

# 3D model kuće

---

Ilijević, Dominik

Undergraduate thesis / Završni rad

2019

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://um.nsk.hr/um:nbn:hr:200:667742>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2025-04-02**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

**Stručni studij**

**3D Model Kuće**

**Završni rad**

**Dominik Ilijević**

**Osijek, 2019.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju**

Osijek, 12.07.2019.

**Odboru za završne i diplomske ispite****Imenovanje Povjerenstva za obranu završnog rada  
na preddiplomskom stručnom studiju**

<b>Ime i prezime studenta:</b>	Dominik Ilijević
<b>Studij, smjer:</b>	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
<b>Mat. br. studenta, godina upisa:</b>	AI4472, 26.09.2018.
<b>OIB studenta:</b>	58247491966
<b>Mentor:</b>	Izv. prof. dr. sc. Alfonzo Baumgartner
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Izv.prof.dr.sc. Tomislav Keser
<b>Član Povjerenstva:</b>	Dr. sc. Tomislav Galba
<b>Naslov završnog rada:</b>	3D model kuće
<b>Znanstvena grana rada:</b>	<b>Informacijski sustavi (zn. polje računarstvo)</b>
<b>Zadatak završnog rada</b>	Potrebno je napraviti 3D model vlastite kuće ili neke druge obiteljske kuće u stvarnom mjerilu. Koristiti OpenGL ili neki drugi okvir za rad s 3D grafikom. Omogućiti kretanje po tome 3D modelu koristeći tipke i/ili miš i/ili a, te pomoću odredjenih tipki omogućiti otvaranje/zatvaranje prozora i/ili vrata.
<b>Prijedlog ocjene pismenog dijela ispita (završnog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	12.07.2019.
<i>Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:</i>	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 26.09.2019.

**Ime i prezime studenta:**

Dominik Ilijević

**Studij:**

Preddiplomski stručni studij Elektrotehnika, smjer Informatika

**Mat. br. studenta, godina upisa:**

AI4472, 26.09.2018.

**Ephorus podudaranje [%]:**

0

Ovom izjavom izjavljujem da je rad pod nazivom: **3D model kuće**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Alfonzo Baumgartner

i sumentora

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 .....	1
1.1. Zadatak završnog rada .....	1
2.KORIŠTENE TEHNOLOGIJE.....	2
2.1. C++ programski jezik .....	2
2.2. OpenGL i GLUT .....	2
2.3. Visual studio programsko okruženje .....	3
3. FUNKCIJE .....	5
3.1. Glavni dio programa.....	5
3.2. Dio programa zadužen za iscrtavanje .....	7
3.3. Korisnički dio programa.....	10
4. IZVEDBA PROGRAMA.....	14
5. ZAKLJUČAK .....	17
LITERATURA.....	18
SAŽETAK.....	19
ABSTRACT .....	19
PRILOZI.....	21

# 1.UVOD

U radu je izveden prikaz 3D modela obiteljske kuće koristeći se OpenGL API-jem (engl. *Application Programming Interface*). Prije same izrade bilo je potrebno izmjeriti sve dijelove kuće, čije vrijednosti u stvarnom mjerilu trebaju odgovarati vrijednostima u programu. Program omogućuje kretanje u prostoru te otvaranje i zatvaranje vrata koristeći se tipkama na tipkovnici. Tehnologije korištene pri izradi su OpenGL (engl. *Open Graphics Library*) i GLUT (engl. *OpenGL Utility Toolkit*), a korišteni programski jezik je C++. U nadolazećim poglavljima bit će opisane korištene tehnologije pri izradi, glavne funkcije programa, te način na koji program funkcionira.

## 1.1. Zadatak završnog rada

Zadatak završnog rada je napraviti 3D model obiteljske kuće u stvarnom mjerilu koristeći se OpenGL-om ili nekim drugim okvirom za rad s 3D grafikom. Treba omogućiti kretanje oko tog modela i gledanje iz različitih kutova, te omogućiti otvaranje i zatvaranje vrata koristeći se tipkama na tipkovnici.

## 2.KORIŠTENE TEHNOLOGIJE

### 2.1. C++ programski jezik

C++ programski jezik je nastao kao proširenje C programskog jezika sa dodanom notacijom za objektno orijentirano programiranje. C++ nasljeđuje većinu sintakse C-a. Ovaj programski jezik pripada skupini programskih jezika više razine zajedno sa Python-om, Ruby-em, Javom i sl. Najveća mana ovog programskog jezika je to što radi samo na sustavu za koji se kompajlira, iako za svaki sustav postoji veliki broj kompajlera. Kako bi program radio na drugim sustavima (npr. MacOS, Linux), potrebno ga je kompajlirati za svaki sustav posebno, za razliku od Jave, čiji se kôd može izvršavati na svim sustavima ukoliko imaju JVM (engl. *Java Virtual Machine*). Neki od tipova podataka koji se koriste su *int*, *float*, *char*, *bool*, kao i korisnički tipovi podataka – objekti. Kao objektno orijentirani programski jezik glavne značajke su mu polimorfizam, apstrakcija, enkapsulacija i nasljeđivanje.

```
void changeSize(int w, int h)
{
    if (h == 0)
        h = 1;

    float ratio = w * 1.0 / h;

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glViewport(0, 0, w, h);
    gluPerspective(45.0f, ratio, 0.1f, 100.0f);
    glMatrixMode(GL_MODELVIEW);
}
```

Programski kôd 2.1. Primjer C++ kôda

### 2.2. OpenGL i GLUT

OpenGL je okruženje za razvoj interaktivnih i prijenosnih 2D te 3D grafičkih aplikacija. Sadrži velik broj funkcija kako bi se što vjernije prikazao prostor na računalu. Definirane su funkcije za mapiranje tekstura, dodavanje specijalnih efekata, rotiranje i translaciju predmeta, kao

i mnoge druge snažne funkcije za vizualizaciju. Od prve pojave 1992., OpenGL je postao najrašireniji i najviše korišteni 2D i 3D API u industriji. OpenGL radi na svim većim operacijskim sustavima kao što su Mac OS, UNIX, Windows, Linux i BeOS. OpenGL je moguće pozvati sa C, C++, Fortan, Python, Perl i Java programskim jezicima.

