Primjena klijentske skriptne tehnologije u izradi računalne igre

Matić, Krešimir

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

PRIMJENA KLIJENTSKE SKRIPTNE TEHNOLOGIJE U IZRADI RAČUNALNE IGRE

Završni rad

Krešimir Matić

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 19.09.2022.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Krešimir Matić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R 4398, 22.07.2019.
OIB Pristupnika:	14130425510
Mentor:	Prof. dr. sc. Krešimir Nenadić
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Primjena klijentske skriptne tehnologije u izradi računalne igre
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rad:	Kratko opisati kako se klijentska skriptna tehnologija može primijeniti u izradi računalne igre. Opisati funkcionalnosti koje će igra imati. Detaljno opisati postupak izrade igre. Modelirati i izraditi bazu podataka za potrebe pohrane svih potrebnih podataka izrađene aplikacije. Tema rezervirana: Krešimir Matić
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	19.09.2022.
Datum potvrde ocjene od strane Odbora:	22.09.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 23.09.2022.

Ime i prezime studenta:	Krešimir Matić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R 4398, 22.07.2019.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Primjena klijentske skriptne tehnologije u izradi računalne igre**

izrađen pod vodstvom mentora Prof. dr. sc. Krešimir Nenadić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Veoma poznata mobilna igra „Flappy Bird“ je izašla na tržište Android i iOS operacijskih sustava sredinom 2013. godine. No, ni punih godinu dana kasnije i igrica je maknuta sa Google Play trgovine i Apple-ove trgovine aplikacija. Igrica je bila jako popularna u svojem kratkom životnom ciklusu te je inspiracija za završni rad.

Svrha rada je bila istraživanje i korištenje skriptne tehnologije za izradu igrice slične igri „Flappy Bird“ u obliku web aplikacije. Igrica ima sve funkcionalnosti poput originala, isti princip prelaženja prepreka te svakim uspješnim prelaženjem broji rezultat dok se ne zaustavi srazom sa preprekom, srazom na dnu ili vrhu područja za igranje. Rezultat je moguće pohraniti u bazu podataka te je omogućen pregled svih rezultata u obliku ljestvice.

Za izradu web aplikacije koriste se web tehnologije poput HTML-a, CSS-a te JavaScript, te za dodatan dizajn svakog dijela igre koristiti će se Adobe Photoshop, a pohrana i dohvaćanje podataka će biti omogućeno kroz Firebase servis.

Struktura rada se sastoji od poglavlja koji detaljno opisuju postupak izrade igrice te moguću problematiku i slična rješenja. Drugo poglavlje se bavi problematikom i sličnim rješenjima završnog rada pronađena jednostavnom pretragom putem internetskog pretraživača. U ovom poglavlju su opisane pet sličnih aplikacija te su za svako rješenje napisane moguće koristi i primjedbe koje su korištene za unaprjeđenje vlastite web aplikacije. Treće poglavlje opisiva sami dizajn cjelokupne web stranice i strukturu stranice. Naime, web stranica se sastoji od tri osnovna dijela: naslovna, leaderboard i about stranica, u korist jednostavnijem i ljepšem snalaženju. Četvrto poglavlje se bavi opisom svih alata korištenih u izradi web aplikacije i cjelokupno web stranice. Svako potpoglavlje govori o tome kako je točno alat primijenjen i korak po korak opisiva moguće funkcije korištene tim alatom. Peto poglavlje objašnjava rad s aplikacijom, odnosno, objašnjava se sučelje cijele web stranice počevši od same igre koja je interaktivna uz pomoću dvije tipke. Također se objašnjava postupak spremanja podataka u bazu i pregled istih podataka sa desne strane web stranice. Potpoglavlja ovog naslova jesu pojašnjenje testiranja svih mogućih funkcija te je navedeno i potpoglavlje o uklanjanju pronađenih grešaka.

1.1. Zadatak završnog rada

Zadatak završnog rada je kratko opisati kako se klijentska skriptna tehnologija može primijeniti u izradi računalne igre. Opisati funkcionalnosti koje će igra imati. Detaljno opisati postupak izrade igre. Modelirati i izraditi bazu podataka za potrebe pohrane svih potrebnih podataka izrađene aplikacije.

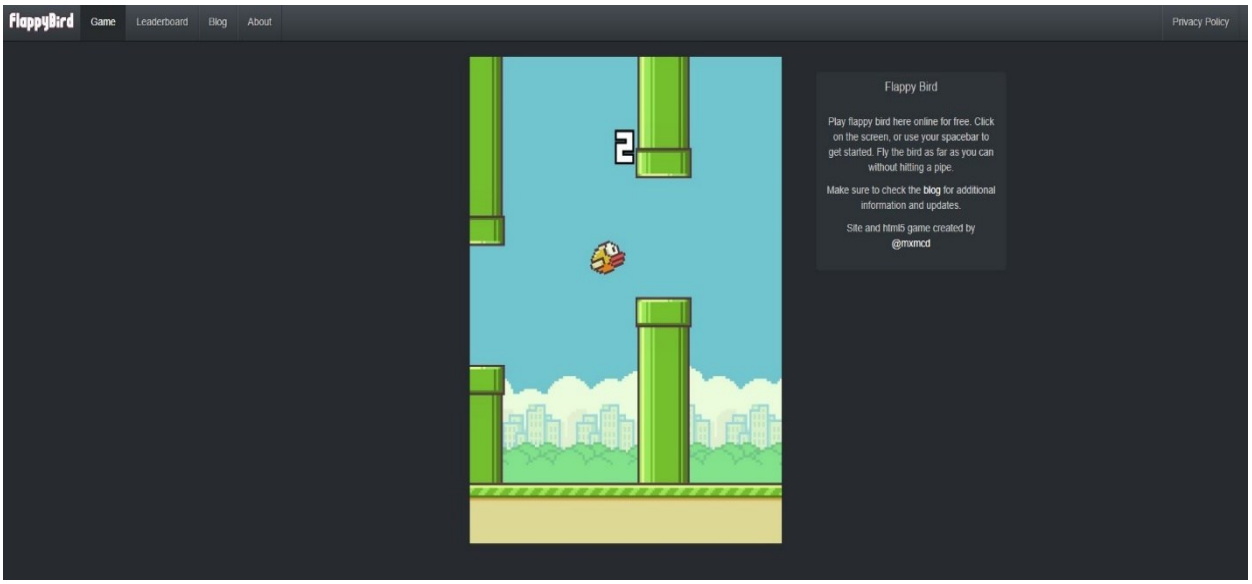
2. PROBLEMATIKA I SLIČNA RJEŠENJA

Tvorac igre Flappy Bird Dong Nguyen imao je 28 godina kada je prvi put napravio aplikaciju 2013. Htio je napraviti nešto jednostavno, ali izazovno s igrom. Radio je kao programer u Hanoju u Vijetnamu i živio je s roditeljima. S 28 godina, Nguyen je proveo vikend smišljajući igricu koja će ga uskoro učiniti vrlo bogatim i koja će postati jedna od najčešće preuzimanih mobilnih igara ikada. Jednostavna mobilna igra je navodno preuzeta više od 50 milijuna puta u svom postojanju i ima vrlo veliku bazu obožavatelja do danas. Osnovne funkcionalnosti igre jesu da dodirom ekrana „ptica“ leti prema gore te sa velikom preciznošću prolazi između prepreka koje se nalaze na donjoj i gornjoj strani ekrana te se beskonačno generiraju. Uspješnim prolaskom svake prepreke brojač bi se povećavao sve dok ne bi došlo do kolizije između „ptice“ i prepreka. Konačni rezultat nije bio moguće uspoređivati sa drugim korisnicima jer nije postojala ljestvica sa najboljim rezultatima. Mnogi su tvrdili da je ovu igricu koja izaziva veliku ovisnost zabranio Google. Flappy Bird nije zabranjen, ali aplikacije još uvijek nema u Googleovoj trgovini Play. Razlog za to je taj što je Flappy Bird uklonio vlastiti kreator Dong Nguyen jer se osjećao krivim zbog toga koliko je igra postala zarazna.

