Fakultet elektrotehnike i računarstva

Unska 3, Zagreb

Dokumentacija laboratorijskih vježbi

Računalna grafika

Marin Maršić

Zagreb, listopad 2016.

SADRŽAJ

1. Prva laboratorijska vježba

1.1. Opis zadatka

Zadatak ove vježbe bio je omogućiti praćenje gibanja objekta po putanji. Putanja je definirana aproksimacijskom B-spline krivuljom trećeg stupnja. Objekt se treba orijentirati u smjeru tangente na krivulju u zadanoj točki.

1.2. Opis implementacije rješenja

Na samom početku programa obavlja se učitavanje svih potrebnih podataka. Učitavaju se kontrolne točke poligona te se nad njima vrše svi potrebni izračuni kao što su računanje točaka krivulje te vrijednosti derivacija u pojedinoj točki. Zatim slijedi učitavanje objekta, njegovo normiranje te izračun tangenti u vrhovima te središtima poligona koje koristimo u modelu usvjetljenja.

Tangente, toče i trokuti (poligoni) definirani su odgovarajućim razredima. Objekti tih razreda sadržani su u modelu. Uz to, u implementaciji se još koristi i biblioteka za rad s vektorima i matricama implementirana na predmetu Interaktivna računalna grafika.

1.3. Pokretanje programa

Za pokretanje programa potrebno je zadati datoteku s definiranim kontrolnim točkama objekta te datoteku s definiranim objektom u .obj formatu. Datoteka s kontrolnim točkama zadana je na način da se u svakom redu nalaze koordinate jedne kontrolne točke međusobno odvojene razmakom (prazninom).

Program podržava dva načina rada: analizu i animaciju. U animaciji se objekt samostalno giba po krivulji dok se pri analizi kontrolira njegovo kretanje uz moguća dodatna iscrtavanja. Definirane su sljedeće opcije koje korisnik može odabrati pritiskom na tipku:

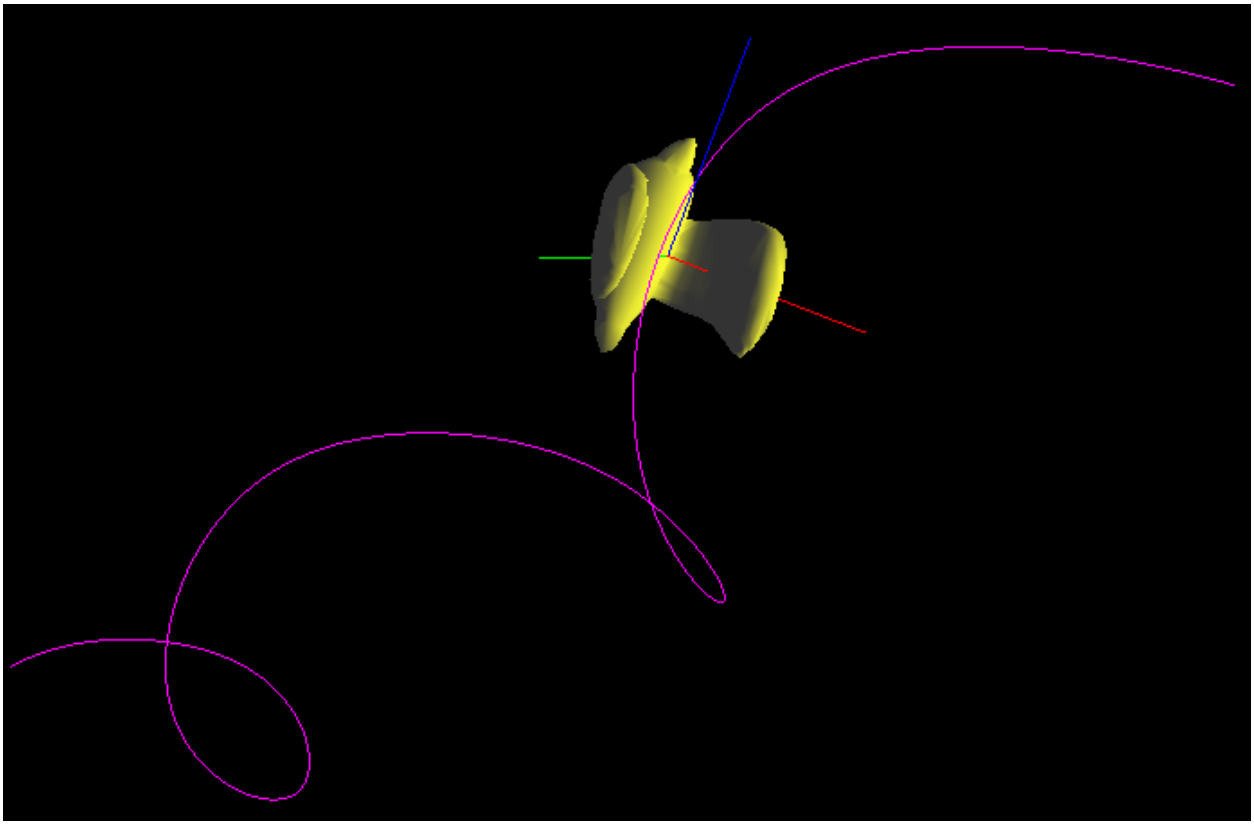
- Pritiskom na tipku 'P' korisnik pokreče jedan ciklus animacije – objekt prolazi jednom cijelom duljinom krivulje.
- Držanjem tipke 'W' objekt se pomiče naprijed po krivulji dok god je ta tipka pritisnuta.
- Držanjem tipke 'S' objekt se pomiče unazad po krivulji dok god je ta tipka pritisnuta.
- Pritiskom na tipku 'K' korisnik pali/gasi iscrtavanje kontrolnog poligona.

