin\left(\frac{y - q}{n}\right) + r$			
	$q =$	<input type="text" value="0"/>	$m =$	<input type="text" value="5"/>	Funkcija: <input type="text" value="sinusna"/>	
	$r =$	<input type="text" value="20"/>	$n =$	<input type="text" value="5"/>		

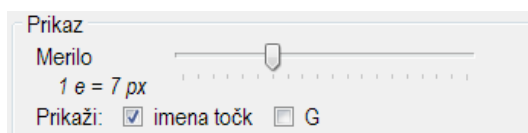
Slika 39: Dodajanje novega objekta



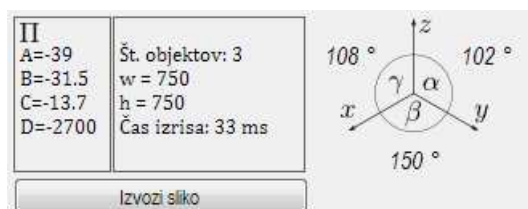
Slika 40: Nastavitve projekcije



Slika 41: Nastavitve kamere



Slika 42: Nastavitve prikaza



Slika 43: Prikaz kotov med osmi in drugi podatki

4.2 Projekcije

V nadaljevanju bom opisal nekaj slik, ki so nastale s programom. Slike so prikazane v podpoglavju *Slike projekcij*.

4.2.1 Kocka

Na slikah od 44 do 51 je prikazana kocka v različnih projekcijah.

4.2.2 Kvader

Slika 52 prikazuje kvader v perspektivni projekciji s prikazanimi imeni točk. Na slikah 53 in 54 je prikazanih nekaj kvadrov v dveh različnih perspektivnih projekcijah (različna goriščna razdalja).

4.2.3 Krožnice

Sliki 55 in 56 prikazujeta krožnice z različnimi premiki in polmeri v izometrični in perspektivni projekciji.

4.2.4 Funkcije

Na slikah 57 in 58 so prikazani trije grafi vseh tipov funkcij, ki jih lahko prikažemo v programu (kvadratna, kubična in sinusna), in sicer v narisu (pravokotna projekcija) ter izometrični projekciji. Nobena funkcija ni premaknjena, kvadratna in kubična sta raztegnjeni v smeri osi y s koeficientom 5, sinusna pa je raztegnjena v smeri osi y in z s koeficientom 5.

Slike 59–62 prikazujejo projekcije grafov kvadratnih in sinusnih funkcij z različnimi premiki in raztegi.

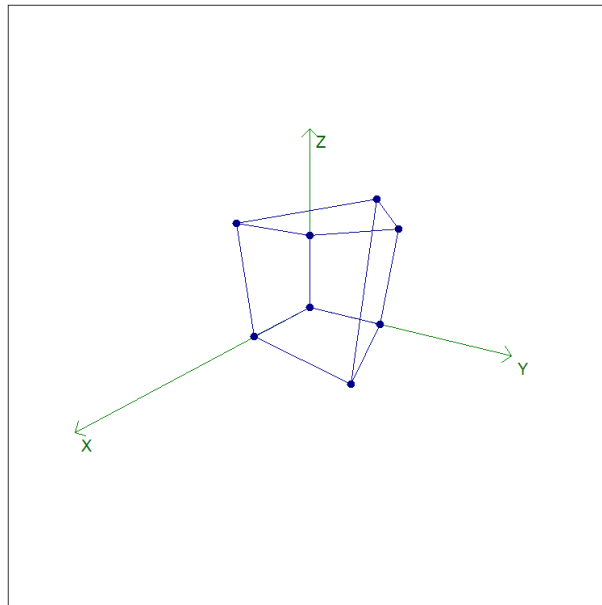
4.2.5 Slabost vzporednih projekcij

Na slikah od 64 do 67 sem prikazal slabost vzporednih projekcij, in sicer nejasnost višine in oddaljenosti objekta. To sem ponazoril s tremi krožnicami (rdeča, modra in oranžna), ki so vzporedne ravnini xy . Na sliki 64, kjer je izbrana izometrična projekcija, se zdi, da so središča vseh treh krožnic kolinearna. Ko kamero premaknemo v pogled tlorisa, lahko vidimo, da to ne drži in da je ordinata središča oranžne krožnice večja od ostalih dveh. To lahko vidimo tudi v trimetrični in perspektivni projekciji, pri kateri je oranžna krožnica tudi večja, kar nakazuje, da se nahaja bližje kameri kot modra.

