

Implementacija mrežne višekorisničke video igre borbe robota u Unreal radnom okviru

Implementation of a network multiplayer robot fighting video game in Unreal engine

Uvod

Unreal Engine je savremeni i svestrani alat za izradu video igara koji korisnicima omogućuje potpuni i nezavisni razvoj video igre u svim aspektima i nivoima. Predstavlja snažnu bazu za kreiranje visoko optimizovanih, kompleksnih igara i danas je industrijski standard. Njegova implementacija sadrži podršku za sve standardne zahteve sadašnjih video igara, kao što su 3D i 2D grafičko renderovanje, simulacija fizike, sistem za izradu korisničkog interfejsa, audio sistem, umrežavanje više igrača, registrovanje ulaza od strane igrača i ostale. Unreal Engine je softver otvorenog koda koji se svakodnevno unapređuje, te sadrži i podršku za najnovije tehnologije poput veštačke inteligencije, praćenje zraka svetlosti (ray tracing) kao i voksele (3D alternativa pikselu). Razlog njegove popularnosti i među visokobudžetnim kompanijama i nezavisnim korisnicima jeste upravo javni pristup njegovom kodu, te ga korisnici mogu neograničeno i bez ograničenja prilagoditi svojim potrebama. Okuplja široku zajednicu koja razvija alate i dodatke, i danas Unreal Engine sa svojim širokim spektrom mogućnosti nije samo softver za programere, već za i sve ostale učesnike u kreiranju jedne igre: dizajnere, skulptore, crtače, arhitekte, animatore, dizajnere zvuka, producente, analitičare.

Ovaj projekat predstavlja istraživanje svih aspekata izrade jedne višekorisničke 3D video igre, sa naglaskom na samo umrežavanje igrača. Tematika video igre je međusobna borba robota na tačkovima, što je i popularna disciplina u stvarnosti. Cilj je potpuno funkcionalna igra gde korisnik bira robota i uz jednostavan interfejs se konektuje u sesiju sa drugim igračima i otpočinje borbu. Unreal Engine je izabran kao radni okvir zbog svoje izvanredne podrške za realizaciju 3D igre: jednostavni uvoz Blender 3D modela u igru, podrška za animaciju modela, implementirana fizika za vozila i kolizije, jednostavna izrada terena i mape i ugrađena podrška za umrežavanje igrača. Zahvaljujući Unreal Engine sistemu za umrežavanje, preko daljinski pozivanih procedura (RPCs), repliciranih svojstava i upravljanju sesijama, rezultat je međusobna sinhronizacija između različitih igrača uz minimalno vreme kašnjenja i stabilnu konekciju.

Summary

Unreal Engine is modern and versatile tool for developing video games that allows users to fully and independently develop video games in all aspects and levels. It represents strong base for creating highly optimized, complex games and today it is industry standard. Its implementation consists of support for all the standard requirements for video games, such as 3D and 2D graphics rendering, physics simulation, system for developing user interfaces, audio system, multiplayer networking, handling inputs and others. Unreal engine is open source software that is continuously evolved, so it contains support for new, cutting edge technologies, such as artificial intelligence, ray tracing, voxels etc. The reason for its popularity among high-budget studios and independent users is the public access to its code, to users can adapt it to their needs indefinitely and without restrictions. It attracts a wide community that develops tools and plugins, and today, Unreal Engine is not only software for game programmers, but also for all other participants in the creation of a game: designers, sculptors, artists, architects, animators, sound designers, producers, analysts.

This project represents the research of all aspects of creating a multiplayer 3D video game, with an emphasis on the networking. The theme of the video game is mutual fight of wheeled robots, which

is also a popular discipline in real world. The goal is a fully functional game where the user chooses a robot to play with and with a simple interface connects to a session with other players and starts to battle. Unreal Engine was chosen as the framework because of its outstanding support for 3D game development: easy import of Blender 3D models, support for animating models, existing physics for vehicles and collisions, easy creation of terrain and maps, and built-in support for player networking. Thanks to the Unreal Engine networking system, via Remote Procedure Calls (RPCs), replicated properties and session management, the result is synchronization between different players with minimal time delay and a stable connection.

Opis problema sa pregledom postojećih resenja

Umrežavanje

U zavisnosti od igre, arhitektura umrežavanja se razlikuje, i na programeru je da koristeći Unreal Engine API prilagodi mrežni sistem svojim potrebama. Korisnički zahtevi vezani za umrežavanje su konkretno: mogućnost klijenta da prvenstveno vrši pretragu postojećih sesija i konektuje se na izabranu, ukoliko nema mogućih sesija kreira novu i preuzima ulogu servera, održava sinhronizaciju u realnom vremenu o svojstvima i akcijama robota. Iako Unreal Engine sadrži arhitekturu umrežavanja, to ne znači da je umrežavanje klijentskih procesa brz i jednostavan način bez dodatnih konfiguracija i algoritama. Procesi nižeg nivoa, kao što su kreiranje i održavanje klijentskih konekcija, slanje paketa, serijalizacija, rutiranje, load-balancing, već si implementirani na nivou radnog okvira, te klijent, odnosno programer, kreira logiku umrežavanja na nivou veće apstrakcije [1]. Rezultat ovog projekta jeste i mrežna arhitektura koja može služiti kao osnova za druge igre sličnih karakteristika.

Modeli

U igri sa malom i jednostavnom mapom, sa konkretnim i očekivanim akcijama i strogo definisanim pravilima, glavni utisak na igrača ostavlja model svog virtuelnog karaktera. Suprotno od estetike modela, potrebno je voditi računa i o njegovoj kompleksnosti. Model robota mora biti optimizovan jer će svaki klijent u svakom trenutku prikazivati i modele svih ostalih igrača u sesiji. Takođe različiti tipovi karaktera moraju biti i dovoljno vizuelno različiti, kako bi igrač jednostavno mogao da prepozna tipove protivnika u žaru borbe. Model mora poštovati i logičke karakteristike karaktera kome pripada, i njihove različite uloge.

