



UNIVERZITET U TUZLI  
FAKULTET ELEKTROTEHNIKE

ak. god. 2017/2018

**ZAVRŠNI PROJEKAT**  
Predmet: *Računarska grafika*

Student:  
Mirza Halilčević (15040)  
Računarstvo i informatika

Profesor:  
dr. sc. Emir Skejić

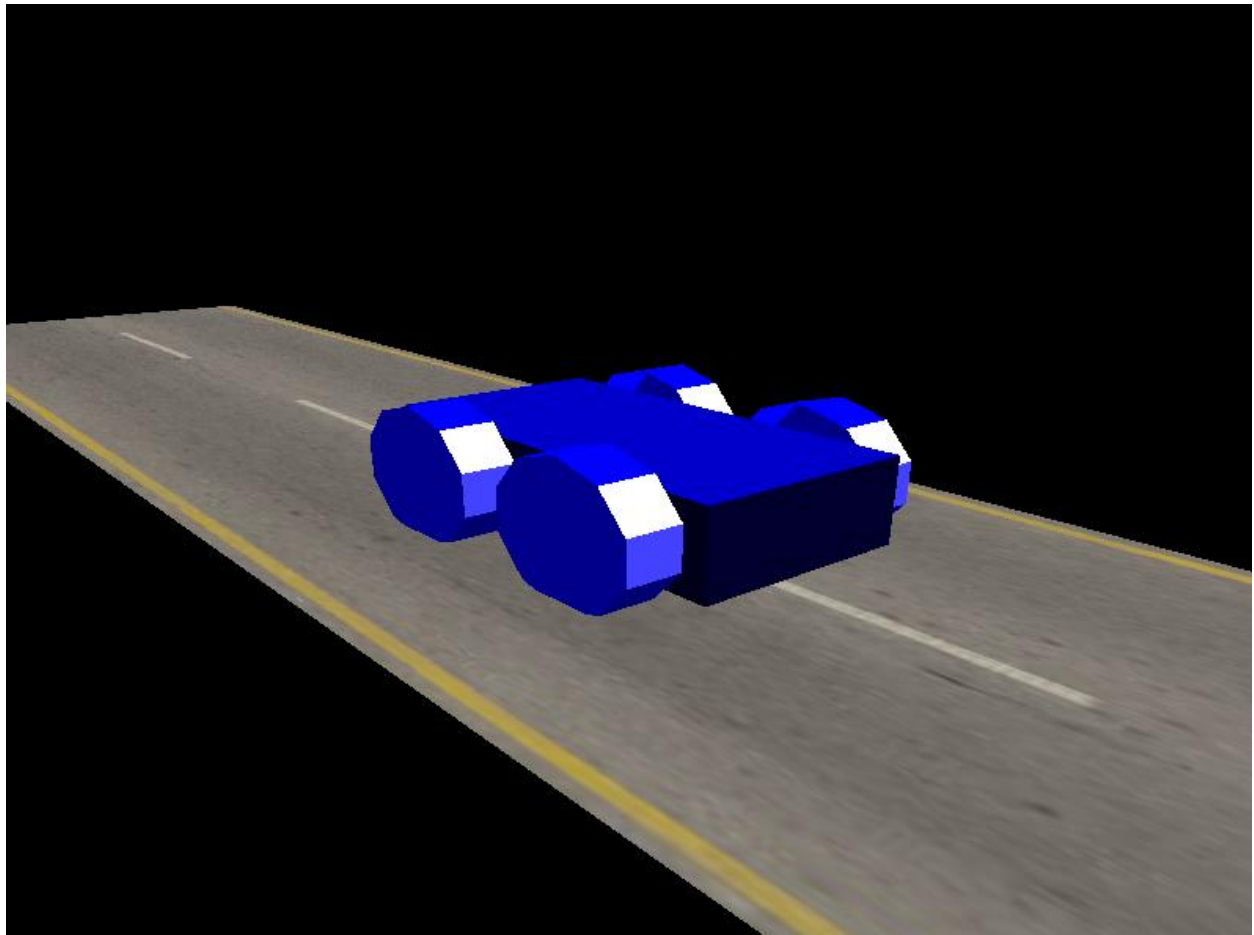
Tuzla, Juni 2018.

## Sadržaj

1 Uvod.....	3
2 3D model.....	4
3 Kamera .....	7
4 Tekstura.....	9
5 Animacije .....	11
5.1 Rotacija.....	12
5.2 Translacija .....	13
5.3 Skaliranje .....	14
5.4 Upravljanje mišem.....	15
6 Osvjetljenje .....	18
7 Materijali .....	21

# 1 Uvod

Tema ovog projekta je 3D model jednostavnog automobila sa animacijama kretanja na cesti. Projekat je rađen u *Code::Blocks* razvojnom okruženju koristeći *OpenGL*, *GLU* i *GLUT* na *Microsoft Windows* operativnom sistemu. Krajnji izgled modela je prikazan na slici 1.



