

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 663

Mobilna igra za vježbanje matematike

Denis Pipalović

Zagreb, svibanj 2022.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

SADRŽAJ

1. Uvod	1
2. Cilj projekta	3
2.1. Mobilna igra	3
2.1.1. Glavni ekran igre	3
2.1.2. Predefinirane postavke	5
2.1.3. Preuzete postavke	6
2.2. Web stranica	8
2.2.1. Izrada postavki	8
2.2.2. Pregled rezultata	8
3. Slične igre	9
3.1. Slična igra 1	9
3.2. Slična igra 2	9
3.3. Slična igra 3	9
3.4. Slična igra 4	9
4. Korištene tehnologije	10
4.1. Java	10
4.2. Node.js	10
4.3. HTML, CSS	10
5. Implementacija mobilne igre	11
5.1. Pozadina	11
5.2. Objekt za skupljanje	13
5.3. Padajući objekti	15
6. Baza podataka	16
6.1. ER dijagram	16

6.2. Opis entiteta i tablica	17
6.3. Relacijski dijagram	19
7. Naslov 4	20
Literatura	21

1. Uvod

Matematika, osim što je predmet koji se predaje u svim razredima osnovnih i srednjih škola, također je i znanost iz područja prirodnih znanosti. Značaj matematike vidljiv je ne samo u drugim poljima područja prirodnih znanosti, već i u svim drugim područjima, posebice u području tehničkih znanosti. Ne poznavanje osnova matematike negativno se odražava na učenikove školske uspjehe kroz cijelo školovanje. Jedan od problema svakako je problem pristupa nastavnika u učenju matematike, odnosno manjak povlačenja paralela sa stvarnim svijetom i pokazivanja primjena matematike; ovakav pristup demotivira učenika u vježbanju matematike i savladavanju čak i osnovnih računskih operacija. Jednom izgublenu motivaciju za neko područje teško je vratiti nazad.

Primjenom gamifikacije u edukaciji učenicima bi se razne teme i ishodi učenja mogle prezentirati kroz zanimljive načine, samim time i povećati zainteresiranost za nekom temom ili predmetom. S obzirom na to da djeca u novije doba koriste pametne uređaje sve više i više, gamifikacija jedan od način kroz koji bi to vrijeme na mobitelu moglo biti potrošeno u edukativne svrhe; kroz zabavu!

S obzirom na navedeno, tema ovog rada je napraviti mobilnu igru za vježbanje matematike u nižim razredima osnovne škole koja će kroz zanimljiv i interaktivan način pružati ugodnije iskustvo vježbanja matematičkih zadataka. Igra je zamišljena kao tzv. „endless runner“ u kojoj igrač skuplja padajuće objekte ovisno o trenutnom zadatku (npr. samo parne brojeve, brojeve veće od zadanog, brojeve djeljive s određenim brojem, samo kvadrate i slično) i sakuplja bodove. Kako igra ne bi postala monotona kroz kratko vrijeme igranja cilj je napraviti da igra postaje progresivno sve teža i teža, no do te mjere da se ne postigne negativan efekt u drugom ekstremu; igra ne smije postati niti previše teška jer bi samim time postala i demotivirajuća! Stoga je cilj da se težina igre dinamički prilagođava mogućnostima učenika. Osim same mobilne

igre, potrebno je izgraditi i jednostavno web sučelje za učitelja kako bi mogao postavljati tipove zadataka koji prate temu onoga što se trenutno predaje na nastavi. Kroz navedeno web sučelje nastavniku će se pružati i mogućnost pregleda rezultata pojedinog učenika te uvid u detalje igre kao što je tip zadatak na kojem je učenik griješio, ukupno vrijeme igranja i slično. Kako bi igra bila dostupna svima koji ju požele igrati, odnosno kako igra ne bi ovisila o postavkama učitelja, postojati će i zadane postavke koje će se moći prilagođavati kroz postavke same igre.

2. Cilj projekta

Cilj projekta je izrada mobilne igre za vježbanje matematike popraćene web stranicom koja služi za zadavanje zadataka i pregled rezultata igre. Cilj mobilne igre napraviti je vježbu matematike zanimljiviju učenicima, dok je cilj web stranice učiteljima olakšati zadavanje zadataka, praćenje i pohranu rezultata učenika na jednostavniji način.

