

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

Soul eater

Izvještaj o projektu

Studentski tim: Marko Banovac
Matej Đanić
Kristijan Vrbanc
Tibor Žukina

Predmet: Virtualna okruženja

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

Sadržaj

1. Kratki uvod	3
2. Pregled područja i problematike	5
3. Osnovne komponente projekta	6
4. Objekti	7
5. Parametri i varijable	8
6. Ekрани	9
7. Poslužiteljske akcije	11
8. Baza podataka	13
9. Zaključak	15
10. Literatura	17

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

Specifikacija projekta

1. Kratki uvod

Tema projekta je izrada interaktivne 2D Desktop računalne igrice koristeći razvojno okruženje Unity te aplikacijsku platformu Heroku, a ciljevi projekta su upoznavanje rada sa Unity razvojnim okruženjem i Heroku aplikacijskom platformom, proučavanje najboljih praksi pri izradi interaktivnih 2D igrica, upoznavanje sa izradom 2D likova, objekata i pozadina te njihovog dinamičkog toka kroz prostor virtualne scene u vremenu i međusobne interakcije putem sudara korištenjem odabranih programa i alata, a krajnji je cilj izrada jednokorisničke 2D interaktivne računalne igre namijenjene zabavi korisnika u kojoj je moguće i dijeljenje rezultata i rangiranje među drugim igračima koji koriste tu igru kreiranjem korisničkog računa i pohrana podataka o korisnicima i rezultatima na poslužitelju u bazi podataka.

Očekivani i planirani rezultat projekta bila je funkcionalna, zabavna i motivirajuća računalna igra Soul eater koristeći Unity game engine te primjenom najboljih praksi, a sve na temelju prethodnih istraživanja postojećih rješenja i tehnologije te razvoja adekvatne pozadinske priče, tematike i pravila uz modeliranje vizualno privlačne grafike.

Riječ je o igri mračne tematike u kojoj igrač upravlja likom demona koji se nalazi na lijevom kraju ekrana te ga može vertikalno pomicati, a prema njemu s desnog kraja ekrana, odnosno scene postoji horizontalni tok same pozadine s kojom igrač nema direktnu interakciju, kao i tok drugih likova i objekata. Likovi i objekti uključuju nasumično generirane elemente koji horizontalno protječu na način da se nužno sijeku s dijelom scene unutar kojeg se demon vertikalno kreće, dovodeći time do sudara, a čine ih duše koje demon mora uhvatiti i pojesti (odatle i naziv igre), anđeli koje mora izbjeći i koji mu oduzimaju snagu te ga nastoje uništiti, nakupine tamne energije koje je poželjno uhvatiti i koje mu vraćaju snagu te štit koji olakšava igru i privremeno čini otpornim na anđele kroz slučajno određeni vremenski interval od počevši od sakupljanja štita, a sve uz uvjet da se u vertikalnom skupu nasumično generiranih elemenata uvijek

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

mora nalaziti barem jedna duša.

Igra će se sastojati od ukupno 14 razina koje će imati različite pozadine (odnosno najmanje 5 pozadina izmjenjivat će se tako da ne bude ista pozadina u dvije uzastopne razine), u svakoj novoj razini kretanje elemenata s desne strane bit će ubrzano i igra time otežana, a igra će završiti porazom ako demon ostane bez sve snage nakon što ga anđeli unište ili kad sakupi ukupno 666 duša, po 50 u prvih 13 razina i 16 u posljednjoj razini.

Iako je sama igra jednokorisnička, korisnik će nakon završetka svake igre imati priliku podijeliti svoj rezultat s drugim korisnicima pomoću jedinstvenog korisničkog imena kojim će se moći registrirati te vidjeti svoj ostvareni rang u odnosu na druge igrače.