Od dana uklanjanja mobilne aplikacije sa svih trgovina nastale su mnoge kopije aplikacije. Prvobitno su nastale kopije same mobilne aplikacije, a kasnije su nastale i web aplikacije iz navedene igre. Do početka 2021. godine sve kopije „Flappy Bird“ aplikacije na web-u su mogle funkcionirati uz pomoć Adobe Flash Player-a, ali nakon prekida podrške radu Flash Player-a trebalo je pronaći novo inovativno rješenje. To rješenje je pristiglo u obliku skriptne tehnologije, i u nekim rijetkim slučajevima Unity Web tehnologije.

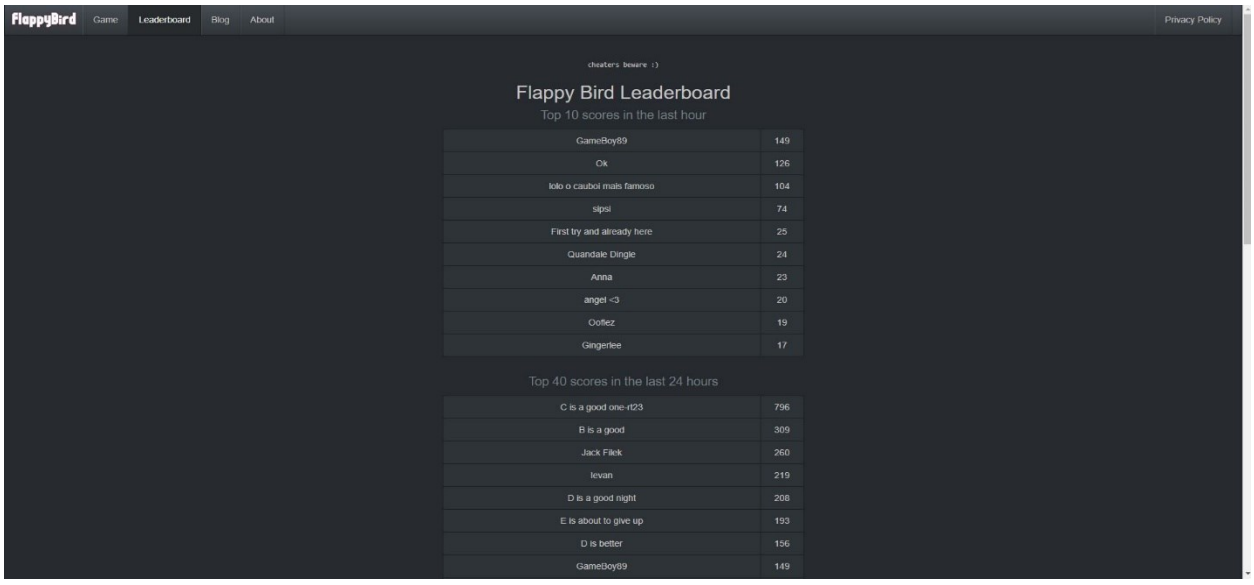
2.1. Flappybird.io

Trenutnim pretraživanjem interneta, ako se upiše „flappy bird“ u tražilici, vrlo vjerojatno se dobije stranica flappybird.io kao prvo ponuđeno rješenje. Ova web stranica omogućava igranje popularne igre te spremanje rezultata. Dizajn web stranice je jednostavan sa tamno sivom pozadinom. Igra se nalazi u kontejneru na sredini stranice te desno od kontejnera se nalaze kratke upute kako igrati igru. Ova stranica je imala velik učinak na dizajniranje vlastite aplikacije zbog svoje jednostavnosti. Stranica također ima navigacijsku traku na kojoj se nalazi kartice za igru, ljestvicu, blog i opis stranice. Svi ovi elementi se mogu vidjeti na slici 2.1.



Slika 2.1. flappybird.io stranica, prema [1]

Dio stranice koji sadrži ljestvicu sa najboljim rezultatima ima sličan dizajn kao i glavni dio stranice, jedina promjena jest što se na sredini stranice nalazi kontejner sa ljestvicom najboljih rezultata umjesto igre. Ljestvica broji deset najboljih rezultata u posljednjih sat vremena i četrdeset najboljih rezultata u posljednja 24 sata. Ovaj dio stranice je prikazan na slici 2.2.

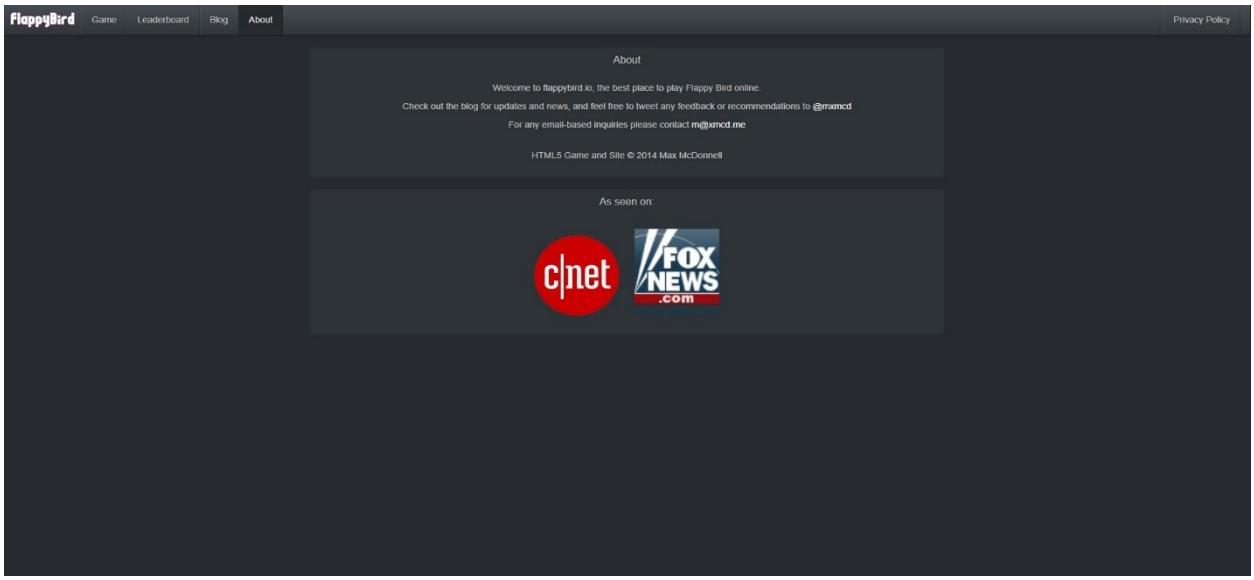


Slika 2.2. Leaderboard dio flappybird.io stranice, prema [1]

Blog je sljedeći dio stranice no pokušajem pristupa toj kratici stranica preusmjerava korisnike na tumblr.com sa porukom da blog dio stranice nije dostupan u trenutnom vremenu.

Posljednji dio stranice jest opis stranice u about kartici prikazan na slici . Dizajn ostaje isti kao na prethodnim primjerima sa tamno sivom pozadinom, a elementi koji se nalaze na ovom dijelu

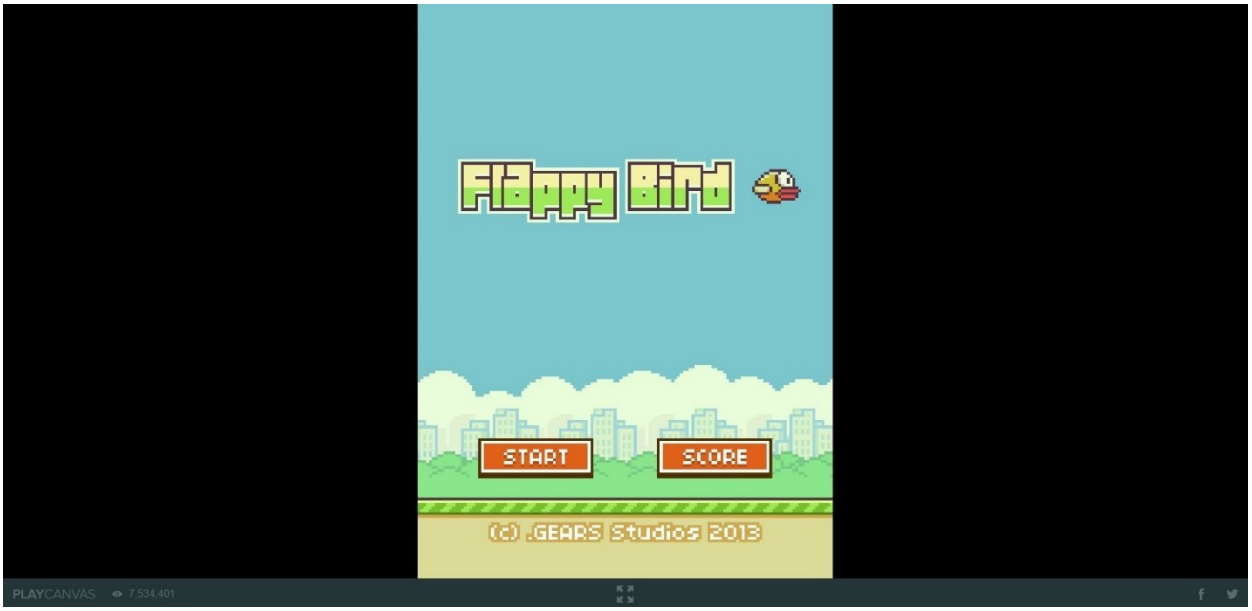
stranice su samo dva div elementa sa kratkim opisima aplikacije i predloženim kontakt računima na društvenoj mreži Twitter i predloženom e-adresom za povratnu informaciju.