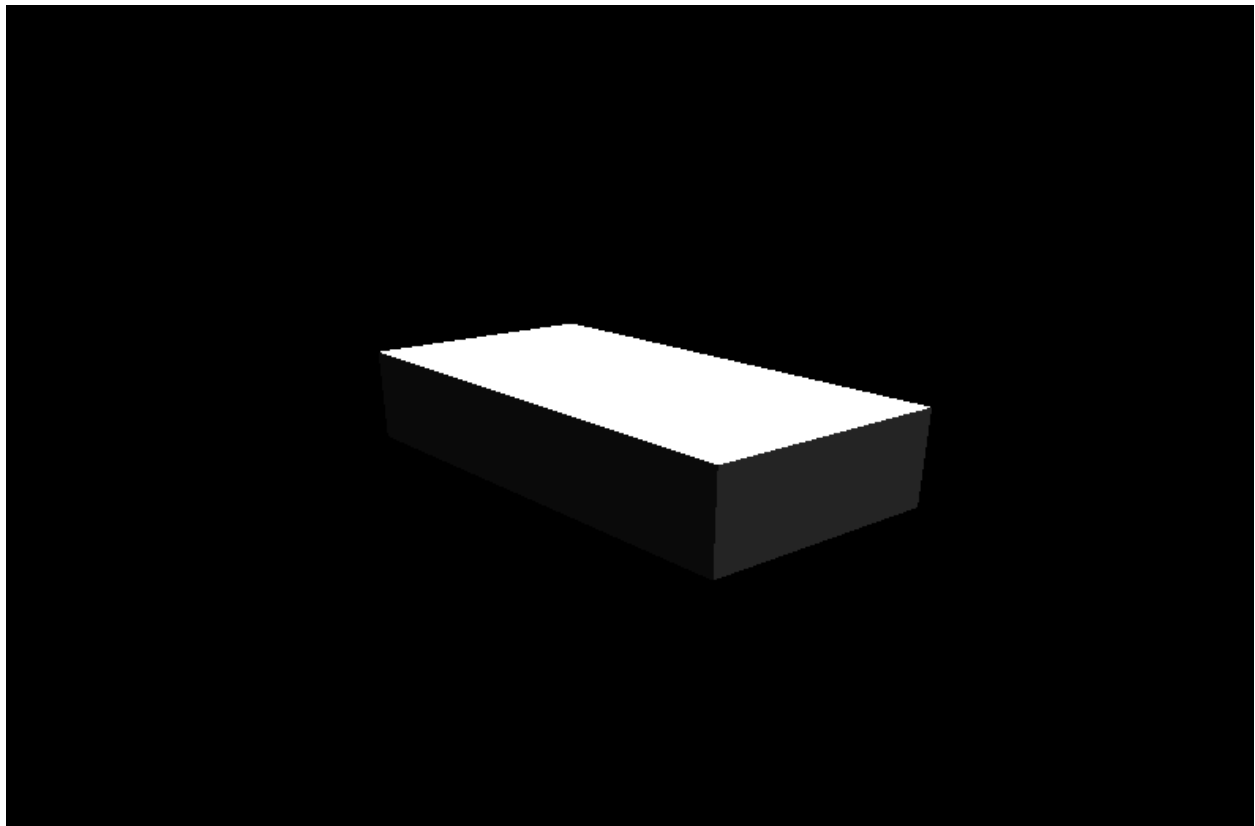
Slika 1. Krajnji izgled scene

## 2 3D model

Tijelo automobila je konstruisano pomoću *glutSolidCube* objekta koji je zatim skaliran kako bi se ostvario izgled kvadra. Programski kod da bi se postiglo navedeno je slijedeći:

```
void display()
{
    ...
    // crta tijelo auta
    glPushMatrix();
    glScalef(2.0f, 1.0f, 4.0f);
    glutSolidCube(1.0);
    glPopMatrix();
    ...
}
```

*Napomena: Tri tačke u kodu (...) označavaju dijelove koji su izostavljeni radi preglednosti.*



Slika 2. Tijelo automobila

Točkovi automobila su sastavljeni od dva objekta, *gluCylinder* kao guma i *gluDisk* kao felga.

