

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD BR. 6254

**Računalna igra s posluživanjem na temelju
podataka stvarnog sustava**

Nina Zuccon

Zagreb, lipanj 2019.

Zagreb, 13. ožujka 2019.

ZAVRŠNI ZADATAK br. 6254

Pristupnik: **Nina Zuccon (0036498292)**
Studij: Računarstvo
Modul: Programsko inženjerstvo i informacijski sustavi

Zadatak: **Računalna igra s posluživanjem na temelju podataka stvarnog sustava**

Opis zadatka:

Posluživanje klijenata je čest i složen problem u različitim sustavima. S druge strane, računalne igre mogu vrlo vjerno simulirati situacije iz stvarnog svijeta. Prikazivanjem određenog problema u formi igre, korisnike se može usmjeriti prema rješavanju složenih problema na zabavan način.

Vaš je zadatak istražiti postojeće računalne igre na principu obrane tornja s jednim mogućim resursom za obranu, koji je potrebno optimalno usmjeriti na odgovarajuće događaje. Podatke stvarnog sustava potrebno je obraditi i prilagoditi primjeni u računalnoj igri, čime će se simulirati intenzitet napada. Na temelju istraživanja implementirajte računalnu igru u kojoj će korisnici braniti toranj optimalnom strategijom posluživanja.

Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 15. ožujka 2019.
Rok za predaju rada: 14. lipnja 2019.

Mentor:



Doc. dr. sc. Marin Vuković

Djelovođa:



Doc. dr. sc. Mirjana Domazet-Lošo

Predsjednik odbora za
završni rad modula:



Izv. prof. dr. sc. Ivica Botički

*Zahvaljujem mentoru doc. dr. sc. Marinu Vukoviću na savjetima
prilikom pisanja ovog rada.*

SADRŽAJ

1. Uvod.....	1
2. Računalne igre temeljene na stvarnim podacima	2
3. Računalna igra FERpocalypse	4
3.1. Implementacija igre.....	5
3.1.1. Korištene tehnologije i alati.....	5
3.1.2. Klasa Game Manager.....	5
3.1.3. Klasa Portal	7
3.1.4. Podaci korišteni u igri.....	8
3.1.5. Problemi s podacima	9
3.1.6. Grafički elementi u igri	10
3.1.7. Scene u igri.....	11
3.1.8. Izlazni rezultati	15
4. Testiranje igre	16
5. Zaključak	19

1. Uvod

Posluživanje klijenata u složenim sustavima čest je problem današnjice. Zahtijeva dobru strategiju posluživanja koju je nerijetko teško postići. Jedan od načina testiranja raznih strategija i stvaranja novih je implementacija stvarnih sustava u računalne igre. Računalne igre razvijene na temelju podataka iz stvarnog svijeta pružaju mnogo mogućnosti. Osim što poboljšavaju doživljaj samog igrača, koji kroz igru „živi“ u simuliranom stvarnom svijetu, dobar su temelj razvoja rješenja modernih problema. Praćenjem korisnika aplikacije i prikupljanjem podataka o njihovim postupcima prilikom igre, može se izgraditi optimalno ili barem optimalnom najbliže rješenje simuliranog problema. Katkad bi za pronalazak odgovarajućeg algoritma u stvarnosti bilo potrebno osigurati mnoštvo resursa. Preslikavanjem stvarnosti u virtualan svijet gdje su mogućnosti bezbrojne u samo nekoliko klikova, pruža nam se uvid u korisnikovu logiku te nam je omogućeno jednostavno prikupljanje statističkih podataka. Računalne igre često se percipiraju samo kao oblik razonode i zabave, dok se rješavanje problema percipira kao kompliciran posao. Spajanjem razonode i posla može se postići učinkovitiji rad uz opušteniji pristup. *Entertainment Software Association* kaže da se godišnje više od 200 milijuna sati svakog dana provede igrajući računalne igre. Prosječan Amerikanac u dobi do 21. godine provede više od deset tisuća sati igrajući računalne igre. [1] Uz toliko uloženog vremena, računalne igre sa stvarnim podacima bi mogle prikupiti mnoštvo podataka u malo vremena. Osim toga, stvaran svijet daje ograničene mogućnosti. Primjerice, ispitivanje pojedinaca oduzima puno vremena i ograničeno je na neki skup ljudi. Možda su to ljudi na istom radnom mjestu, fakultetu, ljudi koji žive u istom gradu i sl. U virtualnom svijetu, spektar ispitnih grupa mnogo je širi, što omogućuje i širi spektar rezultata, više novih ideja i inovativnih rješenja. Računalne igre temeljene na stvarnim podacima pružaju i mogućnost bezbrojnih simulacija s izmijenjenim parametrima koje kao rezultat daju različite ishode s mogućnošću odabira najboljeg. Konkretna problem na kojem bi se mogla primijeniti ovakva tehnologija bila bi preopterećenost nekih ulaza u zgradu.