Slika 2.3. About dio flappybird stranice, prema [1]

2.2. Playcanv.as

Playcanv.as za razliku od flappybird.io nije potpuna stranica nego sadrži samo web aplikaciju. Mnogi portali za igre dohvaćaju „Flappy Bird“ datoteku od ove stranice. Ova verzija igre je najsličnija originalnog mobilnoj aplikaciji te sadrži i nedostatke originale igre kao što je ljestvica najvećih rezultata. Kao što je prikazano na slici 2.4. stranica sadrži samo igru i footer sa linkovima na društvene mreže.



Slika 2.4. playcanvs.as "Flappy Bird", prema [2]

2.3. Flappy Bird – Unity WebGL Player

Ova web stranica omogućava igranje igre bez ikakvih dodatnih elemenata. Igra je izrađena pomoću Unity tehnologije te je implementirana na stranicu. Na slici 2.5. je prikazana web stranica sa elementom u sredini koji služi kao kontejner u kojem je implementirana igra. Ispod kontejnera igre se nalazi traka sa imenom korištene tehnologije i gumb za povećanje igre na cijeli ekran.

Slika 2.5. "Flappy Bird" UnityWebGL Player, prema [3]

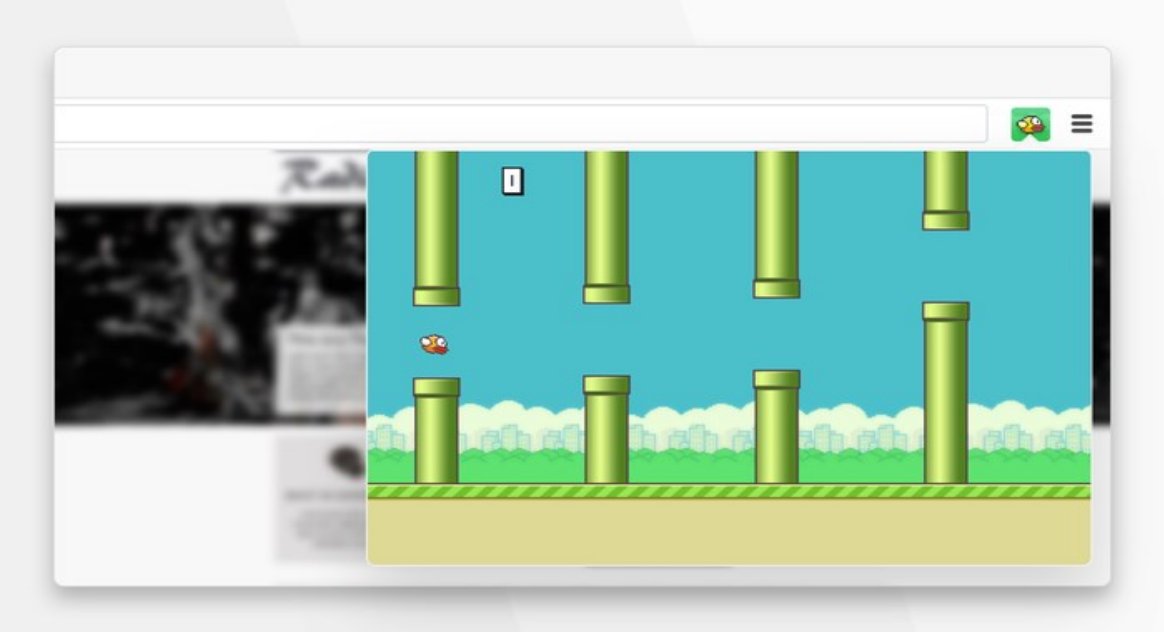
Ova verzija ima najveću nasumičnost prepreka, ali ima brojnih nedostataka. Ne postoji pohrana u ikakvu bazu podataka, te igra sama po sebi nije dobro definirana. Glavna greška koja postoji jest da objekt koji se kreće može letjeti izvan ograničenja ekrana u kojem se nalazi, s obzirom da brojač koji inače radi ako prolazi između prepreka, u tom slučaju ne radi i ostaje na istom broju na kojem je bio otkako se krenulo letjeti izvan granica. Zbog ove uočene greške u završnom radu je odmah bilo uvjetovano da objekt kojim korisnik upravlja ne može letjeti izvan granica kontejnera igre. Kada se igra pokrene u punom ekranu javljaju se novi problemi. Kao što se vidi na slici 2.6., pokretanjem igre u punom ekranu omogućava da se vide sve pozadinske radnje što bi trebale biti sakrivene od korisnika. Može se vidjeti generiranje prepreka, resetiranje donjeg elementa na desnu stranu u svrhu stvaranja iluzije kretanja te odsječenu sliku koja služi kao dio pozadine.

Slika 2.6. "Flappy Bird" UnityWebGL Player u punom ekranu, prema [3]

2.4. Firefox Browser proširenje

Verzija „Flappy Bird“ aplikacije korištene kao proširenje za Firefox internetski preglednik. Prema [4] proširenje je nastalo 2018. godine te je preuzeto od strane 558 korisnika. Nudi isti princip kao original, ali bez metode spremanja konačnog rezultata.

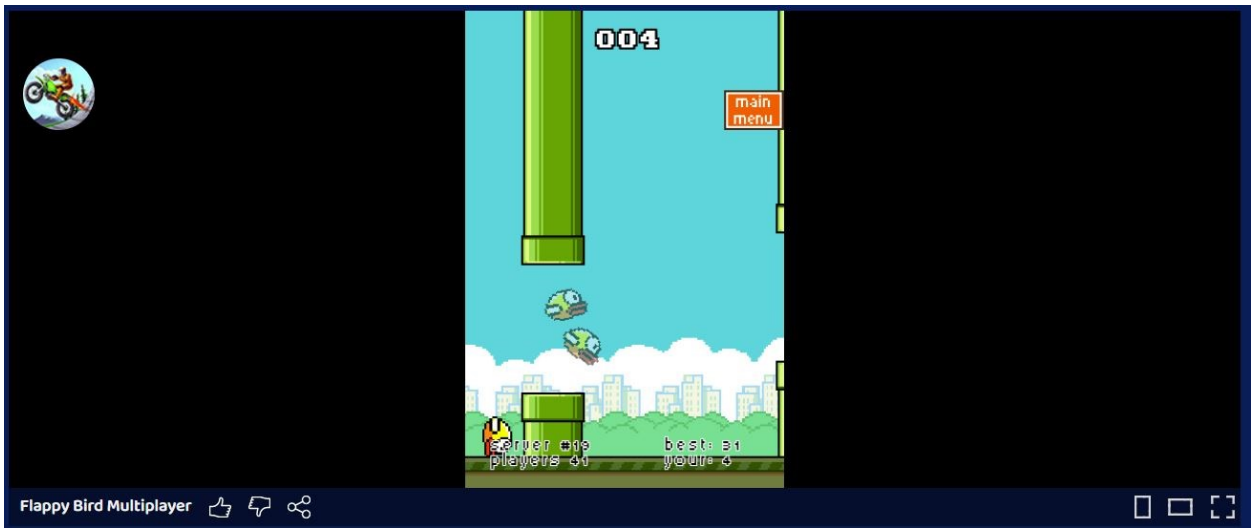
Na slici 2.7. se može primijetiti da se aktivacijom hamburger gumba pored adresne trake, proširenje otvori i moguće je igrati igru. Ovo proširenje predstavlja prvu verziju „Flappy Bird“ aplikacije sa širokim vidnim poljem, ali nije ideja koja će se primjenjivati u radu jer se gubi osjećaj nepredvidljivosti prepreka.



