Detekcija sudara pomoću Unity3d

Tehnička dokumentacija

Verzija 2.0

Studentski tim: Teon Banek

Tomislav Ljubej

Marko Đomlija

Ivan Hakštok

Ana Stepić

Niko Mikuličić

Damir Ciganović-Janković

Nastavnik: prof. dr. sc. Željka Mihajlović

Sadržaj

[1. Opis razvijenog proizvoda 3](#_Toc408434340)

[2. Korišteni alati 3](#_Toc408434341)

[2.1. Blender 3](#_Toc408434342)

[2.2. Unity3d 3](#_Toc408434343)

[3. Tehničke značajke 4](#_Toc408434344)

[3.1. Pojednostavljivanje modela 4](#_Toc408434345)

[3.2. Dodavanje sudarača 6](#_Toc408434346)

[3.3. Performanse 7](#_Toc408434347)

[4. Upute za korištenje 8](#_Toc408434348)

[5. Literatura 9](#_Toc408434349)

Tehnička dokumentacija

# Opis razvijenog proizvoda

Proizvod ovog projekta je program kojim je prikazana mogućnost Unity3d alata u detekciji kolizije. Cilj je bio istražiti koliko dobre performanse se mogu dobiti nad jednim složenijim 3d modelom. Za potrebu ovoga projekta, korišten je detektor čađi tvrtke AVL “Microsoot”.

# Korišteni alati

## 2.1. Blender

Blender [1] je program za modeliranje, oblikovanje modela, izradu animacija te naposljetku i računalnih igara napravljen od strane tvrtke iz nizozemske, Blender Foundations. Spomenuti program ima odličnu potporu za modificiranje 3d modela. U ovom se projektu zato Blender koristio prilikom pojednostavljivanja modela detektora čađi.

## 2.2. Unity3d

Unity3d [2] je program za izradu videoigara (engl. game engine). Prva verzija Unity-a na tržištu je izašla 2005. godine te je bio namijenjen samo za korisnike OS X operacijskog sustava, no od tada se proširio i na druge platforme te ima više od 2 milijuna registriranih korisnika. Osim toga, pogodan je za izvoz i na mobilne operacijske sustave, igraće konzole (Xbox, PlayStation ) te za web aplikacije. Moguće je stvarati igre svih žanrova te je integrirano radno sučelje za 2D i 3D razvoj igara. Gotovo sve potrebno, pri korištenju Unity-a, je na dohvat ruke zbog lakog uvoza, ne samo svojih modela, skripti i materijala, nego i ostalih sredstava (engl. assets) za izgradnju videoigre iz trgovine sredstava kojoj se može pristupiti u svakom trenutku čak bez otvaranja Internet pretraživača. Postoji podrška za sredstva uvezena iz Blendera, Maye, 3ds Max-a, Adobe Photoshopa te mnoge druge. Osim toga, Unity ima podršku za nove tehnologije poput Kinect-a, Occulus Rift-a te Leap Motion-a.

# Tehničke značajke

## 3.1. Pojednostavljivanje modela

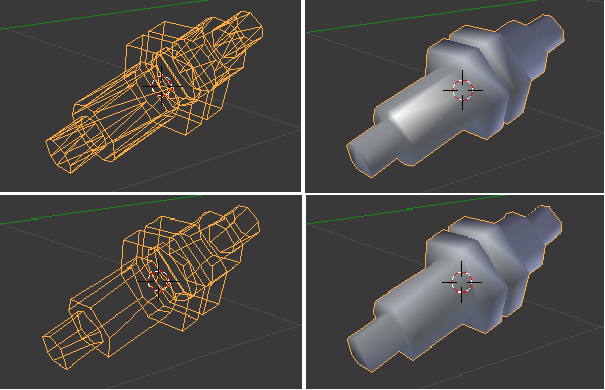
U Blenderu postoji više načina i alata koji mogu poslužiti za pojednostavljivanje korištenog modela. Za potrebe ovog projekta korišten je bio modifikator *Decimate* [3]*.*

Decimate modifikator je modifikator koji smanjuje broj točaka i lica u nekom objektu uz minimalne promjene strukture. Modifikator koristi više algoritama za pojednostavljivanje modela. Mogući algoritmi su urušavanje (eng. *collapse*), poništavanje podjele (eng. *unsubdivide*) i poravnavanje (eng. *planar*). U ovom projektu, zbog specifičnosti modela korišteni su samo algoritmi *collapse* i *planar*.

