Elektrotehnički fakultet

Univerziteta u Beogradu

Katedra za računarsku tehniku i informatiku

Diplomski rad

GraphysX: Grafički-orijentisan simulator mehaničkih sistema uz upotrebu biblioteke PhysX

Mentor	Student
dr. Igor Tartalja	Adrian Đura 0460/05

Beograd, 2010.

Apstrakt

U radu se opisuje edukativni softver GraphysX namenjen vizuelizaciji 3D scene čiji elementi međusobno interaguju pod dejstvom gravitacione ili elektrostatičke sile. Scene čine jednostavni mehanički sistemi koji se sastoje od krutih ili mekih tela. Za proračun kretanja i interakcije objekata koristi se biblioteka PhysX kompanije NVIDIA. Softver sadrži integrisan editor scene u kojem korisnik može, putem grafičkog interfejsa, da definiše atribute scene kao i početni položaj i osobine objekata u sceni. Uz softver se isporučuje nekoliko predefinisanih scena koje demonstriraju njegove mogućnosti i mogućnosti PhysX biblioteke. Softver se može koristiti kao edukativni alat na nižem kursu fizike, dok aproksimativna priroda proračuna implementiranih u biblioteci PhysX sprečava njegovu primenu u simulacijama gde je potrebna visoka preciznost.

Ključne reči: grafički simulator, editor, fizika, PhysX.

Sadržaj

1. Uvod

Softverska industrija i velika konkurencija na svetskom tržištu kućnih računara nameću potrebu za bržim razvojem softvera već više decenija. Široko rasprostranjen komercijalno dostupan moderan softver je praktično nezamisliv bez atraktivne grafike. Ovo naročito važi za video-igre, kod kojih je često potrebno uključiti zakone fizike koji su potrebni za simulaciju kretanja i sudaranja objekata u scenama. Razvoj podsistema koji integriše zakone fizike sa jedne strane zahteva solidno poznavanje klasične mehanike od strane programera, a sa druge značajan broj radnih sati, kako u fazi pisanja, tako i u fazi testiranja. Ova konstatacija je dovela do razvoja većeg broja javno dostupnih biblioteka za vršenje potrebnih fizičkih proračuna (sistem za fiziku – eng. *physics engine*). Neke od tih biblioteka su besplatne, a neke imaju podršku i za hardversko ubrzanje proračuna, poput PhysX [1], od kompanije NVIDIA.

Termin sistem za fiziku se odnosi na biblioteke koje omogućavaju simulaciju interakcije sistema pokretnih tela. Oni se primenjuju u raznim oblastima kao što su računarska grafika, video igre (*Mirror's Edge* [3], *Unreal Tournament* [4]) i filmovi (npr. Apolo 13, [5]). Najveća primena je u video igrama gde se od sistema za fiziku zahteva da vrši simulaciju u realnom vremenu, dovoljno brzo za tečno odvijanje igre. Takođe, s obzirom na to da video igre služe za zabavu, toleriše se gubitak preciznosti simulacije. Standardni način pravljenja kompromisa između brzine i preciznosti jeste pojednostavljenje geometrijskih oblika koji predstavljaju modele tako što se vrši formiranje posebne jednostavne mreže (najčešće sastavljene od trouglova) koja se koristi u proračunima, umesto složene mreže koja se koristi za prikazivanje 3D modela. Jednostavnija mreža se zove još i *koliziona geometrija*.

U radu se opisuje razvoj alata GraphysX. U pitanju je simulator mehaničkih sistema koji korisniku omogućava da posmatra međusobnu interakciju objekata (kretanje, sudari, deformacije, itd) za čije računjanje se koristi biblioteka PhysX, kompanije NVIDIA. Namena ovog alata jeste demonstracija i primena mogućnosti biblioteke PhysX-a, ne samo u okviru industrije igara za koju je prvobitno namenjen, već i u edukativne svrhe. Alat ne eksploatiše sve mogućnosti biblioteke PhysX, već samo one najčešće korišćene, potrebne za simulaciju jednostavnih mehaničkih sistema.

Alat je namenjen korisnicima koji žele da steknu znanja o funkcionisanju mehaničkih sistema interaktivnim sastavljanjem sistema, zadavanjem parametara i posmatranjem kretanja objekata na osnovu zadatih parametara. Korisnik može da u virtuelnom 3D prostoru rasporedi različite vrste geometrijskih tela i vidi kako ona međusobno interaguju u zatvorenom mehaničkom sistemu. Korisnik može da menja sve parametre koji utiču na simulaciju uključujući i smer i intenzitet spoljašnje gravitacione i elektrostatičke sile. Postoji nekoliko oblasti primene alata: može se koristiti u edukativne svrhe, a takođe se može koristiti kao pomoćni alat za razvoj aplikacija u kojima je potrebno rešavanje praktičnih problema iz domena klasične mehanike, bilo da se radi o realnim ili fiktivnim sistemima. Treba imati u vidu da je podržani fizički model pojednostavljen, pa se alat ne može primenjivati u situacijama kada je potrebna visoka preciznost.

U okviru simulatora se nalazi integrisan editor scene, koji korisniku omogućava da zadaje početne pozicije i osobine svih objekata. Korisnik može da učita ili sačuva scene koje je kreirao kao i da ih spaja.

U simulatoru je podržano stvaranje čvrstih i mekih objekata koji imaju razne atribute koji utiču na njihovo ponašanje prilikom simulacije. Pri tom, neki atributi su specifični za čvrste odnosno meke objekte. Simulator podržava kretanje objekata pod dejstvom gravitacione i elektostatičke sile. Kretanje pod dejstvom magnetne sile nije podržano, kao ni stvaranje magnetnog polja usled kretanja naelektrisanih tela.

2. Problem

Javno dostupni besplatni simulatori mehaničkih sistema (npr. Working Model [6], Phyz [7]) su uglavnom značajno komplikovani za korišćenje prosečnom korisniku i zahtevaju prethodnu obuku da bi se upoznali ti alati i njihove mogućnosti. Iako ti simulatori obezbeđuju veliku preciznost simulacije, oni ne koriste specijalizovan hardver za vršenje proračuna. Zbog toga njihov učinak pretežno zavisi od brzine centralnog procesora i u određenim situacijama ne mogu da podrže simulaciju u realnom vremenu čak iako su u pitanju jedostavnije scene. Najveći problem upotrebe ovih alata u školama da bi se učenicima interaktivno prikazala oblast koju trenutno izučavaju je njihova složenost, ali i hardverska zahtevnost koju ne zadovoljava većina računara u školama.

Jedan od ciljeva ovog alata je da omogući prosečnim korisnicima da sastave jednostavne scene i simuliraju ih. U cilju približavanja alata što većem broju korisnika, sastavljanje scene se vrši interaktivno, na intuitivan način, što omogućava i korisnicima koji nisu dovoljno familijarni sa radom na računaru da jednostavno naprave scene, kao na primer profesor fizike u osnovnoj ili srednjoj školi za svoje učenike. Takođe i sami učenici mogu sastaviti svoje scene, menjati vrednosti raznih parametara i videti njihov uticaj. Cela aplikacija je projektovana sa idejom da se što jednostavnije koristi i bez prethodnog izučavanja alata. Aplikacija može da koristi hardversko ubrzanje ukoliko je dostupno i tada može raditi na računarima skromnijih performansi.

Pored navedenog, aplikacija rešava problem demonstriranja mogućnosti biblioteke PhysX u scenariju koji korisnik zadaje, što omogućava programerima da isprobaju i unapred vide da li osnovne mogućnosti biblioteke PhysX zadovoljavaju potrebe njihovih aplikacija. Upotreba biblioteke PhysX dovodi do značajnog smanjenja obima posla za razvoj aplikacije.

3. Funkcionalna specifikacija

U alatu GraphysX korisnik ima mogućnost pravljenja, učitavanja, snimanja i simuliranja scena. U integrisanom editoru scene korisnik može dodavati objekte i podešavati njihove atribute. Kreirane scene se mogu simulirati i može se podešavati brzina simulacije. U podešavanjima alata korisnik može podesiti željenu rezoluciju za prikaz, prečice na tastaturi i jezik.

Simulacija

Simulacija se odvija nad scenom koju čine kamera, opciona čvrsta podloga i skup objekata koji međusobno interaguju, saglasno definisanim zakonima. Objekti u sceni su hijerarhijski organizovani i postoje čvrsti i meki. Čvrsti objekti mogu imati jedan čvrsti roditeljski objekat, tako da su pozicija i rotacija objekta relativni u odnosu na roditelja. Meki objekti ne mogu imati roditelja. U toku simulacije, meki objekti se automatski vežu za objekte sa kojima su u koliziji. Od parametara vezanih za scenu moguće je definisati intenzitet i smer vektora koji definišu polja spoljnjih sila (gravitacione i elektrosatičke) i zadati podlogu koja se prostire u celoj horizontalnoj ravni (Z=0).