Slika 2.7. Mozilla Firefox proširenje, prema [4]

2.5. Flappy Bird Multiplayer – Gameforge.com

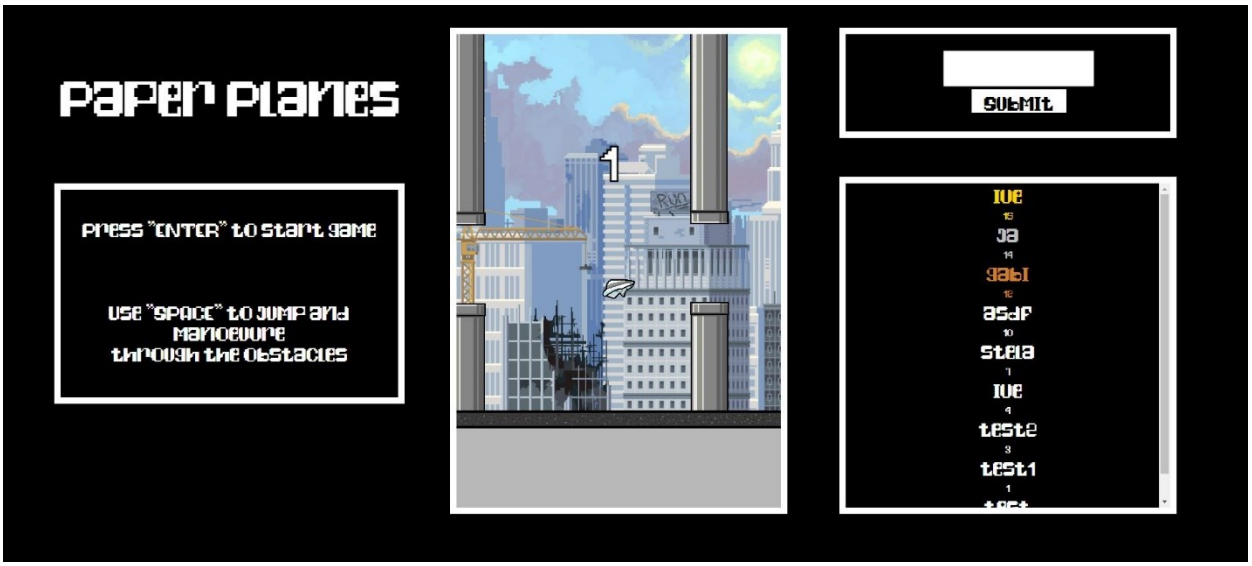
Na web portalu „Gameforge.com“ se može naći verzija popularne igre s mogućnošću igranja protiv drugih korisnika. Naime, kada se započne igra korisnik osim sebe vidi druge transparentne likove koji se natječu u stvarnom vremenu. Igra nema mogućnost spremanja rezultata ni provjeravanja ikakve liste najboljih igrača. Na slici 2.8. je prikazan izgled aplikacije tijekom igranja i mogu se uočiti elementi multiplayer igre. Iako je dobra ideja multiplayer verzije igre, dovoljno je samo implementirati bazu podataka kako bi se vidjela usporedba rezultata različitih korisnika. S obzirom da je stranica web portal za igre, osim ove igre sa desne strane postoji izbornik drugih igara koje portal nudi.



Slika 2.8. "Flappy Bird Multiplayer", prema [5]

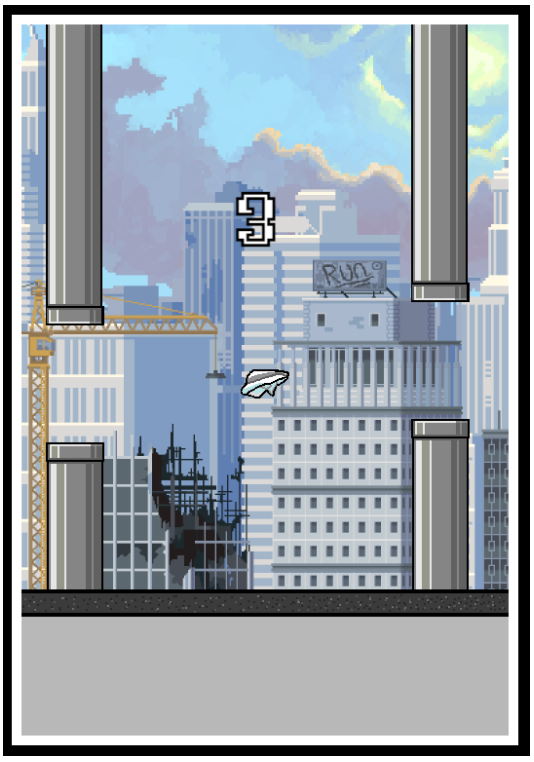
3. DIZAJN WEB STRANICE

Dizajn web stranice se sastoji od definiranja rezolucije igre te određeno smještanje igre na ekranu što se radi uz pomoć HTML-a i CSS-a, a funkcionalnosti igre se definiraju uz pomoć JavaScript tehnologije. Stranica se sastoji od tri dijela, a to su igra, ljestvica najboljih rezultata i upute za igru. Naslovna stranica web aplikacije je veoma jednostavna sa svrhom pružanja samo osnovnih elemenata stranice, a to su sama igra, „Leaderboard“ dio stranice i upute za igru.
Izgled svih elemenata stranice i samim time cijela stranica je prikazana na slici 3.1.



Slika 3.1. Cijela stranica

Glavni dio naslovne strane jest kontejner za igru. Smješten je na sredini same stranice s fiksnim dimenzijama širine od 500 točaka te visine od 700 točaka. Kontejner je relativno malih dimenzija jer ne postoji potreba za većom širinom ili visinom da ne remeti samu srž igre, a to su nepredvidivost prepreka i ograničena kretnja lika. Iz slike 3.2. može se primijetiti da se oko kontejnera nalazi okvir koji pridaje estetici stranice i ispunjava prazni prostor oko kontejnera, a u kontejneru se nalaze svi elementi za igru.



Slika 3.2. Igra

Lijevo od područja kontejnera igre nalazi se naslov igre, slika 3.3. , te ispod naslova su opisane kratke upute kako započeti igru te kako se kretati unutar igre.

Slika 3.3. Upute

Desno od područja kontejnera igre se nalazi ljestvica najboljih rezultata, slika 3.4., gdje su prva tri rezultata obojana zlatnom, srebrenom i brončanom bojom. Iznad ljestvice najboljih rezultata se nalazi blok koji sadrži polje za upis imena te gumb čijom aktivacijom se pohranjivaju ime i ostvareni rezultata u bazu podataka. Unutar ljestvice s najboljim rezultatima omogućeno je pohranjivanje najboljih sto rezultata svih vremena.

Slika 3.4. Ljestvica najboljih rezultata

4. RAZVOJ WEB APLIKACIJE

Razvoj web aplikacije je strogo urađen korištenjem JavaScript tehnologije u kojoj su definirane sve prepreke, lik kojim se upravlja, spremanje u bazu podataka te dohvaćanje podataka u „Leaderboard“ dijelu stranice.

Pri izradi aplikacije korišteni su razne web tehnologije i razvojni alati. Osnovne web tehnologije korištene u radu su HTML (engl. *HyperText Markup Language*), CSS (engl. *Cascading Style Sheets*) i JavaScript, također je korišten Firebase za rad s bazom podataka. Od razvojnih alata korišten je Visual Studio Code, a za izradu svih slika koje su dodane na elemente koristi se Adobe Photoshop.

4.1. HTML

Primarno korišten za određivanje elemenata koji se nalaze na web stranici HTML je također korišten za povezivanje svih datoteka u jednu cjelinu.