2. Računalne igre temeljene na stvarnim podacima

Danas ideja korištenja stvarnih podataka u virtualnom svijetu postaje sve raširenija. Jedan od najboljih načina motivacije čovjeka je pretvaranje danog zadatka u natjecanje. Upravo to omogućuju znanstvene računalne igre koje kroz zabavu korisnika motiviraju na rješavanje nekog složenog problema. Luis von Ahn prvi je predložio ideju igara sa svrhom (*games with a purpose* – GWAP [2]) kako bi iskoristio ljude za pomoć računalima u problemima koje ona sama ne mogu riješiti. On vjeruje da je ljudski intelekt bitan resurs za poboljšanje procesiranja računala. GWAP se opisuje kao podjela posla između ljudi i računala, a koristi se u mnogim računalnim igrama današnjice. Primjerice, igra ESP [2], u kojoj korisnik povezuje slike s ključnim riječima, mogla bi, uz dovoljno velik broj igrača, u nekoliko mjeseci povezati sve Google slike s njihovim ključnim riječima i to koristiti za unaprjeđenje algoritama za pretragu slika i filtriranje neprimjerenog sadržaja [3]. Još jedan primjer takve igre je i Peek-a-Boo [2] koji pomaže računalima locirati objekte u slikama koristeći metapodatke prikupljene tijekom igre korisnika. Zanimljive su i igre koje se koriste u svrhu medicinskih istraživanja. Takva je *EteRNA* [2], igra u kojoj igrači pokušavaju dizajnirati RNA slažući nukleotide. Širok spektar rješenja igrača analizira se u svrhu poboljšanja računalnih modela RNA. Slično funkcionira i *Phylo*[2] koji omogućuje igračima da pokušaju dekodirati kod za genetičke bolesti. Među najpoznatijima je i *Wikidata Game* [2] koja kroz igru daje korisnicima pitanja o osobama, slikama i sl. te na taj način uređuje i odgovarajuće podatke u Wikidata, strukturiranom repozitoriju informacija koje je podrška Wikipediji. Osim toga, valja navesti i *Sea Hero Quest* [4] koji pomoću ljudske prostorne navigacije u dvije minute igranja prikupi informacije o demenciji koje bi znanstvenici klasičnim istraživanjem sakupljali pet sati. Napredna rješenja zahtijevaju mogućnost podrške kompleksnom donošenju odluka na stvarnim i kontinuirano mijenjanim podacima pa je to područje koje je još uvijek u fazi razvoja.

Osim računalnih igara sa stvarnim podacima koje se koriste u svrhu znanstvenih istraživanja, vrlo su popularne i one kojima je svrha zabava i razbibriga. Sve

popularnije postaju igre koje simuliraju stvarna mjesta koristeći se Google Maps API-jem čime se omogućuje bogato iskustvo korisnika. U samo jednom danu moguće je virtualno posjetiti bezbroj različitih mjesta. Osim toga, Google Maps API koristi se i kod *tower defense* igara sa stvarnim mapama, igara za vježbanje sposobnosti uparkiravanja, raspoznavanja poznatih gradova i sl. [5]

Igre sa svrhom imaju doista širok spektar primjene, od sigurnosti, računalnog vida, filtriranja sadržaja, pretraživanja Interneta pa do učenja jezika, medicinskih istraživanja, umjetnosti, genetike i dr. [6]

3. Računalna igra FERpocalypse

FERpocalypse, računalna igra implementirana u sklopu ovog rada, temelji se na posluživanju podataka iz stvarnog sustava – zapisima ulaska ljudi na Fakultet elektrotehnike i računarstva u Zagrebu, u nastavku FER. Korisnik izabire datum koji želi simulirati nakon čega se pri instanciranju razine zapisi učitavaju u igru. Na temelju njih kreiraju se čudovišta (zombiji) koji predstavljaju ulaske ljudi na FER skalirane na vrijeme trajanja razine. Svaka razina predstavlja jedan dan iz danog testnog skupa. Čudovišta se u igri generiraju na šest različitih ulaza (portala). Nakon toga počinju se kretati po svojim putanjama prema središnjem tornju, odnosno pokušavaju ga napasti. Igrač sa središnjeg tornja napada čudovišta s ciljem da ih spriječi da unište toranj, odnosno da obrani FER. Prevođenjem stvarnosti u igru implementiran je problem posluživanja klijenata (u igri: čudovišta) u stvarnom sustavu (FER). Dok korisnik igra svoje poteze, oni se zapisuju u vanjsku datoteku, odnosno pri kraju svake razine, tj. dana u igri, datoteka sadrži zapis o igračevim potezima. Prati se u kojem je trenutku korisnik poslužio kojeg klijenta, u smislu igre prati se u kojem je trenutku napao čudovište s kojeg ulaza. Osim poteza, igra pamti i najbolje rezultate pri svakoj razini kako bi korisnik mogao optimizirati svoje napade, poboljšati svoju efikasnost i pratiti napredak. Daljnjom analizom rezultata korisnika, uspoređivanjem najboljih rezultata igrača s odigranim potezima na pripadnoj razini, može se definirati optimalna strategija posluživanja klijenata u FER-ovom sustavu. Osim toga, igrajući FERpocalypse može se uočiti i kada su najveće gužve te koji su ulazi preopterećeni.



Slika 1: Isječak iz računalne igre FERpocalypse

3.1. Implementacija igre

3.1.1. Korištene tehnologije i alati

Računalna igra FERpocalypse dizajnirana je u radnoj okolini Unity [7] koja omogućuje interakciju s aplikacijom ne samo putem koda, već i putem vizualnih komponenti. Elementi se jednostavno prenose i pozicioniraju na scenu, uz mogućnosti animacija i sličnih dodataka koje pruža Unity. Programski kod pisan je u objektno-orijentiranom jeziku C#.

3.1.2. Klasa Game Manager

Klasa GameManager upravlja razinom i sadrži neke od glavnih funkcija igre. Funkcija Update(), koja se poziva jednom po frameu, provjerava je li korisnik kliknuo mišem na mapu te, ako jest, predaje odredište metka funkciji SpawnBullet().

```
void Update()
{

    totalMonsters.text = "Total monsters: " + MonsterNumber;
    LastMonstersTime.text = "Last monster's real time: " + lastTime;
    if (Input.GetMouseButton(0))
    {

        Destination = Input.mousePosition;
        Destination.z = 10.0f;
        Destination = Camera.main.ScreenToWorldPoint(Destination);
        if (Time.time - shooterTimer > 0.1f) SpawnBullet(Destination);
        shooterTimer = Time.time;
    }
    if (MonsterNumber<=0 && Time.timeSinceLevelLoad>2.0f) GameOver();
}
```

Funkcija SpawnBullet() instancira novi metak te mu postavlja početnu destinaciju na mjesto gdje je toranj. U varijablu klase Projectile sprema krajnju destinaciju metka kako bi ju funkcija Move() klase Projectile, koja pokreće metak, mogla koristiti. Metak se dodaje u listu activeProjectiles. Ona sadrži informacije o

aktivnim metcima kako bi se omogućilo ponovno korištenje istih objekata (GameObject) za nove metke, ako je neki stariji metak neaktivan.

```
public void SpawnBullet (Vector3 dest)
{
    string type = "Bullet";
    Projectile projectile = Pool.GetObject(type).GetComponent<Projectile>();
    projectile.transform.position =
    LevelManager.Instance.EndPort.transform.position;
    projectile.Destination = dest;
    activeProjectiles.Add(projectile);
}
```