Animacije i vizuelni efekti

Zajedno sa modelima, animacije i vizuelni efekti predstavljaju jedan od najupečatljivih detalja u igri. Igrači očekuju realistične animacije u odnosu na zahtevane akcije. Upravljanje vozilom treba se jasno uočiti na samom modelu vozila. Ubrzavanje, kočenje, i biranje smeru treba reflektovati na tačkove vozila, kao i na samo vozilo po zakonima inercije. Glavna odlika koja razlikuje različite tipova robota jeste njihovo oružje, zbog čega je jako bitno ispravno animirati pokrete aktiviranja oružja. Sve spomenute animacije potrebno je izvršiti u realnom vremenu kako bi korisnik imao potpuni doživljaj kontrole robota. U haotičnom okruženju neprestane borbe bitno je da animacije aktivacija različitih tipova oružja budu jasno vidljive i razlikujuće kako bi igrač imao šanse da

izbegne konkretne napade. Unreal Engine sadrži dobru podršku za animiranje dinamičkih modela, gde se model deli na sitnije logičke celine koje je moguće zasebno manipulirati.

Pored animacija, još jedan način kako igrači vizuelizuju događaje u igri su specijalni efekti, koji se aktiviraju kao posledica nekog događaja koji je od izuzetnog značaja. U ovom projektu vizuelni efekti su posledica uspešnog napada jednog robota na drugi. Specijalni efekti jedan su od najčešćih rešenja za problem optimizacije – njihovim smanjenjem se dobija manje iskorišćenje resursa računara.

Fizika vozila

Igrači očekuju da pored ispravnog vizuelnog pokreta točkova i oružja, ti pokreti jasno oslikavaju i fizičku realnost u igri. U tim slučajevima, zapravo je fizička logika ta koja upravlja animacijama. Neki od glavnih izazova pri razvijanju vozila su odnos vozila prema različitom terenu, preprekama, brzini, uglu točkova itd. Zahvaljujući određenim dodacima, u Unreal Engine se dosta fizičke logike konfiguriše parametrima. Bitni parametri vozila koje je moguće podesiti su sam oblik fizičkog tela koje je različito od vizuelnog modela, konfiguracija točkova, trenja, proklizavanja, motora, težišta i mase objekta.

Poseban problem predstavlja definisanje oblika fizičkog tela, koje osim što utiče na ostale fizičke pojave poput inercije, ima i ulogu detektovanja kolizije. Detektovanje kolizije predstavlja i srž same definicije igre, jer je ključna za ispravno funkcionisanje celokupne igre. U brzjoj igri borbe robota, u maloj areni, kolizije se dešavaju često i haotično. Ignorisanjem ili pogrešnim detektovanjem kolizije dolazimo do situacija prolaska robota jedne kroz druge kao i pogrešnim apliciranjem štete od oružja.

Grafički korisnički interfejs

Informacije i stanje igre koje igrači ne mogu zaključiti iz animacija i efekata predstavlja se putem korisničkog interfejsa. U video igrama dva su osnovna tipa korisničkog interfejsa sa drugačijim zahtevima, HUD (Head-up display) i meniji.

HUD predstavlja sve tekstualne i 2D informacije prikazane igraču preko 3D simulacije igre. Glavni zahtevi su jednostavnost, laka uočljivost i razaznalost podataka, prikaz svih potrebnih informacija bez zatrpavanja ekrana suvišnim. U višekorisničkoj igri često podrazumeva i prikazivanje informacija o ostalim igračima u realnom vremenu. HUD treba da ima sve potrebne podatke, a ujedno i da ne ometa i oduzima fokus igračima od same igre.

Meniji predstavljaju zasebni, nezavisniji deo video igre i prate druga pravila i probleme. Osim što služe kao izvor informacija, takođe omogućuju i konkretne akcije. Glavni zahtev je intuitivnost. Igrač treba bez puno razmišljanja i sa lakoćom da dođe do željene akcije. Kod konkretne video igre, meni se prikazuje tik pred ulazak u igru, i služi da korisnik pre konekcije na server izabere željenog karaktera i započne igru. Jedan od glavnih izazova jeste ispuniti spomenute zahteve za rešenje za odabir karaktera.

Dizajn i pravila igre

Kako bi zainteresovala i zadržala igrače, video igra mora da ima balansirana i privlačna pravila. Potreban je cilj, kao i jasne povratne informacije koliko smo mu blizu. Ključna mehanika jeste borba robotima, haotična, brza, nemilosrdna, dinamična. Igrači očekuju fer i balansirano iskustvo, gde svaki tip robota ima podjednake šanse za pobedu, bez očiglednih favorita. Stil igre, umeće i kreativnost trebaju biti jedini parametri koji predviđaju pobedu. Podesivi parametri su brzina, maksimalni životni poeni, manevarske sposobnosti, efikasnost oružja. Vremenom, iskustvom i testiranjem, potrebno je svakodnevno podešavati ove parametre kako bi dostigli balansiranu igru. Takođe je podjednako važno redovno ažurirati igru, ubacivanjem novih tipova robota kao i novih mapa.

Cilj video igre je zabava. To je ujedno konačni izazov pri razvijanju igre, učiniti je zabavnom za ostale igrače. Svi navedeni podnaslovi ponaosob utiču na celokupno iskustvo.

Opis koriscenih tehnologija