- Pritiskom na tipku 'T' korisnik pali/gasi iscrtavanje tangenti poligona u unaprijed definiranim točkama.
- Pritiskom na tipku 'O' korisnik pali/gasi iscrtavanje koordinatnog sustava objekta.

Iscrtavanjem koordinatnog sustava objekta korisnik može pratiti orijentaciju objekta za vrijeme gibanja po putanji i usporediti usmjerenje objekta s smjerom tangente krivulje.

Može se primijetiti da se kretanje objekta izvodi puno sporije ako se ono radi pomoću držanja tipke 'W' u odnosu na slučaj kada je objekt sam animiran. Razlog vjerojatno leži u prekidima izazvanim pritiskom na tipku koji uzrokuju trošenje vremena na njihovu obradu.

Na *slici 1.* možemo vidjeti primjer prikaza koji generira program.



Slika 1: Primjer prikaza programa. Na slici vidimo objekt na jednom dijelu krivulje uz iscrtani koordinatni sustav objekta.

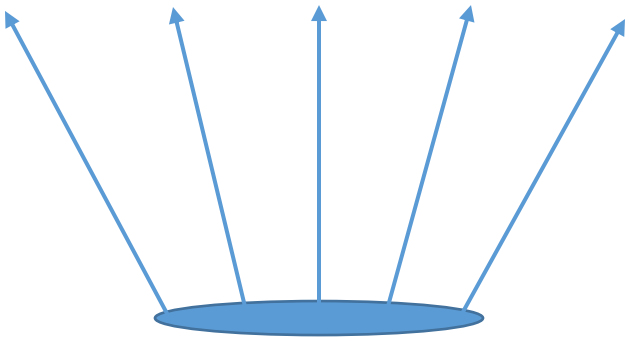
2. Druga laboratorijska vježba

2.1. Opis zadatka

Zadatak ove vježbe bila je implementacija sustava čestica odnosno čestičnog pogona. Čestice su prikazane teksturama, a njihova svojstva definiraju se različitim mogućnostima sustava čestica.

2.2. Opis implementacije rješenja

Sustav je osmišljen tako da prikazuje dim koji izlazi iz tvorničkog dimnjaka. Izvor čestica smješten je na samom vrhu dimnjaka te ima oblik kružnice. Izvor je definiran zasebnim razredom koji definira poziciju (x, y i z koordinate), radijus kružnice unutar koje se čestice nasumično stvaraju, brzinom kojom izbacuje čestice te maksimalnim kutom pod kojim čestice mogu izaći iz dimnjaka. Taj kut otklona omogućuje da čestice bliže centru dimnjaka više idu prema gore, dok se čestice uz rub dimnjaka kreću u stranu te se dobije dojam širenja dima. To možemo vidjeti prikazano na *slici 2.* gdje je prikazano kako određuje smjer gibanja čestice u odnosu na položaj na kojem je stvorena.



Slika 2: Prikaz smjera gibanja čestice u ovisnosti o položaju na kojem je generirana.