Funkcija SpawnMonster() kao parametar prima šifru ulaza, koju prenosi u funkciju Spawn() klase Monster koja po dobivenoj šifri smješta čudovište na pripadni portal (ulaz) i generira putanju nastalog čudovišta.

```
public void SpawnMonster (int id)
{
    string type = "Zombie";
    Monster monster = Pool.GetObject(type).GetComponent<Monster>();
    monster.Spawn(id);
    activeMonsters.Add(monster);
}
```

3.1.3. Klasa Portal

Klasa Portal pridijeljena je svakom portalu i odgovorna je za upravljanje vremenom tijekom kojeg će se čudovišta stvarati na portalima. Lista grupniStart sadrži vremena prolaska ljudi kroz određeni ulaz za dan koji je igrač označio da želi učitati. Lista je sortirana uzlazno, dakle prvi zapis u listi je prvi prolazak osobe kroz neka vrata u tom danu. Provjerava se je li vrijeme proteklo od početka učitavanja razine veće od prvog zapisa u listi te se ako jest, generira putanja i stvara čudovište. Prije te provjere, vrijeme u listi dijeli se s 360 kako bi stvarnih 24 sata bilo skalirano na četiri minute u razini igre. Prilikom svakog stvaranja čudovišta ažurira se zapis na ekranu sa stvarnim vremenom u danu koje to čudovište predstavlja. Prije izlaska iz funkcije prvi zapis iz liste se uklanja kako bi za iduću provjeru bilo spremno sljedeće čudovište koje se treba generirati.

```
void Update()
{
    int br = 0;
    if (grupniStart.Count!=0)
    {
        if((Time.time)>lastZombie )
        {
            if (Time.timeSinceLevelLoad > grupniStart[0] / 360)
            {
                lastZombie = Time.time;
                LevelManager.Instance.GeneratePath(id);

                GameObject.Find("GameManager").GetComponent<GameManager>().
                SpawnMonster(id);
                t = TimeSpan.FromSeconds(grupniStart[0]);

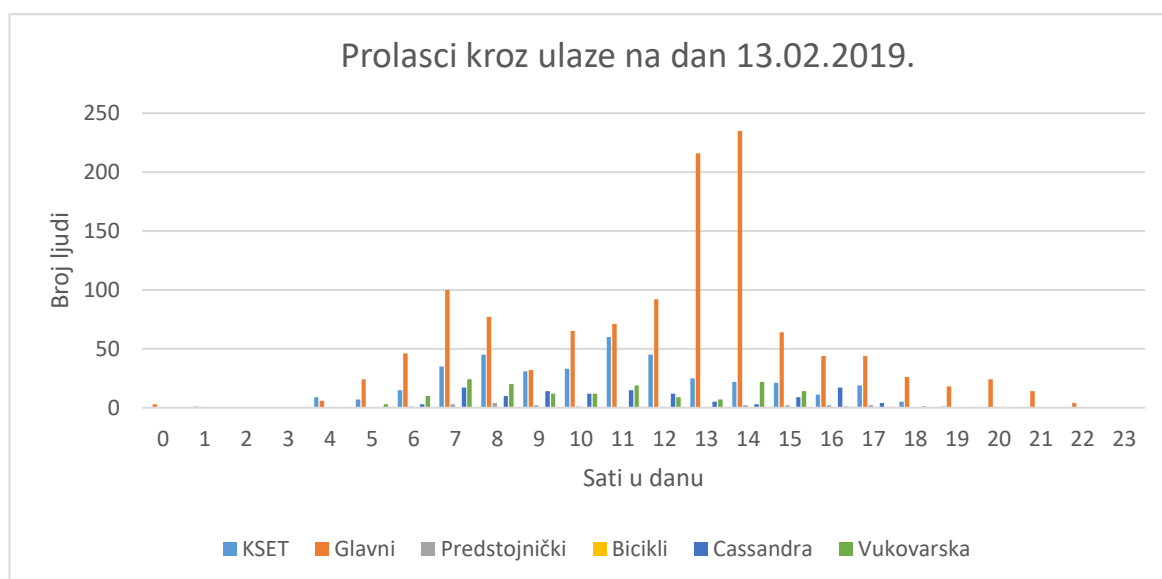
                string answer = string.Format("{0:D2}h:{1:D2}m:{2:D2}s",
                                                t.Hours,
                                                t.Minutes,
                                                t.Seconds);

                GameManager.Instance.LastTime = answer;

                grupniStart.Remove(grupniStart[0]);
            }
        }
    }
}
```

