

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 499

## **Postupci simulacije i prikaza tkanine**

Davorin Gordan Keserica

Zagreb, lipanj 2009.

# Tablica sadržaja

<b>1</b>	<b>Uvod .....</b>	<b>4</b>
<b>2</b>	<b>Fizikalni model tkanine.....</b>	<b>5</b>
2.1	Model opruga i čestica .....	6
2.2	Metode integracije .....	9
2.2.1	Verlet metoda .....	9
2.2.2	Eulerova metoda.....	10
2.3	Stabilizacija fizikalnog modela tkanine .....	11
<b>3</b>	<b>Model tkanine .....</b>	<b>12</b>
3.1	Reprezentacija točke.....	12
<b>4</b>	<b>Prikaz tkanine .....</b>	<b>14</b>
4.1	Izračun fizike.....	14
4.2	Sustav svjetla.....	16
4.3	Izračun normala tkanine.....	16
4.4	Sjenčanje vrhova.....	18
4.5	Spajanje teksture .....	19
4.6	Iscrtavanje tkanine .....	20
<b>5</b>	<b>Primjena vanjskih sile na tkaninu .....</b>	<b>22</b>
5.1	Metoda selektiranja objekata.....	22
5.2	Simulacija djelovanja sile .....	24
5.2.1	Simulacija udarca .....	25
5.2.2	Simulacija vjetra.....	25
<b>6</b>	<b>Prikaz simulacije tkanine .....</b>	<b>26</b>
6.1	Sustav scene .....	26
6.2	Utjecaj promjene parametara .....	28
6.3	Pregled programske podrške .....	29
<b>7</b>	<b>Zaključak.....</b>	<b>31</b>
<b>8</b>	<b>Literatura.....</b>	<b>32</b>

<b>9</b>	<b>Sažetak .....</b>	<b>33</b>
<b>10</b>	<b>Tablica slika .....</b>	<b>34</b>



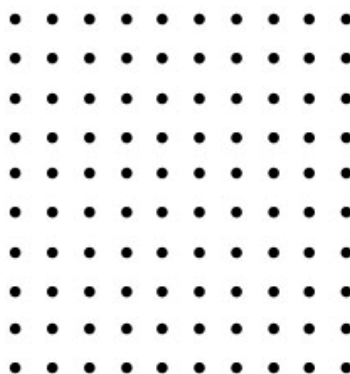
# 1 Uvod

Jedna od zadaća računalne grafike jest prikazivanje i simuliranje dinamičkih procesa poput modela dima, fluida te tkanine o kojemu je riječ u ovome radu. Kako bi se preslikao proces stvaranog svijeta te prikazao pomoću računala potrebno je obavljati simulaciju fizike nad tim dinamičkim procesom kreirajući tako njegov model. Simulacijom se preslikava realna domena svijeta u diskretnu domenu računala tako da pri tome ne gubimo na realnosti ponašanja simuliranog modela. Ova tehnologija je još u razvitku te će u budućnosti omogućiti prikaz složenih fizikalnih pojava u stvarnom vremenu. Razni dinamički procesi se na ovaj način mogu simulirati te tako koristiti u razne svrhe kao što su istraživanja i proučavanja određenih pojava te povećanja realnosti unutar računalnih igara, simulacija i unutar filmske industrije.

Ovaj rad opisuje način prikaza modela tkanine, simulacije njezinog ponašanja uz mogućnost dinamičkog mijenjanja parametara koji ju definiraju. Također će se prikazati i omogućiti interakcija s tkaninom, djelovanjem vanjskih sila na nju kao što su impuls udarca i vjetra. Koristi se OpenGL standard te je sama izvedba tkanine razrađena u programskom jeziku C# koristeći .NET Framework i OpenTK sučelje koje omogućava komunikaciju između programskog jezika i OpenGL-a.

## 2 Fizikalni model tkanine

Da bi se izradila simulacija tkanine potreban je fizikalni model koji ju opisuje. Tkanina je sastavljena od određene sirovine koja je isprepletena čineći tako stabilnu strukturu isprepletenih niti. Ovakav model je kompleksan pa je time i izračun svih sila unutar tkanine velike složenosti. Radi toga prelazi se na diskretizaciju tkanine te kreiranje modela koji ju na jednostavan način opisuje. Na početku potrebno je odrediti sastavne dijelove tkanine te smanjiti njihovu veličinu kako bi se mogla prikazati i kako bi bila iskoristiva u daljnjem postupku modeliranja. Koristi se model tkanine prikazan slikom 2.1, takav model se koristi u ovome radu kao inicijalni kojemu je definirano 100 čestica tkanine. Ovim pristupom smanjuje se kompleksnost, veličina te sama složenost modela tkanine uz očuvanje njezine strukture te dojma realnosti.

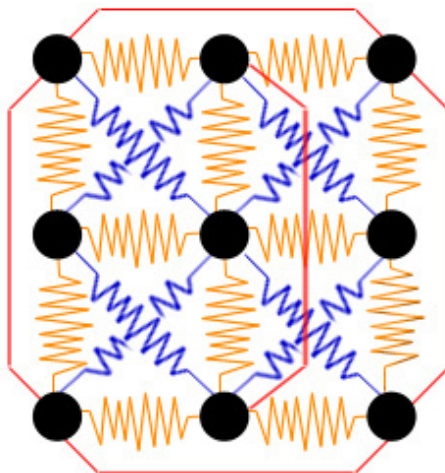


*Slika 2.1 Reprerentacija teksture diskretizacijom točkaka.*

Daljnjom aproksimacijom tkanine koristi se model čestica povezanih oprugama kojima se prezentiraju čestice tkanine te opruge kao niti kojima je tkanina povezana. Prilikom izračuna fizike nad modelom tkanine potrebno je provesti postupak diskretizacije postavljenog fizikalnog modela. Opisane su dvije metode numeričke integracije klasična Eulerova i Verlet. Obje vrste integracije potrebne za izračun fizike su primijenjene u ovome radu te se na kraju koristi Eulerova radi veće efikasnosti kojom se smanjuje potrebno vrijeme izračuna.

## 2.1 Model opruga i čestica

Model unutarnje strukture tkanine se može prikazati sustavom čestica povezanih oprugama (Slika 2.2), gdje je svaka čestica na tkanini povezana okolnim česticama pomoću opruga čineći tako sustav opruga.



*Slika 2.2 Prikaz sustava čestica povezanih oprugama.*

Na slici 2.2 prikazan je sustav opruga, crnom kružnicom predstavljena je čestica tkanine, a linijama u boji opruge. Postoje tri vrste opruga unutar modela tkanine: strukturne, smične i pregibne. Slikom 2.2 prikazane su strukturne opruge (narančastom bojom) koje djeluju horizontalno i vertikalno, smične opruge (plavom bojom) koje djeluju dijagonalno i pregibne opruge (crvenom bojom) koje djeluju između svake tri čestice čineći tako stabilnost od pregiba čestice tkanine. Bez pregibnih opruga model je dovoljno stabilan, ali ako želimo dobiti što realniji prikaz tada ih treba uzeti u obzir.

Ovakvim pristupom na pojedinu česticu djeluju sile okolnih opruga i vlastita sila teže. Tako na središnje čestice djeluje ukupno trinaest sila, od toga dvanaest okružujućih opruga i gravitacijska sila. Na rubne slučajeve tkanine djeluje različiti broj sila ovisno o položaju pojedine čestice te broju susjednih čestica.

Ponašanje čestice i njezino simuliranje je česta radnja u računalnoj grafici koja se bavi prikazom dinamičkog procesa svijeta. Stvaranjem mreža čestica vezanih oprugama i implementiranjem fizikalno temeljenog ponašanja u pozadini, dobivamo složene dinamičke procese. Stoga, najvažniji dio tkanine jest njezina čestica (Slika 2.3) koja ima svoju masu, položaj u prostoru, brzinu i akceleraciju.



*Slika 2.3 Reprzentacija čestice tkanine.*

Položaj:  $x(t)$

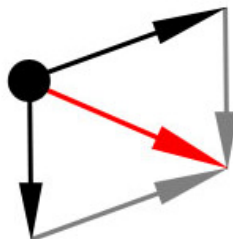
Brzina:  $v(t) = \frac{dx(t)}{dt}$

Akceleracija:  $a(t) = \frac{dv(t)}{dt} = \frac{d^2x(t)}{dt^2}$ ,

kada uzmemo u obzir konstantu akceleraciju tada dobivamo sljedeće izraze:

Brzina:  $v(t) = \int a dt = v_0 + a_p t$

Položaj:  $x(t) = \int v dt = x_0 + v_0 t + \frac{a_p t^2}{2}$



*Slika 2.4 Prikaz djelovanja sila na česticu.*

Sila opruge kojom su čestice vezane izračunava se pomoću Hookovog zakona. Hookov zakon elastičnosti je aproksimacija koja tvrdi da ako neku oprugu rastegnemo za udaljenost



$\Delta x$  od njenog položaja mirovanja, rezultirajuća sila  $F_0$ , koju stvara opruga, je proporcionalna udaljenosti  $\Delta x$  i konstanti opruge  $k$ , a njen smjer je suprotan od smjera pomaka:

$$\vec{F}_0 = -k * \vec{\Delta x},$$

gdje  $k$  predstavlja konstantu opruge, a  $\vec{\Delta x}$  je razlika osnovne duljine opruge  $l_0$  sa njezinom trenutnom  $l_i$ , odnosno:

$$\vec{F}_0 = -k * (\overrightarrow{l_0 - l_i})$$

Kada bi se implementirao samo ovakav model opruga i njezinih čestica puštanjem u titranje on bi oscilirao u beskonačno jer nema gubitka energije u sustavu. Radi toga u realnom svijetu postoji sila prigušenja  $F_b$  koja prigušuje osciliranje ovakvoga modela.

$$\vec{F}_b = -b * \vec{v},$$

gdje  $b$  predstavlja faktor prigušenja, a  $v$  brzinu promjene položaja opruge. Brzinu dobivamo skalarnom projekcijom vektora brzine na jedinični vektor smjera opruge. Uvođenjem ove sile nad modelom postignuto je prigušeno titranje i njegova stabilnost.