Datoteke koje su povezane su *style.css, database.js, leaderboard.js* i *app.js*, a one su povezane u <head> dijelu koda HTML dokumenta. Svrha ovog povezivanja jest da se omogući pokretanje aplikacije bez problema te za primjereno ukrašavanje stranice, što je veoma zahtjevno u samom HTML-u.U <body> dijelu HTML koda su definirani svi elementi koji se trebaju nalaziti na pojedinačnim dijelovima web stranice. Elementi koji su definirani u ovom dijelu jesu: kontejner u kojem se nalazi igra, ljestvica najboljih rezultata, blok za spremanje rezultata i upute za igru. Svi ovi elementi su definirani kao div blokovi sa odgovarajućim imenom klase kako bi se mogli ukrašavati u .css datoteci te kako bi se lako moglo njima pristupiti u JavaScript datotekama.

Igra i svi elementi igre su definirani unutar game-wrapper elementa, a to su „game-container“, „sky“, „ground“, „plane“, „counter“ te granice oko kontejnera. Upute se definirane jednim div elementom, ljestvica najboljih rezultata također jednim div elementom, a blok za spremanje rezultata je definiran sa input elementom i gumb elementom.

4.2. CSS

CSS datoteka je korištena za poboljšavanje izgleda cijele web aplikacije. Unutar datoteke su definirane sve dimenzije i vrijednosti elemenata koji su definirani u HTML dokumentu.

Dimenzije su precizno određene za svaki element i nisu nasumični brojevi. Svaka dimenzija je definirana u svrhu točnosti funkcioniranja aplikacije jer u JavaScript dokumentu su dimenzije jako važne za određivanje granica određenih stanja igre i za detekciju kolizije sa svim mogućim preprekama.

Kontejner igre ima specifične dimenzije od 500 sa 700 točaka te svaki element unutar kontejnera je raspoređen i omjeren u skladu sa ovim dimenzijama. Na primjer element „sky“ ima dimenzije 500 sa 580 točaka dok element „ground“ ima dimenzije 500 sa 150 točaka, te se može primijetiti da ovi elementi popunjavaju cijeli kontejner. Osim „sky“ i „ground“ elemenata, u kontejneru je definiran i treći element po imenu „plane“. „Plane“ element ima najmanje dimenzije od 50 sa 30 točaka jer je to lik kojim će korisnik upravljati u aplikaciji.

Element za koji se možda smatra da bi trebao biti unutar kontejnera igre, ali nije, jest prepreka koja je definirana tek u JavaScript dokumentu. Prepreka je opisana dimenzijama 60 sa 300 točaka te se ne nalazi u kontejneru iz razloga što se ta prepreka treba pomjerati iz desna na lijevo i treba imati učinak kao da dolazi i nestaje neprimjetno iz kontejnera, što ne bi bilo moguće da se nalazi u samom kontejneru te iz tog razloga postoje granice kontejnera koje služe kao okvir u svrhu prekrivanja nastajanja i nestajanja prepreka.

Osim kontejnera igre ukrašeni su i svi ostali elementi navedeni u HTML dokumentu. Pozicija elemenata je određena absolute opcijom. Ova opcija nam omogućava da pomjeramo elemente stranice na bilo koje mjesto uz pomoć left i top opcija. Tako je pomjeren kontejner igre na sredinu, upute za igranje igre na lijevu stranu ekrana te ljestvica sa najboljim rezultatima i element za spremanje imena i rezultata na desnu stranu. Svaki element je ukrašen skladno sa ostalima da bi se održala ravnoteža cijele stranice. Ukrašeni su tako što uz crnu pozadinu, svaki element ima bijeli obrub širine 10 točaka, ako postoji tekst unutar elementa i on je ukrašen bijelom bojom i izabran je poseban font uveden iz dodatnog direktorija unutar završnog rada. Veličina fonta ovisi o veličini elementa u kojem se nalazi te je na primjer font u uputama veći u odnosu sa fontom u ljestvici najboljih rezultata.

4.3. JavaScript

U ovom radu postoje tri JavaScript datoteke koje su međusobno povezane export i import naredbama, koje služe za izvoz i uvoz vanjskog skriptnog sadržaja u neku skriptu. Osnovna datoteka jest *app.js* i u ovoj datoteci se nalaze sve funkcionalnosti i parametri igre. Druge dvije datoteke su *database.js* i *leaderboard.js.* Ove dvije datoteke su povezane jer im je ukupna svrha spremanje, dohvaćanje i prikazivanje imena i rezultata iz baze podataka.

4.3.1. Povezivanje web aplikacije sa Firebase-om

U *database.js* datoteci su definirane poveznice za pristup Firebase bazi podataka koje su dobivene iz uputa tijekom stvaranja baze na Firebase stranici. Firebase je web portal koji sadrži alate i proizvode koji su korisni pri izradi web i mobilnih aplikacija. Alati za rukovođenje aplikacija jesu

baza podataka, daljinska konfiguracija, funkcije na oblaku, hosting web stranice te baza podataka u stvarnom vremenu koja je korištena za ovu web aplikaciju. Dodavanje projekta na Firebase-u je jednostavno, potreban je unos imena projekta te imena direktorija u koje će se spremati podaci. Nakon unosa potrebnih podataka o imenovanju projekta Firebase generira varijable potrebne za spajanje same baze sa aplikacijom te slijedi povezivanje u JavaScript datoteci.

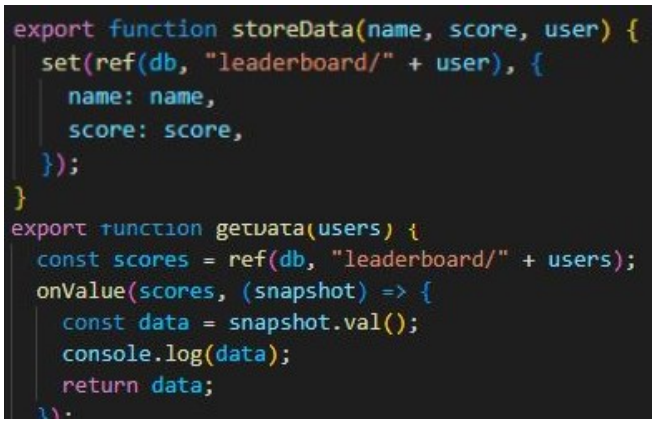
Prvi korak pri povezivanju web aplikacije u *database.js* datoteci jest da se definira Firebase konfiguracija. Unutar konstante „firebaseConfig“ potrebno je putem Firebase portala definirati varijable za API ključ, domenu, URL baze podataka, identifikacijski broj projekta, pošiljatelj poruke te identifikacijski broj aplikacije. Sve ove varijable su dostupne na stranici. Nakon definiranja te konstante potrebno ju je inicijalizirati putem *initializeApp()* funkcije te je spremiti u novu konstantu. Na kraju se poveznica za bazu podataka definira drugom konstantom „db“ te joj se pridružuje vrijednost izlaza funkcije *getDatabase()*. Uspješnim povezivanjem na bazu podataka nam omogućuje rukovanje s istom korištenjem naredbi poput *set()* i *onValue()* koje će biti korištene u *storeData()* i *getData()* funkcijama.

4.3.2. Metode *storeData()* i *getData()*

StoreData() metoda prima parametre „name“, „score“ i „user“. Funkcijom *set()*, koja kao parametar prima *ref()* funkciju, se postavljaju varijable „name“ i „score“ u bazu podataka. Funkcija *ref()* služi za dohvaćanje putanje direktorija i „user“ poddirektorija baze podataka gdje će se spremiti podaci.

getData() metoda prima parametar „users“ iz kojih se u bazi podataka dohvaćaju podaci kao objekti koji sadrže „name“ i „score“. Dohvaćaju se *onValue()* ugrađenom funkcijom koja prima putanju baze te elemente dohvaća kao „snapshot“ vrijednost. „Snapshot“ vrijednosti se spremaju u konstantu „data“ te se ta konstanta vraća pozivom ove funkcije.

Obe funkcije, slika 4.1., su izvožene naredbom export jer se primjenjuju u *leaderboard.js* datoteci, ali se nalaze unutar *database.js* datoteke.