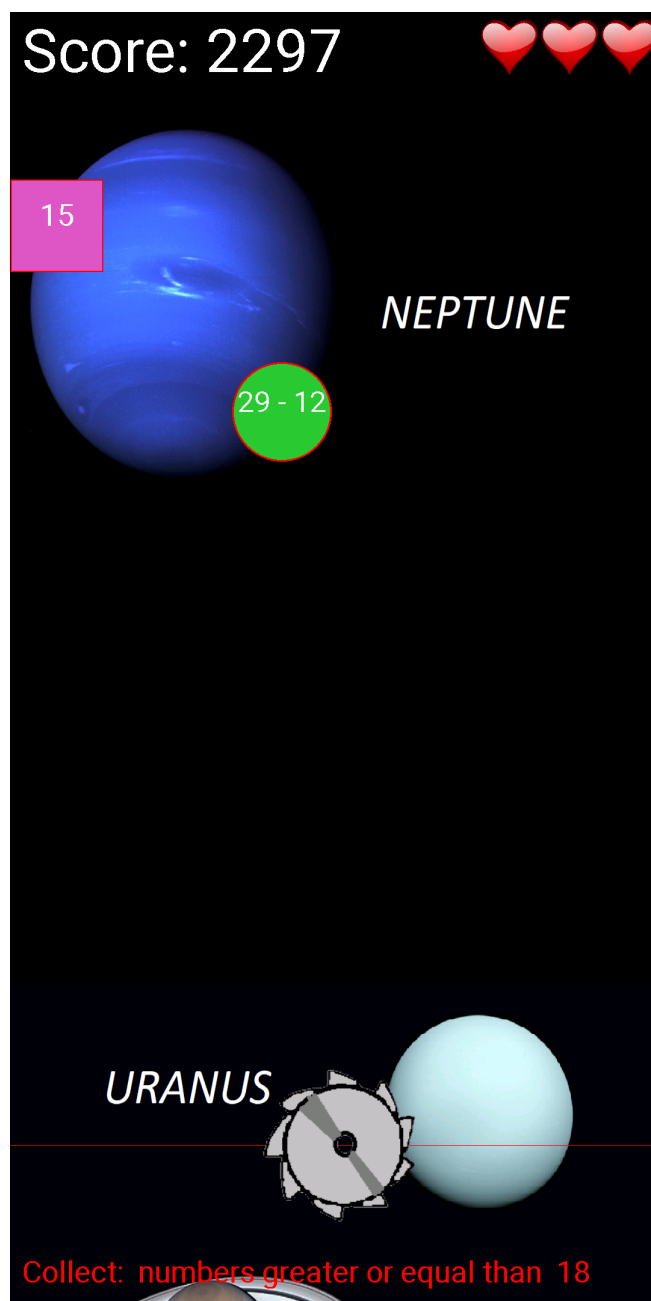
2.1. Mobilna igra

Mobilna igra većinski je dio projekta. Kroz ovu igru učenici nižih razreda na zanimljiviji način mogu vježbati određeno gradivo matematike. Igra osim što se može igrati s predefiniranim postavkama zadataka također omogućava uređivanje istih te preuzimanje postavki koje su napravljene preko web sučelja za učitelja. Ukoliko učenik igra igru sa preuzetim postavkama napravljenim preko web sučelja, podaci o njegovoj igri bit će dostavljeni učitelju po završetku igre. Iz glavnog menija igre možemo pristupiti glavnom ekranu igre i postavkama.