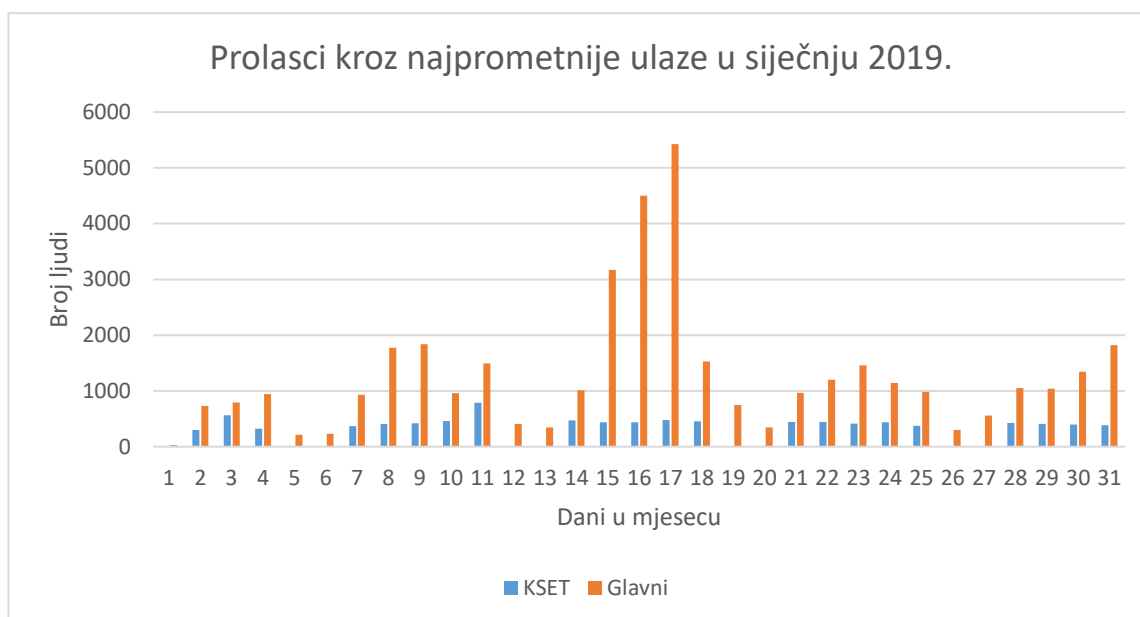
3.1.4. Podaci korišteni u igri

Stvarni podaci na temelju kojih je implementirana aplikacija podijeljeni su u šest tekstualnih dokumenata gdje svaki sadrži zapise s jednog od šest ulaza na FER: glavni ulaz, ulaz kod Kluba studenata elektrotehnike, predstojnički, ulaz kod bicikala, ulaz kod restorana Cassandra i ulaz s Ulice grada Vukovara. Zapisi su uzeti u rasponu od 12.12.2018. do 14.2.2019. Izazov je bio riješiti problem masovnih dolazaka i dugih razdoblja kada na ulaz nitko nije ulazio. Zato je igrice skalirana kako bi jedan dan u igri trajao nešto više od minute, no uz dodatno usporavanje izlazaka čudovišta pri masovnim dolascima kako bi korisnik ipak bio u mogućnosti obraniti toranj. Na grafu ispod, prikazani su prolasci ljudi kroz ulaz na datum 13.02.2019. kroz svih šest vrata, na svaki sat u danu. Na grafu je lako uočljivo kada su najveće gužve i koji su ulazi najopterećeniji.



Slika 2: Graf prolazaka kroz ulaze FER-a na dan 13.02.2019.

Analizom podataka kroz mjesec uočava se da je radnim danima najveća prometnost srijedom i četvrtkom, zanemarujući nagle skokove pri dodatnim događanjima na FER-u.



Slika 3: Graf prolazaka kroz najprometnije ulaze u siječnju 2019.

3.1.5. Problemi s podacima

Kako je vidljivo iz grafa na slici 2, velika su variranja između pojedinih perioda u danu. Najviše ljudi dolazi oko 13 i 14 sati, a nakon toga oko 7 i 8 sati. Mala je dolaznost predvečer i u noćnim satima. Kod realizacije igre problem su bili masovni dolasci koji bi preplavili igrača ukidajući mu bilo kakvu mogućnost za pobjedom. Također, vrijeme kada bi na FER dolazilo manje ljudi, u igri je rezultiralo dugim čekanjem zbog nepojavljivanja novih čudovišta. Kako bi se taj problem sanirao, vrijeme u igri skalira se kako bi se ona izvodila brže, a kod masovnih ulazaka na FER igrač može istim metkom eliminirati čudovišta koja su na identičnim pozicijama kako bi uspio nadjačati masu.

Na slici 3 vidljiv je skok 17.01.2019. kada je na FER-u bilo više od 5000 ljudi, dok primjerice, subotom i nedjeljom u većini slučajeva ne bude više od 500 prolazaka dnevno što rezultira nešto dužim vremenom između stvaranja čudovišta u igri.

3.1.6. Grafički elementi u igri

Sastavni elementi igre su pločice koje čine mapu po kojoj se čudovišta kreću, toranj s kojeg igrač ispaljuje metke, portali, odnosno ulazi s kojih čudovišta dolaze te sama čudovišta. Zelena pločica [8] predstavlja travu, odnosno područje izvan FER-a. Tamno siva [9] predstavlja unutrašnjost FER-a, kao i svjetlo siva [8] koja dodatno označava putanju čudovišta. Zombiji [10] predstavljaju čudovišta u igri koja napadaju FER. Toranj [11] predstavlja igračevo oružje, a portali [12] vrata FER-a. Grafički elementi licencirani su Creative Commons BY 3.0 licencom. [13]



Slika 4: Glavni grafički elementi u igri

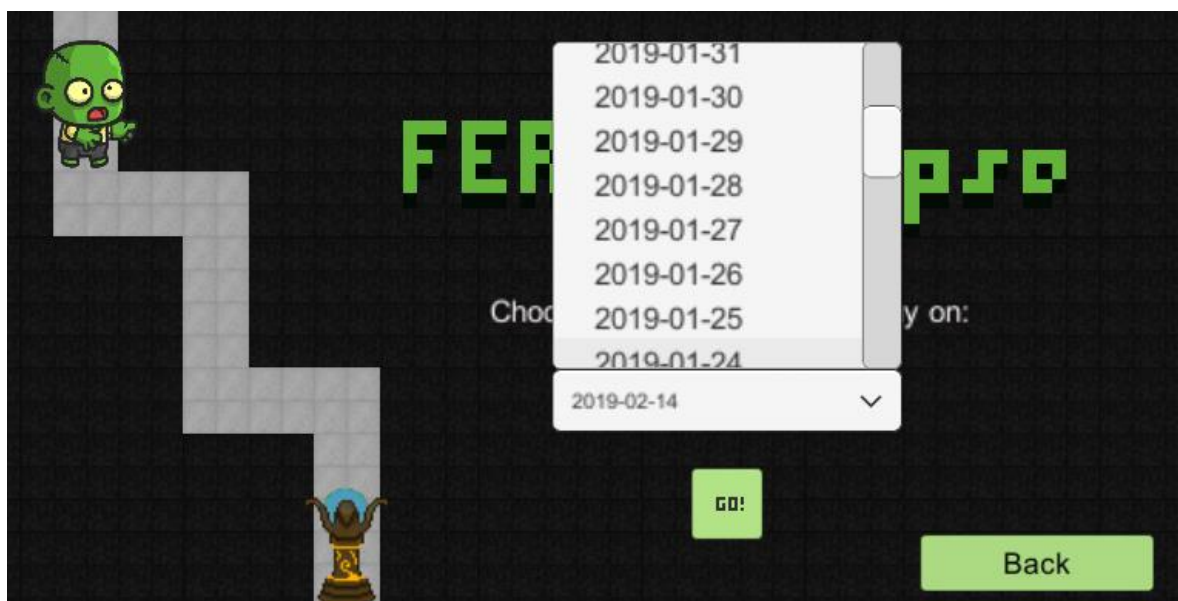
3.1.7. Scene u igri

Igra se sastoji od ukupno pet scena. Prva je scena početnog ekrana koja se sastoji od četiri gumba koji nude različite opcije korisniku.



