

**KREIRANJE OKRUŽENJA I IMPLEMENTACIJA OSNOVNIH ELEMENATA FPS
VIDEO IGRE UNUTAR UNREAL ENGINE-A****CREATING THE ENVIRONMENT AND IMPLEMENTING THE BASIC ELEMENTS OF
AN FPS VIDEO GAME WITHIN THE UNREAL ENGINE**

Milan Mišćević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSKA GRAFIKA

Kratak sadržaj – *Zadatak ovog rada jeste kreiranje okruženja i implementacija osnovnih elemenata FPS video igre unutar Unreal Engine-a. Definisana je razlika i sličnost između dizajna nivoa i dizajna video igre. Objasnjeni su tipovi dizajna nivoa, navedene su njihove mane i prednosti. Detaljno je objašnjen proces kreiranja nivoa, kao i implementacije FPS elemenata u engine-u. Kao krajnji rezultat, dobijen je funkcionalni level video igre unutar kog su ispunjeni svi zahteve propisani pravilima FPS video igara.*

Ključne reči: *Razvoj video igre, dizajn nivoa, dizajn video igre, Unreal Engine, FPS video igre*

Abstract – *The task of this paper is to create an environment and implement the basic elements of an FPS video game within the Unreal Engine. The difference and similarity between level design and video game design is defined. The types of level design are explained, their disadvantages and advantages are listed. The process of level creation, as well as the implementation of FPS elements in the engine, is explained in detail. As a final result, a functional video game level was obtained within which all the requirements prescribed by the rules of FPS video games were met.*

Keywords: *Video game development, game design, level design, Unreal Engine, FPS game*

1. UVOD

Ideja ovog rada jeste kreiranje FPS video igre, počevši od dizajna nivoa pa sve do implementacije osnovnih elemenata FPS video igre. Igra je osmišljena tako da se kroz prvi nivo igrač upozna sa osnovnim kontrolama neophodnim za igru, načinom rukovanja oružjem i na taj način pripremi za složenije nivoe koji bi u budućnosti mogli biti razvijeni. Kroz teorijski deo objašnjena je sličnost i razlika između dizajna nivoa i dizajna video igara i navedeni su osnovni tipovi struktura dizajna nivoa. Ova igra je kreirana kao polu-linearna struktura i namenjena je prvenstveno za pokretanje na računarima, ali uz minimalne izmene komandi može se prilagoditi i drugim platformama poput mobilnih telefona, tableta ili VR uređaja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ratko Obradović, red. prof.

Sam tok igre vrlo je jednostavan, igrač se nalazi unutar napuštene fabrike, elementi koji čine fabriku su raspoređeni tako da usmeravaju igračovo kretanje. Na ovaj način igrač mora da savlada i testira sve komande neophodne za kretanja. Prilaskom prvoj kapiji neophodno je da je otvori, sa tim ostvaruje prvu interakciju. U drugom delu mape nalazi se poligon unutar kog će savladati osnove pucanja i korišćenja oružja. Naučiće kako se menjaju puške, prikuplja municija, nišani i pogađa. Prelaskom ovog nivoa igrač je spreman za kompleksnije nivoe i borbu sa zombijima.

U ovom tekstu biće prikazana metodologija rada pri izradi nivoa kao i implementacije osnovnih elemenata FPS video igre, takođe napraviće se i mali osvrt na teorijski deo vezan za razvoj i dizajn nivoa.

2. TEORIJSKI DEO

Često se tvrdi da je dizajn nivoa podskup dizajna video igara. Mnogi smatraju da dizajn video igara “prevazilazi” dizajn nivoa, te da je rad na dizajnu nivoa na neki način podređen ili jednostavniji od dizajna igara. Sve ove tvrdnje izostavljaju važan aspekt dizajna nivoa: ne treba potcenjivati koliki uticaj dizajn nivoa ima na igru.