GLUT je biblioteka za OpenGL u kojoj su definirane mnoge funkcije koje olakšavaju neke osnovne radnje koje nisu definirane u samom OpenGL-u. Neke od glavnih funkcija su definiranje prozora, kontrola prozora te praćenje unosa tipkovnice i miša. Glavni cilj GLUT-a je pisanje portabilnog kôda između različitih operacijskih sustava a i mogućnost lakšeg učenja samog rada OpenGL-a. Originalna izvedba GLUT-a nije podržana već 20 godina te je zbog tog razloga u ovom radu korištena besplatna alternativa freeGLUT. Osim freeGLUT-a postoje još i druge alternative kao što su GLFW, FLTK, CPW, GLUT i SDL. Pojedine funkcije OpenGL-a kao i GLUT-a bit će objašnjene kroz ovaj rad (Programski kôd 2.2.).

```
glutInit(&argc, argv);

glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);

glutInitWindowSize(1280, 720);

glutCreateWindow("3D Model Kuce");

glEnable(GL_DEPTH_TEST);

glutReshapeFunc(changeSize);

glutDisplayFunc(render);

glutKeyboardFunc(processNormalKeys);

glutSpecialFunc(processSpecialKeys);

glutMainLoop();
```

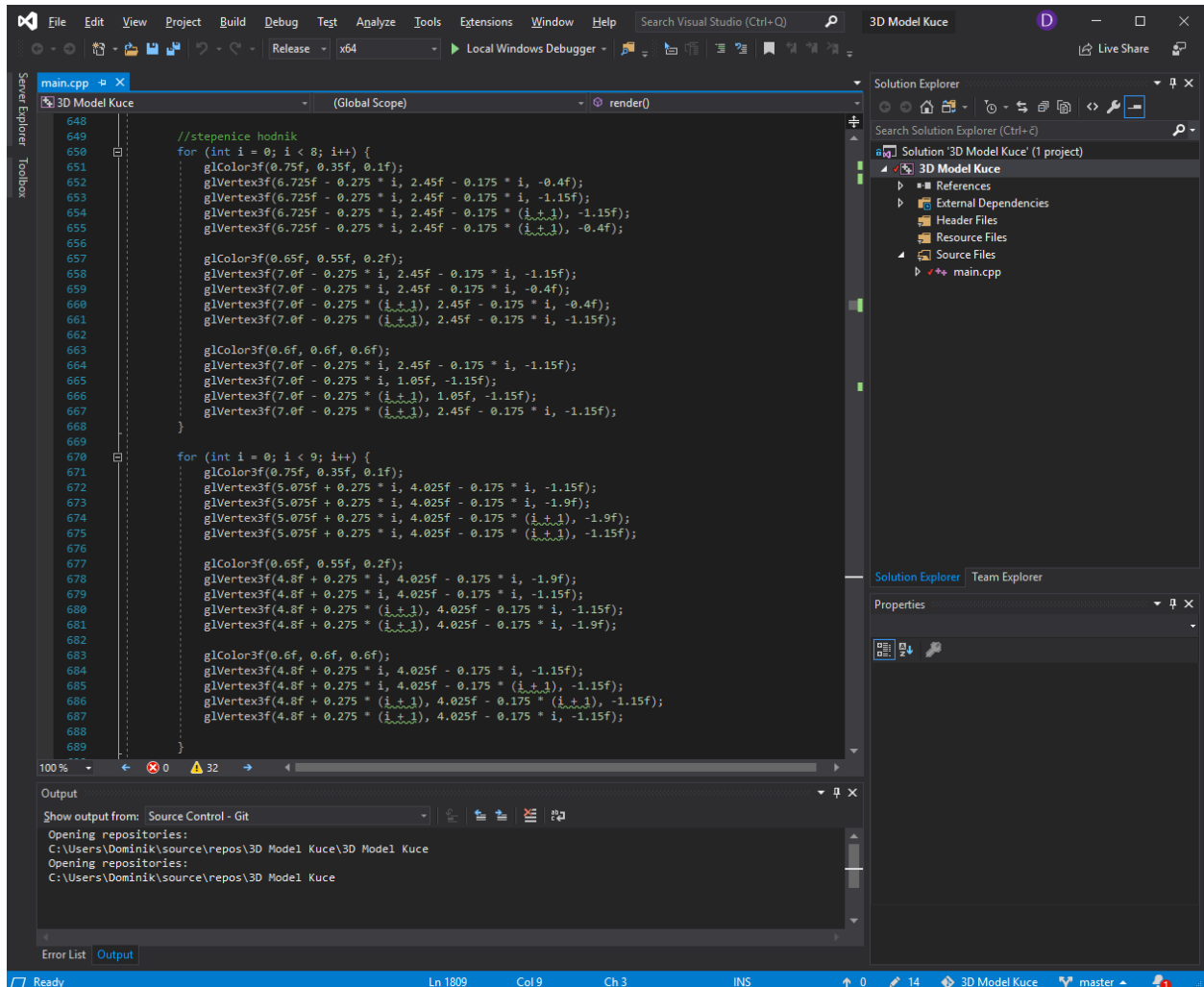
Programski kôd 2.2. Primjer OpenGL i GLUT funkcija

### 2.3. Visual studio programsko okruženje

Visual Studio je IDE (engl. *Integrated Development Environment*) razvijen od Microsofta, te je u potpunosti besplatan. Pogodan je za razvoj programa, web-stranica, web aplikacija, web



usluga i mobilnih aplikacija. Podržava 36 različitih programskih jezika, od kojih su C, C++, Visual Basic .NET, C#, F# ugrađeni jezici. Također su podržani JavaScript, XML, XSLT i CSS. Podrška za ostale jezike moguća je nakon instalacije pojedinih paketa.