Slika 5: Početna scena

Klikom na gumb „Play“ korisniku se učitava nova scena. Iz padajućeg izbornika igrač može odabrati datum koji želi učitati u razinu igre.



Slika 6: Scena biranja razine

Nakon što odabere datum, korisnik klikom na gumb „Go!“ učitava razinu ili, ako se predomisli, klikom na gumb „Back“ može se vratiti na početni ekran.



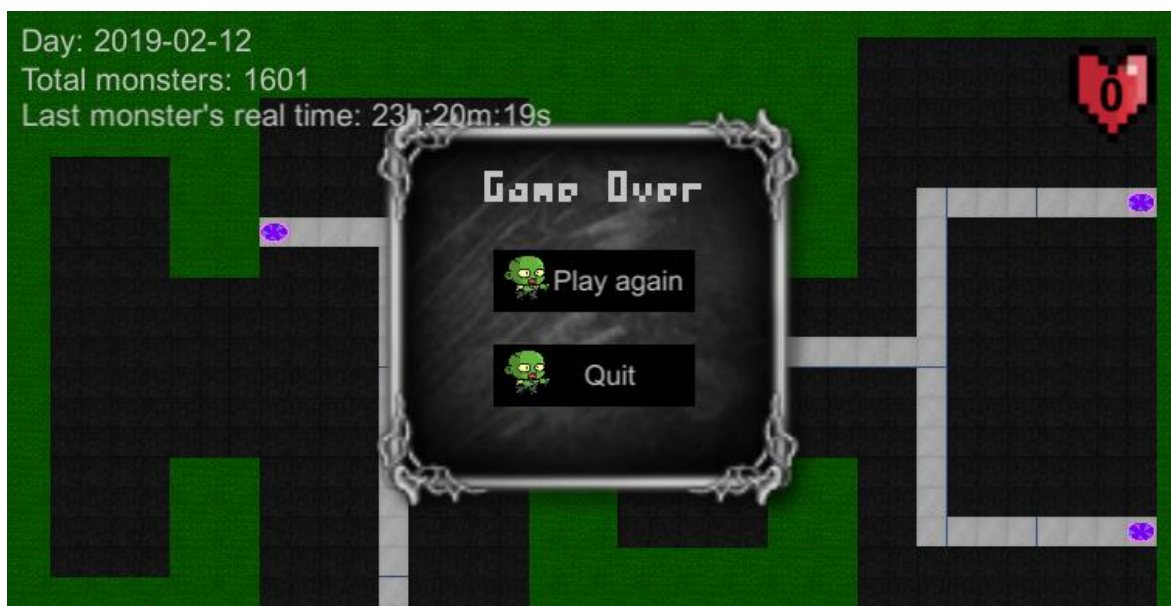
Slika 7: Scena razine

Pri učitavanju razine stvaraju se čudovišta po uzoru na dolaske ljudi na FER izabranog datuma. Čudovišta se kreću po svijetlo sivoj putanji prema tornju. Lijevim klikom miša igrač ispaljuje metke kojima nastoji spriječiti čudovišta da osvoje njegov toranj i time cijeli FER. U gornjem lijevom uglu igrač može vidjeti na kojem danu igra. Odmah ispod ispisan je brojač koji prikazuje koliko je čudovišta ostalo do kraja razine, odnosno do kraja dana. Ispod brojača ispisuje se stvarno vrijeme posljednje stvorenog čudovišta kako bi igrač mogao znati koje vrijeme u danu se trenutno simulira. Krajnji rezultat razine jednak je broju čudovišta koje je korisnik uspio eliminirati tijekom igre. Na početku razine igrač ima petnaest života, a gubi život ako čudovište dođe do tornja prije no što ga on eliminira. Tijekom igre, trenutni se broj života može pratiti u gornjem desnom uglu ekrana. Igra završava kada igrač eliminira sva čudovišta na razini ili kada ostane bez života te se njegov rezultat, ako je eliminirao više čudovišta nego do tad na toj razini, sprema u listu najboljih rezultata.

Po završetku razine na ekranu se pojavljuje skočni prozor koji korisniku nudi dvije opcije. Pritiskom gumba „Play again“ igrač se vraća na scenu odabira datuma ili može izaći iz igre pritiskom gumba „Quit“.