2.1.1. Glavni ekran igre

Na glavnom ekranu igre nalazi se sav takoreći učenicima zanimljiv sadržaj, odnosno mjesto gdje se sama igra i igra. Igra je "endless runner" i tematski je zamišljena kao put iz središta zemlje ka dubokom svemiru, odnosno to je ono što pokazuje pozadina koja putuje prema dolje, što stvara efekt putovanja objekta koji se upravlja prema gore. Objekt koji je upravljani moguće je pomicati lijevo-desno i tako skupljati padajuće objekte koje je potrebno skupiti ovisno o trenutnom zadatku, odnosno izbjegavati one koji krše pravila trenutnog zadatka. Padajući objekti konstanto se stvaraju i njihov sadržaj ovisi o tome koji je trenutni zadatak ili koja je trenutna težina igre. Odnosno svaki

zadatak osim što ima svoj opis ima i određene parametre, tako bi recimo zadatak "skupljaj brojeve veće ili jednake 18", osim navedenog opisa, relativnog broja 18 imao i definirane parametre kao što je minimalna i maksimalna granica iz kojeg seta se brojevi mogu dodjeliti; nadalje, pri većoj težini igre se na padajućem objektu možda neće prikazati broj direktno, već u obliku matematičke operacije, npr. "29 - 12".



Slika 2.1: Ideja igre

Igranjem igre i sakupljanjem objekata koji odgovaraju trenutačnom zadatku

učenici skupljaju bodove; te je cilj po završetku igre imati što veći broj bodova. Završetak igre odgovara gubitku tri života. Bodovanje na primjeru za zadatak "skupljaj parne brojeve" :

- Igrač je pokupio paran broj : +100 bodova
- Igrač NIJE pokupio paran broj: -100 bodova
- Igrač je pokupio neparan broj: Gubitak života

S obzirom da se može dogoditi situacija u kojoj se preklapaju dva objekta, odnosno situacija u kojoj se preklapaju objekt koji se ne smije pokupiti i objek koji se mora pokupiti (u ovom slučaju parni i neparni broj); kako se igrač ne bi osjećao zakinutim nakon gubitka 100 bodova zbog objekta kojeg je morao propustiti kako ne bi izgubio jedan život, igrač je za svakih 10 osvježavanja ekrana nagrađen sa jednim bodom. U konačnici ispada da igrač smije propustiti jedan objekt svakih 15 do 20 sekundi kako bi se poništili bonus bodovi sa izgubljenim bodovima za ne skupljanje odgovarajućeg objekta. Preklapanje dva objekta se ne događa toliko često stoga igrač samim trajanjem igre skuplja dodatne bodove.

Trenutačni broj bodova prikazan je na vrhu zaslona sa lijeve strane, dok je na desnoj strani prikazan broj života koji odgovara broju sličica srca. Na donjem dijelu ekrana moguće je vidjeti trenutačni zadatak. Svaki zadatak traje od prilike pola minute, a promijena zadatka odvija se na način da prvo prestanu padati novi objekti. Kada više nema padajućih objekata na ekranu, slučajnim odabirom se bira sljedeći zadatak te se tekst novog zadatka pokazuje kako u donjem dijelu ekrana tako i na samoj sredini ekrana na nekoliko sekundi kako bi igrač bio u potpunosti svjestan promjene zadatka.

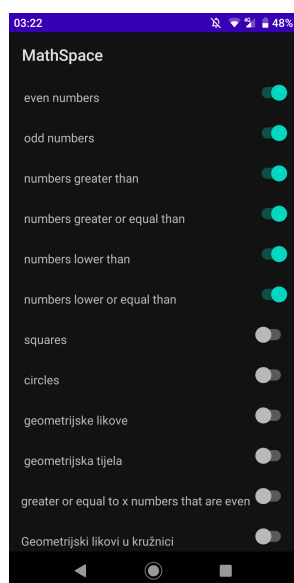
2.1.2. Predefinirane postavke

Kako igra ne bi ovisila isključivo o postavkama koje zadaje učitelj, odnosno kako bi se mogla igrati neposredno nakon instalacije, osmišljene su predefinirane postavke koje sadrže zadatke prilagođene ponajviše prvom razredu osnovne škole. Neki od predefiniranih zadataka su:

- Skupljaj parne brojeve
- Skupljaj neparne brojeve
- Skupljaj brojeve veće od x
- Skupljaj brojeve manje od x

- Skupljaj brojeve veće ili jednake x koji su parni
- Skupljaj kvadrate
- Skupljaj kružnice
- i slično