Slika 2.1. Izgled Visual Studio korisničkog sučelja

### 3. FUNKCIJE

Kako bi program uspješno radio potrebno je u potpunosti razumjeti OpenGL i GLUT funkcije. OpenGL funkcije su u većini korištene za iscrtavanje, bojanje, translaciju i rotaciju, dok su GLUT funkcije korištene za inicijalizaciju prozora i kontrolu korisničkog unosa.

#### 3.1. Glavni dio programa

Glavni dio programa se odnosi na onaj dio unutar *main* funkcije. *Main* funkcija je ona koja se prva pokreće pri pokretanju programa te ima prednost nad svim drugim funkcijama. Svaki C++ program mora imati *main* funkciju kako bi radio. U nastavku će biti opisane funkcije korištene u ovom dijelu programa (Programski kôd 3.1.).

```
int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
    glutInitWindowSize(1280, 720);
    glutCreateWindow("3D Model Kuce");
    glEnable(GL_DEPTH_TEST);
    texture = LoadTexture("door.bmp");
    glClearDepth(1);
    glutReshapeFunc(changeSize);
    glutDisplayFunc(render);
    glutKeyboardFunc(processNormalKeys);
    glutSpecialFunc(processSpecialKeys);
    glutMainLoop();
    return 0;
}
```

Programski kôd 3.1. Main funkcija

*glutInit(&argc, argv);* - inicijalizira GLUT biblioteku, parametri koje funkcija prima se koriste u slučaju unosa putem komandne linije.

*glutInitDisplayMode*(GLUT\_DOUBLE | GLUT\_RGBA | GLUT\_DEPTH); - postavlja inicijalni način prikaza. Postoji niz parametara koje funkcija može primiti. GLUT\_RGBA prikazuje boju (crvena, zelena, plava) i alfa kanal, GLUT\_DEPTH omogućuje *buffer* dubine, GLUT\_DOUBLE omogućuje korištenje dva *buffera* koja se međusobno izmjenjuju sa pozivom *glutSwapBuffers* funkcije.

*glutInitWindowSize*(1280, 720); - definira veličinu prozora, visina će biti 1280 piksela a širina će biti 720 piksela.

*glutCreateWindow*(„3D Model Kuće“); - stvara se prozor sa prethodno definiranim dimenzijama i pod imenom „3D Model Kuće“. Također je i prije korištenja ove funkcije moguće pozvati funkciju *glutInitWindowPosition* koja definira lokaciju gdje će se prozor prikazati.

*glEnable*(GL\_DEPTH\_TEST); - *glEnable* uključuje neki od određenih modova. Također se može i koristiti *glDisable* kojim se može isključiti prethodno uključeni mod. GL\_DEPTH\_TEST parametar omogućuje dubinsko testiranje, tj. koristi se kako se neke površine ne bi vidjele kroz druge površine. Neki od dodatnih modova koji se mogu koristiti su GL\_ALPHA\_TEST, GL\_DITHER, GL\_FOG, GL\_LIGHT, GL\_BLEND i sl.

*glClearDepth* - specificira vrijednost čišćenja dubinskog *buffera*. Vrijednosti specificirane od *glClearDepth* funkcije su u opsegu [0,1]. Također je vrlo važna funkcija koja se koristi kako se površine ne bi vidjele kroz druge površine.

*glutReshapeFunc*(*changeSize*) - funkcija se poziva svaki put kada se prozoru mijenja veličina. Također se poziva i odmah nakon što je prozor inicijaliziran. Parametar *changeSize* je zapravo posebna funkcija koja je definirana za potrebe mijenjanja veličine prozora.

*glutDisplayFunc*(*render*) - postavlja prikaz za trenutni prozor. Poziva se svaki puta kada se prozor mora ponovno iscrtati (kada mu se promjeni veličina, kada mu se pomakne pozicija...). Kao parametar prima *render* što je korisnička funkcija koja je zadužena za iscrtavanje svih površina.

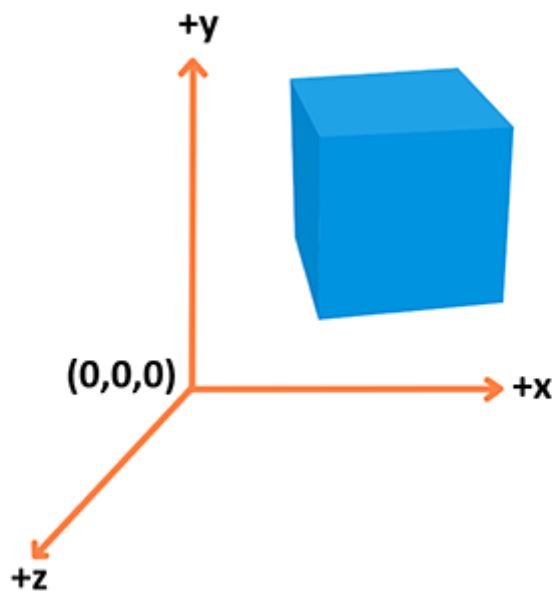
*glutKeyboardFunc*(*processNormalKeys*) - svaki put kada korisnik pritisne tipku na tipkovnici, tipka generira ASCII znak na koji se funkcija može pozvati. Koristi se za dodavanje funkcionalnosti određenim tipkama na tipkovnici. Parametar *processNormalKeys* je funkcija koja upravlja unosom tipka koji služe za izlaz iz programa te otvaranje i zatvaranje vrata.

*glutSpecialFunc(processSpecialKeys);* - ova funkcija je gotovo pa identična sa *glutKeyboardFunc*, samo što se ovdje koriste isključivo posebni znakovi. Ti posebni znakovi su F1-F12, *Page Up*, *Page Down*, *Home*, *End*, *Insert* i tipke strelica.

*glutMainLoop* - postavlja glavnu petlju kako program ne bi završio nakon prvog iscrtavanja.