Često se spominje kako “loš dizajn nivoa može uništiti dobru video igru”, nasuprot tome možemo takođe reći da “loša video igra se ne može spasiti dobrim dizajnom nivoa”. Upravo sa ove dve rečenice možemo postaviti dizajn nivoa i dizajn video igre na donekle jednaku osnovu, barem u smislu uticaja na razvoj same video igre. Jasno je da dizajn video igara i dizajn nivoa nisu ista disciplina, međutim oni očigledno ne mogu postojati jedno bez drugog.

Svaki tok igre je vođen određenim ciljem koji igrač mora dostići. Da bi oni bili implementirani unutar samog nivoa, dizajneru nivoa moraju biti predstavljeni logični izvori ili razlozi zašto baš to mora biti cilj datog nivoa. Ovi razlozi se uglavnom mogu pronaći unutar unutrašnjih ciljeva igre.

Unutrašnji ciljevi igre su veoma slični spoljnim ciljevima, oni opisuju ciljeve visokog nivoa iz kojih možemo izvesti zahteve samog toka igre. Neki od unutrašnjih ciljeva koji su najšestice prisutni unutar video igara su:

- ojačati igrača;
- naučiti igrača kako da se zabavi uz igru;
- nagraditi igrača za istraživanje;
- obezbediti zaraznu i zabavnu igru.

Struktura i metodologija nivoa su veoma važni i mogu imati veliki uticaj na kreiranje samog nivoa. U nastavku ćemo se osvrnuti i na strukturu i metodologiju paralelno jer jedno na drugo imaju veliki uticaj.

Kada donesemo odluku o sadržaju koji će nivo posedovati, moramo takođe odlučiti i koju ćemo strukturu primeniti na nivou. U većini slučajeva ova odluka se donosi prilikom dizajna same video igre, ali je vrlo često da su neki izbori prepušteni dizajneru nivoa, barem u manjem obimu unutar samih nivoa. Ovakva struktura predstavlja tok igre što je u velikoj meri u rukama dizajnera nivoa.

Najčešće se koriste tri tipa strukture:

1. linearna,
2. polu-linearna,
3. nelinearna.

Linearni dizajn nivoa, kao što mu i samo ime kaže, jeste dizajn unutar kog se događaji odvijaju linijski jedan za drugim i postavljeni su tako da ih igrač mora pratiti.

Kada je u pitanju nelinearna struktura redosled radnji u samoj video igri je u potpunosti prepušten igraču. Video igre sa čistom nelinearnom strukturom su zaista retke, ali postoji veliki broj video igara koje u svojim pojedinim segmentima imaju nelinearnu strukturu.

Kao prelazno rešenje između linearne i nelinearne strukture pojavila se polu-linearna struktura. Ova struktura predstavlja određenu hibridnu formu koja omogućava igračima da sami donesu odluku o pravcu svog kretanja ali opet uz određenu dozu usmerenosti. To znači da igrači mogu sami da izaberu stazu po kojoj će se kretati ili zadatak koji žele da izvrše, ali će bez obzira na odabir biti dovedeni do istog kraja ili prelaza ka sledećem nivou.

3. PUCAČINA IZ PRVOG LICA

Pucačine iz prvog lica (*FPS* igre) su video igre u potpunosti orijentisane na borbu. Igrač se bori protiv drugih likova ili igrača, ukoliko je u pitanju online igra, koristeći razne vrste oružja za blisku borbu. Kamera pomoću koje posmatramo ovaj prostor je postavljena tako da imamo utisak da smo upravo mi taj igrač. *FPS* igre predstavljaju najpopularniji žanr među komercijalnim video igrama, sa velikim brojem objavljenih naslova prilagođenih svim platformama. Ovaj tip video igara je veoma zahtevan kada je u pitanju procesorska snaga potrebna za prikazivanje realističnih *3D* okruženja, stoga se smatraju glavnim pokretačem razvoja personalnih računara i konzola.