Dakle nad jednom česticom mase  $m$  povezanom oprugom djeluje ukupna sila  $F_{uk}$  te ju možemo prikazati formulom:

Gravitacijska sila:  $\vec{F}_g = m * \vec{g}$

Sila opruge:  $\vec{F}_0 = -k * \vec{\Delta x}$

Sila prigušenja:  $\vec{F}_b = -b * \vec{v}$

Ukupna sila:  $\vec{F}_{uk} = \vec{F}_g - \vec{F}_0 - \vec{F}_b$

$$\vec{F}_{uk} = m * \vec{g} - k * \vec{\Delta x} - b * \vec{v}$$

Dani sustav čestica povezanih oprugama možemo prikazati sljedećom formulom, tako da računamo ukupnu silu za svaku česticu  $j$ , koju okružuje  $n$  opruga.

$$\vec{F}_{j,uk} = m * \vec{g} - \sum_{i=1}^n (k * \vec{\Delta x}_i + b * \vec{v}_i)$$

Kada izračunamo silu koja djeluje na pojedinu česticu tada preko dvostruke integracije dobivene akceleracije  $a_i$  određujemo poziciju čestice. Kako je limitirano znanje o akceleraciji čestice tijekom cijele simulacije na trenutnu i prethodnu vrijednost, tada prelazimo na model numeričke integracije kako bi odredili približnu vrijednost sljedeće pozicije čestice.

$$x_i = \iint a_i dt dt$$

## 2.2 Metode integracije

Kako je model koji izgrađujemo diskretan, odnosno vrijeme unutar simulacije nije kontinuirano, potrebno je uvesti model numeričke integracije. U ovom radu koristi se vrijeme osvježavanja izračuna fizike pod frekvencijom od 60Hz što je dovoljno za glatku predodžbu simuliranja modela tkanine. Tkanina se može prikazivati i manjom brzinom osvježavanja na zaslonu, ali potrebno vrijeme za postupak izračuna fizike određuje model integracije. Duljim vremenom izračuna fizike, odnosno smanjenjem vremena otipkavanja realnog svijeta, povećavamo stabilnost simuliranog modela, ali time djelujemo i na brzinu izvođenja cijelog programa koja se tada povećava. Ako taj broj prijeđe ispod granice od 25Hz (njime osvježavamo model tkanine) tada više ne možemo vršiti simulaciju u realnom vremenu jer se izračun fizika u pozadini našeg modela ne stigne u potpunosti izračunati. Predstavljena su dva modela integracije, njihove prednosti i mane.

### 2.2.1 Verlet metoda

Verlet metoda numeričke integracije omogućava efikasnu aproksimaciju integracije, njenim uvođenjem potrebno je pamtili prošlu i sadašnju poziciju pojedine čestice. Ovoj metodi nije potrebna informacija o brzini čestice te time skraćuje dodatno izračunavanje njene

vrijednosti. Sljedeći izraz opisuje dobivanje nove pozicije čestice kojoj je za izračun potrebna akceleracija  $a_t$ , prethodna pozicija  $x_{t-\Delta t}$  i vrijeme uzrokovanja  $\Delta t$ .

Sljedeća pozicija točke: 
$$x_{t+\Delta t} = (2 - f)x_t - (1 - f)x_{t-\Delta t} + a_t(\Delta t)^2,$$

gdje  $f$  predstavlja djelić brzine koji se izgubi radi trenja, on se unosi kao broj između (0-1), u ovom radu za  $f$  je isprobana vrijednost od 0,01 kojom je stabilizirana simulacija. Bez parametra  $f$  sustav opruga ne bi gubio energiju te bi ušao u harmonično titranje. Ovaj parametar se koristi umjesto sile prigušenja nad oprugom i samim time stabilizira tkaninu. Verlet metoda djeluje globalnom silom prigušenja preko parametra  $f$  te tako umanjuje moć prikaza malih, za potrebe simulacije, utjecajnih sila. Radi toga je potrebno smanjivati vrijeme otipkavanja  $\Delta t$  kako bi i te sile došle do izražaja. Smanjivanjem  $\Delta t$  simulacija ima popratni efekt radi kojega je potrebna veća količina osvježavanja fizike, što na kraju dovodi do njene slabe iskoristivosti.

## 2.2.2 Eulerova metoda

Klasičan pristup je Eulerova metoda numeričke integracije, a ona glasi:

Izračun brzine:  $\dot{v} = a \quad v_{n+1} = v_n + a_n \Delta t$

Izračun pozicije:  $\dot{x} = v \quad x_{n+1} = x_n + v_{n+1} \Delta t,$

gdje je  $\Delta t$  vrijeme između dvije iteracije izračunavanja fizike nad česticom, odnosno ako se izračun izvodi pod frekvencijom od 60Hz, tada  $\Delta t$  iznosi 1/60 sekundi.

Za implementaciju Eulerove metode potrebno je pamtiiti brzinu pojedine čestice i njezinu poziciju. Ova metoda nema u sebi ugrađenu stabilizaciju kao verlet metoda. Time se omogućuje veća fleksibilnost i mogućnost djelovanja slabih sila koje neće biti prigušene kao kod verlet metode integriranja. Radi toga ovom metodom se mora dodatno izračunavati brzina te implementirati prigušenje nad oprugama tkanine. Povećavanjem dodatnih izračuna unutar Eulerove metode je jeftinije i isplativije od veće količine osvježavanja fizike unutar verlet

metode. Za isti broj točaka tkanine verlet metodom, koja je naizgled praktičnija varijanta, dobivamo šest puta manju isplativost od Eulerove metode.

### **2.3 Stabilizacija fizikalnog modela tkanine**

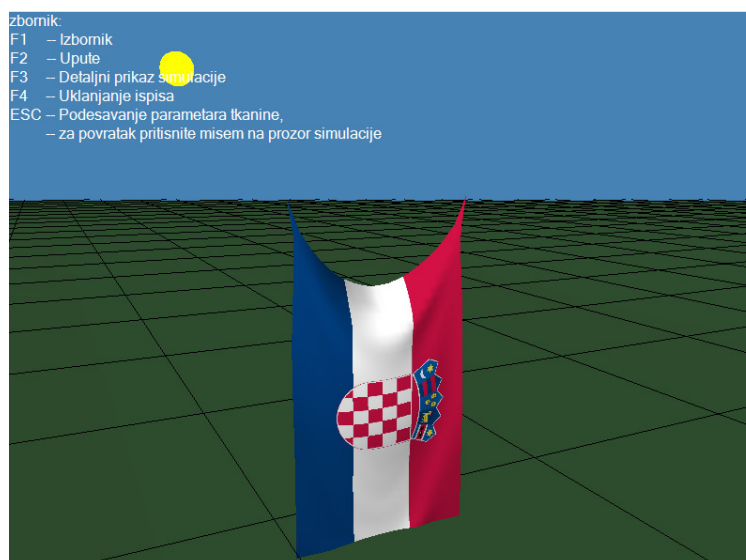
Jedan od najvećih problema koje se javlja unutar simulacije modela rada čestica jest njihova stabilizacija, a razlog tomu jest diskretizacija realnog svijeta. Ovom problemu prilazi se na razne načine ali sve se one svode na uvođenje gubitka energije u sustav kojim osiguravamo konačno prigušenje.

Kada koristimo verlet metodu integracije nije potrebna dodatna stabilizacija kako bismo izbjegli osciliranje fizikalnog modela tkanine. Verlet metoda, na svaku iduću iteraciju izračuna fizike, umanjuje doprinos te iteracije pa time zapravo imitira globalnu silu prigušenja koja se može gledati kao sila otpora zraka.

