# Naloga 4 – Geometrijske transformacije in projekcija

Bistven del grafičnega cevovoda je senčilnik oglišč. Njegova glavna naloga je transformacija 3D koordinat oglišč v prostor zaslona za namen raterizacije. Ta korak se praviloma izvede tako, da se položaji oglišč v senčilniku oglišč zmnožijo z matriko PVM. V matriko PVM zapišemo položaj in orientacijo kamere, parametre projekcijske matrike in transformacijo oglišč (npr. premik oglišč po 3D sceni). Matrika PVM je zmnožek med matrikami:

- P – določa projekcijo (preslikavo iz 3D prostora v 2D prostor), kot je vzporedna projekcija ali perspektivna projekcija,
- V – določa parametre gledišča (položaj in smer gledanja), kar dobimo z matričnim množenjem translacijske in rotacijske matrike,
- M – določa transformacijo geometrijskega objekta, če ga želimo npr. premikati po prostoru, in je praviloma izračunana z matričnim množenjem translacijske, rotacijske in skalirne matrike.

Praviloma imajo vsi geometrijski objekti enako matriko P in V, pri čemer je matrika M za vsak objekt drugačna. To pomeni, da je potrebno za vsak objekt pripraviti novo matriko PVM.

Izris več enakih ali podobnih objektov znotraj grafičnega cevovoda poteka tako, da se geometrijski podatki zgolj enega objekta zapišejo v VBO, potem pa se geometrijski objekt izriše večkrat. Pri tem se pred klicem glDrawArrays objekt premakne, torej se na novo izračuna matrika M in posledično PVM ter nato se s funkcijo glUniformMatrix matrika PVM pošlje v senčilnik oglišč.

V primeru uporabe knjižnice GLM, se lahko za pripravo matrik uporabljajo sledeče funkcije: glm::perspective, glm::ortho, glm::rotate, glm::scale, glm::translate, ki vsaka vrne novo matriko.

## 1. Zahteve naloge

Implementirajte grafično aplikacijo (spletno ali namizno), ki naj prikaže piramido iz vsaj šestih barvnih kock. Pri tem vsako kocko obvezno narišite z drugo barvo. Izris piramide izvedite tako, da boste za izris vseh kock uporabili geometrijske podatke samo prve kocke. Torej boste uporabili samo en skupen VBO z 12-imi trikotniki in boste isto kocko izrisali večkrat, pri čemer boste pred klicem glDrawArray s klicem glUniformMatrix spremenili matriko PV**M**. Kocke boste na podlagi matrike M premikali znotraj piramide.

V naslednjem koraku za lažjo prostorsko predstavo izrišite tudi vodoraven štirikotnik, ki naj predstavlja tla. Tla lahko ponovno izrišete na podlagi VBO kocke, kjer kocko po vertikalni osi skrčite (za matriko M uporabite matriko za skaliranje).

V naslednjem koraku z meniji ali z bližnjicami na tipkovnici oziroma z miško omogočite geometrijske transformacije piramide:

- Povečevanje/zmanjševanje piramide z operacijo skaliranja (scale), kar zapišete v matriko M.

- Premestitev piramide s translacijo (translate), kar zapišete v matriko M.
- Zasuk (rotate) piramide okoli osi X, Y in Z, kar zapišete v matriko M.

V matriko M zapišite vse tri transformacije hkrati kot produkt med njihovimi matrikami.

Na koncu z meniji ali z bližnjicami na tipkovnici oziroma z miško omogočite spreminjanje parametrov gledišča:

- Premikanje kamere po sceni (po vseh treh dimenzijah): v matriko V zapišite translacijsko matriko (glm::translate). Pazite, da premikanje kamere izvede res zgolj premikanje in ne rotacije.
- Spreminjanje smeri gledanja kamere: pri implementaciji si pomagajte z kombinacijo rotacijskih matrik/glm::rotate in translacijskih matrik/glm::translate), kar zapišete v matriko V. Upoštevajte, da rotacija kamere predstavlja rotacijo objektov okrog kamere, za kar uporabite kombinacijo operacij rotacije in translacije na podlagi zapiskov iz predavanj.
- Preklapljanje med pravokotno (ortogonalno) vzporedno projekcijo in perspektivno projekcijo, kar naredite s spremembo matrike P: pomagate si lahko z zapiski iz predavanj ali uporabite funkciji glm::ortho in glm::perspecitve. Ustvarjena matrika P se naj uporabi pri izrisu vseh objektov.

V matriko V hkrati zapišite premik in rotacijo kamere kot produkt med geometrijskimi transformacijami. Upoštevajte pravilni vrstni red glede na zapiske iz predavanj. Enaka matrika V se naj uporabi pri vseh objektih.

Pri tvorbi matrike PVM pazite na pravilno delovanje kombinacij različnih operacij in tudi, da transformacije piramide ne izvajajo transformacij nad tlemi. Pazite na vrstni red operacij, saj boste v matriko M dodali premik kocke znotraj piramide in tudi transformacijo celotne piramide. Najboljši način je predstaviti matriko M kot zmnožek matrik $M_p$ in $M_k$, pri čemer $M_p$ predstavlja transformacijo piramide, $M_k$ pa predstavlja premik kocke znotraj matrike. Matrika PVM se lahko izračuna torej kot $P\,V\,M_p\,M_k$.

Implementacijo lahko izvedete v programskem jeziku C++, JavaScript, C#, Java, pri čemer lahko uporabite ustrezne knjižnice (npr. Qt, OpenTK, JOGL) za izvedbo preprostega uporabniškega vmesnika in sprejemanje uporabnikovih akcij (navadno okno, zajemanje tipk).

Naloga mora biti izvedena brez funkcij fiksnega cevovoda, ki se je uporabljal nekoč v prejšnjem tisočletju. Ukazi, kot so *glBegin*, *glEnd*, *glLight**, *glRotate*, *glScale, glLoadMatrix* torej **niso** dovoljeni. Transformacije implementirajte z matrikami. Pomagate si lahko s knjižnico glm (http://glm.g-truc.net) ali podobno za vas programski jezik, ki omogoča tvorbo matrik za skaliranje, projekcijo, itd. V pomoč so vam lahko predvsem naslednje funkcije (http://glm.g-truc.net/0.9.7/api/a00174.html): *glm::perspective*, *glm::ortho*, *glm::rotate*, *glm::scale*, *glm::translate*. Premikanje kamere obvezno implementirajte z uporabo geometrijskih transformacij. Uporaba bližnjic, kot je **lookAt** se ne upošteva.

Točkovanje:

- osnoven izris piramide in tal z zgolj enim VBO: 1 točka,
- geometrijske transformacije piramide: 2 točki,

- parametri, rotacija in premikanje kamere: 2 točki.