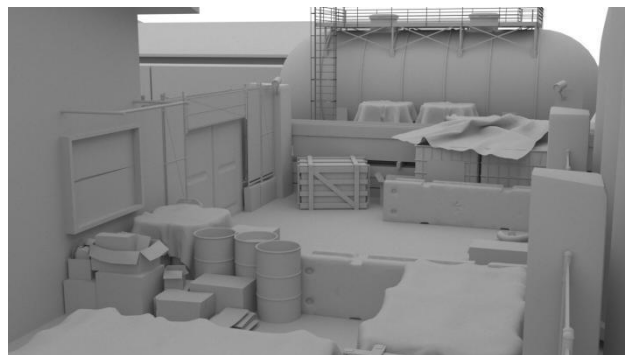
Pored zabave, *FPS* igre su pronašle i edukativnu ulogu u svakodnevnom životu. Od početka 21. veka sa napretkom i razvojem tehnologije *FPS* igre su počele da se koriste za simulacije i treninge za vojne potrebe. Jedna od najznačajnijih *FPS* aplikacija jeste *America's Army*, koja je korišćena za obuku i regrutaciju američke vojske. Njena osnovna namena bila je da obezbedi realističnu simulaciju za upoznavanje regruta sa savremenim vojnim procedurama. Kada su u pitanju nivoi za *FPS* video igre oni su dizajnirani ili za igru sa jednim ili sa više igrača, ali nikako za oboje. Nivoi za jednog igrača imaju tendenciju da budu linearni niz izazova koje igrač mora da savlada

da bi postigao konačni cilj, dok su nivoi za više igrača dizajnirani tako da stvore oblasti za borbu u kojima će se ovi igrači susresti. Prilikom samog procesa dizajniranja nivoa postoji mnogo faktora koje dizajneri moraju uzeti u obzir uključujući izazove, tempo i lakoću navigacije.

U nastavku ovog rada biće detaljno prikazan princip kreiranja nivoa video igre.

4. PRAKTIČAN DEO

Prilikom modelovanja vođeno je računa o geometriji. Svi modeli su rađeni sa četvorougaoanom mrežom poligona, iako *Unreal Engine* pretvara četvorouglove u trouglove. Triangulacija, a pre toga i manipulacija četvorougaoanih poligona, je mnogo lakša od triangulacije i manipulacije mnogouglova. Detalji na sceni kreirani su kombinacijom raznih alata unutar *Editable Poly*-a. Korišćene su opcije *Insert* za uvlačenje ivica, *Chamfer* kako bi se izbegle oštre ivice i kako bi se dobio zaobljeni oblik, *Bridge* kako bi se povezale dve ivice. Izvršene su i manipulacije nad verteksima kako bi se dobili željeni oblici. Pošto scena predstavlja napuštenu fabriku, dominantni materijali su metal, lim i beton. Nivo detalja na njima nije velik. Složeniji objekti su modelovani iz više delova. Ukoliko je to bilo neophodno, više objekata je spajano u jedan. Objekti su namešteni tako da im se po jedan poligon preklapa. Preklopljeni poligoni su obrisani, a svi objekti *Attach*-ovani u jedan objekat. Opcijom *Target Weld* spojili smo vertekse kako ne bi postojale rupe između objekata. Konačan izgled same scene, obogaćene *asset*-ima može se videti ispod teksta. Prilikom samog modelovanja, vodilo se računa o broju poligona, svi suvišni poligoni su uklanjani. Objekti koji su viši i dalji od igrača imaju manji broj poligona kao i detalja.



Slika 1. Izmodelovana scena unutar *3DS Max*-a

Teksture se kreiraju pomoću softvera *Substance Painter* i *Adobe Photoshop CC 2019*. Teksture predstavljaju slike koje nose informacije o materijalima. Svaki materijal se može sastojati iz više mapa. Finalni materijali su kreirani i postavljeni na modele unutar *Unreal Engine*-a, dok su unutar *3ds Max*-a postavljeni materijali kako bi se olakšao rad unutar *Substance Painter*-a i proverio izgled istih. Postavljanjem materijala i dodavanjem *ID*-a objektima unutar *3ds Max*-a, svi objekti sa istim *ID*-om dobijaju istu teksturu, na ovaj način je ubrzan proces postavljanja materijala unutar *Game Engine*-a. Na samom kraju modelovane i teksturisane objekte neophodno je uvesti unutar *Unreal Engine*-a. Uvezeni objekti se ne nalaze na sceni sve dok ih ne prevučemo iz foldera u kojima se