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

2. Pregled područja i problematike

Uz sve veću pojavu složenih i višekorisničkih virtualnih okruženja koja su realizirana u sklopu složenih 3D igrica na različitim platformama kao što su Desktop okruženja, Android i iOS, jednostavne i zanimljive jednokorisničke 2D igrice i dalje imaju nezanemarivu ulogu u svijetu računalnih i mobilnih igrica te postoje slučajevi u kojima su upravo takve igrice doživjele veliki uspjeh uz minimalne ili nikakve troškove razvoja kao što je mobilna igrica Flappy birds ili 2048. Mogućnost stvaranja zanimljive i uspješne računalne ili mobilne igrice moguća je korištenjem jedinstvenog i zanimljivog koncepta tematike i ciljeva koji odudaraju od standardnih i klasičnih modela igrica, a ovdje je primijenjena mračna i neobična tematika demona kojem je cilj uhvatiti i pojesti duše, a izbjeći anđeli koji ga pokušavaju uništiti.

U posljednje vrijeme također raste broj tehnologija i razvojnih okruženja u kojima je igrice niske ili srednje složenosti moguće kreirati bez puno razmišljanja o pozadinskim programskim procesima na nižem nivou, a sve je veći broj i različitih tutorijala i uputa o tome kako implementirati uobičajene obrasce korištene u takvim igricama te postaje sve jednostavnije, ali i sve korisnije ovladati vještinama stvaranja takvih igrica koristeći dostupne alate i razvojna okruženja.

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

3. Osnovne komponente projekta

Sama realizacija projekta i napravljena računalna igra koja čini gotovi proizvod sastoji se od sljedećih komponenti:

- **Klijentski dio aplikacije** - sadrži objekte i grafičke elemente koji se pomiču ili protječu kroz pozadinu i ambijent virtualne scene automatski prema pravilima same igre na temelju određenih parametara i varijabli ili upravljačkim akcijama igrača te ekrane koji se izmjenjuju unutar aplikacije s obzirom na ishod korisnikove igre i njegovih odabira ponuđenih opcija u navigaciji same aplikacije koje su realizirane elementima korisničkog sučelja kao što su tipke. Klijentski dio aplikacije nalazi se u mapi Unity projekta **SoulEater** te je projekt moguće pokrenuti Unity game engine alatom kao što je Unity Hub
- **Poslužitelj** - sastoji se od poslužiteljskih akcija koje služe za interakciju klijentske aplikacije i baze podataka u svrhu postizanja perzistencije podataka u vidu registracije i prijave korisnika, pohrane njegovih rezultata te njihovom usporedbom s rezultatima drugih korisnika dohvaćanjem rang listi. Poslužiteljski dio aplikacije nalazi se u mapi java projekta **Server** te je generiran u okviru Heroku poslužiteljskog razvojnog okruženja
- **Baza podataka** - sastoji se od relacija u kojima su pohranjeni podaci o korisnicima i njihovim rezultatima, a SQL skripta za kreiranje potrebne baze podataka te pripadni ER dijagram nalaze se u mapi **Database** te su kreirani također Heroku poslužiteljskim razvojnim okruženjem na temelju objektnog programskog koda

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

4. Objekti

Pomični objekti se generiraju u vertikalno grupiranim skupovima (stupcima) na desnoj strani ekrana i putuju prema demonu koji se nalazi s krajnje lijeve strane ekrana, u svakom skupu postoji točno određeni broj objekata.