Primjenom Eulerove metoda integracija moramo koristiti silu prigušenja koje djeluje nad domenom opruga. Prigušivanjem oscilacija opruge, unatoč njenoj stabilizaciji, javlja se jedna globalna oscilatorna pojava manjka otpora zraka, koja je neovisna od stabilnosti interne strukture tkanine. Ona je pomoću verlet metode bila utišana jer je pomoću nje djelovalo globalno prigušenje nad svim silama unutar modeliranog sustava. Korištenjem Eulerove metode, mora se naknadno unijeti stabilizacija tkanine silom prigušenja, odnosno uvođenje otpora zraka. Jednostavnom aproksimacijom možemo dobiti da sila otpora zraka djeluje na pojedinu česticu tkanine tako da joj svaki put umanjuje 10% vrijednosti njene brzine. Time dobivamo jednu aproksimativnu silu prigušenja koja osigurava stabilnost simulacije modela tkanine.

### 3 Model tkanine

Model tkanine unutar računalnog programa predstavljena je klasom *cloth*. Ona sadržava svoje vlastite attribute i metode koji ju opisuju te sadrži listu točaka od kojih je sastavljena. Klasa tkanine ima implementirane privatne strukture i metode koje se bave izračunom fizike i sjenčanjem. Javno sučelje tkanine omogućava njeno kreiranje preko konstruktora klase, djelovanje vanjskih sila koje se implementiraju unutar fizike tkanine i iscrtavanje kojim se prikazuje simulacija njenog modela.



*Slika 3.1 Prikaz modela tkanine.*

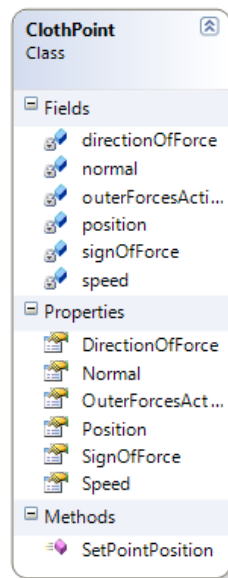
Ovakvim pristupom omogućena je promjena gotovo svih unutarnjih atributa modela tkanine, stvaranje objekata tipa tkanine s različitim parametrima mijenjanjem tako njezine simulacije.

#### 3.1 Reprezentacija točke

Kao što smo rekli čestica je sastavni dio tkanine te je njezina implementacija neophodna za daljnji prikaz tkanine. Skup čestica na tkanini možemo smatrati kao jednu točku tkanine koja će svojim ponašanjem simulirati skupinu čestica. U daljnjem radu skup čestica ćemo

zvati točka tkanine te ćemo nad njom vršiti sve daljnje radnje kao što su pomak te primjena unutarnjih i vanjskih sila.

Točka tkanine predstavljena je klasom *ClothPoint*, svaki objekt tipa točke u sebi sadrži poziciju *position*, brzinu *speed* te vektor normale *normal* (kasnije objašnjeno, potrebno radi sjenčanja tkanine). Točke se kreiraju pri stvaranju tkanine te je moguće zadati broj vertikalnih i horizontalnih točaka. Postavljanje točke obavlja se metodom *SetPointPosition* koja pojedinoj točki tkanine odnosno njegovom objektu postavlja inicijalnu poziciju i brzinu. Masa pojedine točke definirana je globalno za svaku točku tako da je masa jednoliko unificirana.



Slika 3.2 Klasa *ClothPoint* njezini atributi, svojstva i metode.

Nad točkama se obavlja izračun fizike te isctavanje modela tkanine, točke su povezane oprugama jednostavnim adresiranjem, pa nije potrebno pamtiti susjedne točke tkanine unutar samog objekta čime se smanjuje potrebna memorija za pohranu pojedine točka.

Također točka ima definiranu logičku varijablu *outerForceActivated* koja omogućava djelovanje vjetra te udarca na pojedinu točku, a pri tome se koriste statički atribut *signOfForce* i dinamički *directionOfForce*, koji će biti detaljno objašnjeni u djelu rada koji se bavi primjenom vanjskih sila.

## 4 Prikaz tkanine

Za prikaz tkanine potrebno je proći kroz određene korake kako bi ju pripremili za iscrtavanje na zaslon. Jedan od glavnih koraka je izračun fizike koji se obavlja za svaku točku tkanine te opisuje njen fizikalni model. Zatim se pripremaju postavke potrebne za sjenčanje kojoj je uz definiran sustav svjetla potreban izračun normala tkanine te se primjenjuje samo sjenčanje. Nakon definiranog svjetla, tkanini se može dodijeliti boja pojedinom retku ili se na nju povezuje tekstura koja se lijepi na točke tkanine. Nakon svih prethodno navedenih koraka tkanina je spremna za iscrtavanje na zaslon unutar sustava scene.

### 4.1 Izračun fizike

Računanje fizike obavlja se frekvencijom od 60Hz kojom dobivamo glatku reprezentaciju realnog svijeta preslikanu u računalni model. Na početku se definira inicijalno stanje tkanine postavljanjem u sustav scene te definiranjem brzine pojedine točke na nulu. Točke tkanine su interno povezane oprugama te se proširivanjem ili smanjivanjem broja točaka tkanine automatski kreiraju novi objekti točaka i opruge kojima su vezane.

Postoje razni atributi koji definiraju ponašanje i izgled tkanine, a to su:

- Gravitacija
- Masa čestice
- Konstanta opruge
- Prigušenje opruge
- Prigušenje zraka
- Razmak između točaka
- Vrijeme otipkavanja
- Horizontalan i vertikalni broj točaka na tkanini

Kako postoje tri vrste opruga, kao što je rečeno prije, tako se za svaku točku izračunava doprinos sile okružujućih opruga. Kako se radi o indeksiranom pristupu, tako se za svaku točku preko njenog indeksa zna koje su joj okružujuće točke. Okružujuća točka je

predstavljena kao broj koji moramo pribrojiti indeksu određenoj točki kako bi ju indeksirali. Definiran je pobrojani tip relativnih pozicija: gore lijevo, gore, gore desno, lijevo, desno, dolje lijevo, dolje ili dolje desno od trenutne točke tkanine. On predstavlja vrijednosti koju moramo pribrojiti da bi indeksirali točku tkanine ovisno o relativnoj poziciji. Svaka od opruga (strukturna, smična i pregibna) ima definiranu nazivnu duljinu koja se dinamički postavlja pri kreiranju tkanine jer ovisi o definiranom razmaku između postavljenih točaka tkanine. Izračun fizike svodi se na zbrajanje doprinosa svih sila koje okružuju pojedinu točku. Algoritam započinje zbrajajući izračunate sile opruga okružujućih točaka te njihovih prigušenja. Dobivenoj sumi dodamo gravitacijsku silu trenutne točke te dobivamo ukupnu silu  $F_{ui}$  točke  $i$ .

$$F_{ui} = \sum_{j=0, j \neq i}^k (F_{opruga, j} + F_{prigusenja, j}) + m_i * gravitacija$$

Nakon što su sve sile koje djeluje na trenutnu točku izračunata tada preko Eulerove integracije dobivamo novu brzinu točke te njenu poziciju. Te vrijednosti spremamo u objekt točke prepisivanjem prethodno izračunatih vrijednosti. Ovaj postupak obavlja se za sve točke tkanine, osim za točke koje predstavljaju učvršćene dijelove tkanine koje jednostavno preskačemo prilikom izračuna fizike.

Prilikom upisivanja nove brzine radi se aproksimacija prigušenja zraka te se novoj brzini oduzima 10% njene vrijednosti kako bi unijeli stabilnost unutar modela. Ova stabilizacija je različita od stabilizacije odnosno prigušenja nad oprugama, jer iako opruge mogu biti u stabilnom stanju tkanina i dalje može titrati u okomitom smjeru na opruge, a da su pri tome sve sile zadovoljene i uravnotežene.

Svakoj točki tkanine se pored unutarnjih sila modela sustava opruga dodaje i vanjska sila. Jednu od njih smo spomenuli i ona je gravitacijska ali pored nje unose se vektorski iznosi vanjske sile kojom korisnik djeluje na tkaninu (objašnjeno u dijelu koji se bavi utjecajem vanjskih sila). Kada je izračun fizike završio tada imamo spremljene nove vrijednosti brzine pojedine točke te vrijednosti njihove nove pozicije koju treba prikazati odnosno iscrtati na ekranu.



## 4.2 Sustav svjetla

U sceni se uz tkaninu nalazi i sunce koje predstavlja izvor svjetla. Svjetlo može djelovati u svim smjerovima od trenutne pozicije ili usmjereno. Svjetlo je moguće mijenjati attribute preko OpenGL poziva kao što su ambijentalnu, difuznu i zrcalnu komponentu osvjetljenja te poziciju samog izvora svjetla ili njen vektor djelovanja u slučaju kada se radi o usmjerenom svjetlu.



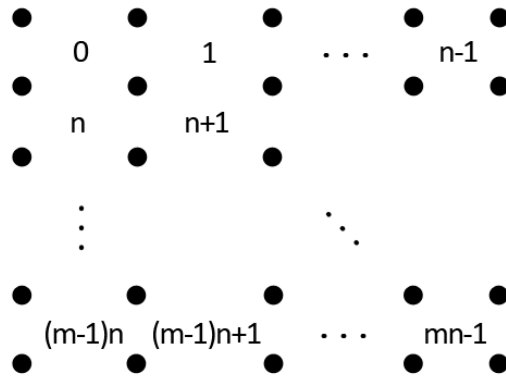
Slika 4.1 Prikaz objekta sunca, usmjerenog i radijalnog djelovanja te parametara koji ga opisuju.