Grafički interfejs

Glavni režim funkcionisanja alata je režim u kome se dodaju objekti, postavljaju parametri i vrši pokretanje simulacije pomoću dugmića na ekranu (slika 1). Kada se miš nađe iznad nekog dugmeta prikazuje se tekst sa opisom tog dugmeta (eng. *tooltip*). U ovom režimu, korisniku na raspolaganju stoje sledeće opcije u grafičkom interfejsu koje se aktiviraju pritiskom levog tastera miša:

Slika 1: Glavni prozor sa grafičkim interfejsom

1. **Pokretanje simulacije nad tekućom scenom**: Kada se simulacija pokrene, dugme za pokretanje simulacije (scenarija) se pretvara u dugme za pauziranje simulacije. Nakon pokretanja simulacije, nije moguće dodavati nove objekte ili menjati postojeće atribute objekata u sceni, sve dok se simulacija ne pauzira ili ne prekine.

2. **Pauziranje pokrenute simulacije**: Aktivno je samo ako je simulacija pokrenuta. Klikom na dugme, pauzira se simulacija i omogućeno je dodavanje novih objekata u scenu koji se pojavljuju u trenutku u kojem su oni dodati, ali ne i menjanje postojećih.

3. **Resetovanje simulacije**: Scena se vraća u prvobitno stanje u kome je bila neposredno pre pokretanja simulacije. Ukoliko su neki objekti naknadno ubačeni u scenu, oni će se pojaviti u sceni u trenutku *t* u kom su ubačeni.

4. **Meni**: aktivira se pritiskom na dugme grafičkog interfejsa. Pritiskom na dugme pojavljuje se meni sa dodatnim opcijama. Ukoliko je simulacija pokrenuta, biće automatski pauzirana dok god se ne izađe iz menija.

5. **Objekti**: prikazuje paletu sa alatima za izbor objekata za ubacivanje u scenu.

6. **Atributi**: prikazuje paletu za menjanje atributa selektovanih objekata. Ukoliko su selektovani objekti iste vrste, biće dostupi svi atributi za menjanje. Ukoliko su selektovni objekti različiti, pojaviće se samo zajednički atributi.

7. **Scena**: prikazuje paletu za izmenu atributa scene.

8. **Dodatni objekti**: prikazuje paletu sa listom dodatnih objekata koji se pojavljuju u trenutku *t* u sceni.

9. **Inkrementiranje/Dekrementiranje brzine protoka vremena**: Promena brzine kojom simulacija teče. Brzina se menja za 0.5 u odnosu na tekuću brzinu kod vrednosti veće od 1. Ukoliko je brzina simulacije manja od 1 korak za promenu brzine biće 0.1 u odnosu na tekuću brzinu.

Palete se sakrivaju ponovnim klikom na dugme (jezičak) kojim su otvarane, desnim klikom miša bilo gde na ekranu ili pritiskom na taster ESC.

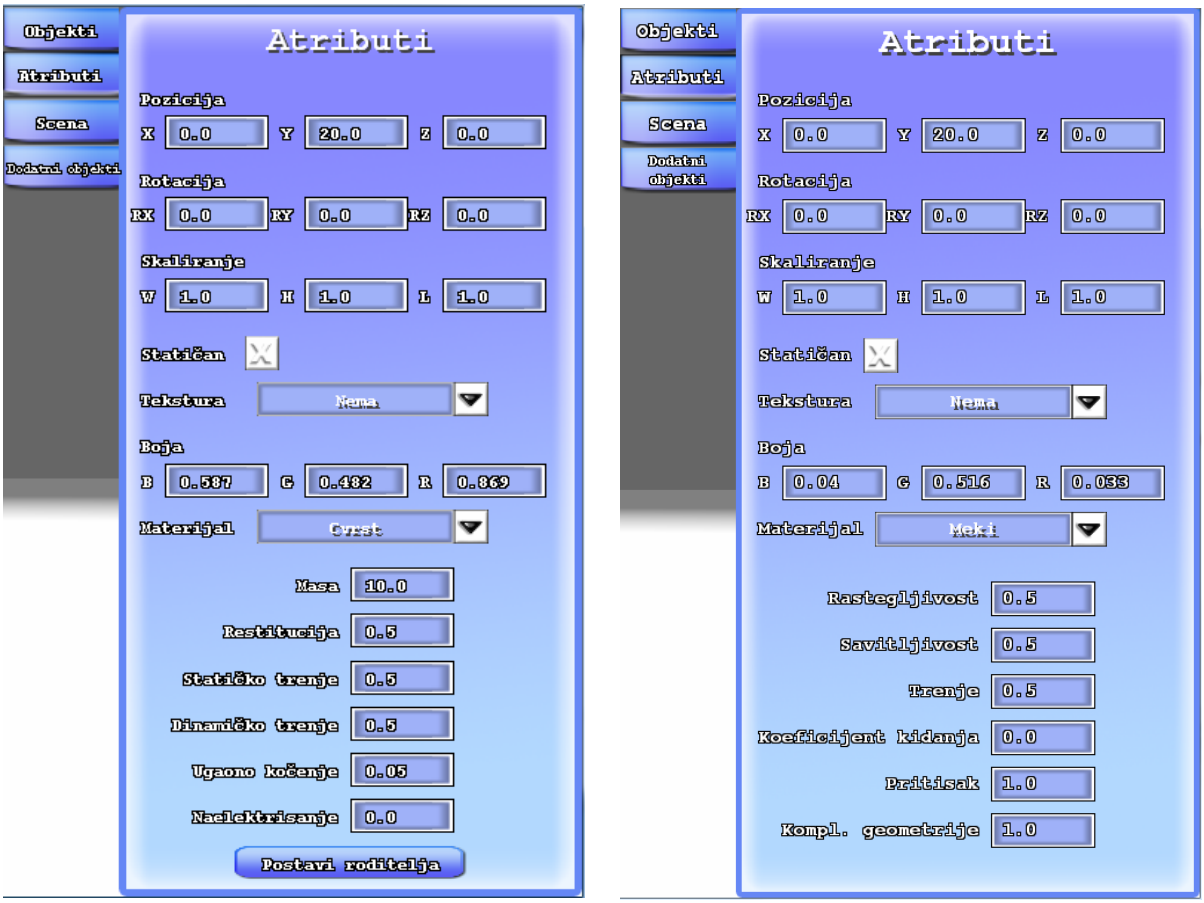
Paleta za dodavanje objekata

Sa palete se mogu izabrati sledeći objekti za ubacivanje u scenu: kocke, tetraedri, prizme i lopte. Objekat se ubacuje u scenu na razumnom rastojanju u pravcu u kome gleda kamera. Ako postoji podloga i ukoliko bi neki deo objekta bio ispod podloge, on se automatski postavlja na podlogu.

Paleta za dodatne objekte

U listi se nalaze objekti koji se naknadno pojavljuju u sceni, u navedenom trenutku *t* nakon pokretanja simulacije. Moguće je izbrisati objekat ili izmeniti njegove atribute. Pritiskom na dugme *Izmeni* aktivira se paleta za menjanje atributa objekta.