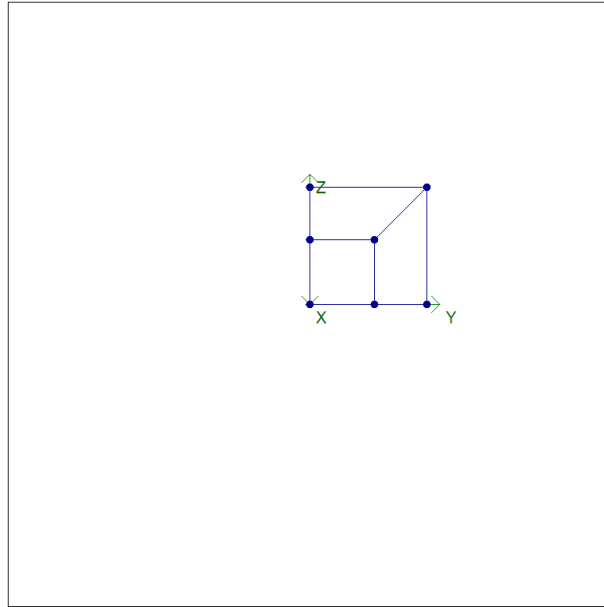
4.3 Pomanjkljivost programa

Ena izmed pomanjkljivosti programa je odsotnost *clippinga*, kar pomeni, da se na zaslonu prikažejo vsi objekti, ki jih projiciramo, torej tudi tisti, ki se nahajajo za projekcijsko ravnino in jih sicer ne bi videli. Ta slabost se pokaže pri perspektivni projekciji, kar lahko vidimo na sliki 63. Prikazana sta kvader z veliko dolžino (v smeri osi y) in sinusna funkcija, ki sta videti deformirano, saj so se projicirali tudi deli za projekcijsko ravnino.

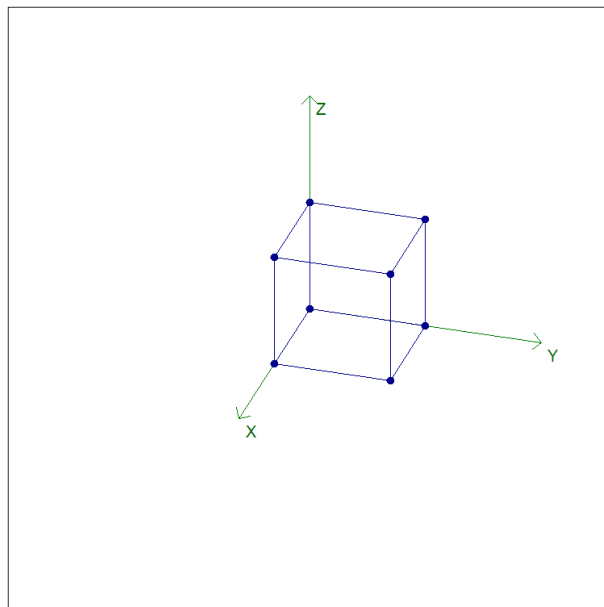
4.4 Slike projekcij



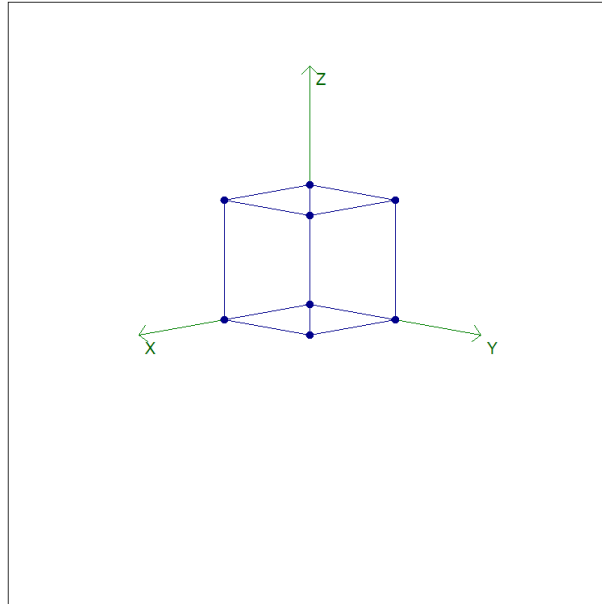
Slika 44: Kocka v perspektivni projekciji



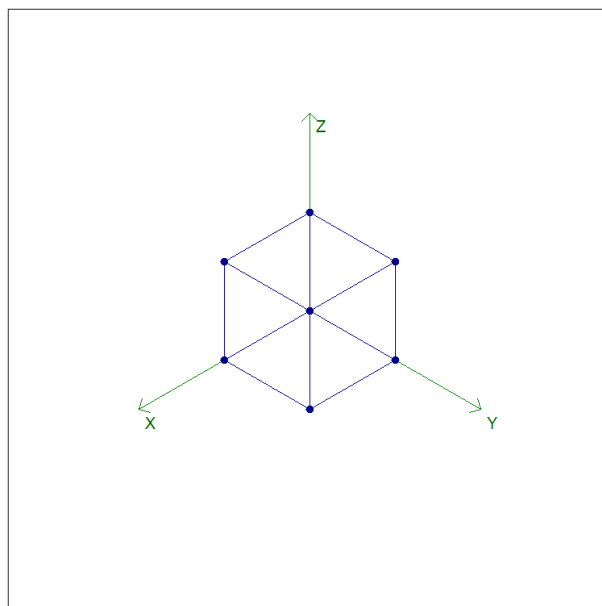
Slika 45: Kocka v perspektivni projekciji (pogled od spredaj)



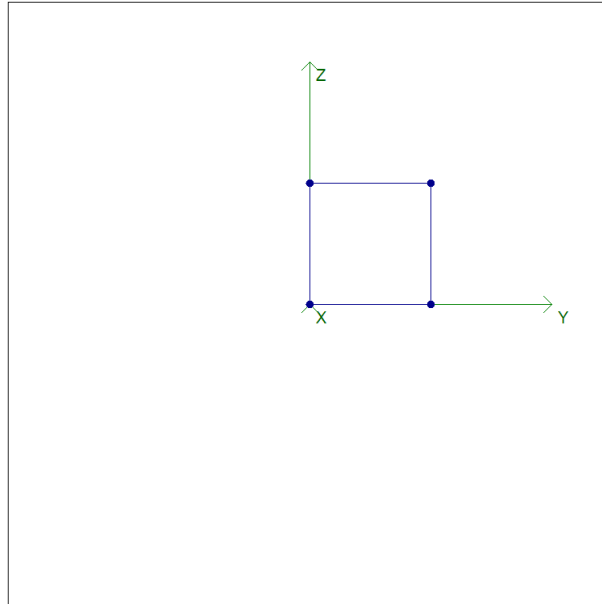
Slika 46: Kocka v trimetrični projekciji