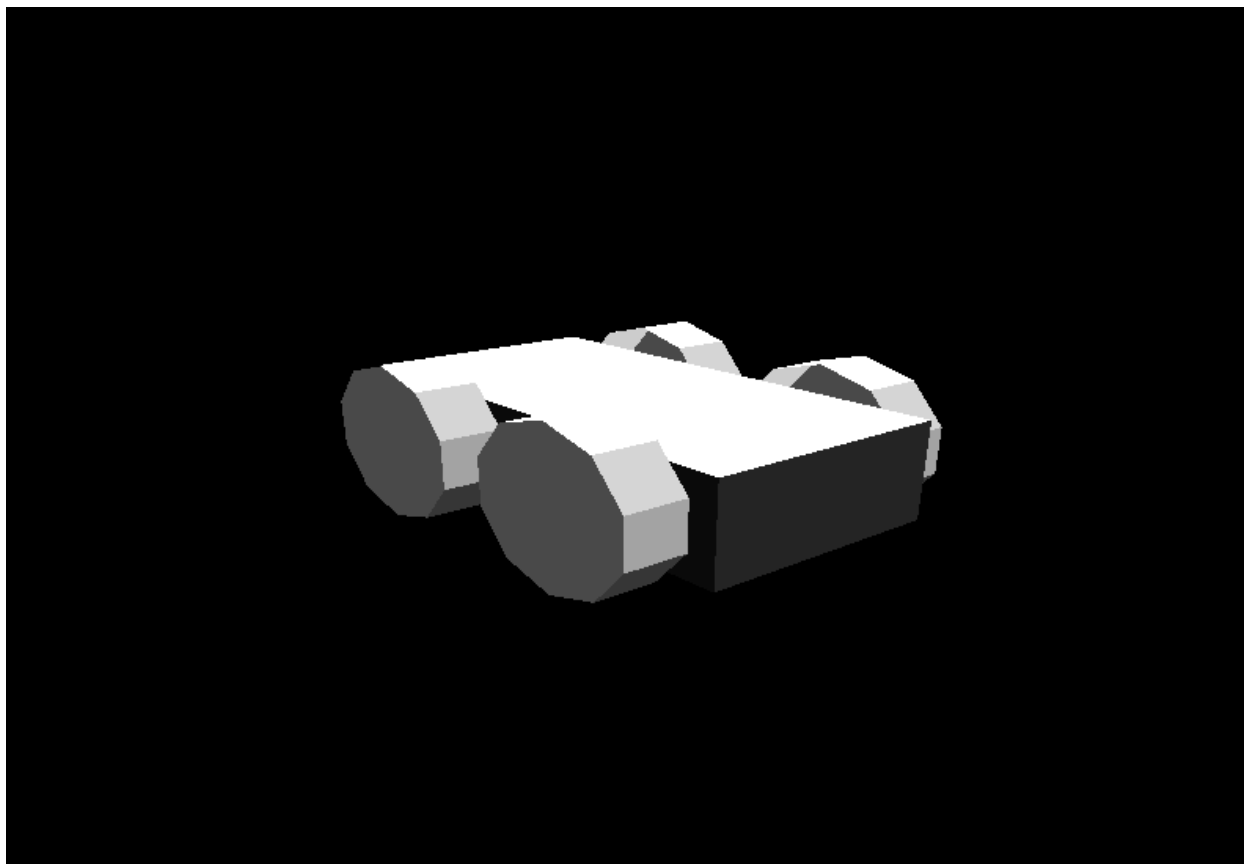
```
// GLU objekti za tockove
GLUquadric* tire_quad = NULL;
GLUquadric* rim_quad = NULL;

void init()
{
    ...
    // kreira potrebne objekte za tockove
    tire_quad = gluNewQuadric();
    rim_quad = gluNewQuadric();
    gluQuadricNormals(tire_quad, GLU_SMOOTH);
    gluQuadricNormals(rim_quad, GLU_SMOOTH);
    ...
}

// crta desni tocak
void draw_right_wheel()
{
    glPushMatrix();
    gluCylinder(tire_quad, 0.8, 0.8, 0.5, 10, 10);
    gluDisk(rim_quad, 0.0, 0.8, 10, 10);
    glPopMatrix();
}

// crta lijevi tocak
void draw_left_wheel()
{
    glPushMatrix();
    gluCylinder(tire_quad, 0.8, 0.8, 0.5, 10, 10);
    glTranslatef(0.0f, 0.0f, 0.5f);
    gluDisk(rim_quad, 0.0, 0.8, 10, 10);
    glPopMatrix();
}

void display()
{
    ...
    // crta tockove
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 1.0f, 0.0f);
    glTranslatef(-1.0f, 0.0f, -1.5f);
    draw_right_wheel();
    glTranslatef(0.0f, 0.0f, 2.5f);
    draw_left_wheel();
    glTranslatef(2.0f, 0.0f, 0.0f);
    draw_left_wheel();
    glTranslatef(0.0f, 0.0f, -2.5f);
    draw_right_wheel();
    glPopMatrix();
    ...
}
```