- **Duša** - generira se na određenoj visini na ekranu kao jedan od objekata u vertikalno usmjerenom skupu, u vertikalnom skupu generiranih objekata mora biti barem jedna duša, u slučaju dodira s demonom on ju uhvati i usisa u mračni portal iza njega koji vodi u pakao, nema nikakve dodatne atribute
- **Anđeo** - generira se na određenoj visini na ekranu kao jedan od objekata u vertikalno usmjerenom skupu, ima atribut štete koji se nasumično određuje u određenom intervalu pozitivnih cijelih brojeva, u slučaju dodira s demonom, energija se smanjuje za iznos koji je jednak atributu štete anđela
- **Tamna energija** - generira se na određenoj visini na ekranu kao jedan od objekata u vertikalno usmjerenom skupu, u slučaju dodira s demonom, energija se povećava za jedan, osim ako već nije maksimalna
- **Štit** - generira se na određenoj visini na ekranu kao jedan od objekata u vertikalno usmjerenom skupu, ima atribut trajanja zaštite u sekundama koji se nasumično određuje u određenom intervalu pozitivnih cijelih brojeva, u slučaju dodira s demonom, oko demona stvara vatreni obruč koji će ga tijekom trajanja zaštite učiniti imunim na sudare s anđelima
- **Demon** - smješten je na lijevom rubu ekrana i ima konstantan horizontalni položaj, a igrač ga može pomicati vertikalno, predstavljen je klasom koja ima samo jednu instancu objekta, ima atribut energije koja kreće od određenog maksimalnog iznosa, smanjuje se u dodiru s anđelima, a povećava u dodiru s tamnom energijom

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

5. Parametri i varijable

Parametri i varijable unutar igre prikazuju se igraču na ekranu, koriste se za naknadno generiranje tablice rezultata (*highscore*) te pojedinim uvjetima određuju ishod igre, kao što je poraz, pobjeda ili prijelaz na iduću razinu.

- **Broj skupljenih duša** - broj duša koje je demon sakupio tijekom igre, počinje od nule, a kada dostigne 666, igra završava pobjedom
- **Razina igre** - razina igre direktno ovisi o broju skupljenih duša, kreće od razine 1 i povećava se nakon svakih 50 duša, dakle dostignuta razina jednoznačno je određena brojem trenutno skupljenih duša te se računa po formuli $\text{ceil}(\text{brojDuša}/50)$, igra ima 14 razina te posljednja razina završava kad je sakupljeno 666 duša, svaka razina ima drugačiju pozadinu, a indeks pozadine koja se koristi određen je formulom $\text{razina} \% \text{brojPozadina}$, svakim prelaskom na višu razinu brzina toka objekata prema demonu se povećava
- **Energija demona** - može se promatrati kao parametar igre ili kao atribut demona, iznos može biti između maksimalnog iznosa i nule, a kad jednom dosegne nulu, igra završava porazom