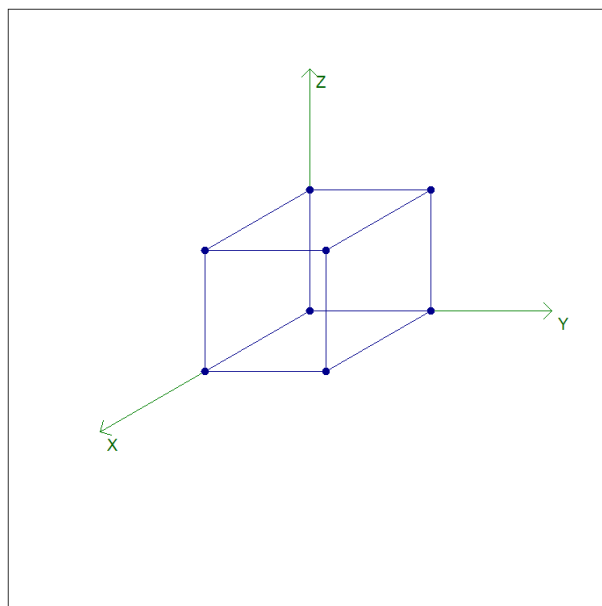
Slika 47: Kocka v dimetrični projekciji



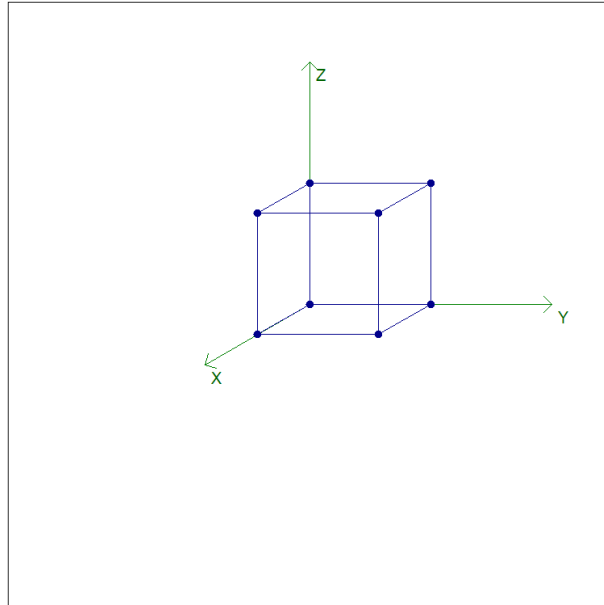
Slika 48: Kocka v izometrični projekciji



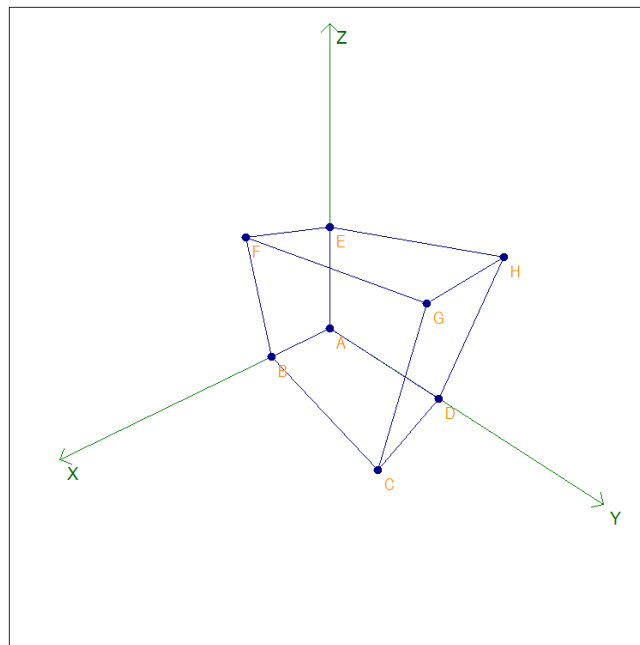
Slika 49: Kocka v pravokotni projekciji (naris)



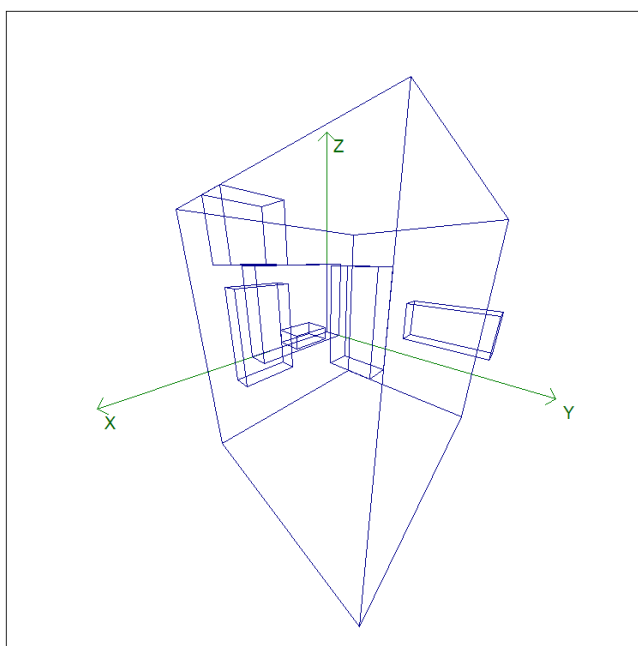
Slika 50: Kocka v kavalirski projekciji ($\alpha = 30^\circ$)