Slika 4.1. Metode *storeData()* i *getData()*

4.3.3. Leaderboard.js

U *leaderboard.js* datoteci su pozivane funkcije iz *database.js* datoteke za rad sa podacima te ova datoteka služi kao medijator između baze podataka i prikaza rezultata na ljestvici. U datoteci je deklarirano prazno polje „scores“ koje se popunjava pozivajući metodu *getData()* u for petlji. Nakon popunjavanja polja poziva se *updateLeaderboardView()* metoda za sortiranje i prikaz elemenata na stranici. U ovoj datoteci se također nalazi i metoda *submitToDB()* za pohranu podataka.

4.3.4. Metoda *updateLeaderboardView()*

Zadaća ove metode jest sortiranje i prikaz elemenata u pravom redoslijedu. Sortiranje se obavlja pozivajući *sort()* funkciju na elemente „scores“. Zatim se deklarira prazno polje elemenata koje će se popunjavati „scoreRow“ elemenetima. U „scoreRow“ elemente se dodaju novonastali div elementi „name“ i „score“ funkcijom *appendChild()*, prolaskom kroz for petlju.
Također u ovoj funkciji, nakon for petlje za dodavanje elemenata, postoje još dvije for petlje za dodavanje boja svakom „elementu“, redoslijedom od prvog do trećeg - zlatna, srebrena, brončana te bijela za ostale elemente.

4.3.5. Metoda *submitToDB()*

Zadaća ove metode jest da pohranjuje ime i rezultat u bazu podataka nakon klika na gumb submit. To se omogućuje putem if naredbe gdje je uvjet da input blok nije prazan, te ako je uvjet zadovoljen poziva se metoda *storeData()* sa odgovarajućim parametrima „name“ i „score“. „Score“ parametar se uvozi iz *app.js* datoteke te predstavlja „count“ varijablu u toj datoteci. Parametar „user“ je lokalna varijabla sa vrijednosti „user“, a mijenja se dodavanje varijable „num“ koja se povećava svakim uspješnim izvođenjem metode. Ova metoda se poziva na *onclick()* funkciju gumba.

4.3.6. App.js

Osnovna JavaScript datoteka za stvaranje svih funkcionalnosti igre. Da bi se uopće moglo početi sa davanjem funkcionalnosti određenim elementima prvo se trebaju ti elementi dohvatiti korištenjem document.QuerySelector-a. Elementi koji se dohvaćaju su „plane“, „gameDisplay“ i „ground“. Nakon dohvaćanja elemenata, sljedeći korak je definiranje varijabli koje će se koristiti kroz cijelu igru. Te varijable su: „planeLeft“, „planeBottom“, „gravity“, „isGameOver“, „gap“, „count“ i prazno polje „pipes“, te se one inicijaliziraju kroz *initVars()* metodu.

4.3.7. Metoda *startGame()*

Zadaća ove metode jest smjestiti element „plane“ na sredinu kontejnera igre te omogućiti spuštanje elementa da bi se dobio efekt gravitacije. Za smještanje elementa na sredinu kontejnera igre potrebno je vrijednostima plane.style.bottom i plane.style.left dodati prethodno definirane varijable planeBottom i planeLeft u točkama. Ove varijable su udaljenosti od ivica do sredine kontejnera igre u točkama. Postepenim oduzimanjem planeBottom varijable sa varijablom gravity, element „plane“ će se spuštati za iznos gravity varijable, to jest dvije točke. Ovim postupcima se dobivaju osnovne funkcionalnosti „plane“ elementa, a metoda je prikazana na slici 4.2.

```
function startGame() {
  updateDisplay();
  plane.style.bottom = planeBottom + "px";
  plane.style.left = planeLeft + "px";
  planeBottom -= gravity;
  if (isGameOver && planeBottom <= 200) {
    planeBottom += 75;
  }
}
```

Slika 4.2. *StartGame()* metoda

4.3.8. *Up()* metoda

Ovom metodom se omogućuje elementu „plane“ da se pomjera prema gore. Pomjeranje prema gore je omogućeno na sličan način kao postavljanje elementa na sredinu kontejnera igre, tako što se na vrijednost „plane.style.bottom“ dodaje „planeBottom“ varijabla koja unutar ove funkcije poprima vrijednost od 75 točaka, a u slučaju da „plane.style.bottom“ se nalazi na više od 520 točaka onda se prekida rad metode dok se element ne spusti na niže područje. Ako se „plane.style.bottom“ nalazi na razini višoj od 520 točaka, varijabla „planeBottom“ će se samo

mijenjati gravitacijom. Ova mjera opreza je postavljena iz razloga da element „plane“ ne može letjeti izvan vidnog polja kontejnera igre kao što je to moguće u jednim od sličnih rješenja.
Tipka koja se treba stisnuti da bi se element pomjerio prema gore jest u ovom slučaju „space“. To ograničenje je postavljeno uz pomoć funkcije *control(e)* koja samo provjerava da li je kod tipke strogo jednak kodu za tipku „space“ te ako jest izvršava se metoda *up()*.
Samim postojanjem ovih metoda korisnik neće moći upravljati elementom „plane“. Da bi bilo omogućeno upravljanje, dodaje se slušatelj događaja koji u slučaju stiska tipke poziva funkciju *control(e)* te ako je šifra stisnute tipe jednaka šifri tipke „space“, element će se pomjeriti prema gore.

4.3.9. Metode *generateObstacle()* i *moveObstacle()*

Ove dvije metode kao glavnu svrhu imaju generiranje elemenata koje će služiti kao prepreke u igri i pomjeranje tih prepreka sa desna na lijevo. Glavna zamisao jest stvoriti div element te dodati ga na kontejner igre. Za definiranje dimenzija prepreka korišten je CSS, a definiranje položaja prepreke se radi u ovoj metodi na način sličan *startGame()* metodi. Najprije su definiranje varijable za horizontalni položaj „obstacleLeft“ i vertikalni položaj „obstacleBottom“. Horizontalni položaj prvo je postavljen izvan samog kontejnera igre, a vertikalnom položaju određujemo nasumičnu visinu s razlikom od 60 točaka. Sljedeći korak je stvaranje div elemenata za gornju i donju prepreku i dodavanje ih u kontejner za igru sa metodom *appendChild()*. Nakon dodavanja ovih elemenata, njima se postavlja položaj da desnu stranu izvan kontejnera igre kako bi se dobio učinak da idu slijedom jedna za drugom te da korisnik ne vidim njihovo generiranje.
Gornja i donja prepreka moraju imati definiran razmak između sebe, te je to postignuto definiranjem varijable „gap“ koja ima veličinu od 60 točaka. Tim razmakom je postignuto da uz generiranje prepreka sa različitim visinama, razmak između njih se nikad ne mijenja. Samim generiranjem prepreka se ništa ne postiže jer se te prepreke generiraju samo u desnom dijelu kontejnera i tu ostaju. Za pomjeranje prepreka služi metoda *moveObstacle()*. Ova metoda se nalazi unutar metode *generateObstacle()* te koristi varijablu „obstacleLeft“ u svom izvršavanju. „ObstacleLeft“ varijabla se postepeno smanjuje za dva cijelih pet točaka, te se položaj „obstacle.style.left“ smanjuje za iznos ove varijable. Ovo vrijedi za donju i gornju prepreku te se ovim postiže pomjeranje obje prepreke s desna na lijevo. Prepreke se ne smiju beskonačno pomjerati u desnu stranu te se to sprječava dodavanjem uvjeta da ako je varijabla „obstacleLeft“ strogo jednaka -50 da se uklanjaju div elementi za prepreke iz kontejnera. Da korisnik web stranice ne vidi stvaranje i brisanje div elemenata, odnosno prepreka, u CSS datoteci je definiran okvir koji ide oko cijelog kontejnera igre sa z-indexom +2.