Slika 3. Tijelo automobila sa točkovima

### 3 Kamera

Na raspolaganju su dvije različite postavke kamere čiji se odabir vrši pritiskom tastera *Space* na tastaturi. Obje varijante su prikazane na slici 4.

```
// odabir kamere
uint8_t camera = 0;

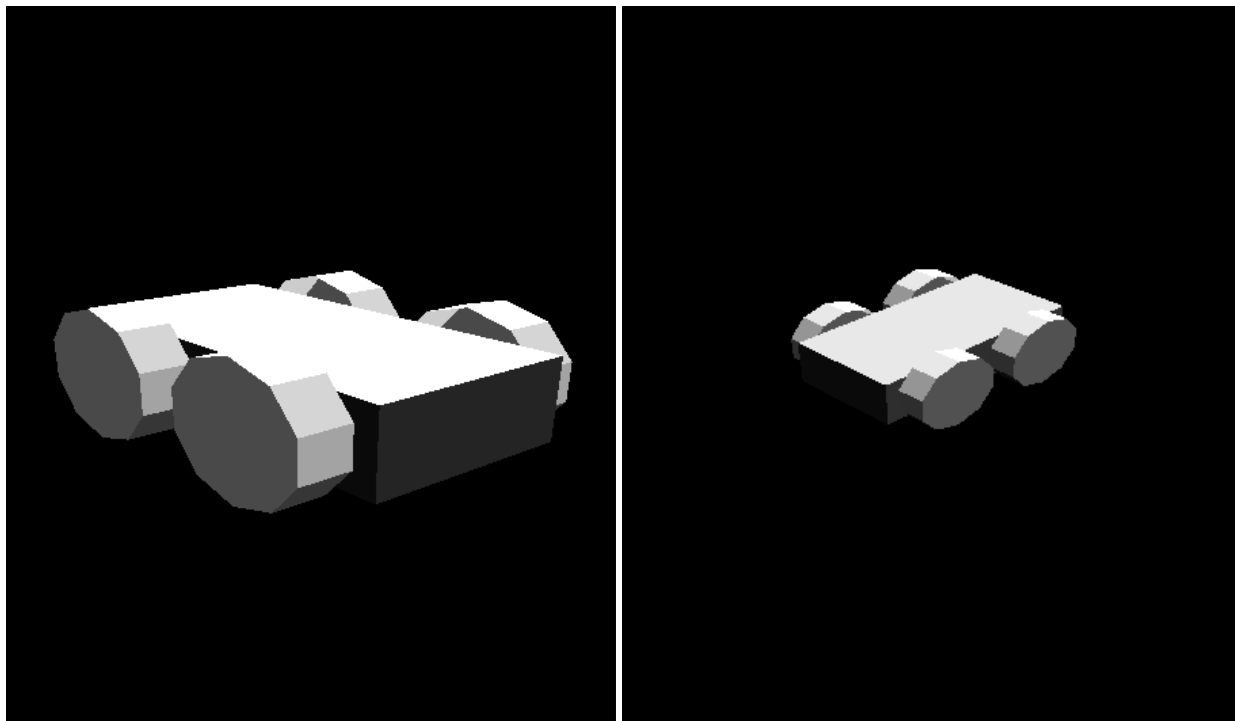
void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case 32:
            // space - mijenja odabir kamere
            camera = (camera + 1) % 2;
            break;
        case 27:
            // esc - program se terminira
            exit(0);
            break;
    }
    glutPostRedisplay();
}

void display()
{
    ...

    // postavlja kameru
    switch (camera) {
        case 0:
            gluLookAt(-5.0, 3.0, 6.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
            break;
        case 1:
            gluLookAt(10.0, 10.0, 10.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
            break;
    }
    ...
}

void init()
{
    ...

    // podesava projekciju
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60, 1.0, 2.5, 30.0);
}
```



Slika 4. Različite postavke kamere



## 4 Tekstura

Da bi predstavili put po kome se automobil kreće korištena je tekstura ceste koja je preslikana na dvodimenzionalan poligon. Tekstura je pomoću programa *GIMP* konvertovana iz *JPEG* formata u C strukturu pod nazivom *gimp\_image* i kao takva korištena sa *GL* funkcijama.

```
// crta put
void draw_road()
{
    // učitava teksturu za put
    static int texture = 0;
    static bool done = false;
    if (!done) {

        glGenTextures(1, (GLuint *)&texture);
        glBindTexture(GL_TEXTURE_2D, texture);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, gimp_image.width,
                    gimp_image.height, 0, GL_RGBA, GL_UNSIGNED_BYTE,
                    gimp_image.pixel_data);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
        done = true;

    } else {

        glBindTexture(GL_TEXTURE_2D, texture);

    }
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
    glEnable(GL_TEXTURE_2D);

    // crta put
    glPushMatrix();
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
    glTranslatef(0.0f, -2.0f, 0.7f);
    glBegin(GL_POLYGON);
        glTexCoord2f(0.0f, 1.0f); glVertex2f(-3.0f, 15.0f);
        glTexCoord2f(0.0f, 0.0f); glVertex2f(3.0f, 15.0f);
        glTexCoord2f(1.0f, 0.0f); glVertex2f(3.0f, -15.0f);
        glTexCoord2f(1.0f, 1.0f); glVertex2f(-3.0f, -15.0f);
    glEnd();
    glPopMatrix();

    glDisable(GL_TEXTURE_2D);
}

void display()
{
    ...
    // crta put
    draw_road();
    ...
}
```



Slika 5. Korištena tekstura ceste



Slika 6. Tekstura ceste preslikana na scenu

## 5 Animacije

Animacije se uključuju/isključuju pomoću tastera miša na slijedeći način:

- lijevi klik za animaciju sa rotacijom
- srednji klik za animaciju sa translacijom
- desni klik za animaciju sa skaliranjem

```
// varijable za praćenje stanja animacije
bool rotate_animation = false;
bool translate_animation = false;
bool scale_animation = false;

void mouse(int button, int state, int x, int y)
{
    if (state == GLUT_UP)
        return;

    switch (button) {
    case GLUT_LEFT_BUTTON:
        // uključuje/isključuje animaciju sa rotacijom
        rotate_animation = !rotate_animation;
        break;
    case GLUT_MIDDLE_BUTTON:
        // uključuje/isključuje animaciju sa translacijom
        translate_animation = !translate_animation;
        break;
    case GLUT_RIGHT_BUTTON:
        // uključuje/isključuje animaciju sa skaliranjem
        scale_animation = !scale_animation;
        break;
    }
}
```

Nakon što je animacija uključena, njeno ponašanje zavisi od položaja miša. U tu svrhu je korišteno nekoliko globalnih varijabli koje predstavljaju parametre za animaciju. Njihova vrijednost će se mijenjati u zavisnosti od položaja miša (poglavlje 5.4).

```
// smjer animacije
uint8_t direction = 1;

// brzina animacije
float speed = 0.0f;

// parametri za animaciju
GLfloat angle = 0.0f;
GLfloat position = 0.0f;
GLfloat scale_factor = 1.0f;
```