Sama čestica također je definirana zasebnim razredom. Osim varijabli koje pamte njezin položaj i brzinu po pojedinim komponentama koordinatnog sustava, čestica sadrži veličinu koja određuje životni vijek čestice, koeficijent prozirnosti teksture, veličinu same čestice te vrijeme kada je čestica stvorena. Životni vijek čestice omogućuje da se ona ukloni iz skupa podataka nakon određenog vremena. Time simuliramo apsorpciju čestice dima u zraku. Kako čestica biva starija, tako ona postaje sve veća i sve prozirnija te tako dobivamo dojam miješanja dima sa

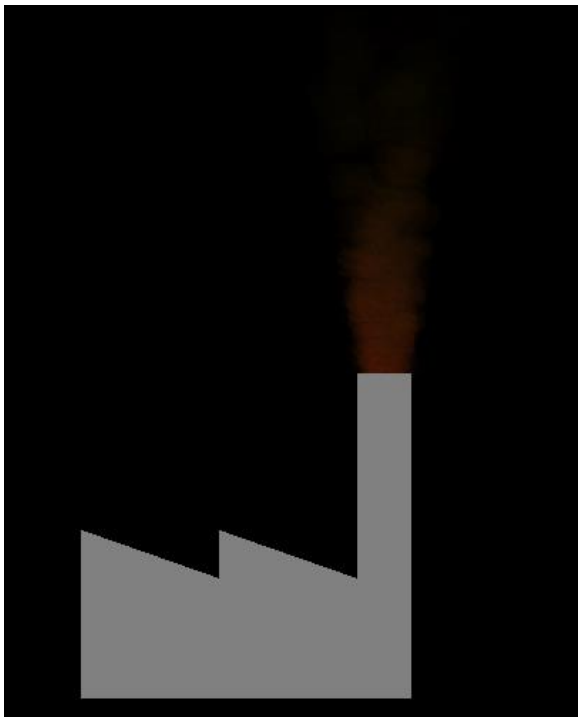
zrakom. Veličina ovisi linearno o duljini života čestice, dok prozirnost ovisi kvadratno. Te odnose sam odabrao isprobavanjem kada sustav izgleda najuvjerljivije.

Model sustava definira učestalost stvaranja čestica (npr. svakih 200 milisekundi) te maksimalan i minimalan broj čestica koje se generiraju u svakom periodu. U svakom periodu generiranja čestica nasumično se odabere broj čestica koje će se generirati iz danog intervala te se generira svaka čestica na nasumično odabranoj poziciji unutar kružnice.

2.3. Izvođenje programa

Pri pokretanju programa nije potrebno definirati nikakve parametre, oni su unaprijed definirani. Animacija se pokreče pritiskom na tipku 'S'.

Primjer izvođenja programa može se vidjeti na *slici 3.* ili na videu dostupnom na adresi https://drive.google.com/open?id=0BwpGIEKGljusYTNLWG1iYm9jM1U u koji je dodana glazba radi boljeg ugođaja.



Slika 3: Prikaz izvođenja programa koji simulira dim.

3. Treća laboratorijska vježba

3.1. Ideja

Ideja ove samostalne vježbe bila je stvaranje računalne 2D igrice koristeći alat MonoGame. MonoGame je open source, cross-platform alat koji omogućuje pokretanje igre na različitim platformama. U igrici igrač skuplja hranu koja pada s neba.

3.2. Opis implementacije

Implementacija funkcionalnosti igre odvija se u Game razredu koji sadrži sljedeće metode:

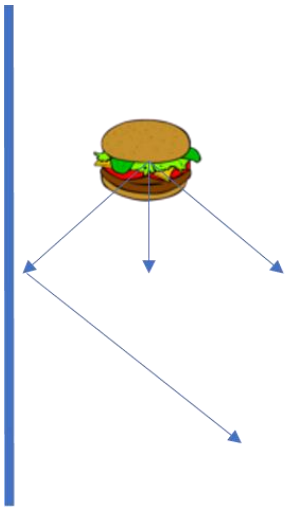
- Initialize() – za inicijalizaciju svih potrebnih stavki programa potrebnih za njegovo daljnje izvođenje.
- LoadContent() – u kojoj se učitavaju svi potrebni sadržaji igrice poput slika, tekstura, glazbe, zvukova…
- Update(GameTime gameTime) – poziva se 60 puta u sekundi (ako ne definiramo drugačije) i služi za obavljanje svih izračuna.
- Draw(GameTime gameTime) – također se poziva 60 puta u sekundi (ako ne definiramo drugačije) i služi za iscrtavanje objekata s parametrima koje smo ranije izračunali.
- UnloadContent() – služi za otpuštanje učitanog sadržaja kako bismo smanjili pritisak na memoriju ukoliko nam sadržaj više nije potreban.