Paleta za menjanje atributa objekata



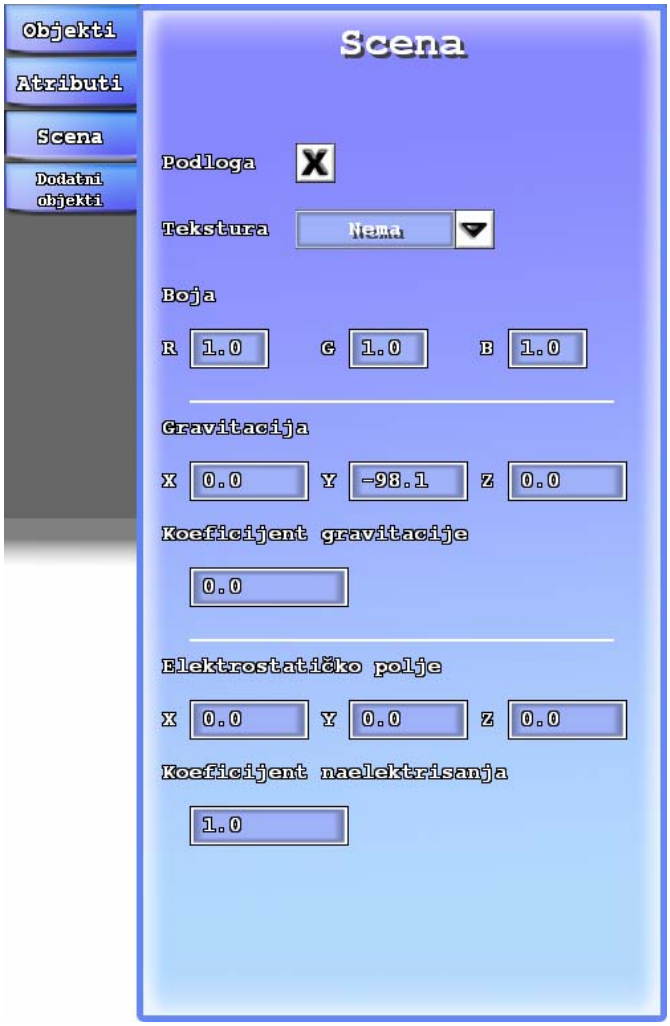
a) *b)*

Slika 2: Paleta za menjanje atributa objekata: a) čvrsti objekti b) meki objekti

Paleta za menjanje atributa objekata je prikazana na a)
b)

Slika 2. U zavisnosti od tipa selektovanih objekata, na paleti će se pojavljivati samo zajednički atributi selektovanih objekata. Postoje 3 različita slučaja kod prikazivanja palete u zavisnosti od objekata koji su selektovani: različiti objekti, samo čvrsti objekti, samo meki objekti. Pritiskom na dugme Postavi roditelja bira se roditelj objekta klikom na neki objekat u sceni. Za promenu teksture i materijala postoji lista iz koje je moguće izabrati željenu teksturu ili materijal. Za promenu da li je objekat statičan ili pokretan označava se odgovarajuće polje. Ostali parametri se menjaju unošenjem odgovarajuće vrednosti u polje pored imena atributa.

Paleta za menjanje atributa scene



Slika 3: Paleta za podešavanje atributa scene

Paleta za podešavanje atributa scene je prikazana na Slika 3. Moguće uključiti prikazivanje podloge i podesiti boju i teksturu podloge. Boja podloge se primenjuje i na teksturu ako je zadata. Može se birati jedna od nekoliko predefinisanih tekstura. Moguće je promeniti intenzitet i smer gravitacione i elektrostatičke sile unošenjem vrednosti komponenti vektora sile (x,y,z). Vektori predstavljaju spoljašnje sile (gravitacionu i elektrostatičku). Ukoliko je vrednost svih komponenti nekog vektora 0, isključuje se uticaj odgovarajuće spoljne sile. Nezavisno od spoljašnjih sila, korisnik može da zada koeficijent gravitacije i naelektrisanja. Koeficijent gravitacije utiče na računanje gravitacione sile kojom se tela u sceni međusobno privlače. Ukoliko je on 0, tela sa masom se međusobno ne privlače gravitacionom silom. Analogno ponašanje važi za spoljnu elektrostatičku silu, odnosno koeficijent naelektirsanja.

Manipulisanje scenom u editoru

Korisnik može da menja sadržaj scene sledećim operacijama:

a. označavanje objekata u tekućoj sceni: klikom na neki objekat u sceni on se označava kao selektovan. Ukoliko je prilikom označavanja objekta pritisnut taster SHIFT, objekat će biti dodat u listu selektovanih objekata.

b. dodavanje objekata u tekuću scenu: iz palete ponuđenih objekata, korisnik bira željenu vrstu objekta i postavlja ga na odgovarajuće mesto u sceni.

c. uklanjanje objekata iz tekuće scene: korisnik pritiskanjem tastera DEL na tastaturi uklanja selektovane objekte iz scene.

d. pomeranje, skaliranje i rotiranje objekata u tekućoj sceni: operacije transformacije objekata se primenjuju nad svim selektovanim objektima. Pritiskom na taster X aktivira se akcija pomeranja, taster C akcija rotiranja i taster V akcija skaliranja. Ponovnim pritiskom na taster akcije koja je aktivna vraća se u mod selektovanja. Kada je aktivna neka od ovih akcija, naglašava se izabrana osa na objektu po kojoj se akcija izvršava. Pritiskom na taster Z (Y na srpskoj tastaturi) menjaju se ose (X, Y, Z) po kojoj se akcija primenjuje. Moguće je još promeniti da li da se akcija izvršava lokalno u odnosu na objekat ili u odnosu na ose sveta klikom na taster B. Akcije se vrše pomeranjem miša u levo i desno. Moguće je i direktno uneti vrednosti pomoću palete za menjanje atributa objekata. Navedeni tasteri (Z, X, C, V i B) se mogu promeniti u dijalogu sa opcijama.

e. Menjanje atributa selektovanih objekata: iz odgovarajuće palete korisnik može da promeni atribute željenog objekta

f. kloniranje objekata: operacija kloniranja se obavlja nad izabranim objektima. Pri tom se kod klonova održava međusobni položaj koji postoji kod originala. Kloniranje se vrši pritiskom na kombinaciju tastera SHIFT+C. Prilikom kloniranja, klonirani objekti se označavaju kao selektovani i mogu se pomerati po želji.

g. Postavljanje roditeljskog objekta selektovanim objektima: kada se iz palete za menjanje atributa izabere opcija *Postavi roditelja*, klikom na objekat u sceni taj objekat postaje roditelj selektovanih objekata. Relacija roditelj-dete služi da bi se u editoru olakšalo manipulisanje objektima, pošto je pozicija i rotacija deteta relativna u odnosu na roditelja. U simulaciji veza roditelj-dete je predstavljana jednom čvrstom vezom između ta dva objekta. Samo čvrsti objekti mogu imati roditelja, dok se meki objekti automatski vežu za objekte sa kojima su u koliziji tokom simulacije. Objekti mogu imati samo jednog roditelja. Ukoliko objekat već ima postavljenog roditelja, novi objekat koji je izabran za roditelja će ga zameniti. Ukoliko se, prilikom zadavanja roditelja, klikne na prazan prostor (tj. ne odabere se drugi objekat), selektovanim objektima se ukida roditelj. Objekat ne može biti sam sebi roditelj, niti roditelj može biti objekat koji je u

grupi selektovanih objekata kojima se pridružuje roditelj.

h. Poništavanje poslednje akcije – pritiskom na dugme CTRL+Z poništava se poslednja akcija. Moguće je ponoviti poništavanje i vratiti više akcija unazad.

i. Prikazivanje prečice na ekranu: pritiskom na taster F1 pojavljuje se spisak prečica i njihov opis. Ponovnim pritiskom na taster F1 ili ESC tekst se sakriva.

Kretanje i kamere

Korisnik može slobodno pomerati virtuelnu kameru scenom, korišćenjem miša i tastature. Tastaturom se kamera translatorno pomera napred, nazad i u stranu (tasteri W,S,A,D), a pomeranjem i držanjem desnog tastera miša se vrši promena orijentacije kamere. Moguće je zapamtiti trenutnu poziciju kamere i kasnije se vratiti na tu poziciju. Moguće je zapamtiti do 9 pozicija kamere. Pamćenje se vrši držanjem tastera CTRL i pritiskom na taster 1 do 9. Skok na zapamćenu poziciju kamere se vrši pritiskom na taster sa brojem koji odgovara željenoj poziciji kamere. Zapamćene pozicije se pamte prilikom čuvanja scene.

Opis objekata

Objekat čini njegova geometrija (tj. geometrijski oblik) i svojstva materijala od kojeg je načinjen. Od geometrijskih oblika, u simulatoru su dostupni:

1. kocke (kvadri, odnosno kutije)
2. tetraedri
3. prizme
4. lopte

Od svojstava materijala koji mogu da se postavljaju objektima, dostupna su:

1. Čvrst materijal
2. Meki materijal

Podešavanjem atributa materijala postiže se oponašanje osobina raznih vrsta materijala kao što su čvrst metal, led ili tkanina. Atributi koji se mogu menjati kod materijala su opisani u sledećem odeljku.

Atributi objekata

Atributi koji su zajednički za sve objekte:

X – Udaljenost od koordinatnog početka po X osi

Y – Udaljenost od koordinatnog početka po Y osi

Z – Udaljenost od koordinatnog početka po Z osi

RX – Ugao rotacije oko X ose

RY – Ugao rotacije oko Y ose

RZ – Ugao rotacije oko Z ose

W – Koeficijent za skaliranje širine objekta. Inicijalna vrednost je 1.