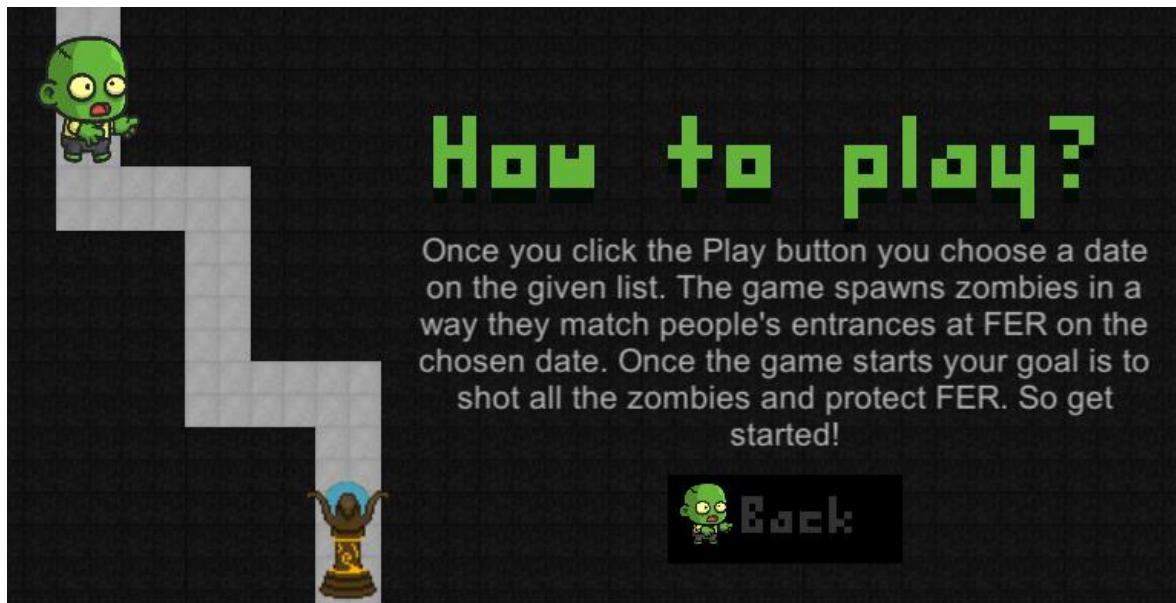


Slika 8: Toranj ispaljuje metak prema čudovištu s glavnog ulaza



Slika 9: Skočni prozor kraja igre

Ako korisnik u glavnom izborniku pritisne gumb „Help“ otvara se nova scena koja ukratko objašnjava ideju i cilj igre nakon čega se korisnik pritiskom gumba „Back“ može vratiti na početnu scenu.



Slika 10: Scena pomoći

U donjem desnom uglu glavnog izbornika korisnik može pritisnuti gumb „High scores“ nakon čega se otvara nova scena s ispisom najboljih rezultata. Prikazan je svaki datum iz skupa ponuđenih uz svoj pripadajući najbolji rezultat, odnosno najveći broj eliminiranih čudovišta postignut na toj razini.



Slika 11: Scena najboljih rezultata

3.1.8. Izlazni rezultati

Dok korisnik igra na nekoj razini, njegovi se potezi zapisuju u vanjsku datoteku. Pri ulasku u razinu zapisuje se da je igra počela, uz datum koji je igrač odabrao te se pri završetku igre ispisuje oznaka kraja kako bi se mogli razlikovati rezultati iz različitih igranja. Kada korisnik pogodi čudovište metkom, u datoteku se zapisuje šifra ulaza s kojeg je čudovište došlo kako bi se mogao identificirati klijent koji je poslužen, odnosno u igri, portal koji je napadnut. Posljednja u retku je oznaka vremena koja označava trenutak u igri u kojem je igrač pogodio čudovište metkom.

```
Started game at level 2019-02-12
Killed monster from gate 3 at 3.58
Killed monster from gate 6 at 4.44
Killed monster from gate 3 at 7.09
Killed monster from gate 6 at 9.77
Killed monster from gate 5 at 12.39
Killed monster from gate 4 at 13.41
Killed monster from gate 6 at 14.24
Killed monster from gate 5 at 17.12
Killed monster from gate 4 at 23.29
Killed monster from gate 1 at 23.89
Killed monster from gate 4 at 25.04
Killed monster from gate 5 at 27.38
Killed monster from gate 1 at 30.74
Killed monster from gate 5 at 31.57
Killed monster from gate 4 at 33.89
Killed monster from gate 1 at 37.82
Killed monster from gate 2 at 38.36
Killed monster from gate 4 at 40.15
Killed monster from gate 5 at 41.55
Killed monster from gate 1 at 43.36
Game ended
```

Slika 12: Primjer ispisa korištene strategije igrača

4. Testiranje igre

Kako bi se proučila raznovrsnost rezultata i kako bi se uočila eventualna podudaranja u logici igrača izabrana je testna skupina od tri sudionika. Svi su igrači igrali na istoj razini (12.02.2019.) te je po završetku njihovih igara analizirana ispisna datoteka.

Sljedeće tri slike prikazuju prvih trideset poteza trojice igrača. Iako su igrači igrali na istoj razini, odnosno čudovišta su nastajala na istim portalima u isto vrijeme, zanimljivo je primijetiti kako su već u početku igranja vidljive razlike u njihovoj strategiji igranja, odnosno ako se problem prenese u stvaran svijet, vidljive su razlike u njihovoj strategiji posluživanja klijenata. Razlike nisu velike, no valja napomenuti kako je za test je izabrana jednostavnija razina, upravo kako bi se vidjeli potezi za koje bi se možda očekivao podjednak algoritam kod većine igrača. Kako je za ovaj test uzeta grupa od tek troje ljudi, može se zaključiti kako bi veći broj igrača dao mnogo različitih rješenja za isti problem, iz kojih bi se na posljetku moglo definirati ono optimalno.

Osim strategije igranja, zanimljiv je i podatak brzine reakcije kod svakog igrača koji se opet razlikuje kod svakog od njih.

Started game at level 2019-02-12
 Killed monster from gate 3 at 5.58
 Killed monster from gate 6 at 7.24
 Killed monster from gate 3 at 9.059999
 Killed monster from gate 6 at 10.56
 Killed monster from gate 5 at 30.38
 Killed monster from gate 4 at 30.66
 Killed monster from gate 1 at 32.56
 Killed monster from gate 5 at 34.02
 Killed monster from gate 4 at 34.44
 Killed monster from gate 1 at 36.5
 Killed monster from gate 5 at 37.78
 Killed monster from gate 4 at 37.9
 Killed monster from gate 1 at 39.94
 Killed monster from gate 4 at 41.94
 Killed monster from gate 5 at 43.1
 Killed monster from gate 1 at 44.36
 Killed monster from gate 2 at 45.2
 Killed monster from gate 4 at 45.68
 Killed monster from gate 5 at 46.68
 Killed monster from gate 1 at 47.98
 Killed monster from gate 2 at 48.86
 Killed monster from gate 4 at 49.62
 Killed monster from gate 5 at 50.68
 Killed monster from gate 1 at 51.78
 Killed monster from gate 2 at 52.54
 Killed monster from gate 4 at 53.02
 Killed monster from gate 5 at 54.1
 Killed monster from gate 1 at 55.5
 Killed monster from gate 2 at 56.82
 Killed monster from gate 5 at 57.64