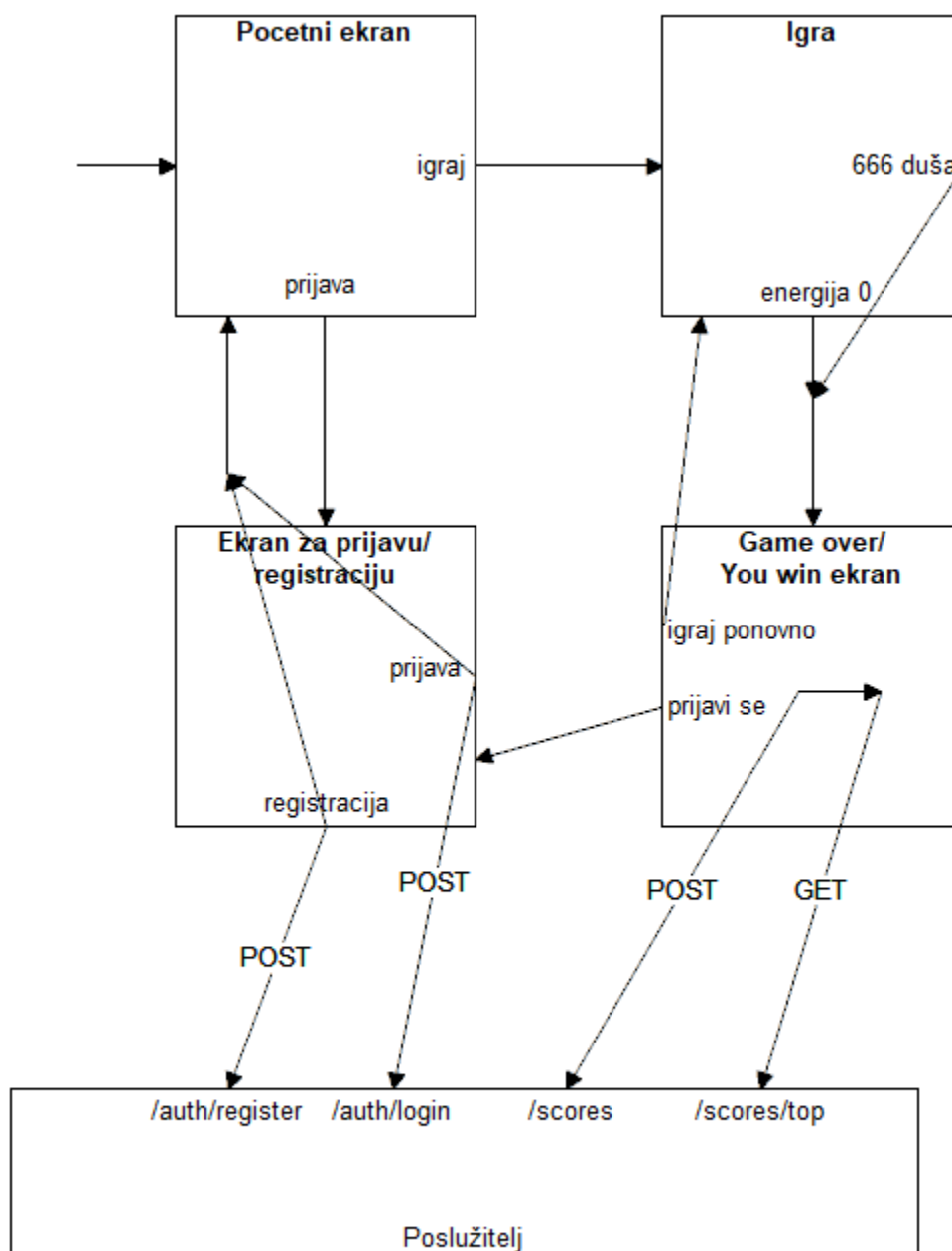
Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

6. Ekрани

Ekрани unutar aplikacije i njihov tok određeni su pravilima i ishodom igre te odabirom opcija na korisničkom sučelju različitih ekrana aplikacije.

- **Glavni ekran** - ekran koji se prikazuje nakon pokretanja aplikacije, s tog ekrana moguće je odabrati ekran za prijavu/registraciju ili započeti igru, a na njega se uvijek vraća nakon uspješne prijave/registracije
- **Igra** - ekran unutar kojeg se odvija igra koja se može pokrenuti s glavnog ekrana ili s Game over/You win ekrana, ako energija dosegne 0 ili broj duša dosegne 666, prelazi se s igre na Game over/You win ekran
- **Game over/You win ekran** - prikazuje se nakon poraza odnosno pobjede igrača, uz prikaz broja duša koji je dostigao, nakon dolaska na taj ekran najprije se poslužitelju šalje ostvareni rezultat korisnika odnosno broj sakupljenih duša, a potom se sa poslužitelja dohvaća i prikazuje *highscore* 10 najboljih korisnika, uključujući i ostvareni rezultat igrača ako se on nalazi među najboljima te najbolji rezultat i rang prijavljenog korisnika, s tog ekrana moguće je započeti novu igru ili odabrati ekran za prijavu/registraciju
- **Ekran za prijavu/registraciju** - prikazuje se ukoliko se opcija prijave odabere na glavnom ekranu ili Game over/You win ekranu, na tom ekranu igrač unosi korisničko ime i lozinku te ako odabere opciju registracije uspješno se registrira ukoliko korisnik već ne postoji te se vraća na glavni ekran, ili ako odabere opciju prijave uspješno se prijavi ukoliko je kombinacija korisničkog imena i lozinke korisna te se vraća na glavni ekran

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20



Slika 1 Dijagram toka aktivnosti

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

7. Poslužiteljske akcije

Iako je moguća igra u *singleplayer* načinu bez prijave i pohrane te spremanja rezultata, poslužitelj pruža i obrađuje zahtjeve za slanje i pohranu rezultata i uvid u rezultate drugih igrača¹. Za najbitnije zahtjeve prikazani su i primjeri tijela poruka/odgovora koji se šalju/generiraju od strane poslužitelja.