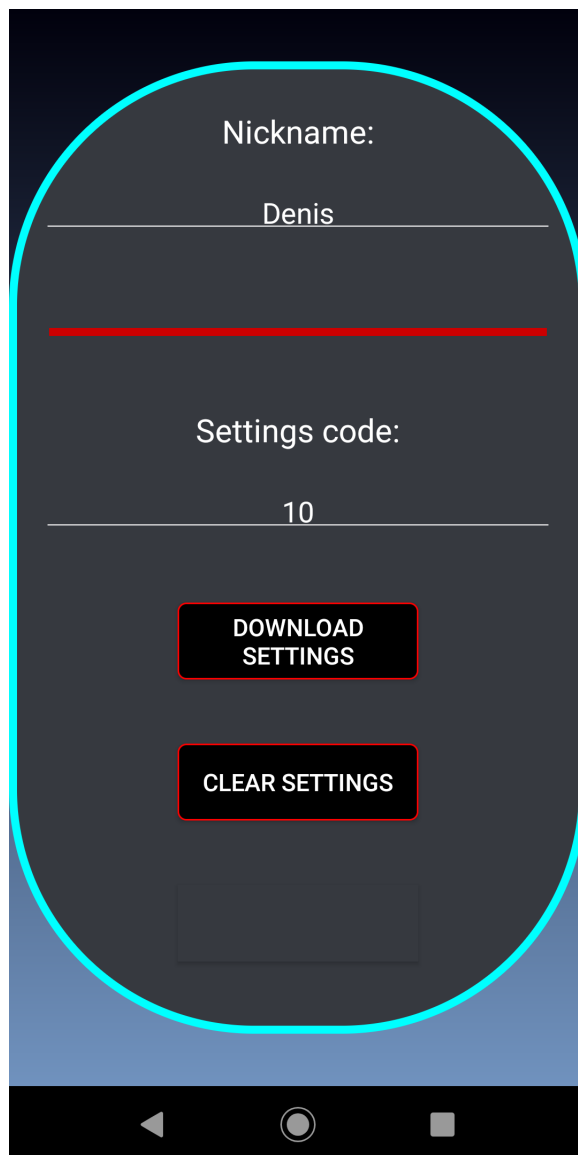
Osim što igra omogućava ove i druge predefinirane postavke, također omogućava i njihovo uređivanje, odnosno njihovo uključivanje te isključivanje; kako bi si učenici samostalno mogli prilagoditi zadatke na koje žele staviti naglasak dok vježbaju. Svako uključivanje, odnosno isključivanje navedenog zadatka ostaje zapamćeno i koristi se pri pokretanju igre.



Slika 2.2: Ideja igre

2.1.3. Preuzete postavke

Osim predefiniranih postavki igra nudi mogućnost preuzimanja postavki izrađenih preko web stranice. Kako bi igrač mogao preuzeti postavke prvo treba postaviti korisničko ime te nakon toga u za to predviđeno polje upisati identifikator postavki koje želi preuzeti i za kraj pritisnuti tipku za preuzimanje postavki.



Slika 2.3: Preuzimanje postavki s interneta

Jednom kad je igrač preuzeo postavke s interneta sada mu se na izbor u glavnom meniju postavki pruža mogućnost odabira koje postavke želi koristiti; one preuzete s interneta ili one predefinirane!



