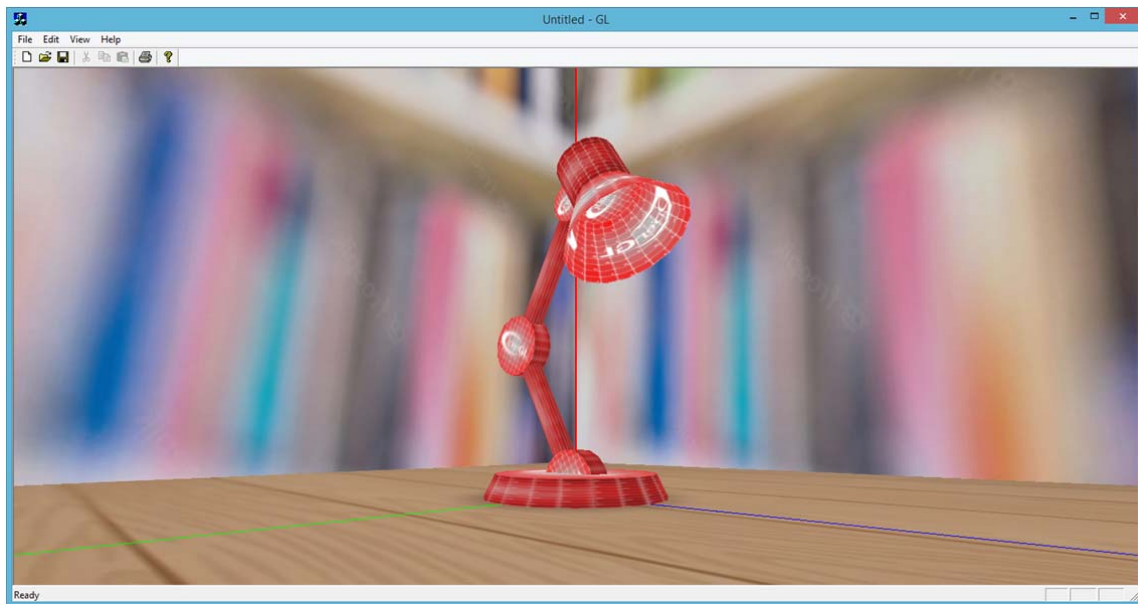


# Računarska grafika

## II kolokvijum

19.01.2023.



Sl.1. Konačni izgled aplikacije

1. Definirati perspektivnu projekciju sa  $FOV = 50^\circ$  i ispuniti funkcije `PrepareScene()`, `DrawScene()` i `Reshape()` odgovarajućim OpenGL funkcijskim pozivima kako bi se omogućilo dalje crtanje. [5 poena]
2. Napisati funkciju `void CGLRenderer::DrawAxes()`, koja crta koordinatne ose dužine 50 jedinica, obojene različitim bojama. Neka je linija duž X-ose plavo, linija duž Y-ose crveno, a duž Z-ose zelena. Preći na sledeću tačku tek kada koordinatne ose budu vidljive. [5 poena]
3. Napisati funkciju `UINT CGLRenderer::LoadTexture(char* fileName)`, koja učitava teksturu sa datim imenom (`fileName`) i vraća ID kreirane teksture. Korišćenjem ove funkcije u okviru `PrepareScene()` učitati teksture: `lamp.jpg`, `side.jpg`, `top.jpg` i `bot.jpg`. Teksture obrisati u `DistoryScene()`. [10 poena]
4. Napisati funkciju `void CGLRenderer::DrawEnvCube(double a)`, koja iscrtava kocku stranice dužine `a`, kojom se uokvirava scena (centrirana na poziciji kamere). Kocka se uvek vidi samo sa unutrašnje strane. Na njenim bočnim stranicama je postavljena tekstura `side.jpg`, a na gornjoj i donjoj `top.jpg` i `bot.jpg`, respektivno. U `DrawScene` funkciji nacrtati ovu kocku, tako da se donja stranica nalazi na XZ ravni. [10 poena]
5. Napisati funkciju `void CGLRenderer::DrawCylinder(double r1, double r2, double h, int nSeg, int texMode, bool bIsOpen)`, koja iscrtava cilindar čiji je poluprečnika jedne baze `r1`, a druge baze `r2`. Visina cilindra je `h`, a broj segmenata kojima se aproksimira omotač `nSeg`. Parametar `texMode` definiše da li se za omotač cilindra koristi gornja (`texMode = 0`) ili donja (`texMode = 1`) polovina teksture `lamp.jpg`, a `isOpen` da li se crtaju baze (ako je `isOpen = false` crtaju se; uvek se koristi gornja polovina teksture). [20 poena]
6. Napisati funkciju `void CGLRenderer::DrawLampBase()`, koja iscrtava osnovu lampe kao cilindar poluprečnika 7 i 8 i visine 2 (vidi sl. 3). [2 poena]
7. Napisati funkciju `void CGLRenderer::DrawLampArm()`, koja iscrtava dva cilindra. Jedan cilindar je poluprečnika 3 (`r1 = r2`) i visine 2, a drugi poluprečnika 1 (`r1 = r2`) i visine 15, uzajamno postavljenih kao na sl. 3. [5 poena]

8. Napisati funkciju `void CGLRenderer::DrawLampHead()`, koja iscrtava glavu lampe sastavljenu od četiri cilindra. Poluprečnici imaju vrednost: 2, 3 ili 6, a visine: 1, 2 ili 5, zavisno od dela (vidi sl. 3). [15 poena]
9. Napisati funkciju `void CGLRenderer::DrawLamp()`, koja iscrtava čitavu lampu koja se sastoji od osnove, dve ruke i glave (vidi sl. 3). Omogućiti rotaciju svakog dela u odnosu na prethodni uvođenjem tri nezavisna ugla rotacije. Uglove definisati kao atribut klase `CGLRenderer`. [12 poena]
10. Iscrtati celu scenu (lampa je na sredini donje stranice kocke okruženja, vidi sl.1) u funkciji `DrawScene` i omogućiti njeno animiranje tako da pritisak na taster:
  - – rotira posmatrača oko Y-ose udesno oko centra scene,
  - ← – rotira posmatrača oko Y-ose ulevo oko centra scene,
  - ↑ – rotira posmatrača naviše u koracima od 0.1 rad do maksimalno 1.5 rad,
  - ↓ – rotira posmatrača naniže u koracima od 0.1 rad do minimalno 0.1 rad,
  - 1, 2, 3, 4, 5 i 6 – rotiraju odgovarajuće delove lampe (1 i 2 prvu ruku, 3 i 4 drugu ruku, a 5 i 6 glavu lampe)
  - + – približava posmatrača centru scene,
  - – udaljava posmatrača od centra scene [16 poena]

Vreme dozvoljeno za završetak kolokvijuma je **120 minuta**. Projekti koji se ne kompajliraju ili ne prikazuju ništa u okviru prozora neće biti ocenjivani. Po završetku, čitav projekat zapakovati u arhivu sa nazivom koji sadrži broj indeksa, ime i prezime (npr. **12345\_Pera\_Peric.zip**), i snimiti na fleš memoriju dežurnog asistenta. Pre kreiranja arhive, iz projekta obrisati **sdf** datoteku, kao i **Debug** i **ipch** direktorijume.



Sl.2. Teksture koje se koriste u sceni (lamp.jpg, side.jpg, top.jpg i bot.jpg)



Sl.3. Rezultati crtanja pojedinačnih funkcija