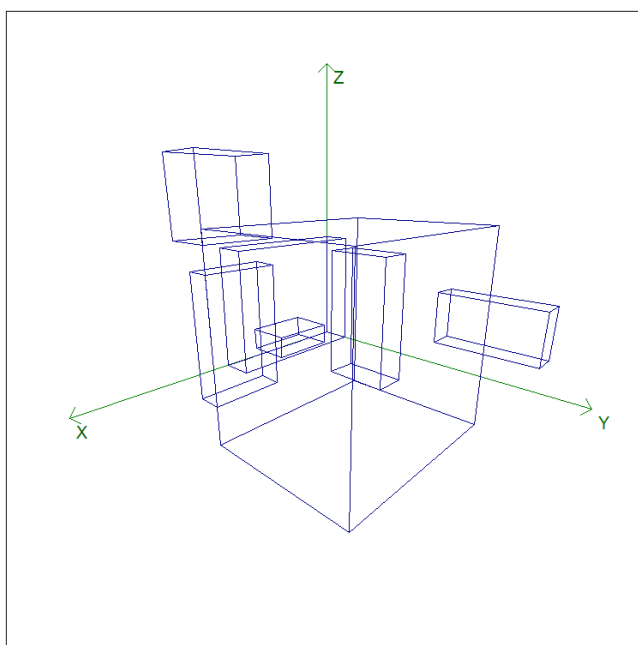
Slika 51: Kocka v kabinetni projekciji ($\alpha = 30^\circ$)



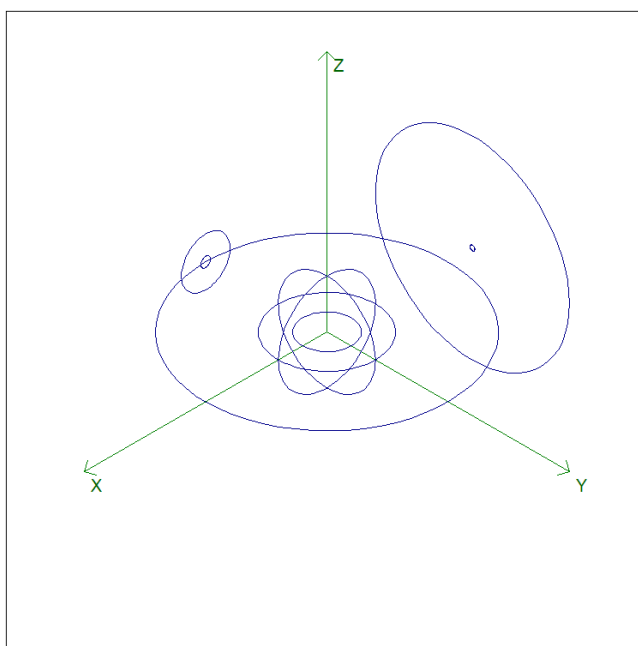
Slika 52: Kvader v perspektivni projekciji



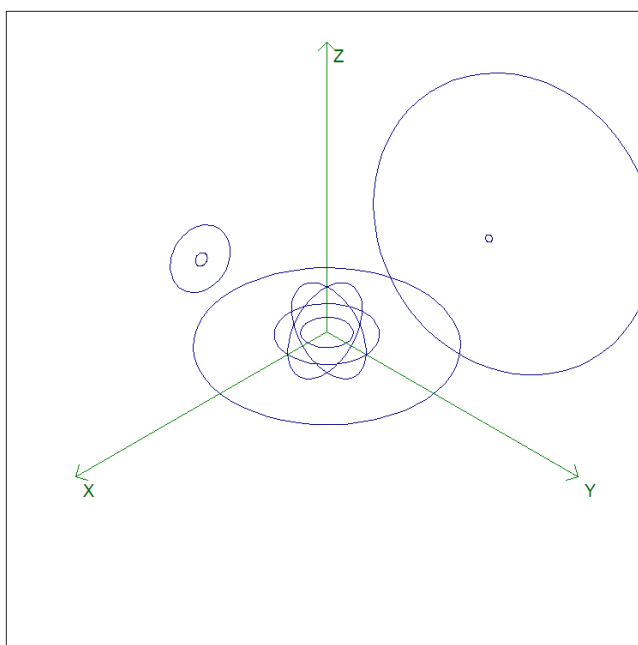
Slika 53: Kvadri v perspektivni projekciji ($f = 50$)