## 5.1 Rotacija

Animaciju rotacije obavljaju točkovi automobila tako što se rotiraju oko svojih osa. Smjer i brzina rotacije zavise od varijabli *direction* i *speed* respektivno.

```
void idle()
{
    // ako je ukljucena, odrzava animaciju sa rotacijom
    if (rotate_animation) {
        if ((angle += (0.001f + speed)) >= 360.0f) angle = 0.0f;
        glutPostRedisplay();
    }
}

// crta desni tocak
void draw_right_wheel()
{
    glPushMatrix();
    // animacija sa rotacijom
    glRotatef(direction ? angle : -angle, 0.0f, 0.0f, 1.0f);
    gluCylinder(tire_quad, 0.8, 0.8, 0.5, 10, 10);
    gluDisk(rim_quad, 0.0, 0.8, 10, 10);
    glPopMatrix();
}

// crta lijevi tocak
void draw_left_wheel()
{
    glPushMatrix();
    // animacija sa rotacijom
    glRotatef(direction ? angle : -angle, 0.0f, 0.0f, 1.0f);
    gluCylinder(tire_quad, 0.8, 0.8, 0.5, 10, 10);
    glTranslatef(0.0f, 0.0f, 0.5f);
    gluDisk(rim_quad, 0.0, 0.8, 10, 10);
    glPopMatrix();
}
```

*Napomena: Podebljani dijelovi koda označavaju dodatak na kod koji je već spomenut u prethodnim poglavljima.*

## 5.2 Translacija

Translaciju obavljaju svi dijelovi automobila tako da se automobil kreće duž puta. Kada stigne do kraja puta kreće ponovo ispočetka. Kao i kod rotacije, smjer i brzina zavise od varijabli *direction* i *speed*.

```
void idle()
{
    // ako je ukljucena, odrzava animaciju sa rotacijom
    if (rotate_animation) {
        if ((angle += (0.001f + speed)) >= 360.0f) angle = 0.0f;
        glutPostRedisplay();
    }
    // ako je ukljucena, odrzava animaciju sa translacijom
    if (translate_animation) {
        position += (0.0001f + (direction ? speed : -speed) / 20.0f);
        if (position > 10.0f)
            position = -15.0f;
        else if (position < -15.0f)
            position = 10.0f;
        glutPostRedisplay();
    }
}

void display()
{
    ...

    // crta put
    draw_road();

    // animacija sa translacijom
    glTranslatef(0.0f, 0.0f, position);

    ...
}
```

## 5.3 Skaliranje

Animacija sa skaliranjem se obavlja nad cijelom scenom, uključujući i put. Kada je uključena, veličinu scene je moguće mijenjati pomoću miša.

```
void display()
{
    ...

    // animacija sa skaliranjem
    glScalef(scale_factor, scale_factor, scale_factor);

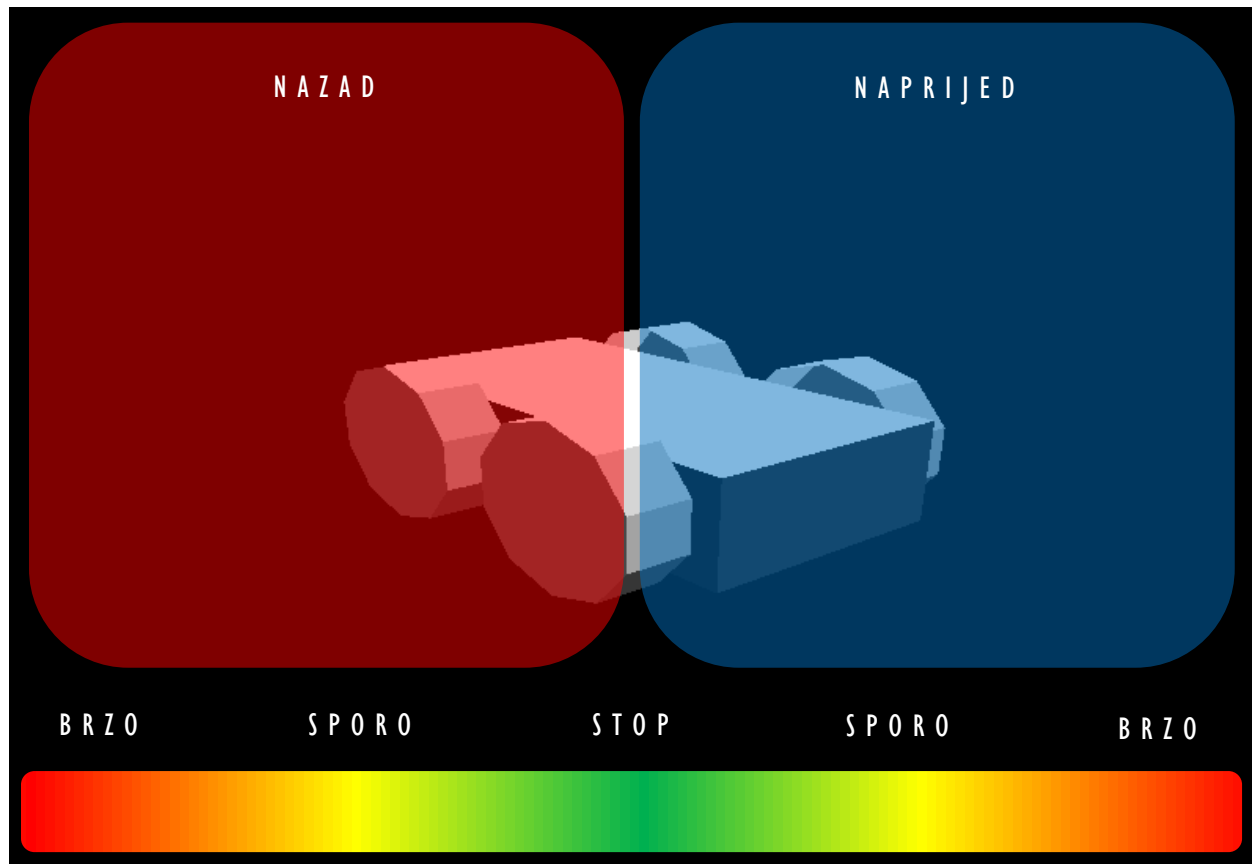
    // crta put
    draw_road();

    // animacija sa translacijom
    glTranslatef(0.0f, 0.0f, position);

    ...
}
```