Slika 2.4: Odabir korištenja predefiniranih postavki



Slika 2.5: Odabir korištenja preuzetih postavki

2.2. Web stranica

Podpoglavlje 1.2

2.2.1. Izrada postavki

2.2.2. Pregled rezultata

3. Slične igre

Poglavlje 2 - prednosti i nedostatci svake, screenshot, url

3.1. Slična igra 1

Podpoglavlje 2.1

3.2. Slična igra 2

Podpoglavlje 2.2

3.3. Slična igra 3

Podpoglavlje 2.3

3.4. Slična igra 4

Podpoglavlje 2.4

4. Korištene tehnologije

Poglavlje 3

4.1. Java

Podpoglavlje 3.1

4.2. Node.js

Podpoglavlje 3.2

4.3. HTML, CSS

Podpoglavlje 3.3

5. Implementacija mobilne igre

5.1. Pozadina

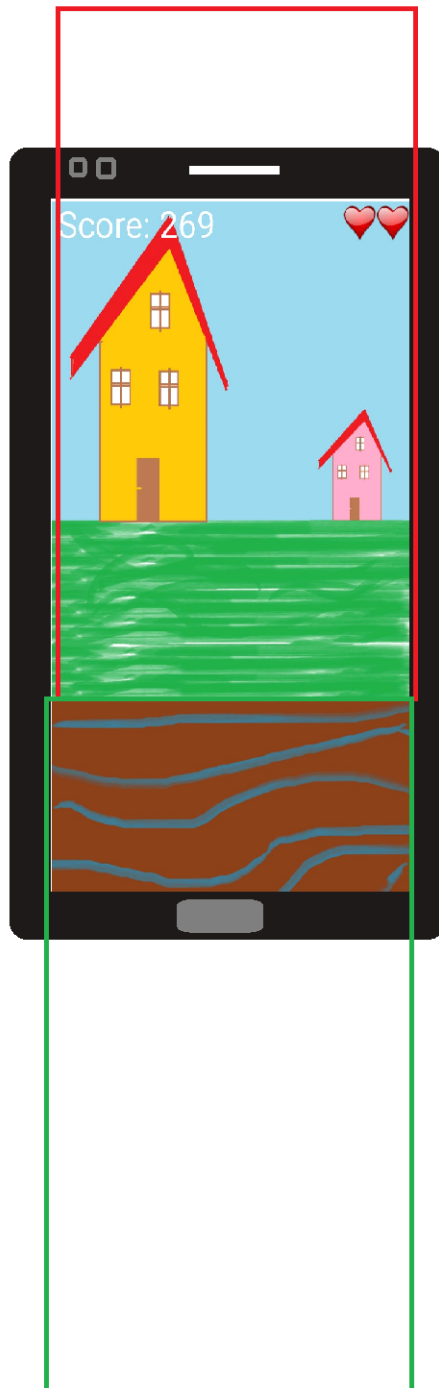
Pozadina je tematski zamišljena kao put iz središta zemlje do dubokog svemira. Mobilna igra je takozvani endless runner, što znači da se potencijalno može igrati beskonačno, odnosno ne postoji način na koji igra završava osim u slučaju da igrač izgubi sve živote, stoga je osim samog efekta "putovanja ka svemiru" potrebno osigurati i beskonačno pomicanje same pozadine.

Pozadina kao takva se u konačnici sastoji od 20ak slika čije spajanje vertikalno daje efekt kontinuiranosti. Dok poznati pokretački strojevi (engl. *game engine*) poput Unity nudi opcije manipulacija pozadinom, kao što je beskonačno ponavljajuća pozadina, ta mogućnost u ovakvom razvoju aplikacije nije moguća, već ju je potrebno iskodirati samostalno. Za početak je potrebno učitati slike pozadina kao Bitmap te ih omotati u objekte kako bi im se dala različita svojstva poput x i y koordinata gdje se trenutno nalaze. Takvi objekti pohranjeni su u polje.

Radi jednostavnosti, svaka slika pozadine se vertikalno i horizontalno skalira na broj piksela koje zaslon mobitela ima; zbog toga, u jednom trenutku na zaslonu se mogu vidjeti maksimalno dvije slike pozadine od jednom. Na samom početku igre uzimaju se prva dva objekta pozadine iz polja. Prvi objekt predstavlja donju pozadinu koja će biti prikazana, a drugi gornju pozadinu. Donju pozadinu potrebno je inicijalizirati na lokaciju sa x koordinatom 0 i y koordinatom 0, odnosno želimo da se pokazuje od početka zaslona vertikalno i horizontalno. Drugu sliku inicijaliziramo opet tako da joj damo x koordinatu 0, no y koordinatu postavljamo na minus vertikalnu veličinu zaslona. Na svako ponovno osvježavanje igre, pozadina se vertikalno spušta ka dolje čime se stvara efekt putovanja igrača prema gore. Spuštanje pozadine prema dolje implementacijski je ostvareno tako što se svakim osvježavanjem zaslona povećava y koordinata obje slike za određenu vrijednost, ovisno o tome kojom

brzinom želimo da "igrač putuje ka svemiru".