H – Koeficijent za skaliranje visine objekta. Inicijalna vrednost je 1.

L – Koeficijent za skaliranje dužine objekta. Inicijalna vrednost je 1.

Statičan – Označava da li je objekat statičan ili pokretan.

Tekstura – Slika koja se iscrtava na objektu. Inicijalno su objekti bez teksture

Boja materijala – Boja objekta

Materijal – vrsta materijala za objekat (čvrst ili mek)

Atributi za čvrsta tela:

Masa – masa objekta. Masa predstavlja meru inertnosti objekta.

Elastičnost (eng. bounciness) – Elastičnost objekta, koja predstavlja meru u kojoj se objektu vraća uložena kinetička energija prilikom sudara sa drugim objektom. Vrednost je realan broj između 0 i 1. Ukoliko je restitucija 0, objekat neće odskakivati prilikom sudara sa drugim objektima ili podlogom. Ukoliko je restitucija 1, objekat se odbija bez gubitka energije (elastičan sudar).

Statičko trenje – trenje između 2 objekta koja se ne pomeraju. Npr. kutija na strmoj ravni: ako je statičko trenje dovoljno veliko, kutija neće skliznuti.

Dinamičko trenje – trenje između 2 objekta koji se jedan prema drugom relativno pomeraju. Npr. u prethodnom slučaju, ukoliko kutija klizi niz rampu. Dinamičko trenje bi trebalo da bude manje ili jednako statičkom trenju, što simulator ne proverava.

Ugaono kočenje – objekat sa manjim ugaonim kočenjem će se brže rotirati u odnosu na objekat sa većim ugaonim kočenjem ukoliko je ista sila primenjena na oba objekta.

Naelektrisanje – količina naelektrisanja koju poseduje dati objekat. Može biti pozitivna i negativna. Ukoliko je naelektrisanje 0, onda na objekat ne utiče elektrostatičke sile koje

proizvode drugi objekti ili spoljnjo elektrostatičko polje.

Atributi za meka tela:

Rastegljivost – Vrednost je realan broj između 0 i 1. Ukoliko je vrednost blizu 1 dobija se telo koje je veoma rastegljivo, a ukoliko je vrednost blizu 0, telo nije rastegljivo.

Savitljivost – Vrednost je realan broj između 0 i 1. Ukoliko je vrednost blizu 1, telo se lako savija (npr. papir) dok se za vrednosti bliže 0 dobija čvršće telo (npr. karton).

Trenje – određuje trenje sa čvrstim telima. Za male vrednosti ovog koeficijenta, telo će sa lakoćom kliziti niz površinu sa kojom je u kontaktu. Sa porastom ovog koeficijenta, otežava se kretanje tela. Simulator će se za simulaciju trenja u potpunosti oslanjati na PhysX. Tačno tumačenje ovog koeficijenta i načina na koji on utiče kod međusobne interakcije dva objekta nije predmet ovog simulatora.

Koeficijent kidanja – koeficijent koji određuje u kojoj meri je rastegljivo telo otporno na kidanje. Neka je X zadata vrednost ovog koeficijenta. Ukoliko je rastojanje između bilo koje dve susedne tačke geometrijskog tela, kojim je modeliran objekat, X puta veće od nominalne, dolazi do kidanja. Kidanje nastaje između svih temena za koje je ovaj uslov ispunjen. Ukoliko je npr. vrednost koeficijenta 1.5 telo će se pokidati kada se deo objekta rastegli za 50% ili više. Da li će do kidanja doći zavisi i od koeficijenta rastegljivosti objekta i sile koja deluje na njega: ako materijal nije rastegljiv, do kidanja (ili bolje reći pucanja) može doći samo za dovoljno velik intenzitet sile. Vrednost 0 znači da se telo ne može pokidati, dok ostale vrednosti označavaju koliko je rastezanje potrebno da bi došlo do kidanja

Pritisak – Označava relativnu vrednost pritiska (u odnosu na atmosferski) u telu i koliko se telo širi u odnosu na inicijalno stanje. Ukoliko je vrednost manja od 1 telo neće moći da zadrži svoju formu i izgledaće izduvano, a ukoliko je vrednosti veća od 1 dolazi do širenja tela (naduvan balon).

Kompleksnost geometrije - koeficijent za finoću mreže za simuliranja mekog tela. Što je vrednost veća to će simulacija biti preciznija.

4. Postojeći algoritmi i alati za razvoj

4.1. nVidia PhysX

PhysX je skup biblioteka za računanje kretanja i kolizije sistema objekata primenom zakona klasične fizike koje je firma nVidia kupila od firme Ageia u februaru 2008. (koja je i sama kupila PhysX tehnogologiju od firme NovodeX 2004 god.).

Opis

Termin PhysX može se odnositi i na hardverski dodatak koja je razvila firma Ageia da bi ubrzala računjanje fizike u igrama. Video igre koje podržavaju hardversko ubrzanje PhysX-a mogu to postići korišćenjem PhysX PPU[1] ili CUDA GPU, rasterećujući centralni procesor od fizičkih proračuna, čime se znatno ubrzava rad aplikacije ili igre.

Biblioteke su trenutno dostupne za sledeće platforme:

- Windows
- Linux (32-bit)
- Apple Mac OS X
- Nintendo Wii
- Sony PlayStation 3
- Microsoft Xbox 360

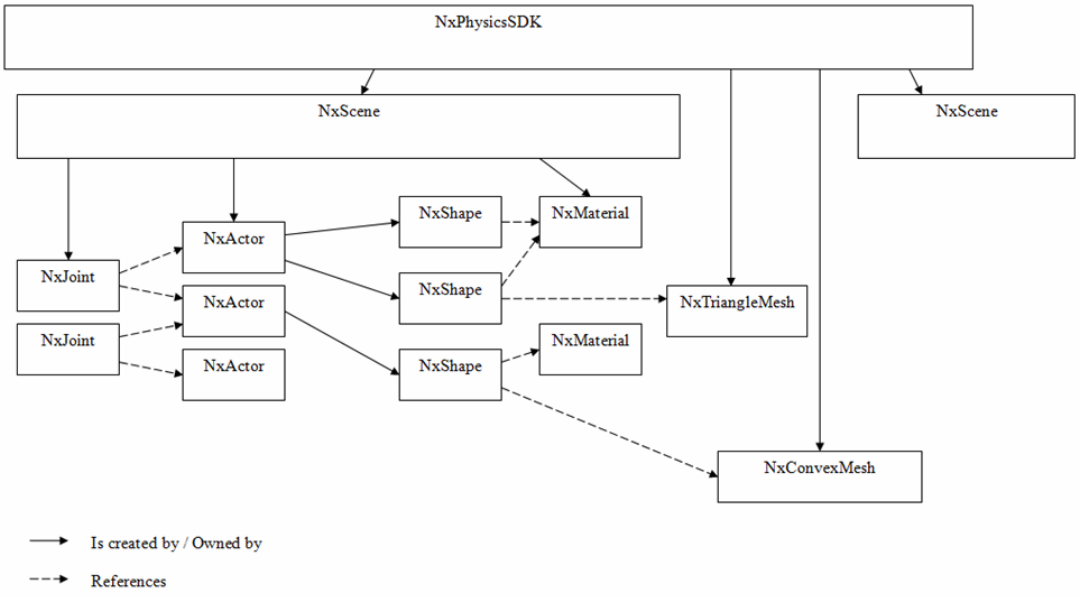
Trenutno najpopularnija okruženja za igre kao što su Unreal Engine 3 [9], Gamebryo [10], Unity [11] koriste PhysX tehnologiju za računanje fizike.

Da bi program koji koristi Nvidia PhysX mogao da se pokrene na računaru mora biti instaliran *PhysX System Software* u kome se nalaze drajveri za PhysX. Za rad programa nije potrebno imati Nvidia grafičku karticu sa podrškom za PhysX ali u tom slučaju će program raditi bez hardverske podrške. Prilikom instalacije novih drajvera za grafičke kartice firme nVidia automatski će se instalirati i ovi drajveri. Trenutno hardversko ubrazanje za PhysX podržavaju samo grafičke kartice nVidia od serije GeForce 8 i novije.

[1] PPU – procesor za računanje fizike je posebno dizajniran da preuzima fizička računanja od procesora. Proizvođači grafičkih kartica su od 2006. godine rešili da implementiraju ovaj procesor u grafičkim karticama

Arhitektura PhysX sistema

U PhysX sistemu može postojati više scena. Svaka scena sadrži objekte tipa NxActor i svaki od njih može da bude sastavljen od više NxShape objekata koja predstavljaju geometriju tela. NxShape objekat može biti prost geometrijski oblik (NxBoxShape, NxSphereShape, NxCapsuleShape) ili kompleksna mreža trouglova (NxTriangleMesh, NxConvexMesh). Materijali (NxMaterial) su jedinstveni za scenu i njih može deliti više NxShape objekata. NxActor objekti mogu biti spojeni sa vezama NxJoint koje imaju parametre koje definišu tip veze.