### 3.2. Dio programa zadužen za iscrtavanje

U ovome dijelu programa se radi iscrtavanje pojedinih površina smještenih u 3D prostoru. 3D prostor u programu je definiran sa 3 varijable: x, y i z, od kojih svaka varijabla predstavlja jednu os (Slika 3.1.).



Slika 3.1. 3D koordinatni sustav[3]

*glLoadIdentity* - zamjenjuje trenutnu matricu sa jediničnom matricom, tj. omogućuje korištenje funkcija translacije i rotacije.

*gluLookAt* - funkcija koja omogućuje mijenjanje kuta gledanja. Prima 9 parametara od kojih su prva 3 x, y i z koordinate pozicije sa koje se gleda, druga 3 parametara su x, y i z koordinate točke u koju se gleda, a zadnja 3 parametara definiraju smjer vektora prema gore (određuje koja se varijabla postavlja za os prema gore).

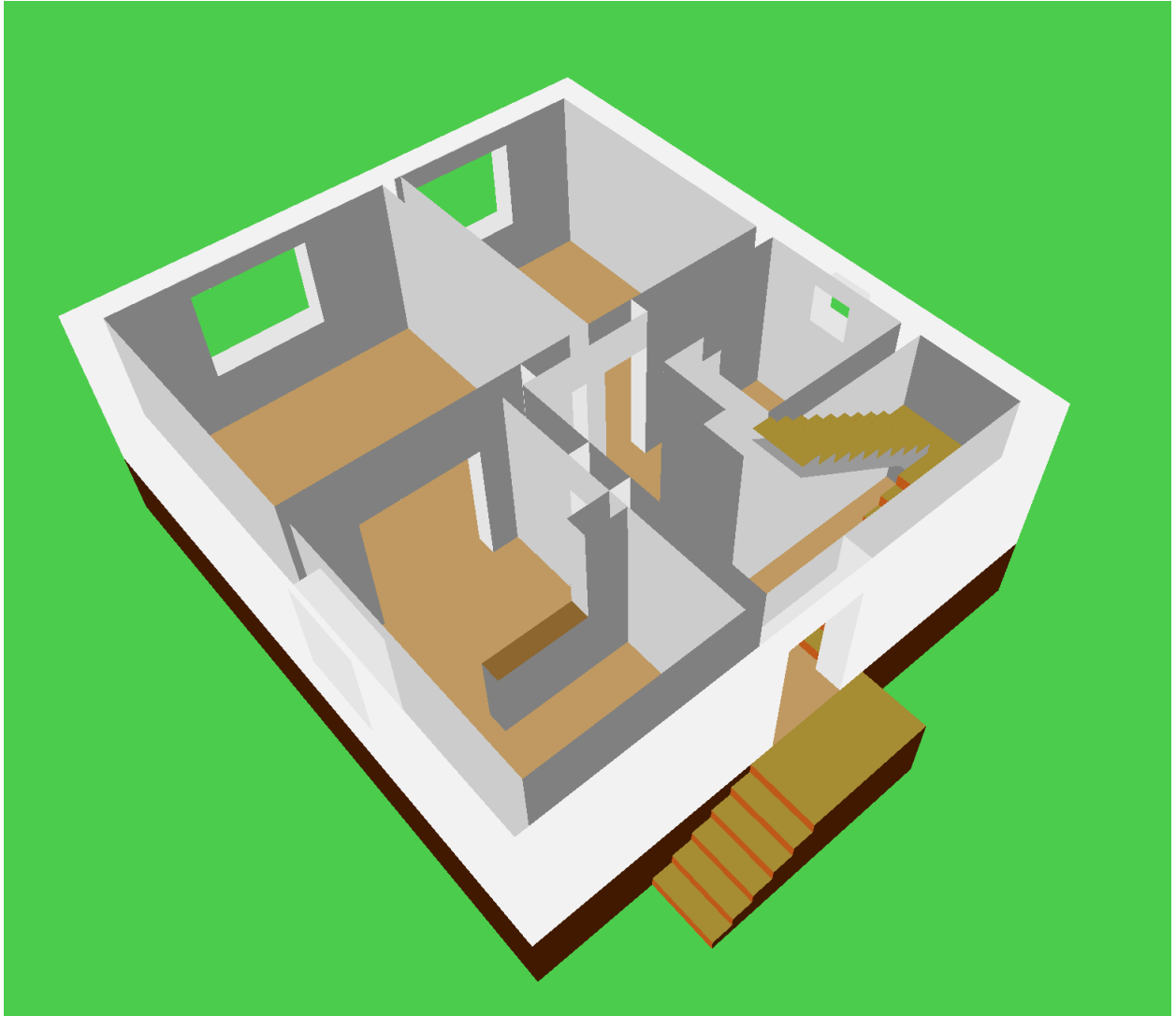
*glBegin* - funkcija unutar koje se definiraju vrhovi između kojih će biti iscrtana površina. Nakon što su svi vrhovi definirani, potrebno je naznačiti kraj sa pozivom funkcije *glEnd*. Svaki *glBegin* mora imati svoj *glEnd*. Funkcija prima jedan parametar koji označava između koliko vrhova će se površina iscrtavati. Glavni parametri su `GL_TRIANGLES`, `GL_QUADS` i `GL_POLYGON`.

*glColor3f* - funkcija koja postavlja trenutnu boju, tj. sve površine koje budu iscrtane nakon poziva ove funkcije će imati ovu boju. Prima 3 parametra koja označavaju količinu crvene, zelene i plave boje (RGB) kao *float* vrijednosti. Također se može koristiti i funkcija *glColor4f*, koja osim RGB vrijednosti prima i vrijednost alfa kanala. Znak „f“ na kraju označava da su parametri *float* tipa.

