

# RAČUNALNA GRAFIKA

## dokumentacija laboratorijskih vježbi

### 1. Uvod

U ovom dokumentu opisana su programska rješenja izrađena u sklopu laboratorijskih vježbi iz računalne grafike. Prva vježba obrađuje problem učitavanja, prikaz i praćenje krivulje u 3D prostoru. Druga vježba se sastoji od implementacije sustava čestica. Tema treće vježbe je slobodni izbor i u ovom slučaju radi se o simulaciji fizičkog ponašanje tkanine. Program iz svake vježbe koristi funkcionalnosti implementirane u vježbi prije nje.

### 2. Programska implementacija

#### 2.1. Korišteni alati

Tijekom izrade programa korišteni su brojni alati. Za pisanje i izmjenu programskog koda korišten je editor Sublime Text 2. Za prevođenje tog koda korišten je g++ koji je sadržan u paketu MinGW. Za izradu i izmjene 3D modela korišten je Blender, a za izradu tekstura Gimp. Ova dokumentacija pisana je u alatu LibreOffice Writer.

#### 2.2. Korištene biblioteke

Za otvaranje prozora s OpenGL kontekstom korišten je GLFW. Ta ista biblioteka se koristi i za registriranje korisničkom inputa tipkovnicom i mišem. Za dohvaćanje OpenGL ekstenzija korišten je gl3w API. Za matematičke operacije s matricama i vektorima koristi se linmath.h biblioteka. GLFW je licencirana pod zlib/libpng licencom. Biblioteke gl3w i linmath.h su u javnoj domeni.

#### 2.3. Učitavanje modela

Modeli su pohranjeni u *.obj* formatu, te se učitavaju pri pokretanju programa. Učitavanje se svodi na čitanje redaka datoteke i zapisivanje pozicija vrhova, vrijednosti njihovih uv-koordinata i normala. Zatim se od tih vrhova konstruiraju trokuti. Radi sažetijeg zapisa u datoteci vrhovi su indeksirani, no nakon učitavanja ti se podaci deindeksiraju radi lakšeg rukovanja. Podaci se pohranjuju u OpenGL spremnike sadržane u Model objektima.

## 2.4. Učitavanje tekstura

U drugoj i trećoj vježbi implementirano je teksturiranje modela. Teksture su pohranjene u *.tga* formatu i učitavaju se parsiranjem zaglavlja datoteke i čitanjem piksela. Radi prikaza polu-prozirnih tekstura bilo je potrebno podržati 32-bitni zapis koji sadrži alfa kanal. Funkcija za učitavanje podržava i RLE kompresiju.

## 3. Praćenje putanje

### 3.1. Opis programa

Zadatak prve vježbe je aproksimirati uniformnu B-splajn kubnu prostornu krivulju. Ta krivulja se definira u tekstualnoj datoteci. Definijska datoteka se sastoji od skupine kontrolnih točaka. Klasa krivulje sadrži polje kontrolnih točaka i omogućava aproksimaciju pozicije i tangente na krivulji za proizvoljni parametar  $t$ . Krivulja se iscrtava kao niz linija koje povezuju aproksimirane točke. Osim krivulje u sceni se nalaze i dva modela učitanih iz *.obj* datoteka. Jedan model je ravna podloga, a drugi se kreće, prateći krivulju. Za iscrtavanje se koriste jednostavni programi za sjenčanje. Kao dodatne mogućnosti programa dodana je kamera koja prati model koji se kreće niz krivulju. Moguće je iscrtati tračnice koje se proceduralno generiraju ovisno o obliku krivulje.

Pri kretanju modela po krivulji može se uočiti problem zastoja žiroskopskih osi (eng. gimbal lock).

### 3.2. Upute za korištenje

Aplikacija se pokreće pomoću izvršne datoteke. Kamera se može kretati tipkama W, A, S, D i zakretati pomicanjem miša. Tipkom C kamera se postavlja iza modela koji se kreće po krivulji, a tipkom X se vraća u slobodno kretanje. Izgled krivulje se mijenja tipkama R i F.