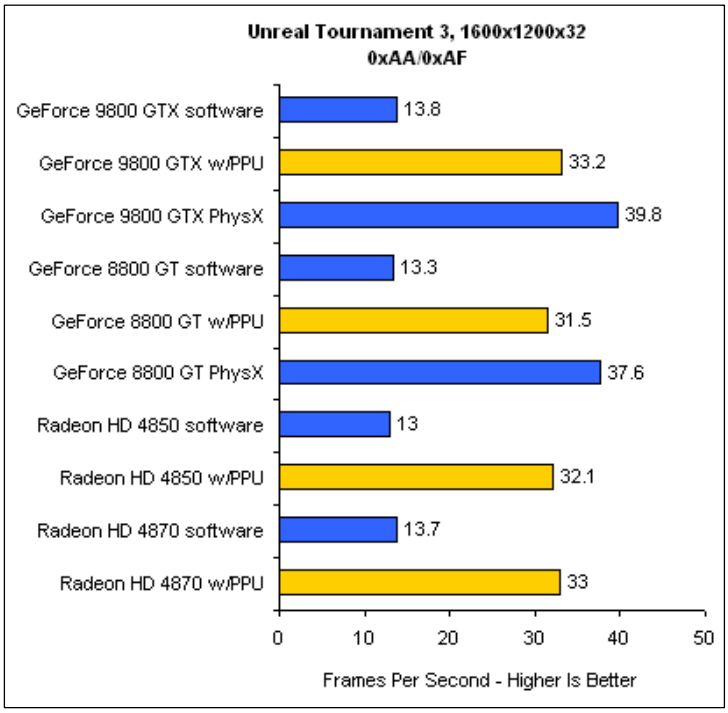


Slika 4: Arhitektura PhysX sistema

U radu se koriste još i NxCloth i ForceField objekti. NxCloth služi za simulaciju mekih tela koja su predstavljena mrežom trouglova. ForceField objekti služe za definisanje polja u kome deluje neka sila. U radu se koristi za simulaciju elektrostatičke i gravitacione sile koja potiče od objekta sadržanih u sceni i simulaciju globalnog elektrostatičkog polja. Ovim objektima može se definisati prostor delovanja i jačina sile. Prostor delovanja može biti sastavljen od više prostih geometrijskih tela.

Performanse

Prilikom korišćenja PhysX-a, na brzinu računanja najviše utiče način na koji se vrše proračuni. Na slici 5 su prikazani rezulati koji su dobijeni testiranjem računanja softverskim putem, pomoću PPU i pomoću GPU.[2]



Slika 5: Rezultati merenja prosečnog broja generisanih slika u sekundi u igri Unreal Tournament 3, za različite grafičke adaptere, sa i bez PhysX hardverskog uređaja (w/PPU – sa PPU karticom, PhysX – sa grafičkom karticom koja podržava harversko ubrzanje PhysX-a).

[2] Rezultati testa i slika su preuzeti sa sledeće adrese: http://www.firingsquad.com/hardware/physx_performance_update/page2.asp

4.2. Ostale biblioteke za računanje fizike

Trenutno ne postoje druge javno dostupne tehnologije za računanje fizike koje podržavaju hardversko ubrzanje. Međutim, postoji veći broj drugih biblioteka za računanje fizike softverskim putem kao što su:

- Havok [12] – najpopularnije okruženje za igre za koji je najavljena hardverska podrška firme ATI. Podržava i simulaciju mekih tela kao i simulaciju uništavanja tela. Većina najpopularnijih igara na tržištu PC računara koristi ovaj sistem za računanje fizike.
- Newton Game Dynamics [13] – besplatni sistem u kojem je bitnija preciznost u odnosu na brzinu računanja. Uglavnom se koristi u nekomercijalnim i akademskim projektima.
- ODE (Open Dynamics Engine) [14] – open source sistem koji podržava simulaciju čvrstih tela i detekciju kolizija. Koristi se u igrama STALKER i World of Goo.
- Bullet [15] – *open source* sistem koji razvija kompanija Sony Computer Entertainment. Koristi se u igri GTA 4 i u još par manje poznatih naslova.

4.3. Alati za razvoj

UltraEngine by Eipix

UltraEngine je grafički 3D sistem razvijen u firmi Eipix iz Novog Sada u čijem razvoju je učestvovao i autor ovog rada. Autorov implementacioni doprinos u razvoju sistema je u oblasti organizacije i prikazivanja scena, integracija sa bibliotekom Nvidia PhysX, upravljanja memorijom i izrade grafičkog interfejsa. UltraEngine je iskorišćen kao osnova za razvoj GraphysX simulatora. Sistem je pisan za Windows platformu i radi sa DirectX grafičkom bibliotekom [16]. Na tržištu trenutno postoje dve video igre koje pokreće ovaj sistem: Pyroblazer i Ziro.

TinyXML

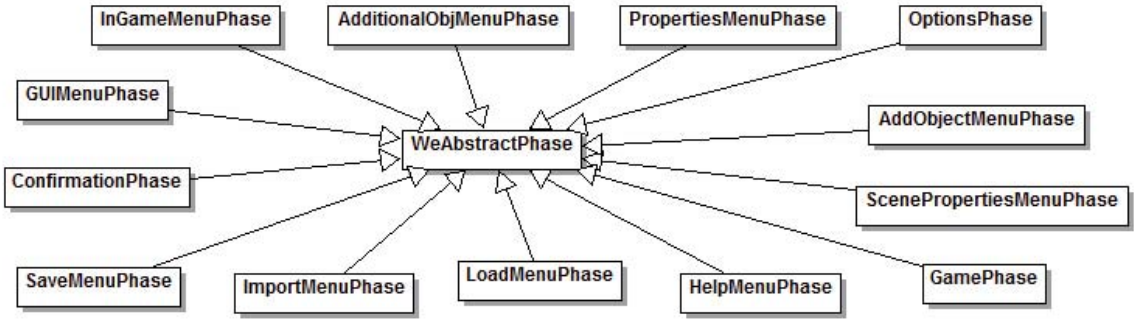
TinyXML [6] je besplatna biblioteka pisana na jeziku C++, koja se statički povezuje (linkuje) sa izvršnim programom i koja omogućava parsiranje XML datoteka. Ona služi da se na jednostavan i brz način čitaju i upisuju podaci u XML datoteke. Ova biblioteka je korišćena za čuvanje i čitanje podataka o scenama.

Microsoft Visual Studio 2008

Microsoft Visual Studio 2008 je okruženje za razvoj konzolnih aplikacija i aplikacija sa grafičkim interfejsom firme Microsoft. Ovo razvojno okruženje omogućava pisanje i prevođenje izvornog koda i korišćenje integrisanog sistema za otkrivanje grešaka. Aplikacija GraphysX je pisana u jeziku C++, korišćenjem ovog razvojnog okruženja.

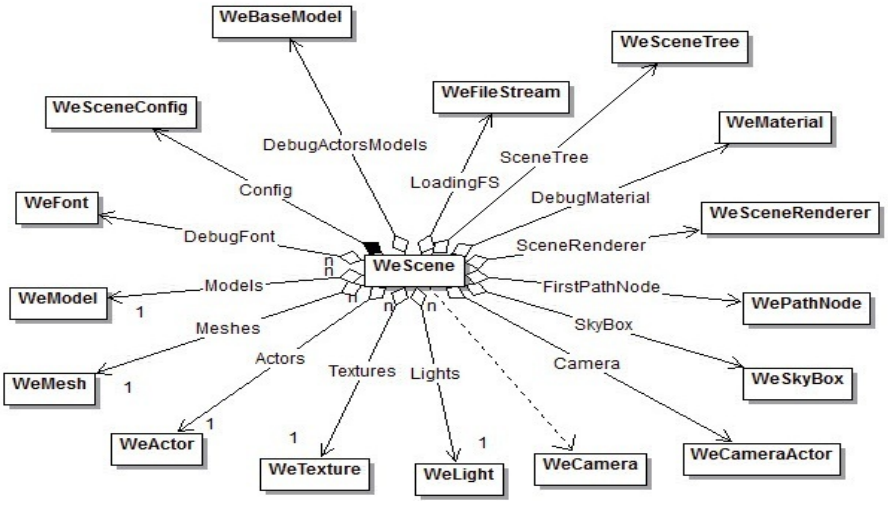
5. Projekat softvera

Alat GraphysX čine klase koje implementiraju različite ekrane. Ekrani definišu režim rada alata i mogućnosti interakcije korisnika. Svaki ekran ima sopstvenu grafičku reprezentaciju. Ekrani su izvedene iz *WeAbstractPhase* i imaju metode za pocetak, kraj, ažuriranje i prikazivanje. Postoji stek aktivnih ekrana i svi ekrani koji se na njemu nalaze se prikazuju, omogućavajući interakciju sa bilo kojim od njih. Ovakva koncepcija omogućava da se putem ekrana upravlja elementima grafičkog interfejsa poput modalnih i nemodalnih dijaloga. O tranziciji, dodavanju ekrana na stek i uklanjanju ekrana sa steka brine se klasa We*PhaseManager.* Dijagram klasa koji odgovara hijerarhiji ekrana je prikazan na slici 6.