Na sljedećoj slici može se vidjeti način iscrtavanja pozadine na ekranu i izvan njega.



Slika 5.1: Pomicanje pozadine

U trenutku kada y koordinata donje slike izađe izvan ekrana (odnosno kada

je vrijednost y osi veća od y dimenzije ekrana) nova donja slika sada je slika koja je bila gore, a novu gornju sliku uzimamo kao sljedeći član polja. Zadnja slika prikazuje svemir pun zvijezdi i nju u polje učitavamo dva puta (dva različita objekta). U trenutku kad smo u igri došli do te slike, tim dvijema slikama se konstanto mijenjaju vrijednosti y osi. Odnosno kad donja slika izađe dolje s ekrana, ponovno se postavljaju vrijednosti y osi za obje slike.

Cijeli način implementacije pomicanja pozadine možemo prikazati sljedećim kodom:

```
1 private void updateBackground() {
2     currentDownBackground.setY(currentDownBackground.getY() + 5);
3     currentUpBackground.setY(currentUpBackground.getY() + 5);
4
5     if (currentDownBackground.getY() > screenY) {
6         currentDownBackground = currentUpBackground;
7         if (currentBackgroundIndex < backgrounds.length - 1)
8             currentUpBackground = backgrounds[++currentBackgroundIndex];
9         else {
10            currentDownBackground = backgrounds[backgrounds.length - 2];
11            currentDownBackground.setY(0);
12            currentUpBackground = backgrounds[backgrounds.length - 1];
13        }
14        currentUpBackground.setY(-screenY);
15    }
16 }
```

5.2. Objekt za skupljanje

Slično kao i pozadinu, objekt koji se kontrolira, odnosno objek s kojim se skupljaju padajući objekt učitani je kao kao Bitmap, odnosno više njih, konkretnije kao dva Bitmap objekta omotana sa objektom klase "Saw". Za razliku od pozadine koja je putovala prema dolje, objekt s kojim skupljamo padajuće objekte je statičan na ekranu, odnosno ne mijenja svoj položaj na ekranu bez korisničke akcije. Pod korisničku akciju u ovom slučaju podrazumijeva se dodir ekrana. Koristeći oblikovni obrazac promatrač (engl. *observer*), pri inicijalizaciji igre subjektu (objektu konkretne klase "View", u ovom slučaju to je "SurfaceView") se postavlja konkretna implementacija apstraktnog promatrača "OnTouchListener". Navedeno sučelje sastoji se od jedne funkcije:

```
boolean onTouch(View v, MotionEvent event);
```

Ova funkcija prima dva parametra, "View v" koji je pokazivač na "View" kojem je događaj dodira dostavljen te parametar "MotionEvent event" koji opi-

suje tip događaja. Primjeri tipova događaja su: "ACTION_DOWN" koji govori da se dogodio pritisak na zaslonu, "ACTION_UP" koji govori da je dodir zaslona otpušten, "ACTION_MOVE" koji govori da se dogodila promjena pozicije između pritiska ("ACTION_DOWN") i otpuštanja ("ACTION_UP") te nekolicina raznih drugih akcija. Navedena funkcija vraća primitivni tip boolean koji u ovom slučaju predstavlja zastavicu koja govori da li je konkretni promatrač "konzumirao događaj do kraja".

U nastavku je prikazan cijeli programski kod potreban za pomicanje objekta za skupljanje. Konkretna implementacija sučelja "OnTouchListener" izvedena je kao anonimni razred, odnosno zbog bolje preglednosti koda koristi se pokratak u vidu lambda izraza, dok "this" predstavlja objekt klase View.