Kako bi se primijetila simulacija tkanine potrebnu ju je osvijetliti i primijeniti sjenčanje. Kada bi tkanina bila jednobojna tada ne bismo primijetili njeno gibanje jer ne bi imali osjećaj dubine predmeta. Radi toga potrebno je definirati normale tkanine nad svakom njenim poligonom kako bismo primijenili sjenčanje i time dobili dojam dubine.

## 4.3 Izračun normala tkanine

Normale poligona statičkog objekta unutar računalne grafike potrebno je izračunati samo jedanput i to na samome početku. Tkanina s druge strane predstavlja dinamički objekt

sačinjen od točaka koje se gibaju i time čine simulaciju. Za nju je potreban dinamički izračun normala koji se obavlja kroz cijelu simulaciju. Tkaninu možemo reprezentirati slikom 4.2 koja prikazuje odnos poligona tkanine i njenih sastavnih točaka.



Slika 4.2 Predstavlja reprezentaciju tkanine raspodijeljenu na  $n*m$  broja poligona.

Kada se na tkaninu gleda na ovakav način, tada možemo odrediti normale pojedinog poligona tkanine. Kako tri točke određuju ravninu, tada uzimajući u obzir da je i četvrta točka ujedno i točka poligona, odnosno da je u istoj ravnini kao prethodne tri, unosimo malu pogrešku pri izračunu. Ta je pogreška zanemariva, a smanjena je složenost izračuna normala sa  $2nm$  na  $nm$  potrebnih poligona te njihovih normala.

Normale jednostavno izračunamo znajući pozicije svake točke tkanine. Tada iz pojedine tri vektorskim produktom dobivamo normalu poligona. Taj proces radimo za sve parove točaka unutar tkanine dok ne dobijemo sve tražene normale odnosno njih  $mn$  prikazano na slici 4.2. Dan je blok koda koji izračunava vektor normale za određeni poligon tkanine.

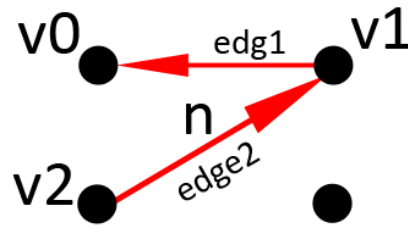
```
v0 = listOfPoints[j * clothWidth + i].Position;
v1 = listOfPoints[(j + 1) * clothWidth + i].Position;
v2 = listOfPoints[j * clothWidth + i + 1].Position;

edge1 = Vector3d.Sub(v0, v1);
edge2 = Vector3d.Sub(v1, v2);

tempNormal = Vector3d.Normalize(Vector3d.Cross(edge1, edge2));
listOfNormals.Add(tempNormal);
```

Preko tri vrha  $v_0$ ,  $v_1$  i  $v_2$  međusobnim oduzimanjem dobivamo dva ruba  $edge_1$  i

edge2. Vektorski produkt tih rubova predstavlja normalu tog poligona sastavljenog od tri vrha. Ova radnja obavlja se za sve poligone tkanine.

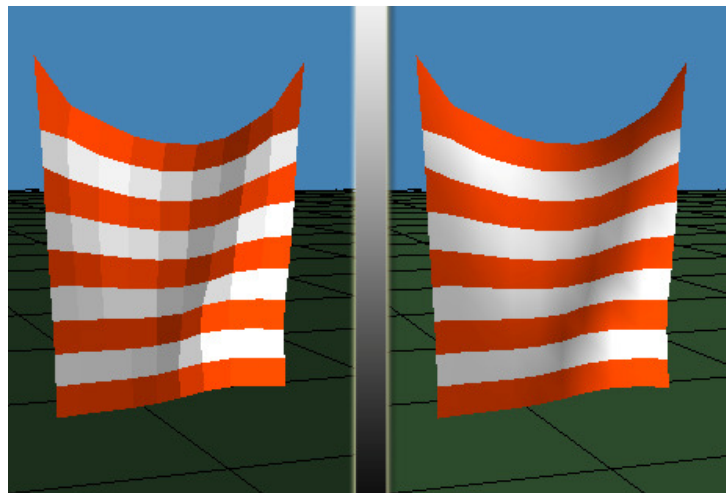


*Slika 4.3 Preko vektorskog produkta između dvaju rubova poligona dobivamo normalu  $n$ -tog poligona.*

Kada izračunamo sve normale poligona, prelazimo na metodu sjenčanja vrhova kojima aproksimiramo glatke prijelaze između sada konstantno osjenčanih poligona.

#### **4.4 Sjenčanje vrhova**

Kada ne bi koristili sjenčanje vrhova dobili bi tkaninu kao na slici 4.4a) zato je potrebno primijeniti ovu metodu kako bi dobili ispravne i glatke prijelaze te učinili tkaninu realnijom (slika 4.4b).

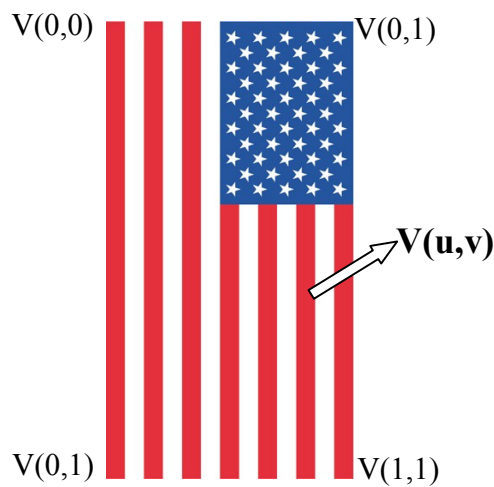


*Slika 4.4 a) Konstantno sjenčanje primijenjeno na poligone tkanine, b) Sjenčanje vrhova.*

Sjenčanje nad vrhovima radi tako da svakoj točki tkanine definiramo zaseban vektor normale. Da bi odredili normalu točke potrebno je poznavati sve normale poligona koji ju okružuju (ovaj korak je prethodno objašnjen). Sumirajući normale okružnih poligona dobivamo vektor normale dotične točke. Prilikom funkcijskog poziva OpenGL-a šaljemo vektor normale i poziciju točke. OpenGL automatski primjenjuje Phongov model osvjetljenja s Gouraudovim postupkom sjenčanja.

#### 4.5 Spajanje teksture

Tekstura daje realnost modeliranom objektu tkanine te je u ovom radu omogućeno dinamičko učitavanje teksture. Tekstura se učitava i preslikava na kvadrat. Tekstura je definirana s dvije koordinate koje se kreću između  $u=[0,1]$ ,  $v=[0,1]$ , funkcijom  $V(u,v)$ . Vrijednost funkcije  $V(0,0)$  predstavlja gornji lijevi vrh preslikane teksture, a  $V(1,1)$  donji lijevi. Odnosno koristi se bilinearna interpolacija.

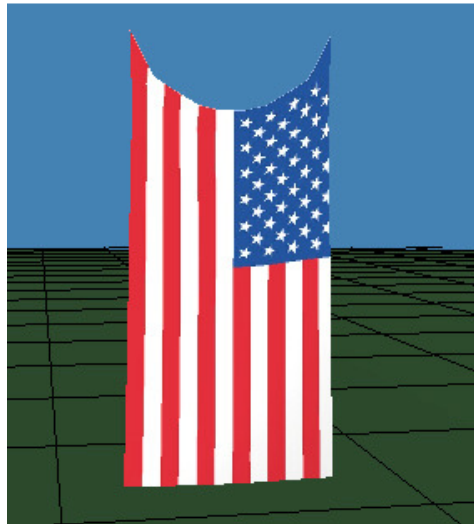


Slika 4.5 Prikaz teksture, bilinearna interpolacija.

Bilinearna interpolacija je vrsta interpolacije kod koje se vrijednost novog slikovnog elementa izračunava na temelju vrijednosti četiriju susjeda: lijevog, desnog, gornjeg i donjeg. Na taj način preslikavamo koordinate slike na koordinate bilo kojeg poligona koristeći donju formulu. U našem slučaju poligoni na koje preslikavamo teksturu su zapravo poligoni tkanine.

$$V(u, v) = (1 - u)(1 - v)V_{0,0} + u(1 - v)V_{1,0} + (1 - u)vV_{0,1} + uvV_{1,1}$$

Tekstura se preslikava na tkaninu tako da na pojedinu točku tkanine lijepi dio teksture. Tekstura se tako skalira, rasteže odnosno deformira kako se mijenjaju pozicije točaka tkanine. Korištenjem bilinearne interpolacije tekstura je vezana za točke tkanine, gdje npr. zadnja točka tkanine dobiva koordinate  $V(1,1)$  od teksture.