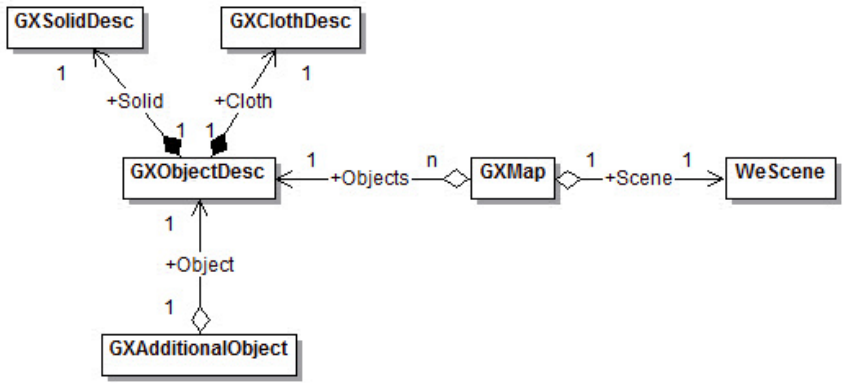
Slika 6: Dijagram ekrana

Za prikazivanje scena zadužena je klasa WeScene iz UltraEngine biblioteke. Ova klasa povezuje i omogućava saradnju većeg broja delova sistema zaduženog (posredno ili neposredno) za formiranje prikaza. Ovakav model povezivanja delova sistema omogućava jednostavnu zamenu jednog ili više delova sistema. Diagram koji prikazuje najvažnije klase povezane sa klasom WeScene je prikazan na slici 7.



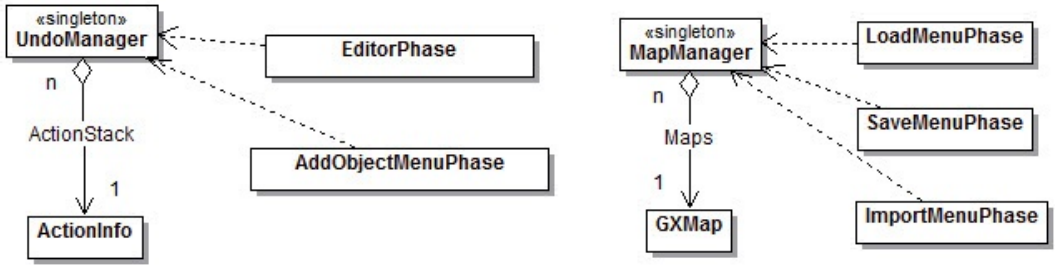
Slika 7: Dijagram WeScene klase

Osnovna klasa za sve geometrijske objekte je GXObjectDesc. Ona sadrži strukture GXClothDesc (za meka tela) i GXSolidDesc (za čvrsta tela), od kojih se u datom trenutku koristi samo jedna (u zavisnosti od toga da li je telo meko ili čvrsto). Klasa GXAdditionalObject predstavlja objekte koji se u nekom trenutku *t* ubacuju u scenu i sadrži objekat GXObjectDesc klase. Klasa GXMap sadrži opis inicijalnog stanja svih modela objekata koji se nalaze u sceni. Prilikom pokretanja (odnosno restartovanja) simulacije, modeli objekata se generišu na osnovu objekata sadržanih u GXMap objektu jer im se, u opštem slučaju, parametri menjaju u toku simulacije. Relacije između navedenih klasa su prikazane na slici 8.



Slika 8: Dijagram relacija između klasa objekata

Osim klasa prikazanih na slici 8, postoje još i klase MapManager i UndoManager koje koriste projektni uzorak Singleton. MapManager vodi računa o učitavanju i čuvanju scena u datotekama. UndoManager pamti akcije korisnika i omogućava povratak na prethodno stanje. Diagram klasa je prikazan na slici 9.



Slika 9: Diagram klasa za Manager objekte

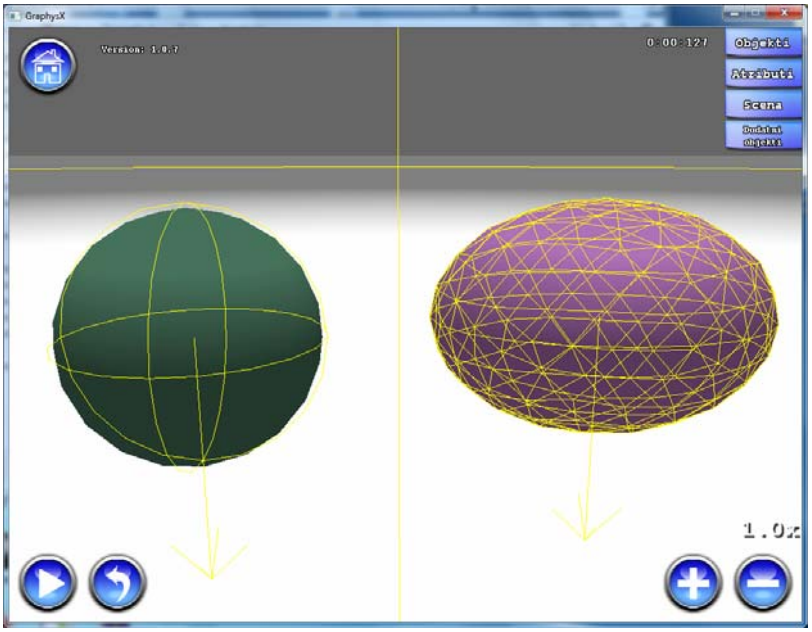
6. Implementacija softvera

Objekti koji se nalaze u sceni i na koje mogu da utiču fizičke sile, mogu biti aktivni i pasivni. Ukoliko na neki objekat ne deluje nijedna sila onda je on pasivan i ne uzima se u obzir prilikom proračuna. Kada na objekat počne da deluje sila (usled kolizije ili dejstva sila drugih objekata) on postaje aktivan i uzima se u obzir prilikom proračuna. Na ovaj način se štedi procersorsko vreme prilikom vršenja proračuna sila, jer se ignorišu objekti koji su pasivni.

Sistemi za fiziku prethodnih generacija su omogućavali animaciju samo čvrstih objekata zbog jednostavnosti i brzine. Moderan hardver omogućava animiranje mekih tela i njihovu interakciju sa drugim telima u realnom vremenu. Meka tela mogu da služe za simuliranje čestica, tečnosti i tkanina. U ovom radu je implementirana samo simulacija tkanina tj. mekih objekata.

Implementacija čvrstih tela

Čvrsta tela su implementirana korišćenjem NxBoxShape, NxSphereShape i NxTriangleMeshShape PhysX objekata (videti odeljak 4.1 – Arhitektura PhysX sistema). Ukoliko je moguće neki objekat predstaviti pomoću kvadra ili sfere onda se koriste odgovarajući NxBoxShape i NxSphereShape. Ako korisnik napravi neki nepravilan objekat, onda se koristi NxTriangleMeshShape koji je zahtevniji kako u pogledu potrebnog memorijskog prostora tako i u pogledu broja operacija prilikom obrade. Na slici 10 je dat primer sfere koje koristi NxSphereShape i nepravilnog tela koje koristi NxTriangleMeshShape. Računanje kolizije tih objekata sa drugim objektima vrši se proverom svih stranica geometrijskog tela pojedinačno sa geometrijom drugih objekata. Upravo zbog toga na performanse značajno utiče broj nepravilnih tela koja se nalaze u sceni.