*Collapse* algoritam koristi težinske faktore dodijeljene pojedinim vrhovima. Algoritam kao parametar prima omjer željenog broja lica i ukupnog broja lica unutar jednog objekta. Zatim se ovisno o težinskom faktoru uklanjaju određeni dijelovi geometrije kako bi se smanjio ukupan broj lica. Ovaj algoritam pogodan je za uklanjanje izoliranih vrhova te parova vrhova koji se nalaze na maloj udaljenosti.

*Planar* algoritam kao parametar prima najmanji kut između normala dvaju susjednih lica unutar nekog objekta. Ako su dva lica u istoj ravnini, razlika između njihovih normala će biti 0° ili 180° (u slučaju dvostranih lica). Kroz lica se prolazi dva puta, jer je moguće nastajanje kumulativnog efekta pri spajanju. Ovaj algoritam pogodan je za pojednostavljivanje zakrivljenih površina, gdje su neravnine prikazane velikim brojem poligona, te također za objedinjavanje višestrukih poligona u istoj ravnini u jedan poligon.

Broj lica u modelu se u konačnici smanjio za otprilike 40 %.

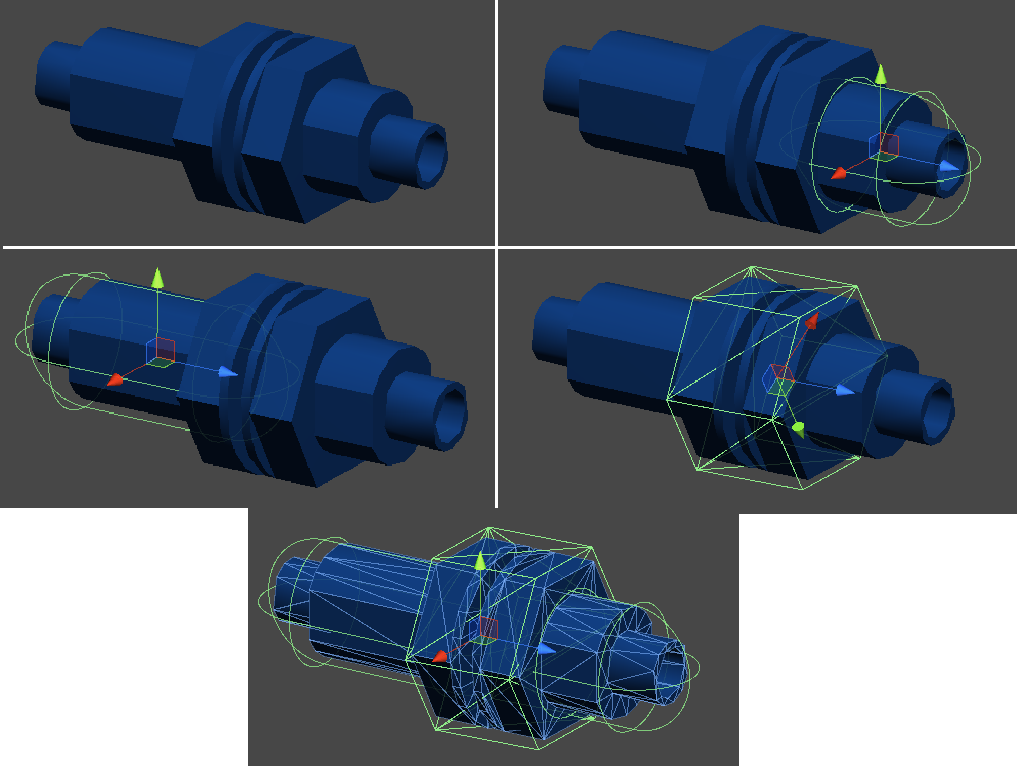


**Slika 1.** Gornje dvije slike prikazuju stanje komponente modela prije korištenja decimate modifikatora, a donje dvije prikazuju stanje nakon. Broj lica (engl. faces) se smanjio s 230 lica na 58 lica, no promjene na obliku modela se gotovo i ne vide.