Slika 13: Ispis korištene strategije prvog igrača

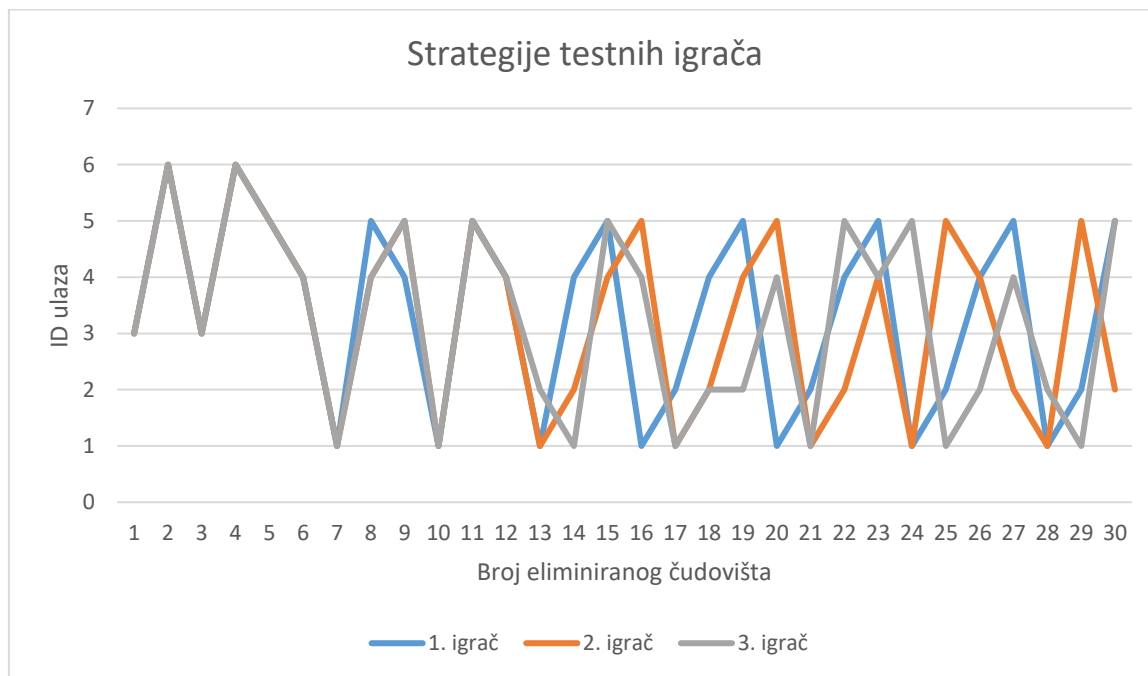
Started game at level 2019-02-12
 Killed monster from gate 3 at 8.24
 Killed monster from gate 6 at 10.24
 Killed monster from gate 3 at 14.04
 Killed monster from gate 6 at 15.82
 Killed monster from gate 5 at 30.82
 Killed monster from gate 4 at 31.16
 Killed monster from gate 1 at 32.72
 Killed monster from gate 4 at 36.82
 Killed monster from gate 5 at 38.26
 Killed monster from gate 1 at 39.7
 Killed monster from gate 5 at 42.98
 Killed monster from gate 4 at 43.42
 Killed monster from gate 2 at 44.62
 Killed monster from gate 1 at 45.36
 Killed monster from gate 5 at 49.22
 Killed monster from gate 4 at 50.14
 Killed monster from gate 1 at 51.6
 Killed monster from gate 2 at 52.3
 Killed monster from gate 2 at 57.24
 Killed monster from gate 4 at 58.1
 Killed monster from gate 1 at 59.96
 Killed monster from gate 5 at 61.26
 Killed monster from gate 4 at 62.3
 Killed monster from gate 5 at 63.92
 Killed monster from gate 1 at 65.34
 Killed monster from gate 2 at 66.26
 Killed monster from gate 4 at 68.22
 Killed monster from gate 2 at 69.64
 Killed monster from gate 1 at 70.7
 Killed monster from gate 5 at 73.08

Slika 15: Ispis korištene strategije trećeg igrača

Started game at level 2019-02-12
 Killed monster from gate 3 at 7.86
 Killed monster from gate 6 at 11.78
 Killed monster from gate 3 at 12.4
 Killed monster from gate 6 at 14.24
 Killed monster from gate 5 at 31.12
 Killed monster from gate 4 at 31.28
 Killed monster from gate 1 at 34
 Killed monster from gate 4 at 36.06
 Killed monster from gate 5 at 37.36
 Killed monster from gate 1 at 38.66
 Killed monster from gate 5 at 40.5
 Killed monster from gate 4 at 41.1
 Killed monster from gate 1 at 42.82
 Killed monster from gate 2 at 44.3
 Killed monster from gate 4 at 46.1
 Killed monster from gate 5 at 47.54
 Killed monster from gate 1 at 48.26
 Killed monster from gate 2 at 48.98
 Killed monster from gate 4 at 50.28
 Killed monster from gate 5 at 51.56
 Killed monster from gate 1 at 52.82
 Killed monster from gate 2 at 55.54
 Killed monster from gate 4 at 56.12
 Killed monster from gate 1 at 58.64
 Killed monster from gate 5 at 60.28
 Killed monster from gate 4 at 60.52
 Killed monster from gate 2 at 61.58
 Killed monster from gate 1 at 62.44
 Killed monster from gate 5 at 64.14
 Killed monster from gate 2 at 64.84

Slika 14: Ispis korištene strategije drugog igrača

Sljedeći graf prikazuje prvih trideset poteza testnih igrača i usporedbu njihovih strategija.



Slika 16: Grafički prikaz strategija testnih igrača

Optimalna strategija mogla bi se definirati analiziranjem poteza igrača koji su uspjeli eliminirati sva čudovišta na razini ili onih koji imaju najbolji rezultat na nekoj razini. Pronalaskom poteza koji se podudaraju kod većine uspješnih igrača mogli bi definirati strategiju koja funkcionira u većini slučajeva.