*glVertex3f* - funkcija koja definira vrh u 3D prostoru. Kao parametre prima 3 *float* vrijednosti koje odgovaraju x, y i z koordinatama. *Float* vrijednosti su korištene jer potrebno obiteljsku kuću predočiti što vjernije i u stvarnom mjerilu. Ova funkcija je najviše korištena funkcija, sa oko 1300 pozivanja, što znači da je definirano oko 1300 vrhova. U većini slučajeva vrhovi su definirani u parovima po 4 vrha koja predstavljaju jedan zid u kući. Programski kôd 3.2. prikazuje skup funkcija koje iscrtavaju površinu zelene boje.

```
glColor3f(0.1f,0.9f,0.1f);
glBegin(GL_QUADS);
    glVertex3f(-100.0f, 0.0f, -100.0f);
    glVertex3f(-100.0f, 0.0f, 100.0f);
    glVertex3f(100.0f, 0.0f, 100.0f);
    glVertex3f(100.0f, 0.0f, -100.0f);
glEnd();
```

Programski kôd 3.2. Primjer funkcije iscrtavanja



Slika 3.3. Iscrtano prizemlje kuće

*glBindTexture* - funkcija koja veže teksturu te omogućuje njeno korištenje. Kao parametre prima `GL_TEXTURE_2D`, i *GLuint* tip podatka koji zapravo predstavlja učitane teksture. Teksture se učitavaju putem *LoadTexture* funkcije.

*glTexCoord2f* - postavlja koordinate trenutne vezane teksture. Kako bi se tekstura uspješno prikazala, potrebno je koristiti funkciju zajedno sa *glVertex3f* funkcijom (Programski kôd 3.4.).

```
glTexCoord2f(0.0f, 0.0f); glVertex3f(3.15f, 4.025f, -2.0f);  
glTexCoord2f(0.0f, 1.0f); glVertex3f(3.15f, 6.025f, -2.0f);  
glTexCoord2f(1.0f, 1.0f); glVertex3f(3.15f, 6.025f, -2.7f);  
glTexCoord2f(1.0f, 0.0f); glVertex3f(3.15f, 4.025f, -2.7f);
```

Programski kôd 3.4. Primjer povezivanja koordinata teksture sa koordinatama vrhova

*glPushMatrix* - OpenGL čuva stog matrica kako bi se brzo primijenile transformacije. Ova funkcija kopira trenutnu matricu (ona matrica koja sadrži pogled na sve prethodno iscertane objekte) na vrhu i gura je na vrh stoga. Tada se sve sljedeće operacije vrše na toj matrici na vrhu stoga. Koristi se za primjenjivanje transformacija na pojedine objekte kako ne bi utjecali na preostale objekte. Kada se završi sa željenim transformacijama potrebno je pozvati funkciju *glPopMatrix* koja briše matricu sa vrha stoga.

*glTranslatef* - funkcija množi trenutnu matricu sa matricom translacije. Matrica translacije je predstavljena sa parametrima koje funkcija prima. U ovom programu glavna namjena ove funkcije je za potrebe rotacije te se koristi zajedno sa funkcijom *glRotatef*.

*glRotatef* - funkcija množi trenutnu matricu sa matricom rotacije. Prima 4 parametra koja odgovaraju kutu rotacije i smjeru vektora oko kojeg se rotira. Kako bi se objekt pravilno rotirao, potrebno je najprije napraviti translaciju do točke oko koje se rotira, rotirati objekt i zatim ponovno napraviti translaciju kako bi se objekt vratio.

### 3.3. Korisnički dio programa

Ovaj dio programa se koristi se za svrhu korisničkog unosa te omogućuje korisniku kontrolu nad programom. Ovdje su definirane funkcije za mijenjanje pozicije gledanja, kuta gledanja, promjena veličine prozora, kontrola unosa tipkovnice.

*LoadTexture* - ova funkcija služi kao priprema teksture iz *.bmp* slike fiksne rezolucije. Tekstura se sprema kao *GLuint* tip podatka koji se zatim veže *glBindTexture* funkcijom. Postoje razne biblioteke kao što su SOIL i GLAUX koje olakšavaju korištenje tekstura u OpenGL-u. OpenGL funkcije korištene unutar funkcije služe za pravilno postavljanje postavki teksture (Programski kôd 3.5.).

```

GLuint LoadTexture(const char* filename){
    GLuint texture;
    int width, height;
    unsigned char* data;
    FILE* file;
    file = fopen(filename, "rb");
    if (file == NULL) return 0;
    width = 256;
    height = 256;
    data = (unsigned char*)malloc(width * height * 3);
    fread(data, width * height * 3, 1, file);
    fclose(file);
    for (int i = 0; i < width * height; ++i){
        int index = i * 3;
        unsigned char B, R;
        B = data[index];
        R = data[index + 2];
        data[index] = R;
        data[index + 2] = B;
    }
    glGenTextures(1, &texture);
    glBindTexture(GL_TEXTURE_2D, texture);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    gluBuild2DMipmaps(GL_TEXTURE_2D, 3, width, height, GL_RGB,
    GL_UNSIGNED_BYTE, data);
    free(data);
    return texture;
}

```

### Programski kôd 3.5. *LoadTexture* funkcija



*CheckDoor* - funkcija koja vraća cijeli broj koji se koristi za određivanje na kojim koordinatama je postavljena kamera. Provjeravaju se x, y i z koordinate kao i trenutni smjer kamere koji je određen varijablama lx i lz. Ukoliko koordinate i smjer kamere odgovaraju određenim uvjetima, funkcija će vratiti broj koji odgovara određenim vratima u kući (Slika 3.6.).

```
int CheckDoor() {
    if (x<4.7f && x>3.55f && y > 1.05f && y < 3.05f && z<2.0f && z>-2.0f && lx<0.5f && lx>-0.5f) {
        return 1;
    }
    else if (x > 1.25f && x<5.25f && y>1.05f && y < 3.05f && z<-2.5f && z>-3.2f && lz<0.5f && lz>-0.5f) {
        return 2;
    }
    else if (x > 3.75f && x<4.5f && y>1.05f && y<3.05f && z>-6.7f && z < -2.4f && lx<0.5f && lx>-0.5f) {
        return 3;
    }
    else if (x > 4.8f && x<5.5f && y>1.05f && y<3.05f && z>-6.7f && z < -2.4f && lx<0.5f && lx>-0.5f) {
        return 4;
    }
    else if (x > 3.7f && x<7.7f && y>1.05f && y<3.05f && z>-4.4f && z < -3.6f && lz<0.5f && lz>-0.5f) {
        return 5;
    }
    else if (x > 1.15f && x < 5.15f && y>4.025f && y<6.025f && z > -2.7f && z < -2.0f && lz<0.5f && lz>-0.5f) {
        return 6;
    }
    else if (x > 0.65f && x < 3.8f && y>4.025f && y<6.025f && z > -3.8f && z < -3.1f && lz<0.5f && lz>-0.5f) {
        return 7;
    }
    else if (x > 2.95f && x < 3.65f && y>4.025f && y<6.025f && z > -6.85f && z < -2.85f && lx<0.5f && lx>-0.5f) {
        return 8;
    }
    else if (x > 4.05f && x < 4.75f && y>4.025f && y<6.025f && z > -6.85f && z < -2.85f && lx<0.5f && lx>-0.5f) {
        return 9;
    }
    else if (x > 3.8f && x < 7.0f && y>4.025f && y<6.025f && z > -4.3f && z < -3.6f && lz<0.5f && lz>-0.5f) {
        return 10;
    }
    return 0;
}
```