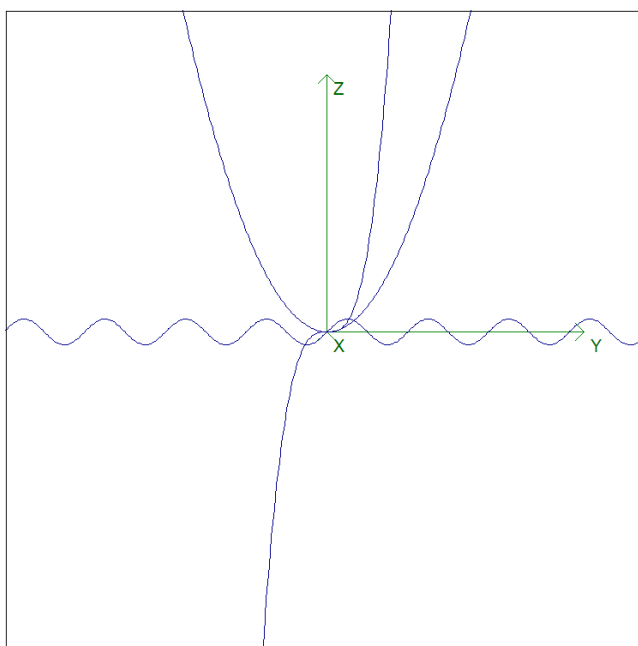
Slika 54: Kvadri v perspektivni projekciji ($f = 100$)



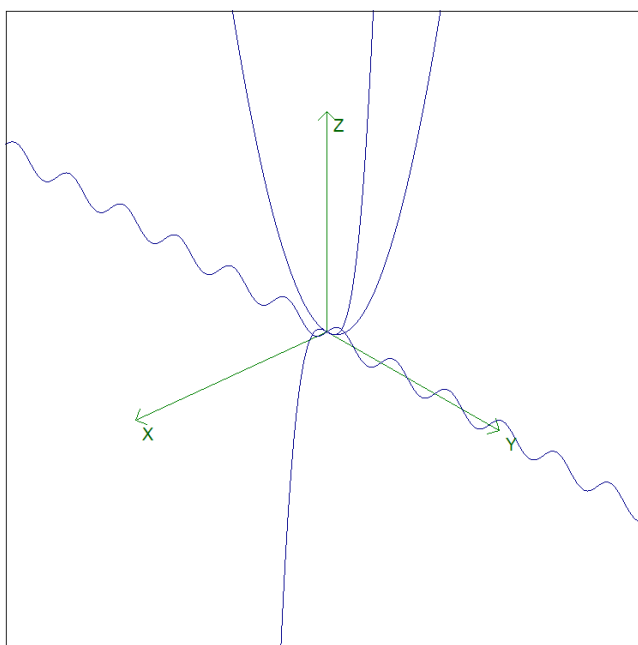
Slika 55: Krožnice v izometrični projekciji



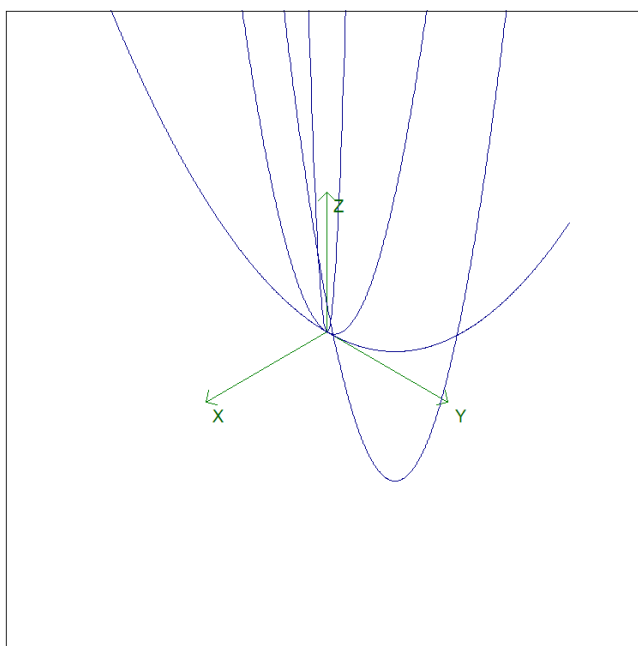
Slika 56: Krožnice v perspektivni projekciji



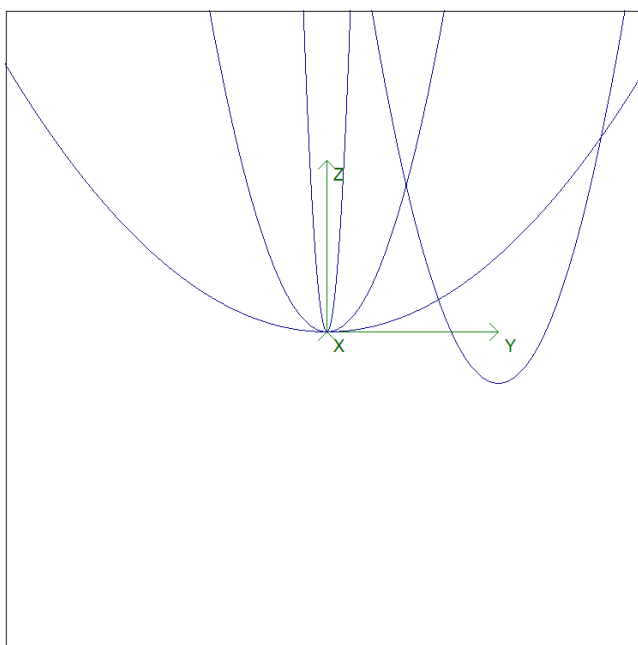
Slika 57: Grafi funkcij v pravokotni projekciji (naris)