## 3.2. Dodavanje sudarača

Geometrija koja definira izgled objekta je obično prekompleksna za korištenje u detekciji i simulaciji kolizije te se za te potrebe koristi druga, jednostavnija geometrija koja sadrži manje vrhova i lica ili se sastoji od nekih primitiva za koje se jednostavno računa kolizija, npr sfera ili kocka. Takva geometrija se onda ne iscrtava već služi samo za detekciju kolizije, nazivamo ju “sudarač”. U razvojnom alatu Unity postoje dvije mogućnosti za postizanje spomenutoga.

Možemo za svaki objekt definirati jedan od primitivnih sudarača, to može biti “box collider”, tj. kvadar, prikladan je za objekte koji otprilike imaju oblik kvadra, “capsule collider”, koji se sastoji od valjka i dvije polukugle na bazama valjka, a prikladan je za većinu duguljastih objekata koji nisu kvadratnog oblika. Osim tih, postoji i “Sphere collider”, sfera, prikladan je naravno za sferične objekte. Uz ove sudarače možemo koristiti i geometrijska tijela napravljena u nekom drugom alatu poput Blendera. Takvi likovi moraju biti konveksni i relativno jednostavni inače performanse simulacije vrlo brzo padaju. Nekada za neki objekt nije prikladan samo jedan primitivni collider već ih treba više. Na primjer, neki objekt koji se sastoji od nekoliko kvadra bi trebao imati nekoliko “box collidera”. To radimo tako da dodamo još objekata u scenu koji će se sastojati samo od jednog primitivnog sudarača te ih postavimo kao djecu glavnome objektu te naravno postavimo položaj, rotaciju i veličinu sudarača da odgovaraju glavnom objektu.



**Slika 2.** Na slici vidimo postupak dodavanja sudarača na složeniji objekt. Prva slika pokazuje objekt bez sudarača a na zadnjoj vidimo sve sudarače (sudarači su iscrtani zelenim linijama). Prva dva sudarača su „capsule collideri“ koji su dodani na prednji i stražni dio objekta dok treći nije dio Unity alata već je jednostavno geometrijsko tijelo koje je napravljeno alatom Blender te pretvoreno u sudarač.

Za složenije modele za koje nije prikladno imati primitivne sudarače možemo koristiti i njihovu geometriju za iscrtavanje. Ovo je jako računski zahtjevno te se treba izbjegavati gdje je god moguće.

## 3.3. Performanse

Scena se sastoji od 59 statičnih sudarača te 324,200 trokuta za iscrtavanje. U sceni je također 800 čestica koje se kreću i odbijaju od sudarača objekta. Čestice nemaju svoje sudarače već se fizika za njih računa zasebno od već postojeće fizike u Unityu pomoću dodatka Unityu „Particle playground“.

Program je testiran na računalu sa intel i7 360QM 2.3 GHz procesorom i Nvidia GeForce GTX 660M grafičkom karticom te se vrti na stabilnih 30 FPSa što je zadovoljavajuć rezultat, no da sudarači nisu statični performanse bi znatno pale. Ručno dodavanje primitivnih sudarača na objekte i pojednostavljivanje objekta je uvelike pomoglo performansi programa. Postoji mogućnost za daljnju optimizaciju tako da se objekt podijeli na još manje dijelove i još više pojednostavi.