- **[POST] /auth/register** – poslužitelju se šalje zahtjev koji sadrži kao parametre korisničko ime i lozinku, a kao rezultat vraća poruku da je registracija (ne)uspješna (ukoliko već postoji korisnik s tim korisničkim imenom). Ovaj zahtjev rezultira generiranjem zapisa o korisniku u bazi podataka (*userId*, *username*, *password*).
- **[POST] /auth/login** – poslužitelju se šalje zahtjev koji sadrži parametre korisničko ime i lozinku, potom generira i vraća jedinstveni autorizacijski token koji se pohranjuje na uređaj korisnika ako je prijava uspješna ili poruku da je prijava neuspješna (ukoliko ne postoji korisnik s tim korisničkim imenom ili je upisana lozinka pogrešna). Autorizacijski token kojeg generira ovaj zahtjev se koristi u svim sljedećim zahtjevima za upisivanje rezultata kako bi se osigurao autorizirani pristup bazi podataka.

Primjer poruke koju vraća ovaj zahtjev nalazi se ispod:

```
{
  "id": 3, "username": "string", "roles": ["USER"], "accessToken":
  "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzdHJpbmciLCJpYXQiOiJlMjM-
  Dg0NjY1MDR9.0XYb82wOkP8RKNhNXMfGGn3ysu9TU0S-
  Ecse0HkbnVYdHQgEYRQYl5pKKLxQuqilq5Gk0fJGS17f_kK0j-
  URwQ", "tokenType": "Bearer"
}
```

¹ Generiranu **API** dokumentaciju moguće je pronaći na sljedećoj poveznici:
<https://virokr.herokuapp.com/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config>

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

Primjer tijela (eng. „*Request body*“) zahtjeva koji se šalju putem **Login/Request** akcija:

```
{
  "username": "John Doe",
  "password": "password123"
}
```

- **[GET] /scores** – Zahtjev koji dohvaća sortiranu listu postignutih rezultata i korisničkih imena igrača koji su ih ostvarili. Lista je sortirana po vrijednosti broja sakupljenih duša od najvećeg prema najmanjem. Ovaj zahtjev ne generira zapise u bazi podataka već ih samo iščitava.
- **[GET] /scores/top** – Slično kao prethodni zahtjev, ali vraća samo 10 najboljih rezultata. Primjer rezultata ovog zahtjeva jest sljedeći:

```
[{"id":3,"user":{"id":5,"username":"test"},"value":1}]
```