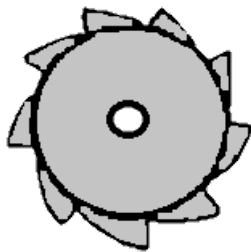
```
1      this.setOnTouchListener((v, event) -> {  
2          if (event.getAction() == MotionEvent.ACTION_MOVE  
3              || event.getAction() == MotionEvent.ACTION_DOWN) {  
4              saw.setX((int) event.getX());  
5          }  
6          return true;  
7      });
```

Na ovom programskom kodu potrebno je uočiti da se kontroliranje našeg objekta može dogoditi doticanjem i pomicanjem prsta na bilo kojem dijelu ekrana, no da sam objekt kojeg kontroliramo neće promijeniti vrijednost svoje x osi, odnosno neće se pomicati vertikalno, već se isključivo mijenja vrijednost položaja na horizontalnoj osi; također je bitno uočiti da za pomicanje objekta nije potrebno stisnuti na objekt te ga povući, već je dovoljno stisnuti na određeno mjesto na ekranu te će se objektov položaj na horizontalnoj osi teleportirati na željeno mjesto. Iako je ovo programsko rješenje vjerojatno najjednostavnije, kasnije će se ispostaviti da je ujedno i najbolje. Ukratko, može se dogoditi da objekt kojeg kontroliramo bude okružen s padajućim objektima koji se ne smiju pokupiti i stoga je ova mogućnost "teleportiranja" poželjna.

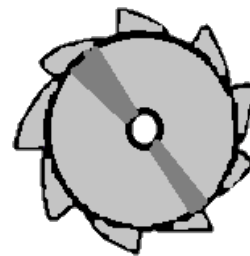
Kao što je već ranije navedeno, objekt za skupljanje koji se kontrolira ne mijenja svoj položaj na ekranu (bez prethodne akcije korisnika) već je potreba učitavanja više slika, odnosno stvaranja više Bitmap objekata nastala u svrhu stvaranja animacije. Animacija se stvara tako što se brzo izmjenjuju slike koje se iscrtavaju na zaslonu. Konkretno, ova animacija sastoji se od dvije slike koje se neprestano izmjenjuju. Prva slika nastala je jednostavnim crtanjem, dok je druga nastala rotacijom za određeni kut te dodavanjem dodatnih detalja.

Sljedeće dvije slike prikazuju objekt s kojim skupljamo padajuće objekte,

odnosno zajedno prikazuju oštricu koja se rotira velikom brzinom.