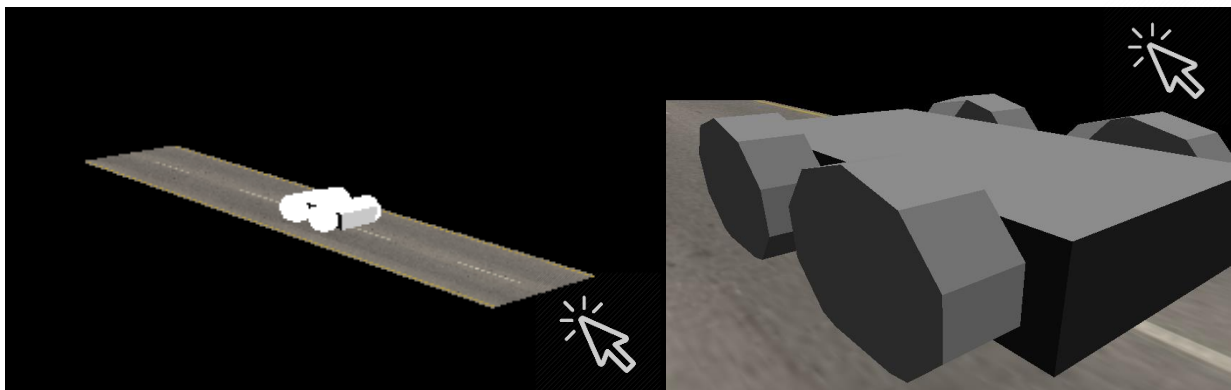
## 5.4 Upravljanje mišem

Pri rotaciji i translaciji, kada je položaj miša na desnoj polovini prozora smjer animacije odgovara kretanju automobila unaprijed. Analogno tome, kada je položaj miša na lijevoj polovini prozora smjer animacije odgovara kretanju automobila unazad. Takođe, brzina animacije je direktno proporcionalna apsolutnom rastojanju pokazivača miša od polovine prozora (slika 7).



Slika 7. Smjer i brzina rotacije i translacije u zavisnosti od položaja miša

Kod animacije sa skaliranjem, veličina scene je proporcionalna položaju miša u vertikalnom smjeru, s time da je faktor skaliranja manji od jedan ako je položaj miša na donjoj polovini prozora, a veći od jedan ako je položaj miša na gornjoj polovini prozora.



Slika 8. Faktor skaliranja u zavisnosti od položaja miša

Programski kod koji ostvaruje navedena ponašanja je slijedeći:

```
// sirina i visina prozora
uint16_t window_width = 800;
uint16_t window_height = 600;

void calculate_animation(int x, int y)
{
    // postavlja smjer animacije
    if (x >= (window_width / 2))
        direction = 1;
    else
        direction = 0;

    // postavlja brzinu animacije
    speed = abs(x - window_width / 2) / 1000.0f;

    // ako je ukljucena, postavlja faktor skaliranja za animaciju sa skaliranjem
    if (scale_animation) {
        scale_factor = abs(y - window_height) / 300.0f;
        glutPostRedisplay();
    }
}

void motion(int x, int y)
{
    calculate_animation(x, y);
}

void passive_motion(int x, int y)
{
    calculate_animation(x, y);
}
```



Da bi se animacije ponašale pravilno pri bilo kojim dimenzijama prozora, potrebno je definisati *reshape callback* funkciju u kojoj se trebaju ažurirati varijable *window\_width* i *window\_height*.

```
void reshape(int width, int height)
{
    glViewport(0, 0, width, height);
    window_width = width;
    window_height = height;
}
```

## 6 Osvjetljenje

Scenu osvjetljavaju dva izvora svjetlosti. Mogu se uključivati/isključivati pomoću tastature na slijedeći način:

- taster *0* uključuje/isključuje prvi izvor svjetlosti
- taster *1* uključuje/isključuje drugi izvor svjetlosti

```
// pomocne varijable za osvjetljenje
uint8_t light0 = 1;
uint8_t light1 = 1;

void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case '0':
            // ukljucuje/iskljucuje prvi izvor svjetlosti
            light0 = (light0 + 1) % 2;
            switch (light0) {
                case 0:
                    glDisable(GL_LIGHT0);
                    break;
                case 1:
                    glEnable(GL_LIGHT0);
                    break;
            }
            break;
        case '1':
            // ukljucuje/iskljucuje drugi izvor svjetlosti
            light1 = (light1 + 1) % 2;
            switch (light1) {
                case 0:
                    glDisable(GL_LIGHT1);
                    break;
                case 1:
                    glEnable(GL_LIGHT1);
                    break;
            }
            break;
        case 32:
            // space - mijenja odabir kamere
            camera = (camera + 1) % 2;
            break;
        case 27:
            // esc - program se terminira
            exit(0);
            break;
    }
    glutPostRedisplay();
}
```

```

void init()
{
    ...

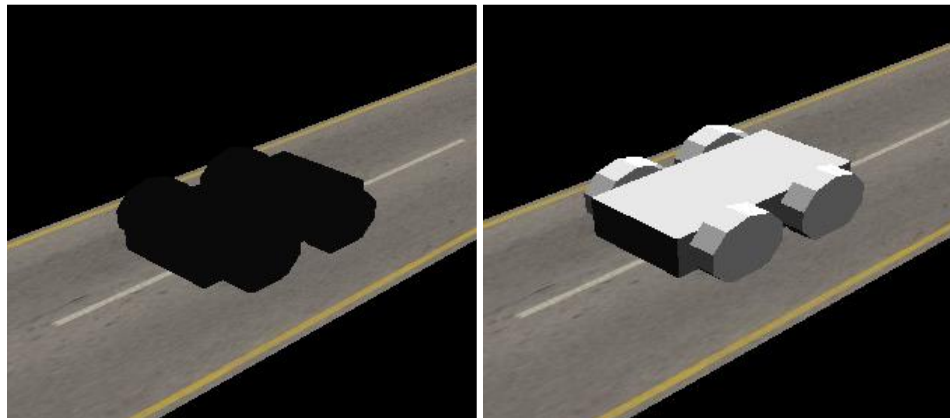
    // prvi izvor svjetlosti
    GLfloat light0_position[] = { 25.0f, 50.0f, 50.0f, 0.0f };
    GLfloat light0_color[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    glLightfv(GL_LIGHT0, GL_POSITION, light0_position);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_color);

    // drugi izvor svjetlosti
    GLfloat light1_position[] = { -25.0f, 50.0f, -50.0f, 0.0f };
    GLfloat light1_color[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    glLightfv(GL_LIGHT1, GL_POSITION, light1_position);
    glLightfv(GL_LIGHT1, GL_DIFFUSE, light1_color);

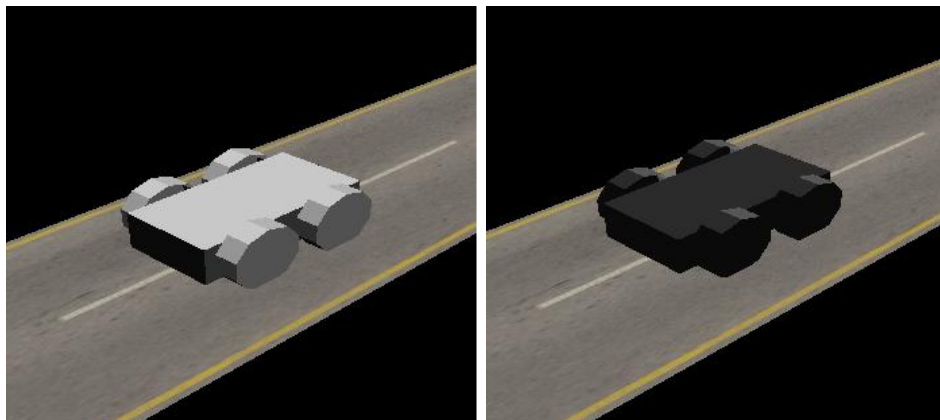
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHT1);

    ...
}

```



Slika 9. Lijevo – oba svjetla isključena, desno – oba svjetla uključena



Slika 10. Lijevo – uključeno samo prvo svjetlo, desno – uključeno samo drugo svjetlo

## 7 Materijali

Na modelu se nalaze dva različita materijala, prvi za tijelo automobila, drugi za točkove. Materijali se uključuju/isključuju na slijedeći način:

- taster 2 uključuje/isključuje prvi materijal
- taster 3 uključuje/isključuje drugi materijal

```
// varijable za ukljucivanje/iskljucivanje materijala
bool mat0_enabled = true;
bool mat1_enabled = true;

void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case '0':
            // ukljucuje/iskljucuje prvi izvor svjetlosti
            ...
            break;
        case '1':
            // ukljucuje/iskljucuje drugi izvor svjetlosti
            ...
            break;
        case '2':
            // ukljucuje/iskljucuje prvi materijal
            mat0_enabled = !mat0_enabled;
            break;
        case '3':
            // ukljucuje/iskljucuje drugi materijal
            mat1_enabled = !mat1_enabled;
            break;
        case 32:
            // space - mijenja odabir kamere
            camera = (camera + 1) % 2;
            break;
        case 27:
            // esc - program se terminira
            exit(0);
            break;
    }
    glutPostRedisplay();
}
```

```

// parametri za materijale
GLfloat mat0[] = {
    0.0, 0.0, 0.5, 1.0, // AMBIENT_AND_DIFFUSE
    0.0, 0.0, 0.0, 0.0, // SPECULAR
    0.0 // SHININESS
};
GLfloat mat1[] = {
    0.0, 0.0, 1.0, 1.0, // AMBIENT_AND_DIFFUSE
    1.0, 1.0, 1.0, 0.0, // SPECULAR
    20.0 // SHININESS
};
GLfloat mat_off[] = {
    0.0, 0.0, 0.0, 0.0, // AMBIENT
    1.0, 1.0, 1.0, 1.0, // DIFFUSE
    0.0, 0.0, 0.0, 0.0, // SPECULAR
    0.0 // SHININESS
};

void set_mat0()
{
    if (mat0_enabled) {
        glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, &mat0[0]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, &mat0[4]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SHININESS, &mat0[8]);
    }
    else {
        glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT, &mat_off[0]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, &mat_off[4]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, &mat_off[8]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SHININESS, &mat_off[12]);
    }
}

void set_mat1()
{
    if (mat1_enabled) {
        glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, &mat1[0]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, &mat1[4]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SHININESS, &mat1[8]);
    }
    else {
        glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT, &mat_off[0]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, &mat_off[4]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, &mat_off[8]);
        glMaterialfv(GL_FRONT_AND_BACK, GL_SHININESS, &mat_off[12]);
    }
}