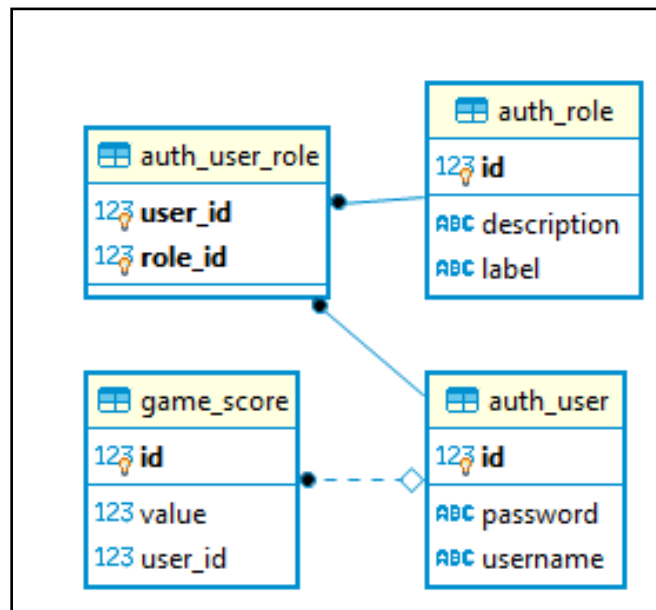
- **[POST] /scores** – zahtjev koji se šalje krajem svake odigrane igre, a kao parametar sadrži ostvareni rezultat u obliku broja sakupljenih duša tijekom igre, a vraća informaciju o uspješnosti pohrane rezultata. Ovaj zahtjev rezultira generiranjem ili ažuriranjem zapisa o najboljem postignutom rezultatu pojedinog korisnika. Rezultat se generira ako korisnik dosad nije ostvario rezultat, tj. ne postoji zapis rezultata trenutnog korisnika u bazi podataka, a ažurira se samo ako je postignuti rezultat veći od postojećeg.
- **[GET] /scores/{id}** – Zahtjev koji vraća jedinstveni rezultat iz baze podataka prema ID-u.
- **[DELETE] /scores/{id}** – Zahtjev koji briše zapis rezultata iz baze podataka (samo za korisnike s administratorskim ovlastima).
- **[GET] /users** – Zahtjev koji vraća listu svih registriranih korisnika.
- **[GET] /users/{id}** – Zahtjev koji vraća korisnika prema ID-u.
- **[DELETE] /users/{id}** – Zahtjev koji briše zapis korisnika iz baze podataka (samo za korisnike s administratorskim ovlastima).
- **[GET] /users/{id}/score** – Zahtjev koji dohvaća rezultat korisnika prema ID-u i rang tog rezultata u odnosu na sve ostale postignute rezultate. Primjer odgovora na ovaj zahtjev jest sljedeći:

```
{"rank":1, "value":1}
```

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

8. Baza podataka

Baza podataka sadrži objekte o registriranim korisnicima, njihovim ulogama i ostvarenim rezultatima. Slika 2 prikazuje *Entity-Relation* dijagram korištene baze podataka. Baza podataka implementirana kao PostgreSQL baza.



Slika 2 ER dijagram baze podataka

- **Korisnik** („auth_user“)– zapis registriranog korisnika
 - **id** (AUTO_INCREMENT, PRIMARY KEY)
 - **username** – jedinstveno korisničko ime
 - **password** – generirani sažetak, tj. *hash* lozinke
- **Uloga** („auth_role“) – zapis o ulozi pojedinog korisnika prema kojoj se određuju točno koje privilegije i ovlasti manipulacije nad podacima korisnik posjeduje
 - **id** (AUTO_INCREMENT, PRIMARY KEY)
 - **label** – naziv uloge
 - **description** – opis uloge

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

- **Rezultat** („*game_score*“) – zapis najvećeg postignutog rezultata za pojedinog korisnika
 - **id** (AUTO_INCREMENT, PRIMARY KEY)
 - **user_id** – strani ključ na tablicu korisnika
 - **value** – vrijednost ostvarenog rezultata
- **Korisnik-Uloga** („*auth_user_role*“) – spojna (eng. „*junction*“) tablica koja sadrži zapise o posjedovanju uloga za svakog korisnika. Kao zadana (početna) vrijednost, svaki novoregistrirani korisnik posjeduje ulogu „*User*“
 - **user_id** – strani ključ na tablicu korisnika
 - **role_id** – strani ključ na tablicu korisnika

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

9. Zaključak

Sve planirane stavke projekta kvalitativno su dovršene kako je planirano i svi planirani elementi klijentske i poslužiteljske arhitekture te integracije između tih arhitektura uspješno su isplanirani i implementirani, no s obzirom na kvantitetu i opseg značajki igrice najvjerojatnije ne bi bila spremna za plasiranje na stvarno tržište u kojem bi u dovoljnoj mjeri privukla stvarne korisnike, a bila bi moguća i nadogradnja sigurnosti samoga sustava.