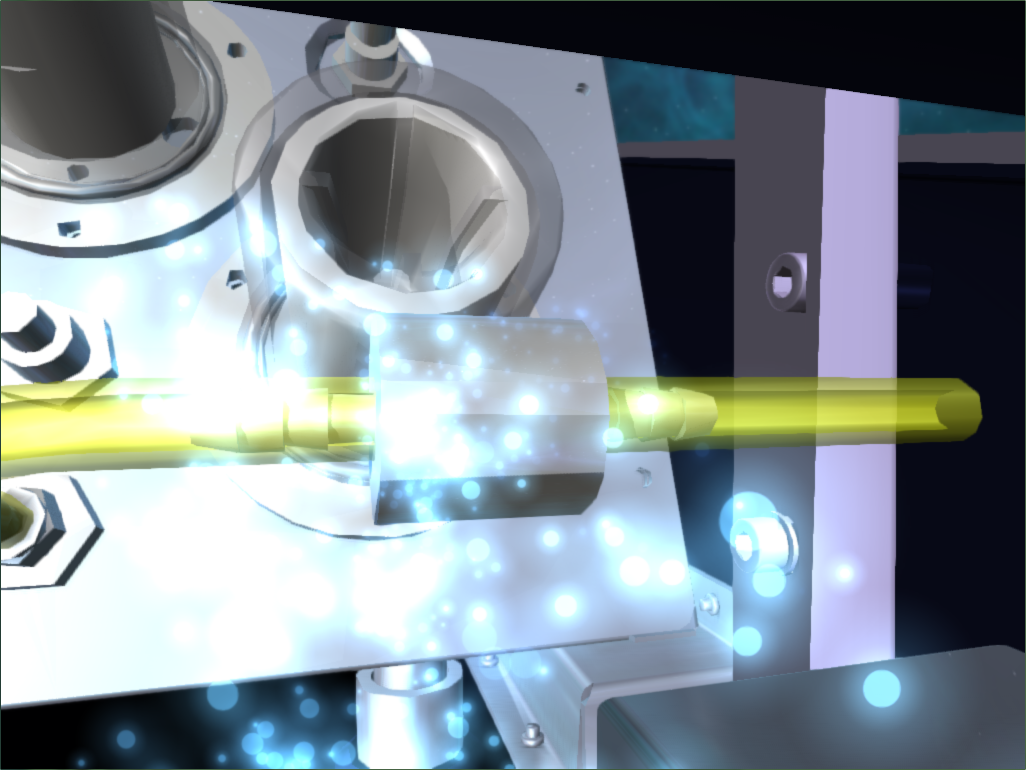
Program je demonstriran „letom“ (engl. flytrough) kroz model te je pušten roj čestica koje se s njime sudaraju. Unity ima podršku za Oculus Rift [4] te je, osim običnog načina korištenja programom, omogućeno i korištenje uz pomoć toga uređaja.

# Upute za korištenje

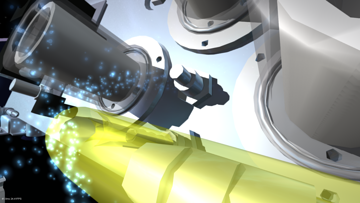
Aplikacija projekta se pokreće dvoklikom na izvršnu datoteku. Otvara se prikaz uređaja i pokreće se tok čestica kroz taj uređaj. Korisniku su ponuđene sljedeće kontrole.

* *Escape* tipka resetira aplikaciju na početno stanje.
* *Enter* tipka uključuje i isključuje prikaz broja sličica u sekundi na ekranu
* Tipka za razmak zaustavlja izvođenje ako je u tijeku, ako je zaustavljeno tada nastavlja s izvođenjem.
* Pritisak lijeve strelice usporava brzinu izvšavanja simulacije.
* Pritisak desne strelice ubrzava brzinu izvšavanja simulacije.

Slike u nastavku prikazuju pokrenutu aplikaciju.



Slika 3. Ulazak čestica u uređaj.



**Slika 4.** Slika zaslona za vrijeme izvođenja aplikacije



**Slika 5.** Slika zaslona za vrijeme izvođenja aplikacije

# Literatura

1. Blender Foundation, <http://www.blender.org/>
2. Unity Technologies, <http://unity3d.com/>
3. Blender Foundation, <http://wiki.blender.org/index.php/Doc:2.6/Manual/Modifiers/Generate/Decimate>
4. Palmer Luckey, <https://www.oculus.com/>