Slika 3.6. *CheckDoor* funkcija

*ProcessSpecialKeys* - funkcija koja definira radnje za koje se vrše pritiskom na pojedinu posebnu tipku na tipkovnici. Zapravo mijenja vrijednosti globalnih varijabla, čijim mijenjanjem se dobivaju određeni rezultati kao što su mijenjanje pozicije ili smjera kamere te otvaranje i zatvaranje vrata. Funkcija se poziva unutar glavnog dijela programa kao parametar funkcije *glutSpecialFunc*.

*ProcessNormalKeys* - za razliku od *ProcessSpecialKeys* funkcije ovdje se mogu koristiti svi znakovi sa tipkovnice. Ovdje su definirane radnje izlaza iz programa pritiskom tipke *ESC* te promjena *bool* stanja varijable određenih vrata pritiskom tipke *ENTER* (Programski kôd 3.7.).

```

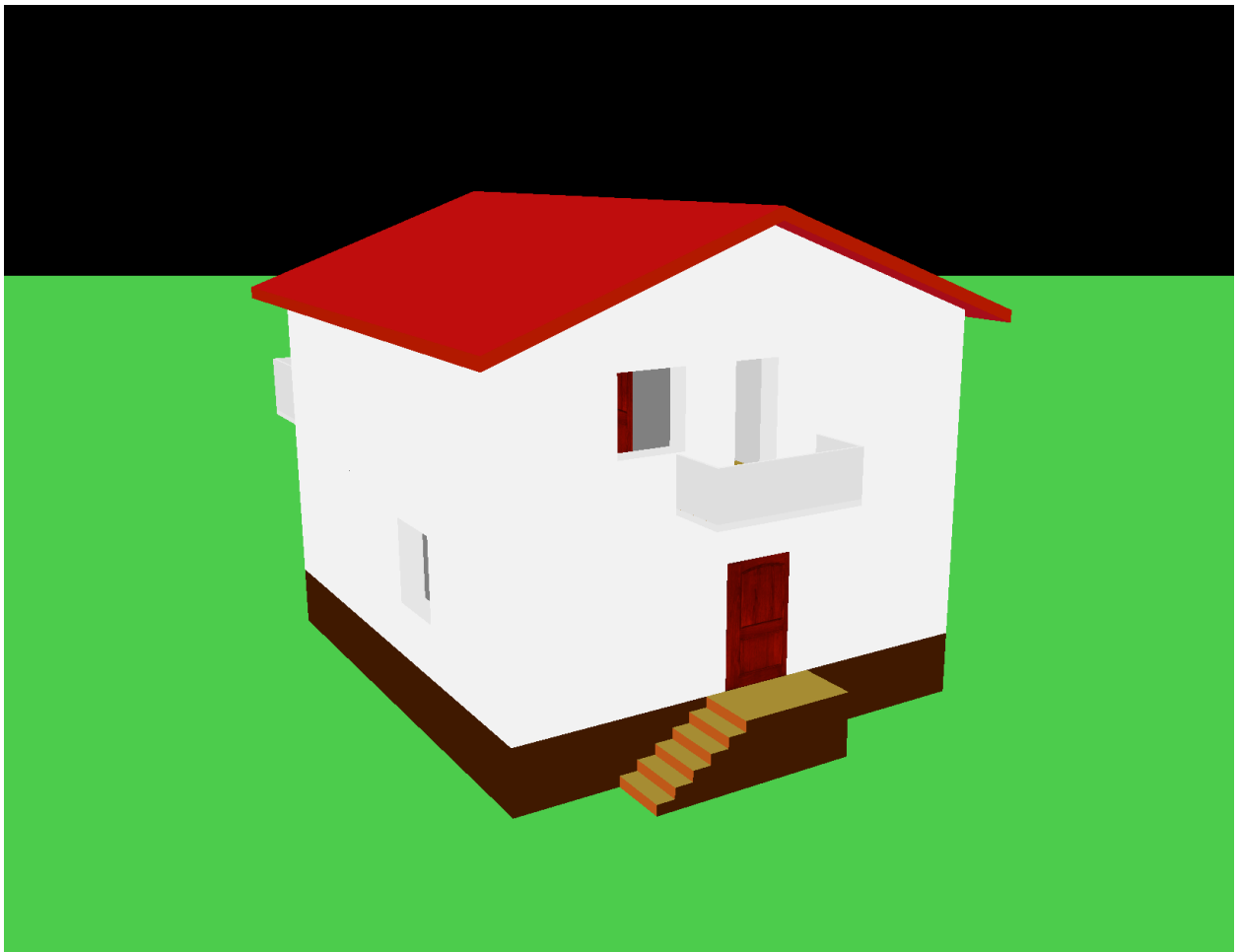
void ProcessNormalKeys(unsigned char key, int x, int y) {
    if (key == 27) {
        exit(0);
    }
    else if (key == 13) {
        doorNumber = CheckDoor();
        switch (doorNumber) {
            case 1: doorUlazna = !doorUlazna;
                break;
            case 2: doorKuhinja = !doorKuhinja;
                break;
            case 3: doorDnevna = !doorDnevna;
                break;
            case 4: doorSobaDolje = !doorSobaDolje;
                break;
            case 5: doorKupatilo = !doorKupatilo;
                break;
            case 6: doorGym = !doorGym;
                break;
            case 7: doorKupatiloGore = !doorKupatiloGore;
                break;
            case 8: doorSoba1 = !doorSoba1;
                break;
            case 9: doorSoba2 = !doorSoba2;
                break;
            case 10: doorRadnaSoba = !doorRadnaSoba;
                break;
        }
    }
    glutPostRedisplay();
}