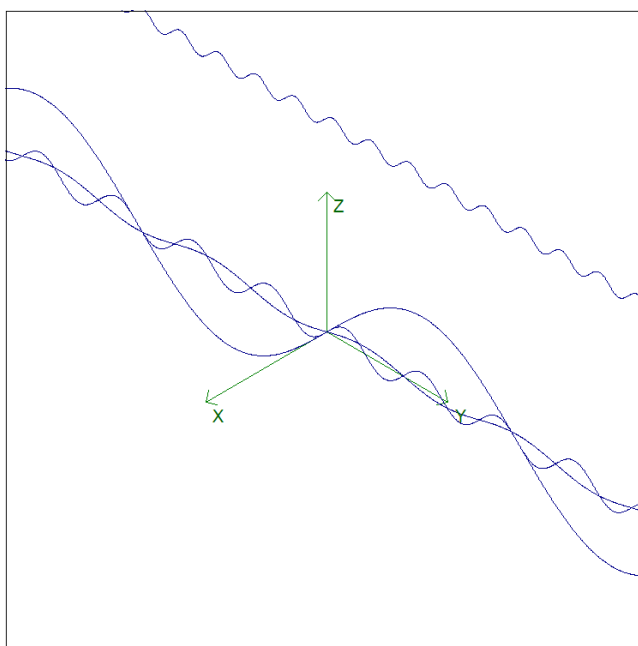
Slika 58: Grafi funkcij v izometrični projekciji



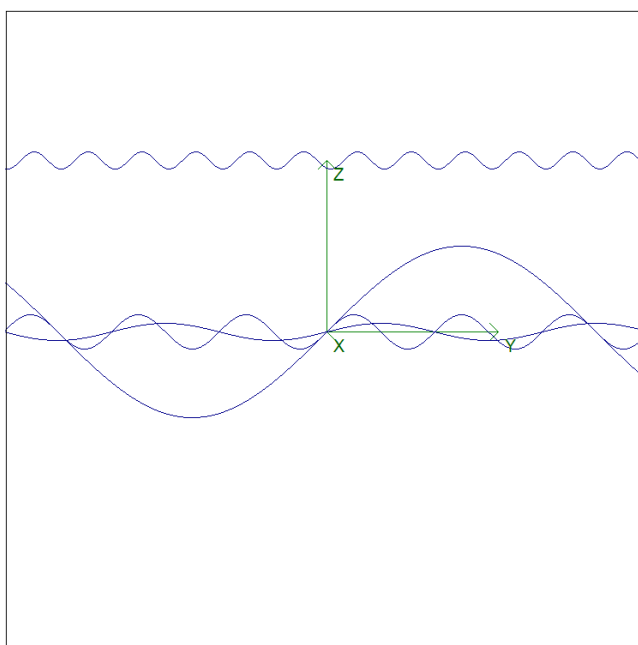
Slika 59: Grafi kvadratnih funkcij v izometrični projekciji



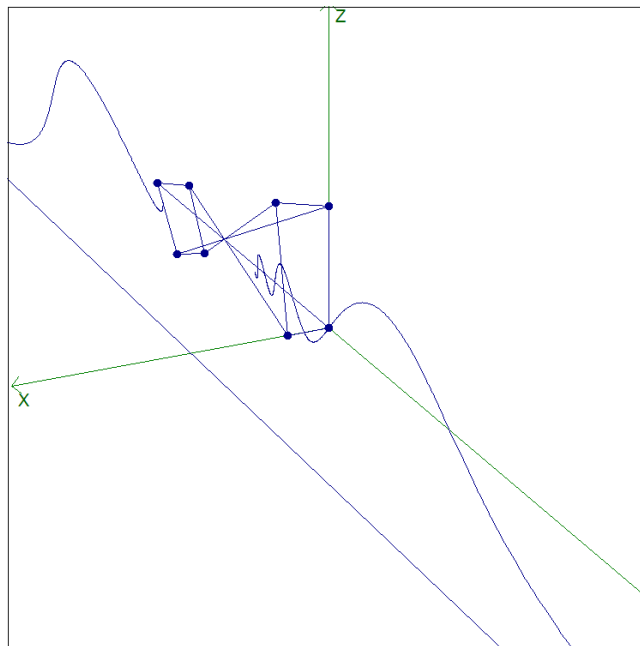
Slika 60: Grafi kvadratnih funkcij v pravokotni projekciji (naris)



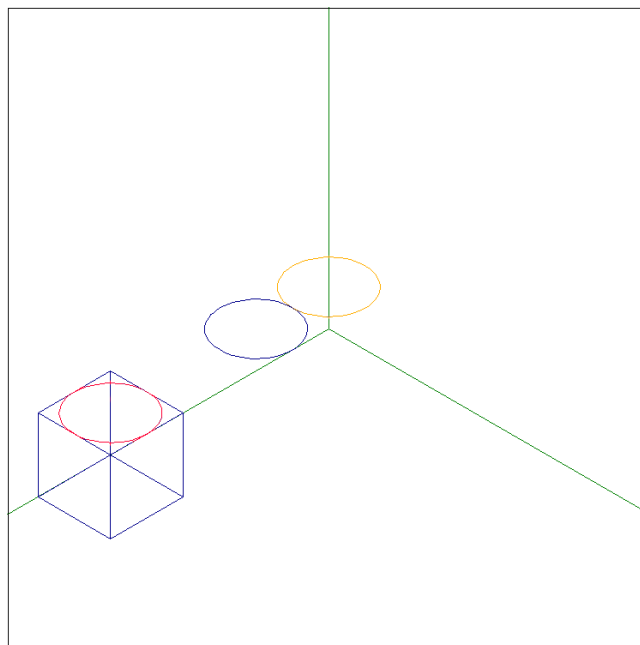
Slika 61: Grafi sinusnih funkcij v izometrični projekciji