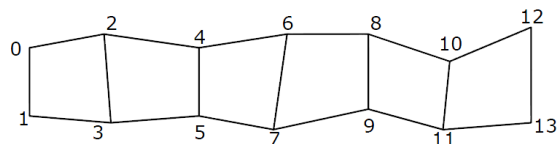


*Slika 4.6 Prikaz preslikane teksture na točke tkanine.*

#### **4.6 Iscrtavanje tkanine**

Za prikaz tkanine potrebno je proći sve do sada navedene korake. Izračun fizike i proračun normala vrhova potrebno je odrediti prije poziva iscrtavanja objekta. Nakon svih ovih postupaka dobivamo prikaz koji se osvježava te pri tome ponovo izračunava pod frekvencijom od 60Hz.

Tkanina se iscrtava pomoću funkcijskog poziva OpenGL-u zvanom „QUAD\_STRIP“, on se koristi jer se njime izbjegavaju ponovna iscrtavanja već postojećih točaka modela tkanine, te se time smanjuje i sama složenost iscrtavanja.



*Slika 4.7 Prikaz redoslijeda slanja vrhova prilikom korištenja „QUAD\_STRIP“ načina iscrtavanja primitiva.*

Brzina prolaska kroz izračun ovisi o veličini tkanine u ovom radu postignuta je veličina od 1000 točaka tkanine što je i više nego dovoljno za realnu i ugodnu simulaciju tkanine za koju je dovoljno već 100 točaka.

Prilikom poziva iscrtavanja tkanine obavlja se sljedeći blok koda. On se obavlja za svaku točku tkanine (i,j).

```
GL.TexCoord2(i / (float)clothWidth, j / (float)clothHeight);
GL.Normal3((Vector3)listOfPoints[j * clothWidth + i].Normal);
GL.Vertex3((Vector3)listOfPoints[j * clothWidth + i].Position);

GL.TexCoord2(i / (float)clothWidth, (j + 1) / (float)clothHeight);
GL.Normal3((Vector3)listOfPoints[(j + 1) * clothWidth + i].Normal);
GL.Vertex3((Vector3)listOfPoints[(j + 1) * clothWidth + i].Position);
```

Prvo se iscrtava prvi red tkanine, a zatim drugi do zadnjeg reda. U bloku koda možemo vidjeti način preslikavanja koordinata teksture pomoću `GL.TexCoord2` i dodjeljivanja normala `GL.Normal3` pojedinoj točki `GL.Vertex3` tkanine. Sustav prikaza tkanine napravljen je tako da je neovisan od broja točaka i veličine dane teksture. Točkama tkanine jednostavno pristupamo preko generičke liste `listOfPoints` koja sadrži objekte tipa `ClothPoint`.

## 5 Primjena vanjskih sile na tkaninu

Kako bi se prikazala simulacija i interakcija modela tkanine potrebno je simulirati djelovanje vanjskih sila na nju. Jedna od vanjskih sila koja je uvijek prisutna je gravitacijska sila. Kada se pokrene simulacija, možemo vidjeti kako gravitacijska sila djeluje na tkaninu tako što se počinje rastezati te se pri djelovanju sila opruga i sila prigušenja stabilizira. Tkanina se sada nalazi u 2D prostoru djelovanja sila, jer na početku simulacije ni jedna sila ne djeluje u smjeru različitom od ravnine u kojoj se nalazi tkanina.

U ovom radu implementirane su dvije vrste sila: vjetar i udarac. Svako od njih možemo mijenjati jačinu, prostor djelovanja te njihov smjer djelovanja. Jedan od pristupa je takoreći klasični pristup određivanja smjera djelovanja vanjske sile. On se postiže projiciranjem pravca u smjeru sile te određivanjem udaljenosti između tog pravca i pojedine točke tkanine kako bi odredili nad kojom točkom djeluje dotična sila. Taj način zahtjeva puno kalkulacije i izračuna te aproksimacije, što sve zajedno otežava prikaz i simulaciju tkanine u realnom vremenu. Ovdje se koristi metoda selektiranja objekata na prizoru kako bi odredili nad kojim točkama te iz kojeg smjera trebamo primijeniti vanjsku silu.

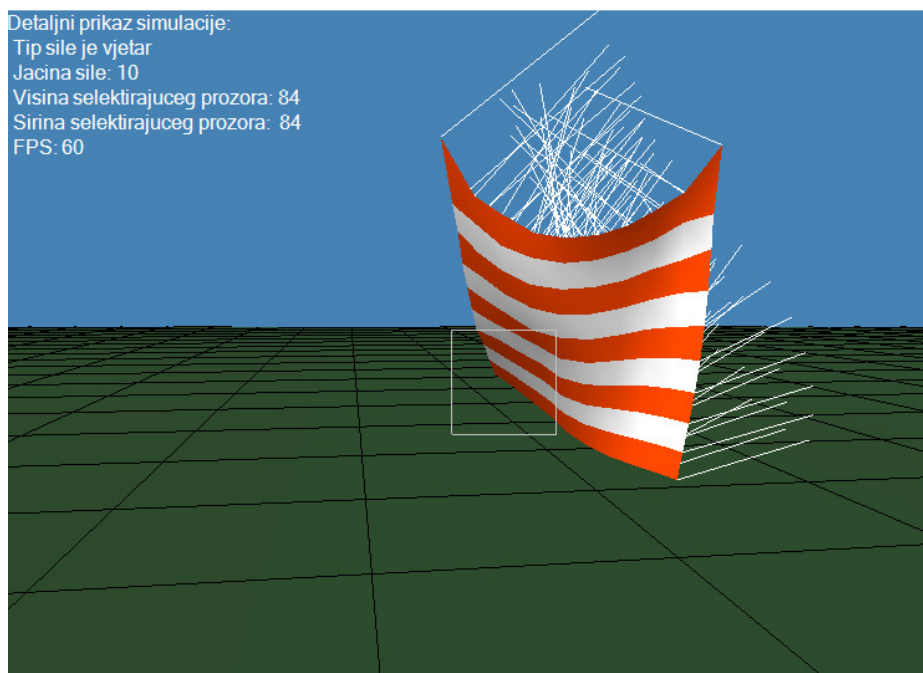
### 5.1 Metoda selektiranja objekata

Postoje ugrađene funkcije unutar OpenGL-a koje omogućavaju imenovanje objekata na sceni te ekstrakciju imena vidljivih objekata unutar prikaza. Prilikom pristupa ovoj metodi potrebno je iscrtavanje prikaza, ali radi bržeg i efikasnijeg iscrtavanja odbacuju se nepotrebne informacije sa scene kao što su tekstura, poligoni, sjenčanja modela tkanine i ostali objekti na sceni koji ne zaklanjaju objekt modela tkanine.

Prilikom generiranja takvog prikaza obavlja se jednostavno iscrtavanje točaka tkanine pozivom `GL.Begin(BeginMode.Points)`. Potrebno je prilikom slanja pozicije vrhova pojedinog objekta OpenGL-u postaviti i njegovu identifikaciju (u ovom slučaju naš objekt je zapravo točka tkanine) to se obavlja funkcijskim pozivom `GL.PushName(j * clothWidth + i)`, gdje ime predstavlja indeks točke tkanine. Kada smo definirali imena svim točkama

tkanine obavlja se iscrtavanje s funkcijskim pozivom `GL.RenderMode(RenderingMode.Select)`. Ovaj način iscrtavanja nije vidljiv korisniku već se iscrtava samo radi brze ekstrakcije imena vidljivih objekata unutar definiranog prizora.

Kako bi izvukli, odnosno spremili podatke o imenima objekata, iz prikaza moramo definirati spremnik, `GL.SelectBuffer(selectBufferSize, selectBuffer)`, u koji će se pohraniti imena dohvaćena ovim pristupom iscrtavanja te njihove udaljenosti od ravnine promatranja. Ovim pristupom dobivamo imena objekata koji se nalaze na cijelom prizoru no ako želimo ograničiti prostor djelovanja metode selektiranja tada moramo definirati `PickMatrix`. Pomoću `PickMatrix` vršimo rezanje trenutnog prikaza pravokutnikom visine `selectHeight` i širine `selectWidth` te pozicije toga pravokutnika unutar prikaza. Za poziciju pravokutnika unutar prikaza koriste se koordinate miša ( $x, Height - y$ ). Time ograničavamo metodu selektiranja tako da prihvaća samo nađene objekte unutar pravokutnika koji svojom visinom i širinom okružuje koordinate kursora miša.