U ovoj funkciji se također nalazi uvjet za prekidanje igre, odnosno nalaze se uvjeti ako položaj elementa „plane“ se poklapa sa položajima prepreka ili ako položaj elementa „plane“ bude jednak sa položajem elementa „ground“, da se igra završi metodom *gameOver()*, ali ako se „obstacleLeft“ nalazi na točno 220, varijabla „count“ se povećava za jedan te to označava konačni rezultat. Posljednji uvjet metode *generateObstacle()* jest u slučaju da boolean varijabla „isGameOver“ nije istinita da se generiraju nove prepreka svake tri sekunde. Obe metode se vide na slici 4.3.

```
function generateObstacle() {
  let obstacleLeft = 500;
  let randomHeight = Math.random() * 60;
  let obstacleBottom = randomHeight;

  const obstacle = document.createElement("div");
  const topObstacle = document.createElement("div");

  if (!isGameOver) {
    obstacle.classList.add("obstacle");
    topObstacle.classList.add("topObstacle");
  }
  pipes.push(obstacle);
  pipes.push(topObstacle);

  gameDisplay.appendChild(obstacle);
  gameDisplay.appendChild(topObstacle);

  obstacle.style.left = obstacleLeft + "px";
  topObstacle.style.left = obstacleLeft + "px";
  obstacle.style.bottom = obstacleBottom + "px";
  topObstacle.style.bottom = obstacleBottom + gap + "p

  function moveObstacle() {
    obstacleLeft -= 2.5;
    obstacle.style.left = obstacleLeft + "px";
    topObstacle.style.left = obstacleLeft + "px";
    if (obstacleLeft === -50) {
      clearInterval(timerId);
      gameDisplay.removeChild(obstacle);
      gameDisplay.removeChild(topObstacle);
    }
    if (
      (obstacleLeft > 205 &&
        obstacleLeft < 275 &&
        planeLeft === 225 &&
        (planeBottom < obstacleBottom + 150 |
          planeBottom > obstacleBottom + gap
      planeBottom <= 0
    ) {
      gameOver();
      clearInterval(timerId);
    } else if (obstacleLeft === 220) {
      count++;
      updateDisplay();
    }
  }
```

Slika 4.3. Metode *generateObstacle()* i *moveObstacle()*

4.3.10. *GameOver()* metoda

Da bi se prekinulo generiranje novih prepreka i zaustavila kontrola elementa „plane“ postoji potreba za metodom koja to može učiniti. Metoda mijenja stanje varijable „isGameOver“ na istinito te to služi za prekid generiranja novih prepreka kroz uvjet u prethodno definiranoj funkciji. Funkcija također uklanja slušač događaja za stiskanje tipke za kontroliranje elementa „plane“ te je omogućen slušač događaja za tipku „Enter“ za ponovno pokretanje igre.

4.3.11. *InitGame()* metoda

Putem ove funkcije se inicijaliziraju funkcije za pokretanje igre, tj. unutar ove funkcije se „isGameOver“ varijabla postavlja na neistino te se pozivaju metode *startGame()* i *generateObstacle()*. Vrijednosti varijabli koje se inicijaliziraju se mogu vidjeti na slici 4.4.

```
function initVars() {
  planeLeft = 225;
  planeBottom = 255;

  gravity = 1.5;
  isGameOver = false;
  gap = 420;
  count = 0;
  pipes = [];

  plane.style.bottom = planeBottom;
  plane.style.left = planeLeft;
}
```

Slika 4.4. Inicijalizacija varijabli kroz metodu *initVars()*

4.3.12. *Start()* metoda

Ova metoda, na slici 4.5., ima ulogu pozivanja svih funkcija za pokretanje igre. Uz if uvjet da je pritisnuta tipka jednaka šifri za tipku „Enter“ i da je varijabla „isGameOver“ istinita, poziva se funkcija *setTimeout()* za metode *gameOver()*, *initVars()* i *initGame()*.

```
function start(s) {
  if (s.keyCode === 13 && isGameOver) {
    setTimeout(() => {
      gameOver();
      initVars();
      initGame();
    }, 1000);
  }
}
```

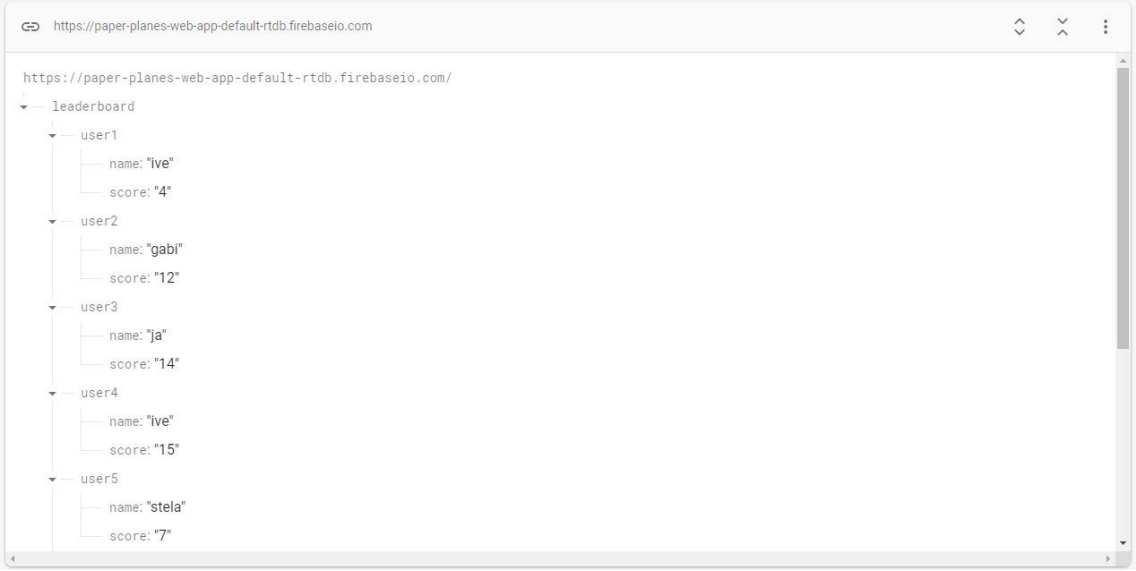
Slika 4.5. *Start()* metoda

4.4. FireBase

Za potrebu spremanja i dohvaćanja podataka potrebna je baza podataka, a tu svrhu ima FireBase. FireBase je jednostavno koristit jer se podaci ne trebaju spremati lokalno već ima opcija na stranici koja jednim klikom omogućava spajanje web aplikacije za bazu podataka što nudi.

Spajanje web aplikacije s FireBase bazom podataka je moguće putem skriptne tehnologije te je proces spajanja pojašnjen u prethodnom poglavlju. Nakon uspostavljanja veze između aplikacije i Firebase-a, na portalu je moguće odabrati tip baze podataka koji se želi implementirati. U ovom slučaju korištena je baza podataka u stvarnom vremenu. Također je moguće koristit uobičajenu bazu podataka za pohranjivanje objekata za kojih nema potrebe istovremeno dohvaćanje na web stranicu ili mobilnu aplikaciju. Kao što se vidi na slici 4.6. u dijelu stranice sa konzolom je moguće pregledati sve podatke koji su spremljeni u hijerarhijskom prikazu te je također omogućeno jednostavno upravljanje istima. Podatke je moguće dodavati i brisati unutar konzole. Ručno

brisanje podataka se lako može izvršiti klikom miša ili tipkom za brisanje. Ručno dodavanje podataka ima svoje poteškoće jer dodavanjem određenog podataka nije moguće birati tip podataka te aplikacija definiranim funkcijama nije u mogućnosti pročitati dohvaćeni objekt, već se dobije greška pri izvršavanju *database.js* dijela aplikacije i dolazi do prekida rada ljestvice rezultata.



Slika 4.6. Firebase baza podataka u stvarnom vremenu

4.5. Visual Studio Code

Razvojni alat korišten za pisanje koda ove web aplikacije jest Visual Studio Code. VS Code ima jednostavno sučelje za rad sa mnogim programskim jezicima ili označnim jezicima. Novu datoteku je lako stvoriti i ona na samom početku ima inačicu za odabir sintakse koja će se koristiti u tom dokumentu. Nakon odabira sintakse, dokument automatski dobiva odgovarajuće proširenje. Ovim razvojnim alatom su stvorena pet dokumenata koji se koriste u web aplikaciji, točnije su stvoreni jedan HTML dokument, jedan CSS dokument i tri JavaScript dokumenta. Za svaki jezik omogućen je olakšan input zbog raznih razloga od kojih se najviše istječu dovršavanje imena naredbe, nuđenje više naredbi koje počinju istom ključnom riječi te prikazivanje sintaktičkih grešaka prilikom pokretanja aplikacije. Pokretanje same aplikacije ili općenito koda je omogućeno kroz Run meni u kojem se može odabrati da li se želi pokrenuti uz ili bez otklanjanja mogućih grešaka. Kada se odabere opcija koja se želi, u primjeru web aplikacije, ako postoji više instaliranih internet preglednika na uređaju, VS Code daje upit kojim preglednikom korisnik želi pokrenuti kod. Dok je aplikacija pokrenuta, ako dođe do izmjene koda, korisnik ne mora opet pokretati aplikaciju, već može samo stisnuti kružić za osvježavanje koda te će se stranica automatski osvježiti sa novim kodom.