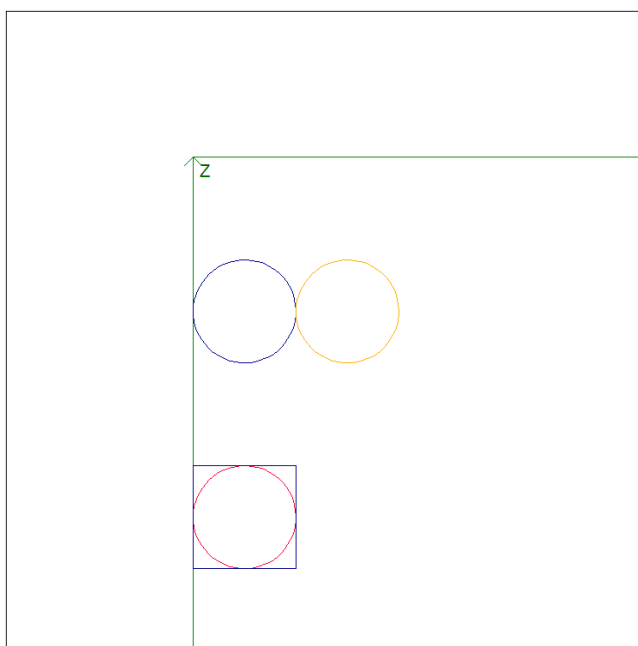
Slika 62: Grafi sinusnih funkcij v pravokotni projekciji (naris)



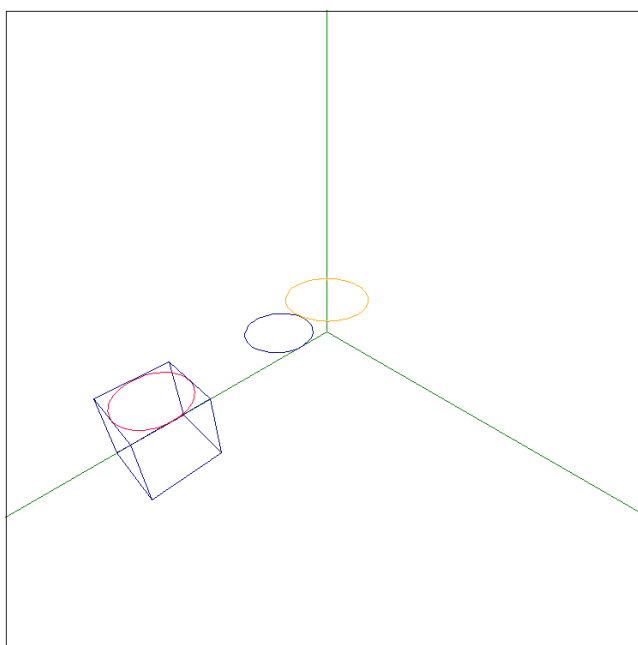
Slika 63: Pomanjkljivost programa pri perspektivni projekciji



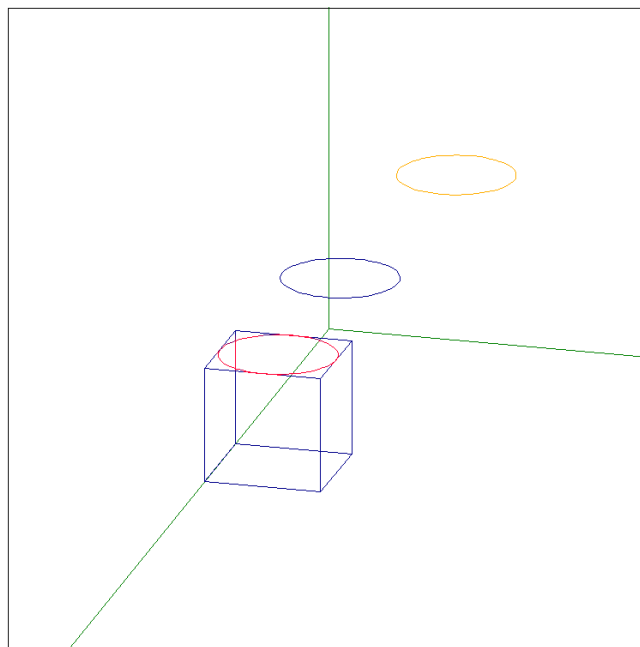
Slika 64: Krožnice v izometrični projekciji



Slika 65: Krožnice v pravokotni projekciji (tloris)