*5.1 Prikaz djelovanje vanjske sile na tkaninu primjenom metode selektiranja, na slici se vidi pravokutnik koji dohvaća indekse točaka tkanine.*

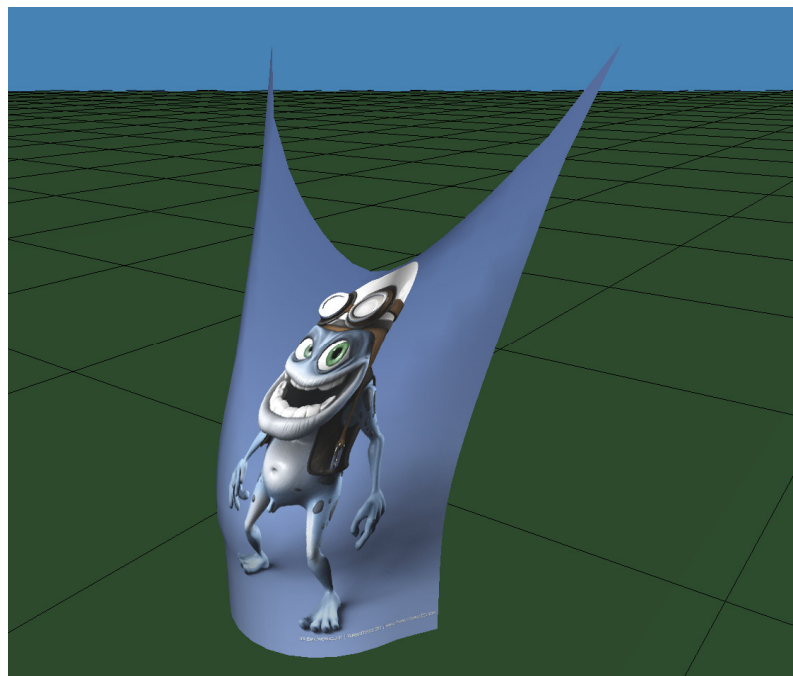
```
Glu.PickMatrix(x, Height - y, selectWidth, selectHeight, viewport);
```



Pozivom `int numberOfHits=GL.RenderMode(RenderingMode.Render)` dohvaćamo broj pogodaka odnosno broj spremljenih imena unutar `selectBuffer`, te pomoću tog broja možemo proći kroz `selectBuffer` i dohvatiti indekse točaka nad kojima trebamo primijeniti vanjsku silu. U ovom radu omogućeno je dinamičko proširivanje dimenzija pravokutnika. Tako povećavamo područje djelovanje sile, te njenog pomaka u ovisnosti o koordinatama kursora miša.

## 5.2 Simulacija djelovanja sile

Prethodno definiranom metodom određujemo nad kojim ćemo točkama primijeniti silu. Smjer sile je projekcija pogleda na tkaninu. Za određivanje smjera sile potrebni su nam vektori očista i gledišta trenutne pozicije i orijentacije kamere unutar scene. Oduzimanjem vektora očista od vektora gledišta dobivamo vektor koji je okomit na trenutni prizor i određuje smjer sile. Kako bi se omogućilo gledanje djelovanja sile s različitog mjesta od trenutne pozicije i usmjerenja, moguće je odabrati fiksnu poziciju vjetra te se tada ta pozicija, odnosno njezina transformacijska matrica, sačuva kako bi se prilikom potrebe iscrtavanja metodom selekcije mogla koristiti za određivanje smjera dolaska vanjske sile, a time i točaka nad kojima djeluje.



*5.2 Promatranje djelovanja sile na tkaninu sa različitim pozicija pozicioniranjem kamera po sceni.*

Moguće je pojačavanje ili smanjivanje sile te promjena smjera djelovanja. Sile su izražene u Newtonima te se lako mogu pribrajati ukupnoj sili koja djeluje na pojedinu točku tkanine. Prilikom odabira točaka nad kojima će djelovati sila trebamo na neki način dojaviti tim objektima da su pod utjecajem vanjske sile i iznosom njezinog vektora. Za to se koristi logička varijabla *outerForceActivated* kojom naznačujemo pojedinom objektu kako u idućoj iteraciji izračuna sile treba dodati impuls vanjske sile jačinom i smjerom definirane od strane korisnika *directionOfForce* koja je također zapisana unutar pojedinog objekta.

### 5.2.1 Simulacija udarca

Simulacija udarca je zapravo simulacija impulsa sile koji djeluje nad označenim točkama tkanine. Točke nad kojima djeluje impuls pronalazimo prethodnom metodom te im njihovim objektima podižemo logičku varijablu kojom naznačujemo djelovanje vanjske sile nad njom. Svakoj od njih definira se vektor sile te preko podignute oznake određujemo da se u idućoj iteraciji treba pribrojiti vektor vanjske sile silama koje djeluju na dotičnu točku tkanine. Možemo reći da smo oživjeli dotičnu točku tkanine. Kada se vektor impulsa sile pribroji ostalim silama tada se logička vrijednost djelovanja impulsa postavlja na laž, kako bi se simulirao samo impuls koji predstavlja kratki udarac. Prilikom djelovanja impulsa na centar tkanine možemo uočiti stvaranje kružnog vala koji se rasprostire po tkanini.

### 5.2.2 Simulacija vjetra

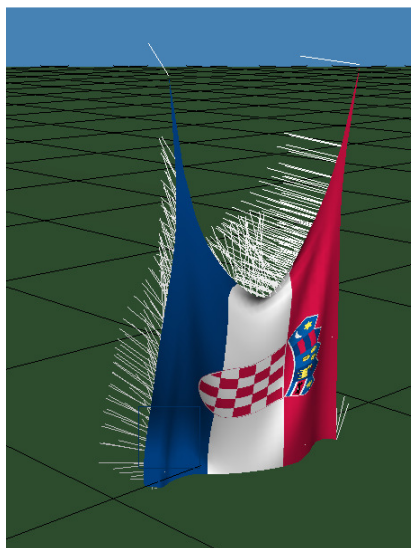
Simulacija vjetra djeluje slično kao i kod udarca ali s tom razlikom da se vanjska sila koja djeluje na tkaninu evaluiraju prilikom svakog prolaska kroz izračun fizike, a ne jedanput kao kod simuliranja udarca. Impuls sile vjetra se također pribraja silama koje djeluju na pojedinu točku te joj je jačina definirana od strane korisnika. Prilikom zaustavljanja vjetra impuls sile se više ne pridodaje pojedinoj točki tkanine te na nju djeluju samo unutarnje sile i gravitacijska sila. Promjenom smjera, jačine i pozicije sile možemo djelovati u isto vrijeme na više segmenata tkanine te time simulirati različite tipove vjetra i promatrati interakciju i reakciju fizikalnog modela tkanine.

## 6 Prikaz simulacije tkanine

Prikaz tkanine obavlja se frekvencijom od 60Hz što je i više nego dovoljno za glatku reprezentaciju njene simulacije. Pored sustava scene u kojoj se nalazi tkanina i objekti koristimo i zaseban prozor s postavkama. On omogućava mijenjanje parametara simulacije kojima možemo mijenjati fiziku tkanine te utjecati na način prikaza.

### 6.1 Sustav scene

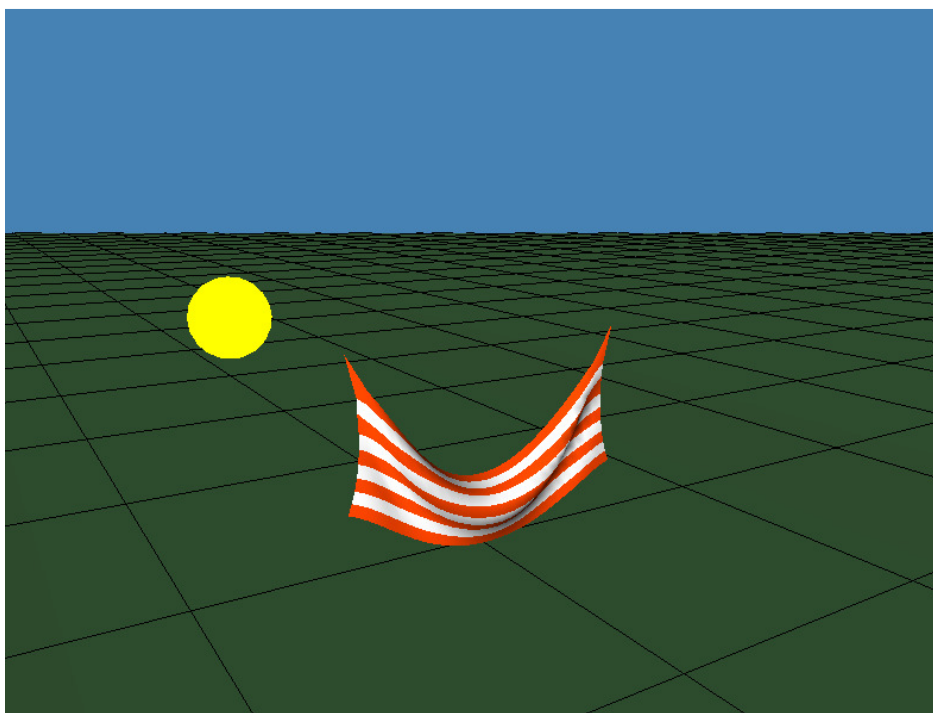
U sustavu scene se uz tkaninu nalazi sunce, pod i nebo. Ostali elementi doprinose osjećaju perspektive. Korisnik može mijenjati načine ispisa i prikaza tkanine prolaženjem kroz izbornik. Pritiskom tipki F1 (Izbornik), F2 (Upute), F3 (Detaljni prikaz simulacije), F4 (Uklanjanje ispisa) i ESC (Podešavanje parametara tkanine) mijenja se ispis unutar simulacije. Izbornik služi za navigaciju i opis pojedinog odabira, u uputama definirane su sve interakcije korisnika sa simulacijom te njihova pojašnjenja. Opcijom uklanjanja ispisa, prikazujemo čistu simulaciju bez okolnog teksta. Pritiskom tipke ESC otvara se prozor s postavkama tkanine te omogućava korisniku promjenu raznih parametara.