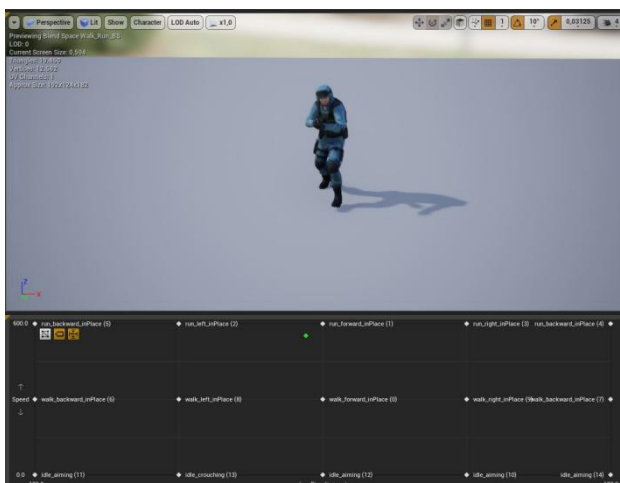
skladište direktno u *Viewport* scene. Svi objekti koji se nalaze na sceni mogu se pomerati, rotirati ili skalirati u zavisnosti od toga kako želimo da scena izgleda. Podloga za scenu je kreirana pomoću *Landscape*-a unutar *Unreal Engine*-a. Ova vrsta podloge opcijom *Sculpt* može da se prilagodi potrebama nivoa, što je učinjeno i unutar našeg nivoa.



Slika 2. Izgled scene unutar *Unreal Engine*-a

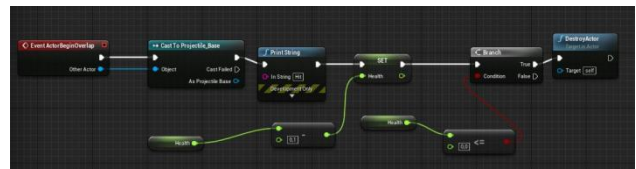
Nakon što je kreirana scena, sredeno je i implementirano kretanje igrača, interfejs i dodati su ostali elementi koji odlikuju *FPS* video igre. Kako bi izgled igrača odgovarao tematici igre i izgledu nivoa, preuzet je karakter kao i animacije sa sajta *Mixamo*. Karakter i animacije su potom uvezene u foldere unutar *ThirdPersonBP*-a. Neophodno je sve ove animacije kao i karakter spojiti unutar jednog *Blueprint*-a, zato je kreiran *Animation Blueprint* iz odeoka *Animation*.

Unutar ovog *Blueprint*-a kreirali smo *State Machine* kako bismo pravilno prelazili između animacija. Da bi animacija tekla kako treba kreiran je *Bland Space* koji se nalazi unutar *Animation* menija. On nam služi da se naprave fini prelazi između kretanja levo/desno, napred/nazad i stanja mirovanja. Animacije su raspoređene po šemi koja se može videti na slici ispod.



Slika 3. Šema animacija kretanja igrača

Nakon sredenog kretanja, dodato je oružje koje karakter ima na raspolaganju. Dodate su dve puške i regulisan je njihov rad. Puške su povezane direktno sa rukom karaktera pomoću opcije *Add Socket*, a potom su pozicionirane tako da odgovaraju animaciji kretanja.



Slika 4. Nodovi koji regulišu zdravlje protivnika

Kao i u svakoj video igri, igrač se mora susresti sa određenim preprekama. Često su te prepreke u vidu barijera na samom terenu, ali one mogu biti i u vidu drugih igrača. Kako je teško izvodljivo kreirati *online* igru, odlučeno je da se kreiraju *AI (Artificial Intelligence)* protivnici, protivnici sa veštačkom inteligencijom, koji će konstantno napadati igrača. Svi protivnici će biti istog izgleda, u vidu zombija. Kada ih igrač pogodi sa određenim brojem metaka oni će nestati.