Slika 5.2: Originalna slika oštrice



Slika 5.3: Zarotirana originalna slika oštrice sa dodatnim detaljima

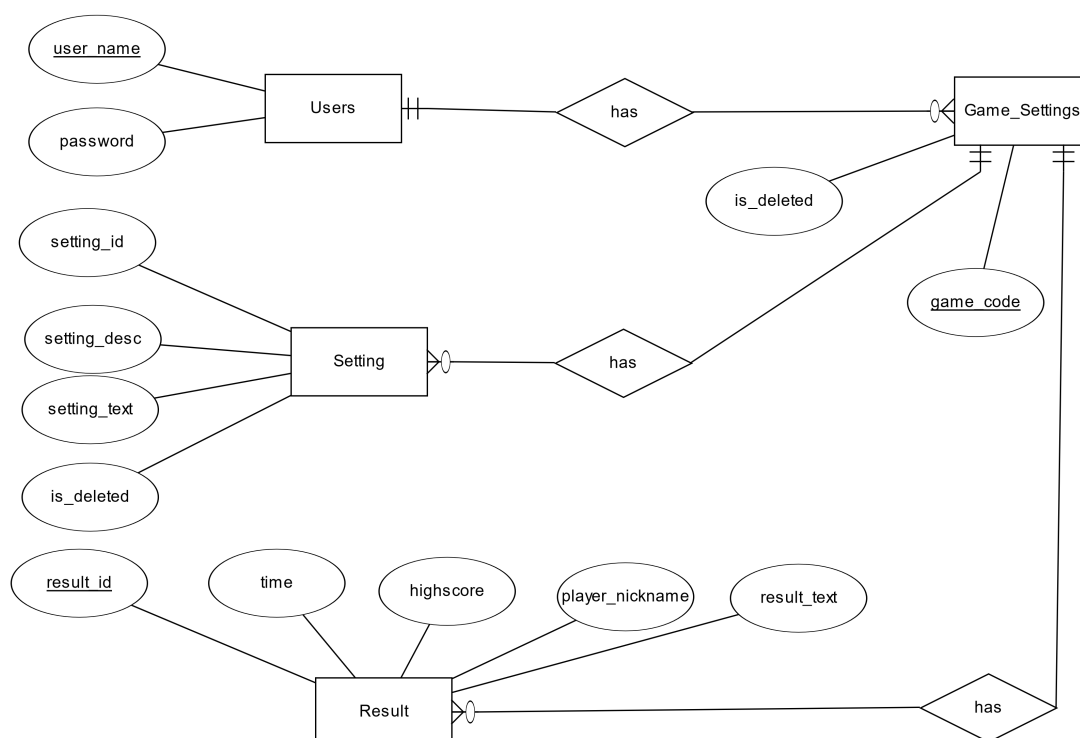
5.3. Padajući objekti

6. Baza podataka

Baze podataka su organizirane kolekcija podataka koje omogućavaju laki pristup podacima, uređivanje podataka i upravljanje istih. U sklopu ovog projekta koristio se SQL tip baze podataka, konkretnije PostgreSQL. PostgreSQL je besplatni, open-source sustav za upravljanje bazom podataka (SUBP) iza kojeg je više od 30 godina aktivnog razvoja.

6.1. ER dijagram

Sljedeća slika prikazuje ER dijagram baze podataka.



Slika 6.1: ER dijagram baze podataka

6.2. Opis entiteta i tablica

Users Ovaj entitet označava korisnika koji koristi web aplikaciju, odnosno web sučelje za učitelja. Sadrži attribute: `user_name` i `password`. Atribut `user_name` predstavlja korisničko ime korisnika, dok atribut `password` označava zaporku koja se ne sprema u "plain-textu", već kao rezultat hash funkcije. Ovaj entitet u vezi je *One-To-Many* s entitetom `Game_Settings` preko atributa `user_name`.

Users		
<code>user_name</code>	VARCHAR	jedinstveni naziv korisnika
<code>password</code>	VARCHAR	rezultat hash funkcije nad zaporkom

Game_Settings Ovaj entitet označava jedan "game room". Sadrži attribute: `game_code` i `is_deleted`. Ovaj entitet u vezi je *Many-To-One* s entitetom `Users` preko atributa `user_name`, u vezi *One-To-Many* s entitetom `Setting` preko atributa `game_code` te u vezi *One-To-Many* s entitetom `Result` preko atributa `game_code`.

Game_Settings		
<code>game_code</code>	INT	jedinstveni idenfitikator <code>game_settings</code>
<code>is_deleted</code>	INT	zastavica koja govori jesu li "game room" obrisani
<code>user_name</code>	VARCHAR	oznaka korisnika kojem "game room" pripada

Setting Ovaj entitet označava jednu postavku. Sadrži attribute: `setting_id`, `setting_desc`, `setting_text` i `is_deleted`. Ovaj entitet u vezi je *Many-To-One* s entitetom `Game_Settings` preko atributa `game_code`.

Setting		
setting_id	INT	jedinstveni idenfitikator za Setting
setting_desc	VARCHAR	tekst koji opisuje cilj zadatka
setting_text	VARCHAR	postavka formatirana tako da bude razumljiva mobilnoj igri
is_deleted	INT	zastavica koja govori je li postavka obrisana
game_code	INT	oznaka Game_Settings kojem Setting pripada

Result Ovaj entitet označava jedan rezultat igranja igre na mobitelu uz preuzete postavke. Sadrži attribute: result_id, time, highscore, player_nickname te result_text. Ovaj entitet u vezi je *Many-To-One* s entitetom Game_Settings preko atributa game_code.

Result		
result_id	INT	jedinstveni idenfitikator za Result
time	VARCHAR	vrijeme pohrane rezultata u bazu podatka
highscore	VARCHAR	ostvareni rezultat na igri
player_nickname	VARCHAR	nadimak igrača
result_text	VARCHAR	JSON polje s događajima igre
game_code	INT	oznaka Game_Settings kojem Result pripada

6.3. Relacijski dijagram

Sljedeća slika prikazuje relacijski dijagram baze podataka.



Slika 6.2: Relacijska shema baze podataka

7. Naslov 4

Poglavlje 4

LITERATURA

Mobilna igra za vježbanje matematike

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Mobile game for practicing math

Abstract

Abstract.

Keywords: Keywords.