```

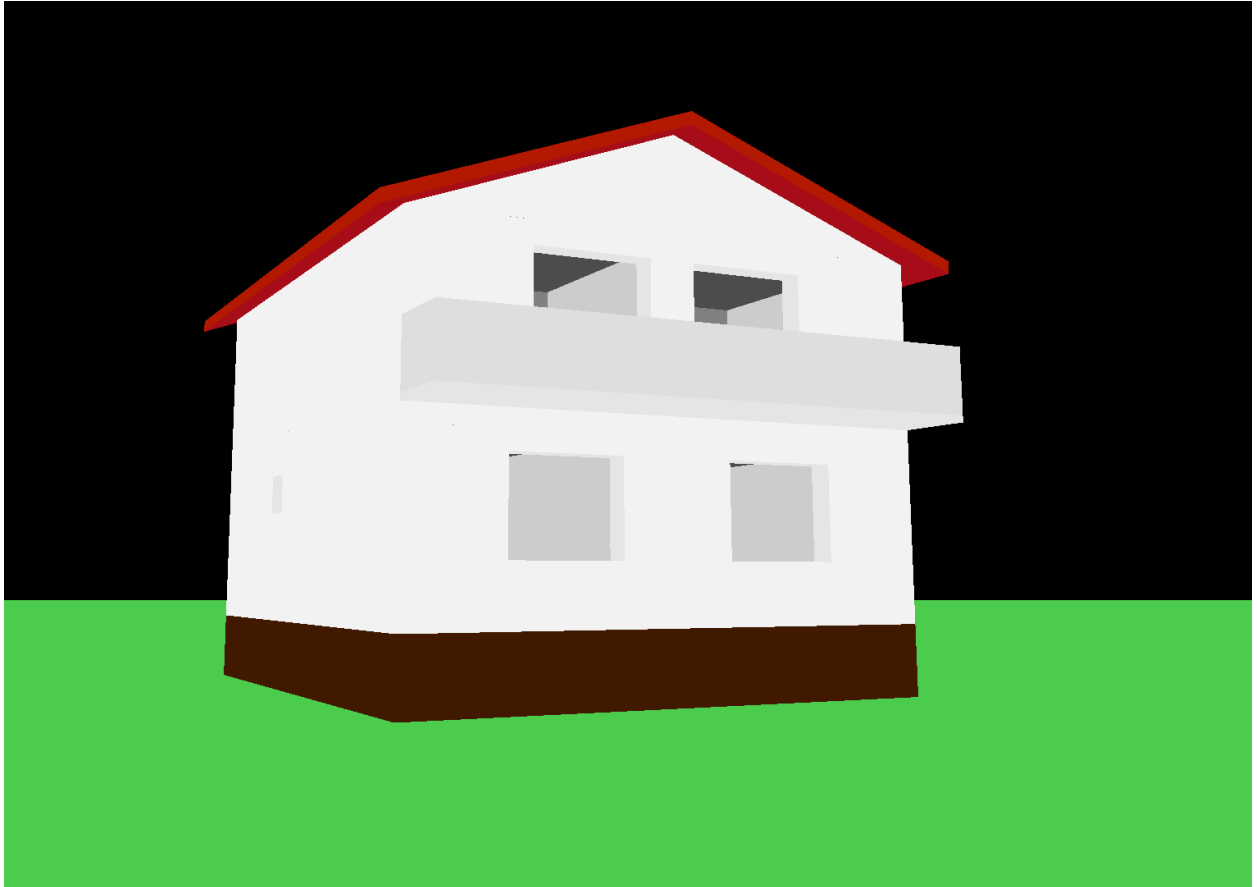
Programski kôd 3.7. *ProcessNormalKeys* funkcija

## 4. IZVEDBA PROGRAMA

U ovome dijelu rada će biti pojašnjeno kako program radi. Pokretanjem programa otvara se novi prozor koji sadrži 3D model kuće (Slika 4.1.). Korisnik se nalazi na početnom položaju koji je određen funkcijom promjene položaja i smjera kamere. Položaj kamere, kao i njen smjer u odnosu na horizont, se mijenja putem unosa tipke strelica, gdje određeni smjer umanjuje ili uvećava vrijednost globalnih varijabli  $x$  i  $z$ . Nakon promjene vrijednosti, smjer i položaj kamere se mijenja te se scena ponovno iscrtava. Kut gledanja prema gore i dolje se mijenja na sličan način putem tipaka *Home* i *End* (Slika 4.2.). Vrijednost varijable  $y$ , koja je zadužena za visinu pozicije kamere, se uvećava tipkom *Page Up*, a umanjuje tipkom *Page Down*.