*Slika 6.1 Prikaz modela tkanine prilikom detaljnog prikaza simulacije.*

Najzanimljivija opcija je pod detaljnim prikazom simulacije unutar koje se iscrtavaju normale točaka tkanine i pravokutnik djelovanja sile. Unutar ovog načina iscrtavanja također se ispisuje neki od parametara fizike poput tipa trenutne sile, njezine jačine, visina i širina pravokutnika djelovanja sile i frekvencija osvježavanja prikaza.

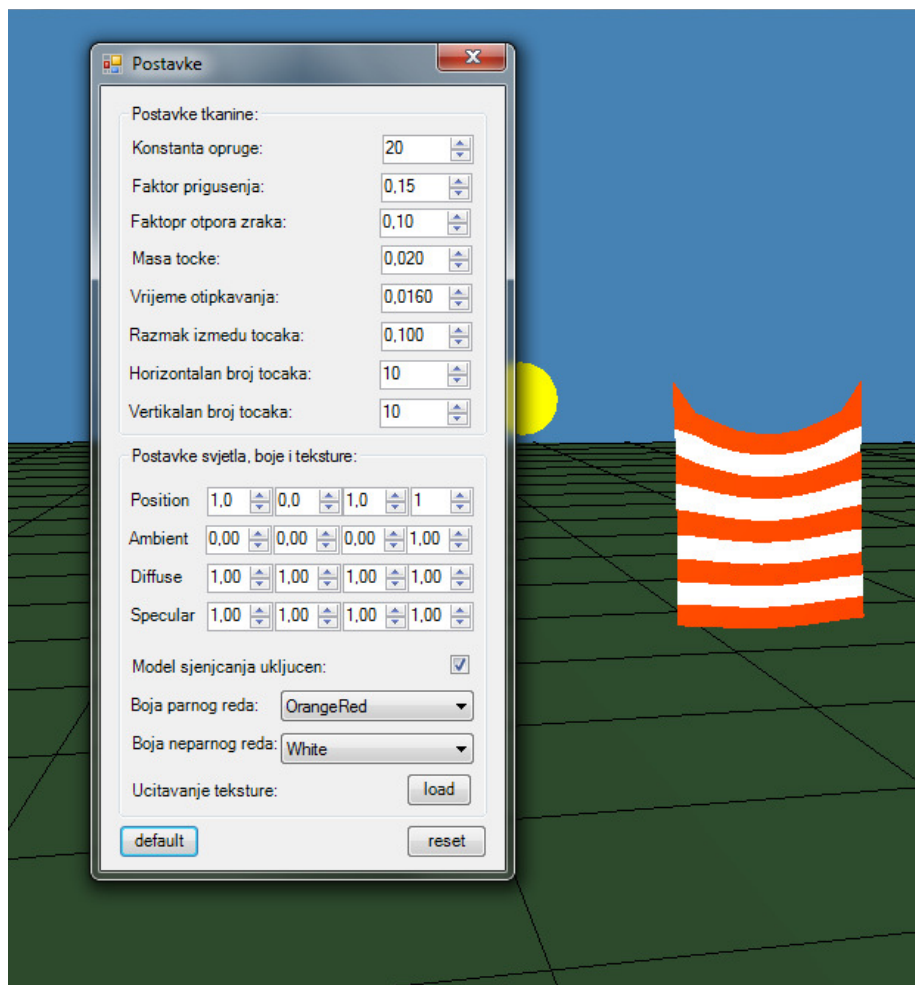
Za kretanje kroz prostor koriste se slova: W (naprijed), S (unazad), A (bočno lijevo), D (bočno desno), SPACE (povećavanje trenutne visine) i C (smanjivanje trenutne visine). Rotacija pogleda obavlja se mišem. Slovom 'F' mijenja se način djelovanja na tkaninu između dva tipa vanjskih sila, vjetra ili udarca. Jačinu te sile mijenja se vrtnjom kotačića na mišu. Držanjem tipke 'CTRL' može se pomicati kursor miša te tako i mijenjati smjer djelovanja sile. Za uključivanje/isključivanje fiksne pozicije djelovanja sile pritišće se tipka 'ENTER'. Pritiskom lijeve tipke na mišu djeluje se silom na tkaninu u smjeru projekcije, a desnom se taj smjer okreće tako da sila djeluje u suprotnom smjeru. Konačno za povećanje ili smanjenje prostora djelovanja sile koriste se tipke +/-.



*Slika 6.2 Prikaz modela tkanine, sunca, tla i neba.*

## 6.2 Utjecaj promjene parametara

Rad i simulacija tkanine obavlja se u realnom vremenu. Promjenom parametara prikazanim slikom 6.3 automatski djelujemo na model tkaninu kojoj se dinamički mijenjaju unutarnje strukture fizike ili način prikaza promjenom osvjetljenja, boje ili teksture tkanine.



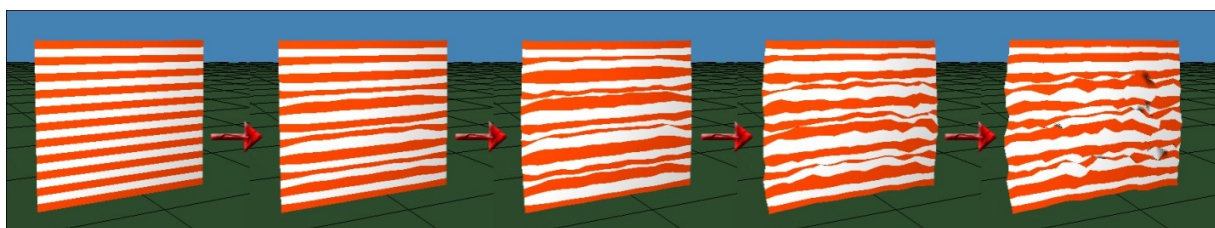
Slika 6.3 Prikaz prozora za postavke tkanine te njegovih inicijalnih vrijednosti parametara.

Inicijalne vrijednosti atributa tkanine namještene su za stabilan i realan prikaz rada simulacije tkanine. Omogućena je promjena parametara fizike unutar modela kao što su konstanta opruge, faktor prigušenja opruge, faktor otpora zraka i masu točke. Strukturne dijelove tkanine promjenom atributa broja vertikalnih i horizontalnih točaka te njihovog međusobnog razmaka. Te vrijeme otipkavanja koji je ključan element unutar numeričke

integracije. Njegovim povećanjem unosimo veliku pogrešku unutar samog izračuna fizike te tako može doći do eksplozije modela tkanine. Smanjivanjem iznosa utječemo na izračun fizike kojemu je vrijednost zanemarivog doprinosa što vodi do zaustavljanja simulacije u vremenu.

Moguće je mijenjati boju parnog i neparnog retka tkanine, ili učitati teksturu tkanini preko definirane slike. Model sjenčanja se također može mijenjati između konstantnog i Gouraudovog sjenčanja. Svjetlost se također može mijenjati po njegovim atributima od ambijentalne, difuzne i zrcalne komponente te same pozicije svjetla (koje utječe na pomak sunca unutar scene).

Svi parametri su ograničeni kao ne bi došlo do grešaka tijekom izvođenja simulacije ali je ostavljena mogućnost promjene parametara van sigurnosnih mjera kako bi se mogla uočiti nestabilnost i krhkost fizike te utjecati na njezinu pogrešnu interpretaciju. Kako bi nastavili ili pokrenuli novu simulaciju koriste se tipke *reset* i *default*. Gumb *reset* se koristi za ponovno pokretanje simulacije modela tkanine s parametrima koji su definirani unutar trenutnog prozora postavki. *Default* gumb se koristi za vraćanje inicijalnih postavki tkanine. Primjer simulacije nestabilnog modela na slici 6.4 kojeg dobivamo promjenom parametra u nepovoljne uvjete.