```

```

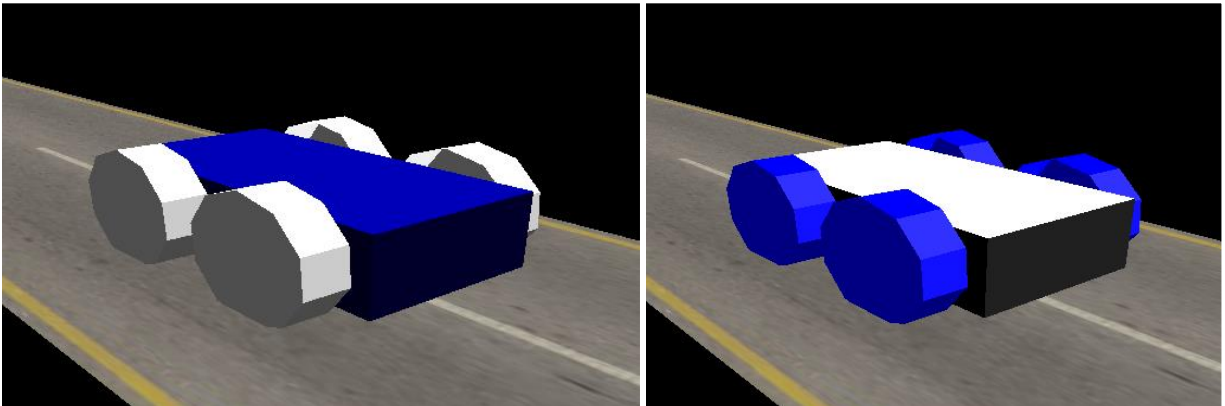
void display()
{
    ...

    // crta tijelo auta
    set_mat0();
    glPushMatrix();
        glScalef(2.0f, 1.0f, 4.0f);
        glutSolidCube(1.0);
    glPopMatrix();

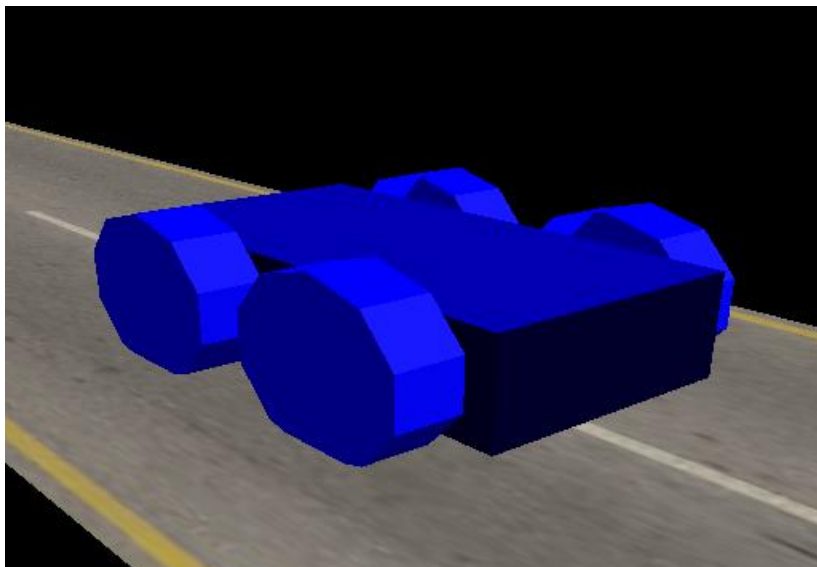
    // crta tockove
    set_mat1();
    glPushMatrix();
        glRotatef(90.0f, 0.0f, 1.0f, 0.0f);
        glTranslatef(-1.0f, 0.0f, -1.5f);
        draw_right_wheel();
        glTranslatef(0.0f, 0.0f, 2.5f);
        draw_left_wheel();
        glTranslatef(2.0f, 0.0f, 0.0f);
        draw_left_wheel();
        glTranslatef(0.0f, 0.0f, -2.5f);
        draw_right_wheel();
    glPopMatrix();

    ...
}

```



Slika 11. Lijevo – uključen samo prvi material, desno – uključen samo drugi materijal



Slika 12. Uključena oba materijala