Također je moguće promijeniti putanju krivulje mijenjanjem koordinata u ulaznoj datoteci *path.txt*.

## 4. Sustav čestica

### 4.1. Opis programa

U drugoj vježbi obrađuje se tema sustava čestica. Čestice su zapravo dva poligona koja čine kvadrat i imaju teksturu koja im daje boju i oblik. U ovom programu sustavi takvih čestica iskorišteni su za prikaz pahulja snijega, plamena vatre i oblaka dima u sceni. Svaka čestica je instanca klase koja sadrži informacije o položaju, brzini i izgledu. Te informacije se mijenjaju kroz vrijeme. Čestice imaju svoje vrijeme života i nakon što to vrijeme

istekne one se vraćaju na početnu poziciju i osvježavaju se varijable. Tim pristupom se recikliraju istekle čestice i sprječava se prečesto stvaranje i brisanje čestica. Kvadrati koji služe za prikaz čestica se zakreću tako da normala njihove površine pokazuje prema očistu kamere. Na čestice utječe vjetar čiji smjer i količinu je moguće mijenjati tijekom izvođenja programa. Kako vjetar ne bi "otpuhao" većinu pahulja izvan prostora scene izvor čestica je pomaknut u suprotnom smjeru od puhanja vjetra. Također, pahulje se zaustavljaju pri sudaru s podlogom.

## 4.2. Upute za korištenje

Način pokretanja kamere ne razlikuje se od prve vježbe. Osim tipki za pomak omogućene su tipke Q i E za reguliranje vjetra.

## 5. Simulacija tkanine

### 5.1. Opis programa

Treća vježba je nadogradnja na drugu vježbu. Iako koristi mnoge funkcionalnosti vježbe iz sustava čestica, pristupa se rješenju na nešto drugačiji način. U ovom programu čestice se računaju pomoću Verlet integracije. Verlet integracija je vrlo pogodna za rad s česticama jer ne zahtijeva spremanje vektora brzine za svaku česticu. Tema vježbe je simulacija tkanine, a to se svodi na simulaciju mreže povezanih čestica. Za simulaciju kretanja čestica koriste se dvije formule. Jedna formula je temeljni zakon gibanja (drugi Newtonov zakon).

$$\vec{a} = \frac{\vec{F}}{m}$$

Za svaki vremenski pomak zbroje se sve sile koje djeluju na pojedinu česticu i ta suma se podijeli s njezinom masom. Tako dobivenu akceleraciju zatim koristimo za izračun nove pozicije pomoću formule za Verlet integraciju.

$$\vec{x}_{n+1} = \vec{x}_n + (\vec{x}_n - \vec{x}_{n-1}) + \vec{a} dt^2$$

Nakon toga akceleracija čestice se resetira na nulu jer se primjenom te formule pretvorila u pomak, a samim time i implicitno u brzinu. S navedenom formulom čestica bi se, nakon što na nju utječe impulsna sila, nastavila jednoliko kretati. Kako bi simulirali mali gubitak brzine zbog npr. otpora zraka dodajemo faktor prigušenja  $b$ .

$$\vec{x}_{n+1} = \vec{x}_n + (\vec{x}_n - \vec{x}_{n-1})(1-b) + \vec{a} dt^2$$

U drugoj vježbi se koristio varijabilni pomak, koji je jednostavan za napraviti programski i sastoji se od množenja pomaka u prostoru s količinom vremena proteklom od prošlog iscrtavanja slike. To nije uvijek najbolje rješenje. Verlet integracija radi najbolje s konstantnim vremenskim pomakom zbog toga je bilo potrebno odvojiti vremenski pomak fizičke simulacije od vremena iscrtavanja. To je postignuto akumuliranjem vremenskog pomaka svaki puta kada bi se slika iscrtala i računanja fizičkog pomaka tek nakon što je proteklo odgovarajuće vrijeme, a ne pri svakom iscrtavanju. Tako se dobivaju precizniji rezultati integracije, a brzina simulacije više ne ovisi o brzini iscrtavanja. Više detalja dostupno je na poveznici [5] na kraju ove dokumentacije.