*Slika 6.4 Prikaz raspada modela tkanine, sljedeća slika je prazan pogled jer je brzina raspada eksponencijalna.*

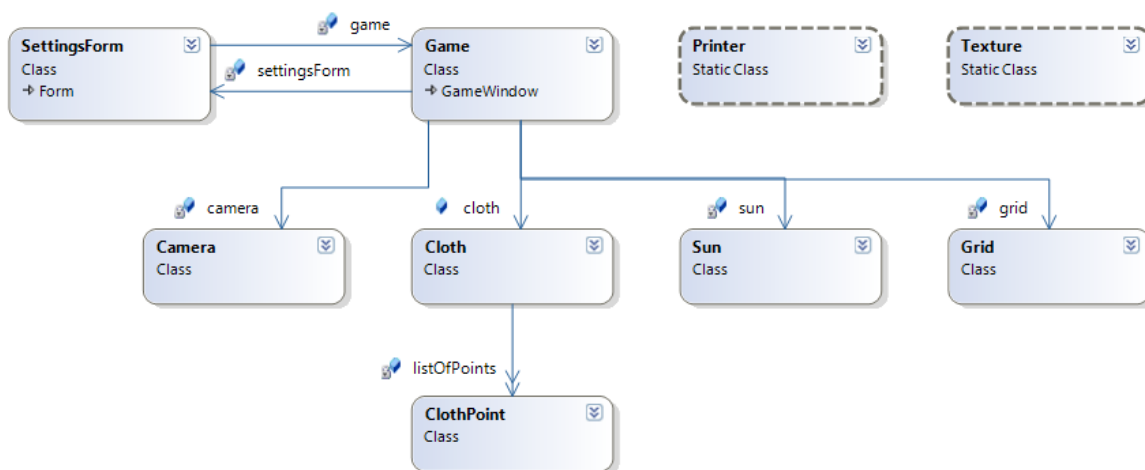
### **6.3 Pregled programske podrške**

Unutar programa definirani su dva prozora u jednom je moguće mijenjati postavke, a u drugom se vrši sama simulacija modela tkanine. Prozori su pokrenuti u različitim dretvama

kako jedno drugome ne bi narušavali efikasnost izvođenja. Između tih dviju dretvi vrši se međusobna komunikacija kako bi se postiglo dinamičko mijenjanje parametara simulacije. Korištenjem OpenTK klase `game` omogućava se preko metode `Run(60.0, 60.0)` definiranje frekvencije osvježavanja prikaza i frekvencije prihvatanja ulaznih signala (tipkovnica, miš).

Promjenom parametra unutar prozora s postavkama automatski djelujemo na stvoreni objekt tkanine. Sve postavke vrše se dinamički unutar objekta tkanine osim promjena broja točaka na tkanini. Promjenom broja točaka tkanine objekt tkanine se uništava te se kreira nova tkanina s postavkama koje su prethodno definirane zajedno s promijenjenim brojem točaka koje želimo da tkanina posjeduje.

Tkanina je odvojena od ostatka programa zapakirana u zasebnu klasu sa stvorenim javnim sučeljima za kreiranje, promjenu parametara te njeno iscrtavanje. Takav model tkanine je spreman za implementaciju i korištenje u drugim projektima. Na slici 6.5 predstavljen je dijagram klasa i njihovih međusobnih odnosa unutar izgrađenog programa za simuliranja modela tkanine.



Slika 6.5 Predstavlja dijagrama klasa unutar programa za simulaciju tkanine.

## 7 Zaključak

Simuliranje dinamičkih pojava iz stvarnog svijeta unutar računalnog sustava otvara nove horizonte računalne grafike i animacije. Računalom se možemo u realnom vremenu prikazati podosta kompleksan matematički, odnosno fizikalni model. Pri kreiranju ove simulacije osjetio sam značajnost ispravno preslikanog fizikalnog modela te moć njegovog djelovanja. Programiran i grafički reprezentiran model dinamičkog procesa počinje se ponašati kao stvarno tijelo koji splotom fizikalnih definicija stvara efekte realnog svijeta.

Pristupom aproksimacije i otipkavanja realnog svijeta možemo dobiti gotovo identične događaje kao što bi one bile u stvarnom svijetu. Mogućnost ovakvog zavaravanja svodi se na aproksimaciju kojoj pripomaže tromost ljudskog oka i ljudska percepcija. Oni se mogu iskoristiti za prikaz određene simulacije jer ih je lako zavarati. Ovom brzinom razvoja računalne grafike pronaći ćemo se unutar računalne simulacije, a da pri tome uopće ne primijetimo razliku između nje i realnog svijeta koji nas okružuje.



## 8 Literatura

- [1] Željka Mihajlović, ZEMRIS, FER,  
[http://www.zemris.fer.hr/predmeti/ra/predavanja/5\\_dinamika.pdf](http://www.zemris.fer.hr/predmeti/ra/predavanja/5_dinamika.pdf), lipanj 2009.,
- [2] The Open Toolkit library 0.9.8, <http://www.opentk.com/>, lipanj 2009.,
- [3] OpenGL Tutorials, <http://jerome.jouvie.free.fr/OpenGL/Tutorials1-5.php>, lipanj 2009.,
- [4] Verlet integration, [http://en.wikipedia.org/wiki/Verlet\\_integration](http://en.wikipedia.org/wiki/Verlet_integration), lipanj 2009
- [5] OpenGL Picking Made Easy,  
<http://web.engr.oregonstate.edu/~mjb/cs553/Handouts/Picking/picking.pdf>, lipanj 2009.

## 9 Sažetak

### Postupci simulacije i prikaza tkanine

Ovaj rad opisuje način simulacije i prikaza modela tkanine. Drugo poglavlje opisuje fizikalni model tkanine definiranog preko sustava čestica povezanih oprugama. Izrađene su dvije vrste numeričke integracije te opisana njihova funkcionalnost i učinkovitost. Također su opisane metode potrebne za stabilizaciju simulacije. U trećem poglavlju prikazana je računalna interpretacija modela tkanine, njezinih atributa i metoda.

Četvrto poglavlje bavi se prolaskom kroz korake koji su potrebni za prikaz simulacije modela tkanine. Za prikaz potrebno je proći kroz izračun fizike, definiranje sustava svjetla, određivanje normala točaka tkanine te spajanje boje ili teksture poligonima modela tkanine. Peto poglavlje je posvećeno primjeni vanjskih sila na model tkanine. Implementirana su dva tipa sila: impuls udarca i sila vjetra. Šesto poglavlje opisuje sustav scene, utjecaj promjene parametara tkanine i načine interakcije korisnika sa simulacijom modela tkanine.

Ključne riječi: simulacija modela tkanine, sustav čestica povezanih oprugama, metoda selektiranja, verlet integracija

### Abstract

#### Procedures for cloth simulation and rendering

This paper describes methods for simulation and rendering cloth. Second chapter describes physical model of cloth defined through particle string system. There are two methods of numerical integration represented in this paper. Their functionality and efficiency are also presented with methods for making simulation stable. Third chapter shows computer interpretation of cloth model his attributes and methods.

Fourth chapter deals with passing through steps needed for rendering cloth simulation. For that purpose we need to calculate physics, define light system, calculate and determine normal vectors of cloth points and finally coloring and texturing cloth model. Fifth chapter is dedicated for applying outer force influence in cloth simulation. There are two type of force implemented in this simulation: impulse impact and wind force. Sixth chapter describes scene, effects generated by changing cloth parameters and interaction between user and cloth simulation.

Key words: cloth simulation, particle spring system, picking method, verlet integration

## 10 Tablica slika

Slika 2.1 Reprezentacija teksture diskretizacijom točaka.....	5
Slika 2.2 Prikaz sustava čestica povezanih oprugama.....	6
Slika 2.3 Reprezentacija čestice tkanine.....	7
Slika 2.4 Prikaz djelovanja sila na česticu.....	7
Slika 3.1 Prikaz modela tkanine.....	12
Slika 3.2 Klasa ClothPoint njezini atributi, svojstva i metode.....	13
Slika 4.1 Prikaz objekta sunca, usmjerenog i radijalnog djelovanja te parametara koji ga opisuju.....	16
Slika 4.2 Predstavlja reprezentaciju tkanine raspodijeljenu na $n*m$ broja poligona.....	17
Slika 4.3 Preko vektorskog produkta između dvaju rubova poligona dobivamo normalu $n$ -tog poligona.....	18
Slika 4.4 a) Konstantno sjenčanje primijenjeno na poligone tkanine, b) Sjenčanje vrhova.....	18
Slika 4.5 Prikaz teksture, bilinearna interpolacija.....	19
Slika 4.6 Prikaz preslikane teksture na točke tkanine.....	20
Slika 4.7 Prikaz redoslijeda slanja vrhova prilikom korištenja „QUAD_STRIP“ načina iscertavanja primitiva.....	20
5.1 Prikaz djelovanje vanjske sile na tkaninu primjenom metode selektiranja, na slici se vidi pravokutnik koji dohvaća indekse točaka tkanine.....	23
5.2 Promatranje djelovanja sile na tkaninu sa različitih pozicija pozicioniranjem kamera po sceni.....	24
Slika 6.1 Prikaz modela tkanine prilikom detaljnog prikaza simulacije.....	26
Slika 6.2 Prikaz modela tkanine, sunca, tla i neba.....	27
Slika 6.3 Prikaz prozora za postavke tkanine te njegovih inicijalnih vrijednosti parametara.....	28
Slika 6.4 Prikaz raspada modela tkanine, sljedeća slika je prazan pogled jer je brzina raspada eksponencijalna.....	29
Slika 6.5 Predstavlja dijagrama klasa unutar programa za simulaciju tkanine.....	30