Što se tiče značajki igre, moguća proširenja bilo bi dodavanje više različitih tematika igre koje bi korisnici mogli podešavati u skladu s vlastitim preferencijama, koje bi se nasumično pokretale ili koje bi kreirale virtualna okruženja i scene koje čine više jedinica igre poredanih prema težini. Primjerice, nakon što igrač u klasičnom načinu igre prijeđe posljednju razinu i prikupi 666 duša, umjesto završetka igre i pobjede, igra bi mogla početi ispočetka u većoj težini sličnoj normalnom tijeku razina igrice, ali bez postojanja štitova, ili bez postojanja tamnih energija koje demonu mogu vratiti energiju izgublenu u sudaru s demonima, čime bi se drastično smanjio broj igrača koji dostignu krajnji rezultat i ostvare pobjedu, a što bi igračima predstavljalo veću motivaciju za postizanje boljih rezultata i natjecanje s drugih korisnicima.

Kako bi se povećala raznolikost igrice, mogla bi postojati mogućnost demona da tijekom igrice prikuplja pentagrame koji bi se pohranjivali na njegovom računu te služili kao valuta za kupnju dodatnih pojačanja poput produljenog trajanja štita ili mogućnosti uništavanja nadolazećeg anđela umjesto njihovog izbjegavanja, a pentagrami bi se također mogli moći kupovati za pravi novac putem nekog integriranog vanjskog sustava za obradu plaćanja. Igrica bi mogla biti proširena dodatnim elementima koji bi ju dodatno otežali i učinili izazovnijom i zanimljivijom, poput svetaca koji bi se povremeno generirali umjesto anđela i koji bi mogli u potpunosti uništiti ili paralizirati demona (onemogućiti privremeno njegovo kretanje i izbjegavanje anđela) ili mu prekinuti ili skratiti djelovanje štita. Izmjena različitih pozadina virtualne scene, pozadinske glazbe i ambijentalnih zvukova bilo bi još jedno poželjno proširenje same igre.

Također, sama ograničena implementacija liste rezultata koja prikazuje 10 najboljih igrača te trenutnom igraču vraća samo njegovu poziciju mogla bi biti

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

zamijenjena listom koja se može cijela pregledati iz aplikacije postupnim učitavanjem sa poslužitelja tijekom listanja, ili listom u kojoj je moguće pretraživati listu rezultata po korisničkom imenu, rednom broju rezultata ili rasponu rezultata.

S obzirom na sigurnosnu nadogradnju, pohrana lozinki korisnika trebala bi biti realizirana uz pohranu dodatnih nasumično generiranih tzv. salt nizova znakova za lozinku svakog korisnika umjesto samo uz pohranu hasha lozinke korisnika, što dodatno otežava pokušaje njihovog probijanja. Također, uz registraciju svakog korisnika trebala bi se tražiti email adresa koja bi bila pohranjena u bazi podataka i pomoću koje bi se potvrdnim kodom ili linkom verificirao svaki račun tijekom registracije, kako ne bi moglo doći do automatiziranog kreiranja velikog broja računa. U budućnosti bi trebala biti implementirana i mogućnost registracije i prijave administratorskih korisnika koji bi imali ovlasti brisati pojedine korisnike, dohvat liste svih korisnika i brisanje pojedinih rezultata iz liste rezultata, što je trenutno sve realizirano samo na poslužiteljskoj strani te za to ne postoji opcija u korisničkom sučelju niti su pripremljeni sigurnosni mehanizmi zaštite koji bi bili primijenjeni u slučaju uvođenja tih dodatnih funkcionalnosti.

Soul eater	Virtualna okruženja
Izvještaj o projektu	Datum: 20/12/20

10. Literatura

- [1] <https://devcenter.heroku.com/categories/reference> - *Heroku* platforma
- [2] <https://swagger.io/docs/> - Generirana *API* dokumentacija
- [3] <https://docs.spring.io/spring-framework/docs/current/reference/html/> - *Spring* radni okvir
- [4] <https://www.postgresql.org/docs/> - PostgreSQL baza podataka
- [5] https://www.baeldung.com/learn-spring-security-course?utm_source=github&utm_medium=social&utm_content=general&utm_campaign=lss – Implementacija autorizacije (registracija, prijava...)
- [6] <https://spring.io/guides/tutorials/rest/> - *Rest* arhitektura web servisa