Slika 4.1. Slika 3D modela kuće

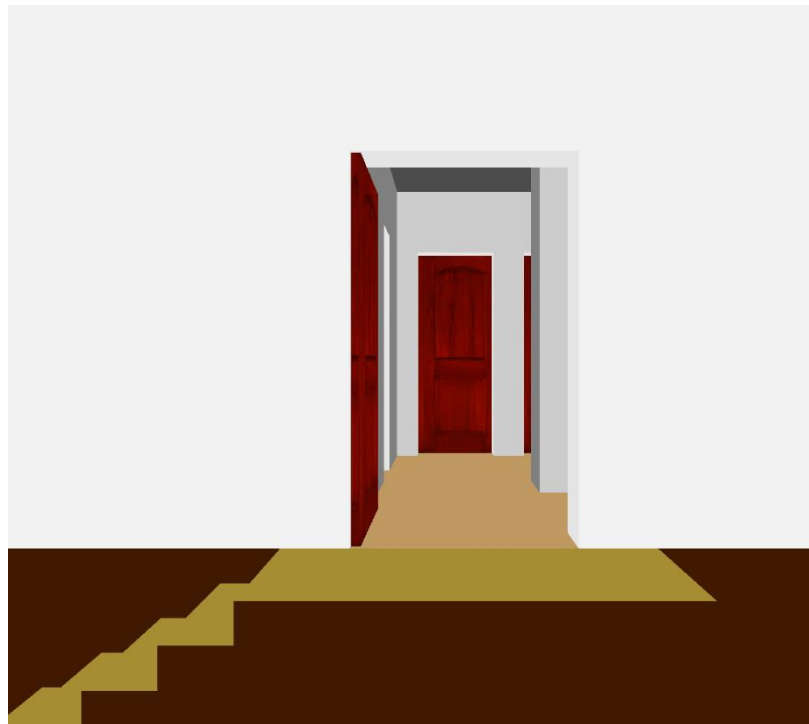


Slika 4.2. Slika 3D modela kuće

Osim mijenjanja smjera i kuta gledanja kamere, korisnik može i otvarati ili zatvarati vrata. Preduvjet za otvaranje određenih vrata je da se kamera nalazi na koordinatama u blizini vrata te da je smjer gledanja kamere okrenut prema vratima. To je napravljeno iz razloga što u kući postoje vrata koja su vrlo blizu te nije dovoljno provjeravati samo koordinate kamere, jer bi se u tom slučaju otvorila oba vrata, pa se zato mora provjeravati i smjer gledanja kamere. Ako su prethodni uvjeti zadovoljeni, pritiskom tipkom *ENTER* mijenja se stanje globalne varijable za ta određena vrata, te se scena ponovno iscrtava. Pošto je stanje varijable promijenjeno, pokreće se petlja zadužena za rotaciju i translaciju tih vrata te su vrata iscrtana kao otvorena (Slika 4.3., Slika 4.4.).



Slika 4.3. Vrata prije otvaranja



Slika 4.4. Vrata nakon otvaranja

## 5. ZAKLJUČAK

U završnom radu rađena je računalni program koji prikazuje trodimenzionalni model obiteljske kuće u stvarnom mjerilu. Omogućeno je kretanje u prostoru oko tog modela te promatranje iz raznih kutova. Za izradu programa korišten je OpenGL koji je zadužen za iscrtavanje, a za implementaciju OpenGL funkcija je korišten programski jezik C++. GLUT je korišten kao alat koji olakšava komunikaciju sa OpenGL-om te omogućuje kontrolu programa unosom putem tipkovnice. Prije same izrade programa svaki dio kuće morao je biti precizno izmjeren i zabilježen. Program je izrađen koristeći se razvojnim okruženjem Visual Studio. Za razvoj samog programa mogle su se koristiti i druge biblioteke za rad sa OpenGL-om, kao što su GLFW i SDL, s kojima bi se dobili slični rezultati.

Program također ima osobnu primjenu te se koristi za vizualizaciju trenutnih i nadolazećih renovacija na obiteljskoj kući.

## LITERATURA

[1] OpenGL, The Khronos Group, dostupno na: <https://www.khronos.org/opengl/wiki/>, lipanj 2018.

[2] GLUT dostupno na: <https://www.opengl.org/resources/libraries/glut/spec3/node1.html>, lipanj 2018.

[3] Developer Mozilla, dostupno na: <https://mdn.mozillademos.org/files/13326/mdn-games-3d-coordinate-system.png>, lipanj 2018.

[4] freeGLUT, dostupno na: <http://freeglut.sourceforge.net/>, lipanj 2018.

## **SAŽETAK**

U završnom radu razvijen je program za iscrtavanje trodimenzionalnog modela obiteljske kuće u stvarnom mjerilu. U svrhu izrade programa korišteni su OpenGL i GLUT knjižnica, koji su realizirani putem C++ programskog jezika. OpenGL služi za definiranje vrhova u prostoru te vrši njihovo iscrtavanje, dok GLUT omogućuje lakše povezivanje sa OpenGL-om i pruža dodatne mogućnosti unosa u program. Korisniku je omogućeno promatranje 3D modela iz različitih položaja i kutova koristeći se tipkama na tipkovnici. Korisnik također može otvoriti ili zatvoriti pojedina vrata ukoliko se nalazi u blizini tih vrata. Programski kôd programa je portabilan te radi na svim operacijskim sustavima.

Ključne riječi: 3D model, kuća, OpenGL, GLUT, C++

## **3D HOUSE MODEL**

### **ABSTRACT**

In this final paper, a program for rendering a three-dimensional model of a family home was developed in real scale. For the purpose of developing the program, OpenGL and GLUT library were used, which were realized through the C++ programming language. OpenGL serves to define the vertices and to enable their rendering, while GLUT make sit easier to connect with OpenGL and provides additional input options to the program. The user is enabled to view the 3D model from different positions and angles by using the keyboard keys. The user can also open or close individual doors if he is located near the individual door. The program code is portable and works on all operating systems.

Key words: 3D model, house, OpenGL, GLUT, C++



## **ŽIVOTOPIS**

Dominik Ilijević rođen je 10.3.1997 u Đakovu. Pohađao je Osnovnu školu „Vladimira Nazora“ u Đakovu. Nakon toga pohađa Srednju strukovnu školu Antuna Horvata u Đakovu smjer Tehničar za mehatroniku. Godine 2015. upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku – stručni studij elektrotehnike smjer Informatika.

## **PRILOZI (CD)**

Prilog 1. Kompletan kod aplikacije