

3. Sile i oslonci

3.1. Upravljanje silama u konstrukciji (forces.cpp)

3.1.1. Pristup čvorovima konstrukcije

```
struct Node {  
    float x, y;  
};  
  
extern Node nodes[100];  
extern int nodeCount;
```

Sile u konstrukciji moraju biti vezane za postojeće čvorove. Pošto čvorove definiše druga osoba u projektu, ovde se oni ne definišu ponovo, već se koriste pomoću ključne reči extern.

Na ovaj način:

- Svi delovi programa rade nad istim podacima;
- Izbegava se dupliranje struktura;
- Održava se podela rada između članova tima.

3.1.2. Struktura sile

```
struct Force {  
    int nodeId;  
    float magnitude;  
    float angle;  
};
```

Struktura Force opisuje jednu silu u konstrukciji:

- nodeId - indeks čvora na koji sila deluje;
- magnitude - intenzitet sile (fiksna u ovom projektu);
- angle - pravac sile u radianima.

Fiksna jačina sile je korišćena jer projekat nije fokusiran na proračun, već na grafički prikaz i interakciju.

3.1.3. Čuvanje sila

```
static Force forces[100];  
  
static int forceCount = 0;
```

Sve sile se skladište u statičkom nizu maksimalne veličine 100. Promenljiva forceCount prati broj trenutno dodatih sila. Ovakav pristup je jednostavan i dovoljan za zahteve projekta.

3.1.4. Pronalaženje čvora na osnovu klika miša

```
int findNode(float mx, float my)  
{  
    const float eps = 0.05f;  
  
    for (int i = 0; i < nodeCount; i++) {  
        float dx = mx - nodes[i].x;  
        float dy = my - nodes[i].y;  
        if (dx * dx + dy * dy < eps * eps)  
            return i;  
    }  
    return -1;  
}
```

Ova funkcija proverava da li je korisnik kliknuo dovoljno blizu nekom čvoru. Koordinate klika miša (mx, my) se porede sa koordinatama svakog čvora, pri čemu dx i dy predstavljaju razliku (rastojanje klika od čvora) po x i y osi. Umesto računanja stvarnog rastojanja koristi se kvadrat rastojanja radi efikasnosti. Ako je čvor pronađen, vraća se njegov indeks, dok se u suprotnom vraća vrednost -1, koja se koristi kao signal da validan čvor ne postoji.

3.1.5. Ograničenje ugla sile (snap na $\pi/4$)

```
float snapAngle(float angle)  
{  
    float step = M_PI / 4.0f;  
    return round(angle / step) * step;  
}
```

Prema smernicama projekta, dozvoljeni su samo uglovi koji su umnošci od $\pi/4$.

Funkcija:

1. Deli ugao osnovnim korakom;
2. Zaokružuje na najbližu dozvoljenu vrednost;
3. Vraća korigovani ugao.

Na ovaj način se sprečava unos proizvoljnih pravaca.

3.1.6. Dodavanje sile

```
void addForce(float mx, float my, float angle)
{
    int node = findNode(mx, my);

    if (node == -1)
        return;

    forces[forceCount].nodeId = node;
    forces[forceCount].magnitude = 1.0f;
    forces[forceCount].angle = snapAngle(angle);
    forceCount++;
}
```

Sila se može dodati samo ako postoji čvor ispod kursora. Ako findNode vrati -1, funkcija se prekida i sila se ne dodaje. Time se obezbeđuje validnost podataka u sistemu.

3.1.7. Iscrtavanje sila

```
void drawForces()
{
    glColor3f(1.0f, 0.0f, 0.0f);

    for (int i = 0; i < forceCount; i++) {
        Node n = nodes[forces[i].nodeId];

        float dx = cos(forces[i].angle) * 0.15f;
```

```

        float dy = sin(forces[i].angle) * 0.15f;

        glBegin(GL_LINES);

            glVertex2f(n.x, n.y);

            glVertex2f(n.x + dx, n.y + dy);

        glEnd();

    }
}

```

Sile se prikazuju kao linije koje polaze iz čvora u pravcu delovanja sile. Koordinate $n.x$ i $n.y$ predstavljaju položaj čvora, dok se pomoću trigonometrijskih funkcija iz ugla sile računaju pomeraji kraja strelice, dx i dy . Dužina prikazane sile je konstantna i nema fizičko značenje, jer je cilj ovog dela projekta vizuelni prikaz i interakcija, a ne statički proračun.

3.2. Upravljanje osloncima (supports.cpp)

3.2.1. Struktura oslonca

```

struct Support {
    int nodeId;
    float angle;
};

```

Oslonac je vezan za jedan čvor i ima definisan pravac reakcije. Ne poseduje intenzitet jer se ne vrši statički proračun.

3.2.2. Čuvanje oslonaca

```

static Support supports[100];
static int supportCount = 0;

```

Oslonci se čuvaju u statičkom nizu, analogno silama. Promenljiva `supportCount` prati broj dodatih oslonaca.

3.2.3. Dodavanje oslonca

```

void addSupport(float mx, float my, float angle)
{
    int node = -1;
    const float eps = 0.05f;

```

```

for (int i = 0; i < nodeCount; i++) {
    float dx = mx - nodes[i].x;
    float dy = my - nodes[i].y;
    if (dx * dx + dy * dy < eps * eps)
        node = i;
}

if (node == -1)
    return;

supports[supportCount].nodeId = node;
supports[supportCount].angle = snapAngle(angle);
supportCount++;
}

```

Oslonac se može dodati samo ako postoji čvor, pri čemu se pronalaženje čvora vrši direktno u funkciji poređenjem pozicije klika sa koordinatama čvorova. Ugao oslonca se zatim ograničava na dozvoljene vrednosti pomoću funkcije snapAngle.

3.2.4. Iscrtavanje oslonaca

```

void drawSupports()
{
    glColor3f(0.0f, 0.0f, 1.0f);

    for (int i = 0; i < supportCount; i++) {
        Node n = nodes[supports[i].nodeId];
        float s = 0.05f;

        glBegin(GL_TRIANGLES);
            glVertex2f(n.x, n.y);
            glVertex2f(n.x - s, n.y - s);

```

```
        glVertex2f(n.x + s, n.y - s);  
    glEnd();  
}  
}
```

Oslonci su prikazani kao trouglovi radi lakše vizuelne identifikacije. Promenljiva `s` definiše veličinu simbola i nema fizičko značenje.

3.3. Zajedničke deklaracije (utils.h)

```
void addForce(float mx, float my, float angle);
```

```
void drawForces();
```

```
void addSupport(float mx, float my, float angle);
```

```
void drawSupports();
```

U ovom fajlu se nalaze deklaracije funkcija za dodavanje i iscrtavanje sila i oslonaca. Ove funkcije se pozivaju iz glavne petlje programa, zbog čega moraju biti dostupne drugim modulima, dok njihova implementacija ostaje sakrivena u odgovarajućim .cpp fajlovima.