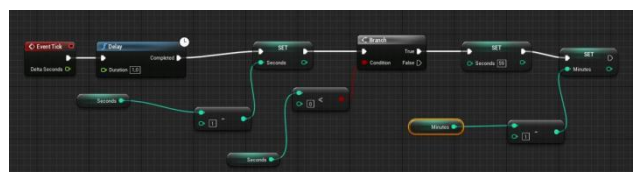
Za kreiranje protivnika napravljena je nova *Blueprint Class*-a u vidu *Character*-a unutar koje je postavljen *Mesh* zombija. U *Event Graph*-u ove klase postavljeni su nodovi pomoću kojih je definisano koliko puta igrač mora pogoditi zombija da bi on nestao sa scene. Protivniku (zombiju) je pomoću *float* varijable dodeljeno zdravlje. Svaki metak koji pogodi zombija smanjiće mu zdravlje za 10% sve do momenta dok zdravlje ne bude imalo vrednost 0, tada će zombi nestati. Kako bi sama igra bila zanimljivija, protivnici konstantno napadaju igrača.

Interfejs je izrazito bitan tokom igranja same igre. On igraču daje osnovne informacije o tome koliko metaka ima na raspolaganju, koliko je zdravlje njegovog igrača, koji zadatak mora da ispuni ili u kom delu mape se njegov igrač nalazi. Upravo zbog toga se mora voditi računa i o dizajnu samog interfejsa.

Da bi igrač u svakom trenutku znao gde se nalazi implementirali smo i minimapu koja se nalazi u gornjem desnom uglu interfejsa. Za implementaciju minimape neophodno je dodati još jednu kameru koja će snimati igrača od gore (*TOP* pogled) i praviti *Scene Capture*.

Unutar interfejsa prikazano je i vreme koje igrač ima na raspolaganju kako bi savladao nivo. Na ovaj način dodatno smo ograničili igrača. Da bismo implementirali tajmer neophodno je unutar interfejsa postaviti dva *text box*-a, jedan koji će ilustrovati minute i drugi koji će prikazivati sekunde.

Za svaki od ovih *text box*-ova neophodno je kreirati varijablu tipa *Integer* unutar *ThirdPersonGameMod*-a. Nakon kreiranja varijabli neophodno je napraviti *Blueprint* koji će omogućiti funkcionisanje ovog tajmera. *Blueprint* je osmišljen tako da smanjuje sekunde, potom kada dođe do 0 sekundi, broj minuta smanji za jedan.



Slika 5. Implementacija tajmera pomoću nodova

5. ZAKLJUČAK

Video igre su sastavni deo budućnosti. Osim za zabavu koriste se u edukaciji, medicini, arhitekturi, čak i kao simulatori za vojne potrebe. Kroz razvoj ovog projekta prošireno je znanje kada je u pitanju dizajn nivoa za video igre i stečeno je novo znanje u *Node Base* programiranju unutar *Unreal Engine*-a. Kreirana je osnova za igru koja može da ima velik potencijal i uz razvoj novih nivoa, različite složenosti, može da se probije na tržištu.

6. LITERATURA

[1] An Architectural Approach To Level Design, Christopher W. Totten, George Mason University, 2014.

[2] Beginning Game Level Design, John Feil and Marc Scattergood, Thomson Couse Technology PTR, 2005.

[3] Game Development Essentials, Jeannie Novak, Delmar, 2012.

[4] Level Design Concept, Theory and Practice, Rudolf Kremers, 2009.

Kratka biografija:



Milan Mišćević rođen je 29.11.1997. godine u Osijeku, Republika Hrvatska. Završio je osnovnu školu „Zmaj Jova Jovanović“ u Rumi, potom gimnaziju „Stevan Puzić“ takođe u Rumi. Školske 2016/2017 upisuje osnovne akademske studije Animacija u inženjerstvu na Departmanu za Računarsku grafiku, Fakulteta tehničkih nauka u Novom Sadu. Diplomirao je 2020. godine i od tada je zaposlen na Fakultetu tehničkih nauka kao saradnik u nastavi na Departmanu za opšte discipline u tehnici, uža naučna oblast Računarska grafika.