5. Zaključak

Računalne igre temeljene na stvarnim podacima postaju sve popularnije. Osim načina zabave, predstavljaju i inovativne ideje prema rješenju raznih problema. Protežu se kroz područja medicine, znanosti, umjetnosti i mnoga druga. One predstavljaju suradnju čovjeka i računala, gdje svaki obavlja svoj dio posla, ali i suradnju ljudi koji mogu putem Interneta zajedno raditi na rješavanju nekog problema. U mnoštvu rješenja odabire se ono optimalno. Računalne igre s posluživanjem na temelju stvarnih podataka nekog sustava taj sustav simuliraju i omogućuju igračima da putem interaktivne aplikacije daju prijedloge za problem posluživanja klijenata. U velikim sustavima posluživanje klijenata čest je problem te se ovakvim načinom brže i efikasnije dolazi do novih algoritama.

FERpocalypse je računalna igra napravljena na temelju stvarnih podataka koji predstavljaju ulazak ljudi u Fakultet elektrotehnike i računarstva u određenom periodu vremena. Mapa igre je tlocrt FER-a na čijim se ulazima stvaraju čudovišta po uzoru na dolaske ljudi u zgradu u stvarnom svijetu. Igrač brani FER uz pomoć tornja s kojeg puca metke kako bi se zaštitio od čudovišta. Prilikom svake odigrane igre zapisuju se igračevi potezi radi analize strategije igrača koja bi se mogla primijeniti na realan sustav.

U budućnosti, ovaj rad se može poboljšati ili nadograditi. Sustavi današnjice konstantno rastu te će potreba za pronalaženjem optimalnog rješenja posluživanja klijenata biti sve veća. U FERpocalypse se može uvesti valuta koju bi igrač dobivao kad pobijedi razinu. Valuta bi omogućavala unaprjeđenje oružja, odnosno metke s većom brzinom, dosegom i sl. Mogla bi se postići ravnopravnost ulaza tako da s onih manje zastupljenih dolaze „jača“ čudovišta. Primjerice, ulaz koji generira više čudovišta imao bi „slabija“ čudovišta koja bi igrač mogao eliminirati jednim metkom, dok bi manje zastupljen ulaz imao „jača“ čudovišta koja bi se mogla eliminirati tek s tri metka. S napretkom igrača svaka bi razina mogla postajati teža tako da joj čudovišta budu otpornija na metke. Osim toga, mogla bi se generirati različita čudovišta na svakom ulazu kako bi ih igrač mogao razlikovati tijekom igre. Kako bi se stvorila kompetitivnost među igračima mogla bi se implementirati online ljestvica s najboljim rezultatima.

LITERATURA

[1] Luid Von Ahn, Laura Dabbish, Designing Games With A Purpose, kolovoz 2008., http://delivery.acm.org/10.1145/1380000/1378719/p58-ahn.pdf?ip=31.147.157.197&id=1378719&acc=OPEN&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&__acm__=1560428471_0672348d9742b6f7590f12966c86bbbf, 09.06.2019.

[2] Human based computation game, https://en.wikipedia.org/wiki/Human-based_computation_game, 27.05.2019.

[3] Doug Wightman, Crowdsourcing Human-Based Computation, 16.10.2010., <https://static1.squarespace.com/static/519d10a2e4b090350a2b66a0/t/51c33cd8e4b03a05fb22549d/1371749592822/p551-wightman.pdf>, 02.06.2019.

[4] Sea Hero Quest, <http://www.seaheroquest.com/site/hr/>, 10.06.2019.

[5] Aleks Buczkowski, Top 10 Google Map games of all times, 13.08.2015., <https://geoawesomeness.com/10-top-google-maps-games-of-all-times/>, 03.06.2019.

[6] Luis von Ahn, Games with a Purpose, <http://www.cs.cmu.edu/~biglou/ieee-gwap.pdf>, 10.06.2019.

[7] Unity, <https://unity.com/>, 12.06.2019.

GRAFIČKI OBJEKTI

[8] Pločica trave, <https://opengameart.org/content/grass-tile-0>

[9] Pločice betona, https://opengameart.org/sites/default/files/Tiles_v9.zip

[10] Ikona zombija, <https://opengameart.org/content/mini-zombie>

[11] Ikona tornja, <https://opengameart.org/content/tower-defense>

[12] Ikona portala, <https://opengameart.org/content/portal-animation>

[13] Licenca, <https://creativecommons.org/licenses/by/3.0/>

Računalna igra s posluživanjem na temelju podataka stvarnog sustava

Sažetak

Računalne igre s posluživanjem na temelju stvarnih podataka nekog sustava taj sustav simuliraju i omogućuju igračima da putem interaktivne aplikacije daju rješenja za problem posluživanja klijenata. U velikim sustavima posluživanje klijenata čest je problem te se ovakvim načinom brže i efikasnije dolazi do novih algoritama. Cilj rada bio je napraviti računalnu igru na temelju stvarnih podataka koji predstavljaju ulazak ljudi u Fakultet elektrotehnike i računarstva u određenom periodu vremena. Igrač s tornja metcima gađa čudovišta kako bi obranio FER. Prilikom svake odigrane igre zapisuju se i spremaju igračevi potezi radi analize strategije igrača koja bi se mogla primijeniti na realan sustav.

Ključne riječi: računalna igra, posluživanje klijenata, algoritam, analiza strategije

Serving Game based on Real System Data

Abstract

Computer games with serving based on real system data simulate that system and allow the players to give solutions to client serving problems over an interactive application. In large systems serving clients is a common issue and this kind of problem solving allows us to find new algorithms faster and more efficient. The aim of the thesis was to build a computer game based on real data which represented people entering the Faculty of Electrical Engineering and Computing in a specific period of time. The player shoots projectiles at monsters from the tower to protect the building. During each game the player's strategic moves are collected and saved for analysis of the player's strategy which could be applied to the real system.

Key words: computer game, client serving, algorithm, strategy analysis