Objekti koji se pojavljuju u igrici su hrana za skupljanje i sam igrač s vrećom za skupljanje. Postoje tri vrste hrane koja se pojavljuje:

- Obična hrana čije skupljanje igraču donosi bodove. Ukoliko igrač promaši ovu vrstu hrane, gubi jedan od života.
- Zdrava hrana koja igraču daje jedan dodatni život ukoliko ju skupi.
- Otrov čije skupljanje uzrokuje gubitak života, ali igraču se dodaje veći broj bodova zbog hrabrosti.

Hrana se generira iznad ekrana igrice u pravilnim razmacima. Smjer kretanja na početku je isključivo u smjeru prema dolje, dok nakon nekog vremena kretanje dobije i horizontalnu komponentu. Takvo gibanje može uzrokovati da se hrana odbije od ruba ekrana što je prikazano na *slici 4.* Čestice se gibaju jednolikom brzinom u tijekom određenih vremenskih perioda. Ta

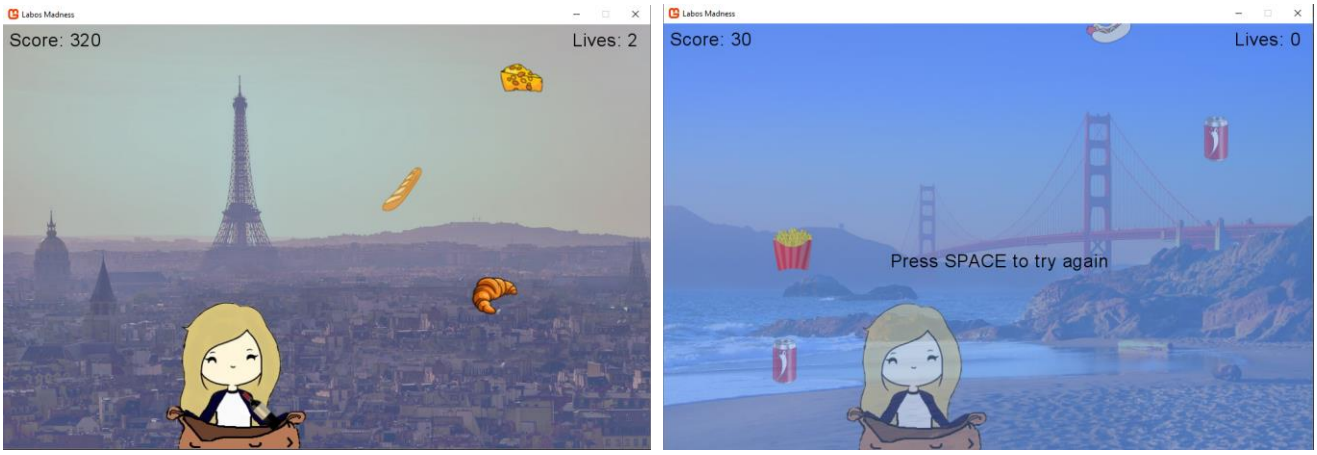
brzina definirana je na razini igrice te se ona periodički ubrzava kako igra odmiče. Detekcija kolizije vrši se jednostavnom bounding-box metodom. Mane ove metode javljaju se pri velikim brzinama kretanja hrane kada se može dogoditi da se zbog diskretnog uzorkovanja ne registrira sudar dvaju objekata. Taj problem može s riješiti na dva načina. Prvi je da igricu dovoljno otežamo kako bi bili sigurni da igrač nikad neće doći do brzina pri kojima se ovaj problem javlja. Drugi način bio bi da odredimo presjecište vektora smjera gibanja jednog objekta s drugim.



Slika 4: Primjer gibanja objekta i refleksije o rub okvira igre

U igrici postoje tri tematske scene koje se izmjenjuju nakon što igrač izgubi sve živote. Svaka scena predstavlja jednu državu. Pozadina prozora predstavlja jedan od gradova iz dane države, a hrana predstavlja karakterističnu hranu za tu kulturu. Tako u igrici imamo San Francisco u kojem igrač skuplja hamburgere, pržene krumpiriće, hot-dogove te gazirano piće u limenci. U Veneciji igrač skuplja espresso, tjesteninu, gelato ili pizzu, dok se u Parizu jedu sir, kroasani, francuski kruh i pije vino.

Igrica ima dva stanja: stanje igre u kojem igrač igra i stanje mirovanja u koje se odvija između dvaju igranja (nakon što igrač izgubi sve živote). Ova dva stanja prikazana su na *slici 5.*



Slika 5: Primjer izgleda igrice.