Slika 10: NxSphereShape i NxTriangleMeshShape geometrije

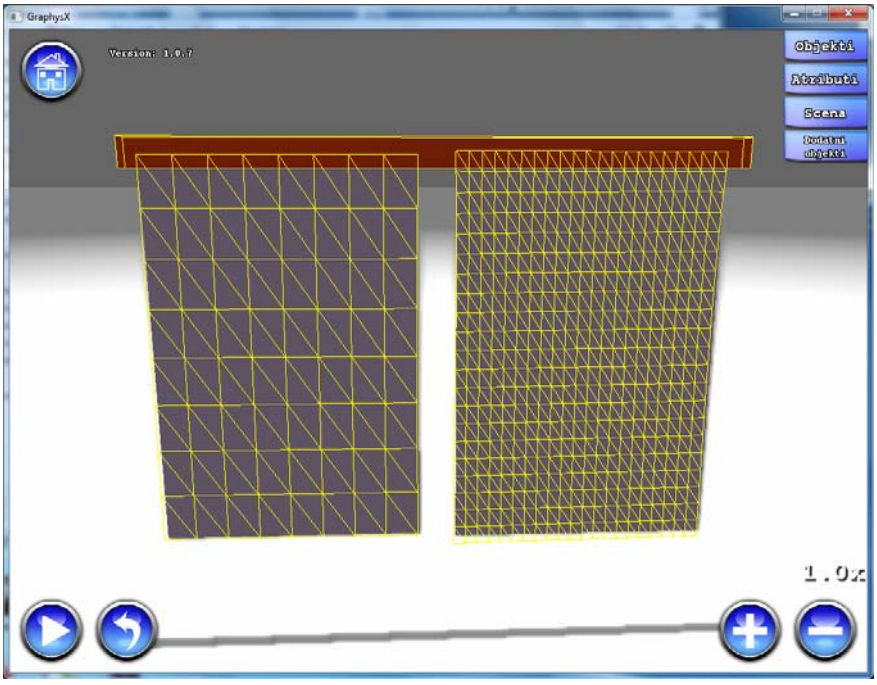
Atributi čvrstih objekata se predstavljaju pomoću jedinstvenih deljenih materijala. Jedan materijal može da koristi više objekata, ukoliko oni imaju zajedničke karakteristike. U najgorem slučaju broj materijala je jednak broju objekata u sceni.

Implementacija mekih tela

Za razliku od čvrstih tela, meka tela se mogu deformisati i njihovo ponašanje će zavisiti od atributa i finoće kojom je geometrijska mreža predstavljena. Pošto je objekat u početku predstavljen pomoću jednostavnog geometrijskog tela, da bi se tokom interakcije sa ostalim objektima u sceni dobilo realističnije ponašanje, potrebno je usložniti geometriju objekta (na primer, kao posledicu gužvanja ili savijanja objekta). Meka tela su implementirana pomoću NxCloth objekata. Za usložnjavanje geometrije objekata koristi se funkcija grafičke biblioteke DirectX `D3DXTessellateNPatches`, koja deli svaki trougao u sastavu geometrije objekta sa zadatim faktorom, a zatim generiše mrežu trouglova pomoću kojih će se simulirati meko telo. Faktor usložnjavanja je mera kojom se definiše povećanje složenosti geometrije objekata i može da se zada nezavisno za svaki objekat. Za svako meko telo se prvo izračuna broj segmenata (*numSeg*) koji bi on trebalo da ima na osnovu njegove veličine. Faktor usložnjavanja se zatim množi sa brojem segmenata i dobija se konačna vrednost koja se koristi kao parametar funkcije `D3DXTessellateNPatches`. Broj segmenata u odnosu na veličinu objekata se računa sledećim empirijski utvrđenim izrazom, koji je u primerima upotrebljenim za testiranje simulatora dao zadovoljavajuće rezultate:

```
float numSeg = 5 + (10 + maxSize) / 10;
```

Gde je `numSeg` broj segmenata i `maxSize` veličina najveće strane objekta.



Slika 11 Primer različitih faktora usložavanja

Implementacija gravitacije i naelektrisanja

Kao što je rečeno u odeljku 4.1, gravitaciona i elektrostatička sila se računaju pomoću objekata polja sile (*eng. ForceField*). Objekti polja sile su objekti koji imaju definisan prostor u kome se oseća prisustvo sile. Njima se pridružuje programsko jezgro (eng. kernel) koje definiše funkciju za računanje sile koja deluje u određenom delu tog prostora. Prostor objekata polja sile može da se definiše pomoću jednostavnih geometrijskih oblika (poput sfere, kvadra...), koji se mogu kombinovati, usled čega se dobija složeniji oblik prostora. Za jezgra koja služe za računanje sile koja deluje može da se koristi već gotovo rešenje u vidu LinearKernel klase ili se može definisati posebno jezgro za računanje sile. U ovom radu su korisćena LinearKernel jezgra, a oblik polja gravitacione i elektrostatičke sile je definisan sferom.

Gravitacija jednog objekta je predstavljena pomoću jednog objekta polja sile, pridruženog tom objektu, koji generiše silu koja privlači ostale objekte. Njegova sila privlačenja je proporcionalna masi pridruženog objekta. Naelektrisanje objekta je predstavljeno pomoću dva pridružena objekta polja sile koji generišu odbojnu i privlačnu silu (po jedan za privlačenje i odbijanje). Na nivou aplikacije su definisane kolizione grupe za svaku vrstu objekta i njihovih objekata polja sile. Koncept kolizionih grupa je objašnjen u narednom odeljku. Svakoj kolizionoj grupi se na osnovu vrednosti naelektrisanja, mase, sile globalnog elektrostatičkog polja i globalne gravitacione sile postavlja atribut da li na nju deluje zadata grupa objekata polja sile.

Nedostatak realizacije privlačne odnosno odbojne sile, bazirane na objektima polja sile, leži u činjenici da se za objekte na koje deluje dati objekat polja sile ne računa rezultujuća sila, već rezultujuće ubrzanje. Praktična posledica je nedosledno ponašanje simulacije u slučaju prisustva obe podržane sile. Na primer masivan naelektrisan objekat A će biti privučen (ili odbijen) istim ubrzanjem kao i manje masivan naelektrisan objekat B od nekog naelektrisanog objekta C (pod pretpostavkom da su na istom rastojanju od objekta C). U realnosti, ubrzanje koje se saopštava objektima A i B pod dejstvom elektrostatičke sile bi moralo da zavisi i od njihove mase (u ovom slučaju, objekat A bi imao manje ubrzanje).

Implementacija i vrste kolizionih grupa

Svaki objekat koji se nalazi u sceni pripada nekoj kolizionoj grupi. Kolizione grupe služe za definisanje interakcije između objekata, odnosno prisustvo kojeg objekta može da utiče na kretanje drugih objekata. Pošto postoji više vrsta objekata, postoje i različite kolizione grupe. Od grupa za objekte postoje:

- NORMAL_OBJECT (objekti koji nemaju masu ili naelektrisanje),
- GRAVITATIONAL_OBJECT (objekti sa sopstvenom masom),
- ELECTRIC_POSITIVE_OBJECT (pozitivno naelektrisan objekat),
- ELECTRIC_NEGATIVE_OBJECT (negativno naelektrisan objekat).

Pored grupa objekata, postoje i grupe za različite objekte polja sile koje služe za računanje sila:

- GRAVITATIONAL_FF (gravitacija objekata)
- ELECTRIC_P_POSITIVE_FF (privlačna sila za pozitivno naelektrisana tela)
- ELECTRIC_P_NEGATIVE_FF (odbojna sila za pozitivno naelektrisana tela)
- ELECTRIC_N_POSITIVE_FF (privlačna sila za negativno naelektrisana tela)
- ELECTRIC_N_NEGATIVE_FF (odbojna sila za negativno naelektrisana tela)
- ELECTROSTATIC_P_FF (privlačna sila za globalno elektrostatičko polje)
- ELECTROSTATIC_N_FF (odbojna sila za globalno elektrostatičko polje)

Na normalne objekte (objekte bez mase i naelektrisanja) ne treba da deluju nikakve sile pa se zato ignorišu sve sile isključivanjem interakcije između kolizione grupe NORMAL_OBJECT i svih ostalih. Slično tome, na pozitivno naelektrisan objekat sa masom mogu da utiču sledeće kolizione grupe: ELECTRIC_N_POSITIVE_FF (za privlačenje objekta), ELECTRIC_P_NEGATIVE_FF (za odbijanje objekta), ELECTROSTATIC_P_FF (elektrostatičko polje za privlačenje) i GRAVITATIONAL_FF (za privlačenje gravitacione sile). Segment izvornog koda koji ilustruje primer postavljanja kolizionih grupa je prikazan na kodu 1. U tabeli 1 je prikazano između kojih grupa je dozvoljena interakcija.