Slika 66: Krožnice v perspektivni projekciji



Slika 67: Krožnice v trimetrični projekciji

5 ZAKLJUČEK

Cilj moje naloge je bil ustvariti program, ki bo geometrijske objekte v prostoru prikazal na računalniški zaslon, kar mi je tudi uspelo. Nastali program sicer ni namenjen prikazovanju zahtevnejših objektov ali velikega števila objektov, saj algoritem ni optimiran. Program bi lahko izboljšal predvsem na nivoju uporabniškega vmesnika in nastavitev, prav tako bi za prikazovanje na zaslonu lahko uporabil katero izmed naprednejših knjižic, ki omogočajo tudi glajenje (ang. *anti-aliasing*), kar bi proizvedlo kvalitetnejšo končno sliko. Pomembna izboljšava bi bila tudi uvedba rezanja (ang. *clipping*) pri perspektivni projekciji, kar bi pomenilo, da se objekti, ki se nahajajo za projekcijsko ravnino, ne bi prikazali na zaslonu. Hkrati bi lahko program prilagodil do te mere, da bi bil uporaben pri pouku matematike, saj bi dijakom zagotovo olajšal dojemanje in razumevanje geometrije v prostoru.

Obe predvidevanji, ki sem ju imel, sem preveril in izkazali sta se za resnični. Opisal sem različne tipe projekcij in primerjal njihove pogloblitve značilnosti, pri tem pa ugotovil, da ima vsak tip projekcije določene prednosti in slabosti.

6 DRUŽBENA ODGOVORNOST

Tematika, ki sem jo obravnaval v raziskovalni nalogi, je med dijaki malo poznana, ker pa je prisotna pri različnih predmetih, bi bilo dobro, da bi dijake o njej bolje poučili. S programom, ki sem ga sestavil, bi jim projekcije lahko lažje predstavili in poskrbeli za večje razumevanje.

7 VIRI IN LITERATURA

Literatura

- [1] CEDILNIK, A. 2006. *Matematični priročnik*. Radovljica: Didakta.
- [2] ČERMELJ, P. 1999. *Premica in ravnina v 3D-prostoru*. [Online]. Ljubljana: Fakulteta za strojništvo Univerze v Ljubljani. Dostopno na naslovu:
<http://www2.lecad.si/vaje/resitve/10.11/index.htm> [22. 12. 2012]
- [3] GDI. 2012. *About GDI+ (Windows)* [Online]. Dostopno na URL naslovu: [http://msdn.microsoft.com/en-us/library/windows/desktop/ms533798\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms533798(v=vs.85).aspx) [7. 1. 2013]
- [4] GUID, N. 2009. *Računalniška grafika*. [Online]. Dostopno na URL naslovu: http://graph-srv.uni-mb.si/cgai/slo/RG_dokumenti/RG-Projekcije.pdf [22. 12. 2012]
- [5] PREBIL, I. 2002. *Opisna geometrija*. Ljubljana: Fakulteta za strojništvo.
- [6] PROSEN, M. 1999. *Vektorji ; Merjenje v geometriji*. Ljubljana: DZS.
- [7] SAJOVIC, O. 1975. *Terminološki slovar opisne geometrije*. Ljubljana: Državna založba Slovenije.
- [8] SRŠEN, L. 1989. *Opisna geometrija*, Ljubljana: Fakulteta za strojništvo Univerze E. Kardelja v Ljubljani.

Slikovno gradivo

- [9] DIMETRIČNA. [Online]. Dostopno na URL naslovu: http://www.educa.fmf.uni-lj.si/izodel/sola/2002/di/brecl/gradivo/TVA1_datoteke/image130.jpg [15. 1. 2013]
- [10] HOGARTH. [Online]. Dostopno na URL naslovu: http://en.wikipedia.org/wiki/Satire_on_False_Perspective [15. 1. 2013]
- [11] HIŠA. [Online]. Dostopno na URL naslovu: <http://gluedideas.com/content-collection/>

- Radfords-cyclopedia-of-construction-Vol-1-mechanical-drafting/
images/Pictorial-Drawing-1101.jpg [15. 1. 2013]
- [12] IZOMETRIČNA. [Online]. Dostopno na URL naslovu: http://www.educa.fmf.uni-lj.si/izodel/sola/2002/di/brecl/gradivo/TVA1_datoteke/image128.jpg [15. 1. 2013]
- [13] MARIO. [Online]. Dostopno na URL naslovu: <http://www.emunova.net/img/tests/106.gif> [15. 1. 2013]
- [14] PAPERBOY. [Online]. Dostopno na URL naslovu: http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/projection/projection_images/paperboy.png [15. 1. 2013]
- [15] PENROSE. [Online]. Dostopno na URL naslovu: http://upload.wikimedia.org/wikipedia/commons/3/34/Impossible_staircase.svg [15. 1. 2013]
- [16] PORTAL. [Online]. Dostopno na URL naslovu: <http://learningworksforkids.com/wp-content/uploads/portal2-screenshots-7.jpg> [15. 1. 2013]
- [17] RAPHAEL. [Online]. Dostopno na URL naslovu: http://upload.wikimedia.org/wikipedia/commons/9/94/Sanzio_01.jpg [15. 1. 2013]
- [18] SIMS. [Online]. Dostopno na URL naslovu: <http://interactive.usc.edu/thesis2007/papers/barfly/800px-TheSims.jpg> [15. 1. 2013]
- [19] VIŠINA. [Online]. Dostopno na URL naslovu: http://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/IsometricFlaw_2.svg/800px-IsometricFlaw_2.svg.png [15. 1. 2013]
- [20] VRIES. [Online]. Dostopno na URL naslovu: <http://www.relewis.com/img/devries-1-16.jpg> [15. 1. 2013]