Osim preciznijeg izračuna pozicije čestica unaprijeden je i način iscrtavanja. Dok su se u drugoj vježbi čestice iscrtavale jedna po jedna, i to je zahtijevalo više od 5000 poziva funkcija za crtanje po okviru, u ovoj vježbi je to napravljeno u samo jednom pozivu funkcije po sustavu čestica. To se postiže prijenosom podataka o položaju svih čestica u program za sjenčanje odjednom i korištenjem tehnike *Hardware Instancing*. Kao rezultat je moguće nacrtati puno više čestica odjednom.

Za simulaciju tkanine potrebno je još definirati i ponašanje veza između čestica u mreži. Ta veza nastoji držati udaljenost između dvije čestice konstantnom. Kada se dvije povezane čestice previše približe ili udalje potrebno ih je pomaknuti za korekcijski vektor. To se ponovi nekoliko puta.

$$\vec{c} = \frac{1}{2} \vec{v} \left( 1 - \frac{d}{\|\vec{v}\|} \right)$$

$\vec{c}$  – korekcijski vektor  
 $\vec{v}$  – vektor između dvije čestice  
 $d$  – udaljenost pri mirovanju  
 $\|\vec{v}\|$  – trenutna udaljenost

Ponašanje simulirane tkanine ovisi o načinu na koji njezine čestice povežemo u mrežu. Dovoljno je povezati susjedne čestice po osima paralelnima s rubovima tkanine. Da bi se postiglo realnije ponašanje i spriječilo smicanje čestice se povezuju sa susjednima i dijagonalno. Mogu se povezati, ne samo susjedne, nego i čestice nakon susjednih. Time se dobiva ponašanje nalik gumi jer se mreža više opire savijanju.

Za prikaz tkanine potrebno je izraditi poligone koji povezuju čestice tkanine. Njihovom konstrukcijom određuje se i normala. Normala trokuta nam služi da bi mogli simulirati ponašanje tkanine na vjetru. Vjetar za razliku od gravitacije ne djeluje jednako na svaku česticu, već djeluje jače na one čestice čiji je trokut okomitiji na smjer vjetra.

Kolizija tkanine s okolinom postiže se na sličan način kao i veze između čestica same tkanine. Kada se neka čestica nalazi unutar kolizijske sfere ili izvan područja scene potrebno ju je vratiti za određeni korekcijski vektor. Pri kolizijom s podlogom briše se bilo kakva brzina koju čestica ima, kako tkanina ne bi klizila po podlozi. Trenutna simulacija ne sprječava da tkanina prolazi kroz samu sebe, pa je moguće pomalo neobično ponašanje.

## 5.2. Upute za korištenje

Pomicanje kamere se postiže kao i u prošle dvije vježbe. Tipke Q i E služe za reguliranje količine vjetra. Tipka X u potpunosti uklanja vjetar. Iz kuta scene pritiskom lijeve tipke miša moguće je podići kuglu kojom se može utjecati na tkaninu. Učvršćeni vrhovi tkanine mogu se otpustiti razmaknicom, te se tada jedan od vrhova može podići lijevom tipkom miša.

## 6. Poveznice

- [1] [http://www.zemris.fer.hr/predmeti/ra/laboratorijske\\_vjezbe.html](http://www.zemris.fer.hr/predmeti/ra/laboratorijske_vjezbe.html)
- [2] <http://www.glfw.org/>
- [3] <http://github.com/skaslev/gl3w>
- [4] <http://github.com/datenwolf/linmath.h>
- [5] <http://gafferongames.com/game-physics/fix-your-timestep/>