Kod 1: izvorni kod za inicijalizaciju kolizionih grupa.

```
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, ELECTRIC_P_POSITIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, ELECTRIC_P_NEGATIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, ELECTRIC_N_POSITIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, ELECTRIC_N_NEGATIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, GRAVITATIONAL_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, ELECTROSTATIC_P_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(NORMAL_OBJECT, ELECTROSTATIC_N_FF, false);

WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, ELECTRIC_P_POSITIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, ELECTRIC_P_NEGATIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, ELECTRIC_N_POSITIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, ELECTRIC_N_NEGATIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, GRAVITATIONAL_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, ELECTROSTATIC_P_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(GRAVITATIONAL_OBJECT, ELECTROSTATIC_N_FF, false);

WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, ELECTRIC_P_POSITIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, ELECTRIC_P_NEGATIVE_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, ELECTRIC_N_POSITIVE_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, ELECTRIC_N_NEGATIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, GRAVITATIONAL_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, ELECTROSTATIC_P_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_POSITIVE_OBJECT, ELECTROSTATIC_N_FF, false);

WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, ELECTRIC_P_POSITIVE_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, ELECTRIC_P_NEGATIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, ELECTRIC_N_POSITIVE_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, ELECTRIC_N_NEGATIVE_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, GRAVITATIONAL_FF, true);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, ELECTROSTATIC_P_FF, false);
WePhysics::Instance.SetGroupCollisionInteract(ELECTRIC_NEGATIVE_OBJECT, ELECTROSTATIC_N_FF, true);
```

Interakcije između grupa	NORMAL_OBJECT	GRAVITATIONAL_OBJECT	ELECTRIC_POSITIVE_OBJECT	ELECTRIC_NEGATIVE_OBJECT
GRAVITATIONAL_FF		X		
ELECTRIC_P_POSITIVE_FF				X
ELECTRIC_P_NEGATIVE_FF			X	
ELECTRIC_N_POSITIVE_FF			X	
ELECTRIC_N_NEGATIVE_FF				X
ELECTROSTATIC_P_FF			X	
ELECTROSTATIC_N_FF				X

Tabela 1: Prikaz dozvoljenih interakcija između grupa.

Format sačuvanih scena

Scena se čuva u tekstualnom formatu u xml datoteci. Ovim se omogućava jednostavno proširenje formata u kojem je sačuvana scena. Format datoteke je dat u prilogu A.

7. Zaključak

U radu je razmatran edukativni alat GraphysX koji vrši simuliranje mehaničkih sistema primenom biblioteke PhysX kompanije nVidia. Rad je imao za cilj da prikaže neke od mogućnosti ove biblioteke kroz razvoj jednostavnog simulatora mehaničkih sistema. Implementirana je aplikacija u kojoj je pored simulatora dostupan integrisan editor scena. Objašnjene su dostupne opcije za manipulisanje objektima i njihovim atributima u editoru scena, kao i značenje atributa za čvrste i meke objekte čime se korisnicima stavlja na raspolaganje alat sa jasnim mogućnostima upotrebe.

GraphysX aplikacija može pomoći studentima da upoznaju biblioteku PhysX i počnu sa njenom upotrebom u svojim budućim projektima. Takođe može biti od koristi u nastavi iz fizike u srednjim i osnovnim školama za objašnjavanje putem primera koje i sami učenici mogu da sastave, a zatim i eksperimentišu nad njima.

U ovom radu nisu podržane neke mogućnosti PhysX-a kao što su meke veze ili opruge. Buduće verzije simulatora treba da omoguće definisanje veznih objekata između druga dva objekata i postavljanje parametara usled kojih će se vezni objekat ponašati kao meka veza ili opruga.

Objekti polja sile trenutno rade sa već imlementiranim (predefinisanim) programskim jezgrima za računanje kretanja objekata, što za posledicu ima netačno računanje kretanja objekata koji istovremeno imaju masu i naelektrisanje. U cilju postizanja doslednih rezultata simulacije, trebalo bi implementirati nova jezgra za računanje, koja bi za objekte na koje deluju objekti polja sile uzimala u obzir njihovu masu prilikom računanja rezultujućeg ubrzanja.

8. Literatura

[1] http://en.wikipedia.org/wiki/Physx

[2] NVIDIA PhysX SDK 2.8 Documentation

[3] Igra koja koristi PhysX - http://en.wikipedia.org/wiki/Mirror's_Edge

[4] Igra koja koristi PhysX - http://en.wikipedia.org/wiki/Unreal_Tournament

[5] Fizika u filmovima - http://www.intuitor.com/moviephysics/goodMovies.html

[6] Working Model – http://en.wikipedia.org/wiki/Working_Model

[7] Phyz - http://en.wikipedia.org/wiki/Phyz

[8] TinyXml biblioteka - http://www.grinninglizard.com/tinyxml/

[9] UnrealEngine 3 - http://en.wikipedia.org/wiki/Unreal_Engine_3

[10] Gamebryo Engine - http://en.wikipedia.org/wiki/Gamebryo

[11] Unity Engine - http://en.wikipedia.org/wiki/Unity_(game_engine)

[12] Havok - http://en.wikipedia.org/wiki/Havok_(software)

[13] Newton Game Dynamics – http://en.wikipedia.org/wiki/Newton_Game_Dynamics

[14] Open Dynamic Engine - http://en.wikipedia.org/wiki/Open_Dynamics_Engine

[15] Bullet - http://en.wikipedia.org/wiki/Bullet_(software)

[16] DirectX - http://en.wikipedia.org/wiki/DirectX

Prilog A

Struktura XML dokumenta scene

Dat je primer dokumenta u kome je sačuvana scena. Scena sadrži tri objekata (statični, čvrsti i meki objekat) od kojih čvrsti objekat ima još jedan čvrsti objekat kao svog sina.

```
<Scena>
	<Gravitacija>0,0,0</Gravitacija>
	<Tekstura>0</Tekstura>
	<Boja>0,0,0</Boja>
	<objekat>
		<Geometrija>Kutija</ Geometrija>
		<Tekstura>0</Tekstura>
		<Boja>0,0,0</Boja>
		<Pozicija>0,0,0</Pozicija>
		<Rotacija>0,0,0</Rotacija>
		<Skaliranje>1,1,1</Skaliranje>
		<Staticno>1</Staticno>
		<Materijal>Cvrst</Materijal>
	</Objekat>
	<objekat>
		<Geometrija>Kutija</ Geometrija>
		<Tekstura>0</Tekstura>
		<Boja>0,0,0</Boja>
		<Pozicija>0,0,0</Pozicija>
		<Rotacija>0,0,0</Rotacija>
		<Skaliranje>1,1,1</Skaliranje>
		<Staticno>0</Staticno>
		<Materijal>Cvrst</Materijal>
		<Restitucija>1</Restitucija>
		<StatickoTrenje>1</StatickoTrenje>
		<DinamickoTrenje>1</DinamickoTrenje>
		<UgaonoKocenje>1</UgaonoKocenje>
		<objekat>
		<Geometrija>Kutija</ Geometrija>
			<Tekstura>0</Tekstura>
			<Boja>0,0,0</Boja>
			<Pozicija>0,0,0</Pozicija>
			<Rotacija>0,0,0</Rotacija>
			<Skaliranje>1,1,1</Skaliranje>
			<Staticno>0</Staticno>
			<Materijal>Cvrst</Materijal>
			<Restitucija>1</Restitucija>
			<StatickoTrenje>1</StatickoTrenje>
			<DinamickoTrenje>1</DinamickoTrenje>
			<UgaonoKocenje>1</UgaonoKocenje>
		</Objekat>
	</Objekat>
	<objekat>
		<Geometrija>Kutija</ Geometrija>
		<Tekstura>0</Tekstura>
		<Boja>0,0,0</Boja>
		<Pozicija>0,0,0</Pozicija>
		<Rotacija>0,0,0</Rotacija>
		<Skaliranje>1,1,1</Skaliranje>
		<Staticno>0</Staticno>
		<Materijal>Elastican</Materijal>
		<Rastegljivost>1</Rastegljivost>
		<Savitljivost>1</Savitljivost>
		<Gustina>1</Gustina>
		<Trenje>1</Trenje>
		<KoefKidanja>1</KoefKidanja>
		<Pritisak>1</Pritisak>
		<KoefPrivTela>1</KoefPrivTela>
		<KoefSudara>1</KoefSudara>
	</Objekat>
</Scena>
```