VS Code također nudi mnoga proširenja, no postoji jedno proširenje bez kojeg rad baze podataka ne bi bio moguć. To proširenje jest Live Server te je ono bilo potrebno prilikom spajanja na FireBase stranicu. Naime, ako je web aplikacija pokrenuta iz lokalnog direktorija, a ne na lokalnom serveru, CORS polica onemogućava spajanje sa Firebase bazom podataka. Live Server stvara lokalni server sa IP adresom preko koje je moguće pristupiti web aplikaciji.

4.6. Adobe Photoshop

Adobe Photoshop je najpoznatiji program za uređivanje i stvaranje slika. Relativno je zahtjevan program za razumijevanje svih funkcionalnosti, ali uz osnovno znanje svaki korisnik se može lako snaći i istraživati nove mogućnosti ovog razvojnog alata. Za potrebe web aplikacije, u Photoshop-u su kreirane četiri različite slike sa odgovarajućim dimenzijama kako bi se mogle implementirati bez grešaka. Kreirane su slike za pozadinu kontejnera igre, odnosno za „sky“ element, zatim za „ground“ element, „plane“ element te za prepreke. Sve slike su kreirane na način, da spojeni u jednu cjelinu, izgledaju kao 8-bitna igra, odnosno svaka slika se sastoji samo od točaka.
Za pozadinu kontejnera igre koristi se slika koja prikazuje grad u ruševinama s tamnijim nijansama. U originalnog igri, pozadina je također grad u puno većoj udaljenosti i velika praznina plavog neba. Ova pozadina sadrži puno više detalja i pridodaje estetici cijele stranice. Na slici 5.2. prikazana je usporedba pozadinske slike iz originalne igre i pozadinske slike ove web aplikacije.

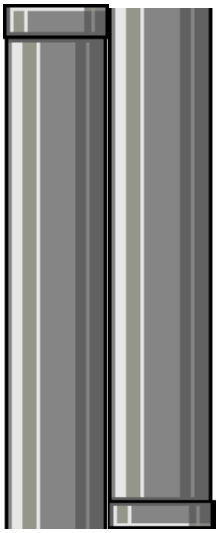
Slika 5.2. Usporedba pozadinskih slika igara

Za sliku „ground“ elementa korištena je slika koja prikazuje tip tvrde površine poput ceste ili pločnika sa raznim nijansama sive boje. Veoma je jednostavan dizajn te time pridodaje ravnoteži kontejnera igre. Također je urađena po uzoru na originalnu igru, samo sa izmjena uzorka gornjeg

obruba i promjena boja. Na slici 4.3. prikazana je usporedba slika korištenih kao „ground“ elementi u originalnog „Flappy Bird“ aplikaciji i u ovoj web aplikaciji.

Slika 4.3. Usporedba slika korištenih kao „ground“ elementi

Za donju i gornju prepreku, slika 4.4., korištene su slične cijevi kao i u originalnoj igri samo sive boje, umjesto jarko zelene boje. Prikazane su i gornja i donja prepreka jer one nisu potpuno iste. Pri završetku osnovne prepreke nije bilo moguće samo okrenuti prepreku već je bilo potrebno i redizajnirati nijanse kako bi se nastavio uzorak s donje na gornju prepreku.



Slika 4.4. Prepreke

Kako bi se stvorio jak kontrast između svih elemenata, a osobito između elementa „plane“ i prepreka, za element „plane“ se koristi slika 4.5., slika papirnog aviona bijele boje sa nijansama sive i plave boje. Obrub elementa je crne boje da još više pridonese kontrastu te da ga korisnik može još bolje razlikovati od prepreka i okoline.

Slika 4.5. "plane" element

Sve slike zajedno stvaraju ravnotežu sive boje, dovoljno dobro da se svaki element može raspoznati jedan od drugog, što je veoma važno sa ovaj tip igre.

5. RAD S APLIKACIJOM

Web stranica u konačnici predstavlja jednostavnu igru sa jednostavnim dizajnom cijele web stranice. Ulaskom na web stranicu uočljive su upute za igranje s lijeve strane, ljestvica sa najboljim rezultatima s desne strane te sama igra u sredini. Prvim učitavanje web stranice igra je u fazi stajanja, tj. prepreke se ne generiraju, brojač je na nuli, te „plane“ element se kreće samo u sredini, s prizorom da leti u beskrajnost. Stiskom tipke „Enter“ s desne strane se krenu generirati prepreke i cilj igre je da „plane“ element prođe kroz razmake između prepreka, a da ne smije dirnuti ni stijenke prepreka. U slučaju sudara sa stijenkama, prepreke na ekranu se brišu te se prestaju generirati. Na brojaču je prikazan konačni rezultat, odnosno broj prijeđenih prepreka te korisnik ima opciju u desnom gornjem kutu da upiše svoje ime te klikom na gumb „submit“ spremi svoj rezultat u bazu podataka te da vidi na kojem se mjestu nalazi na ljestvici sa najboljim rezultatima. Ljestvica se u istom trenutku osvježava jer je rad s bazom podataka u stvarnom vremenu. Ako korisnik želi ponovo igrati samo stisne ponovo tipku „Enter“ i igra se ponovo pokrene, tj. brojač se resetira na nulu i prepreke se ponovo krenu generirati.

5.1. Pronalazak mogućih grešaka

Kako bi se osigurala potpuna funkcionalnosti igre izvršeno je uzastopno testiranje svih interaktivnih dijelova web stranice. Prvo se testirala sama igra. Igra sadrži dva interaktivna dijela i to tipke za start i za pomjeranje „plane“ elementa. Za tipku „Enter“ ustanovljeno je da ako uzastopnim pritiskom ove tipke u veoma kratkom vremenskom periodu igra se zaustavi na nekoliko sekundi te se brojač nikako ne promjeni, ali ako se naknadno klikne tipka „Enter“ samo jednom onda igra normalno funkcionira. Testirane su i granice dokle „plane“ element može doći u visinu, te igra ne dozvoljava prelaženje elementa izvan granica kontejnera igre i uvjet za onemogućavanje pomjeranja elementa prema gore ako se nalazi na razini od 520 točaka je zadovoljen. Testirana je i kolizija između prepreka i „plane“ elementa te je ustanovljeno da „plane“ element sadrži nevidljive granice, tj. slika u elementu ne popunjava dobro sve praznine te se to očituje u nevidljivim dodirnim točkama elementa.

Nakon pronalaženja grešaka igre, testirana je i baza podataka, odnosno testirano je sučelje u koje se upisiva ime, pohrana svih podataka i dohvaćanje svih podataka iz baze. Uzastopnim pokušavanjem spremanja rezultata bez upisa ikakvih znakova ustanovljeno je da je nemoguće pohraniti podatke u bazu ako je varijabla „name“ jednaka null elementu, ali ako je uneseno ime u to polje moguće ga je pohraniti kolikogod puta se klikne na gumb „Submit“. Sa dohvaćanjem

podataka nije bilo problema, svaki put kad se unese novi rezultat odmah se učita na ljestvici rezultata.

5.2. Uklanjanje grešaka

Prva greška koja se pojavila tijekom testiranja jest uzastopnim pritiskanjem tipke „Enter“ igra se nekad pokrene i odmah zaustavi. Ova greška je uklonjena postavljanjem *setTimeout()* funkcije u if-u za provjeravanje da li je stisnuta tipka „Enter“, s najmanjim vremenom od 800 ns. Sljedeća greška jest sa kolizijom „plane“ elementa sa preprekama. Popravak ove greške je jednostavan i bio je potreban mali redizajn postojeće slike u „plane“ elementu tako da je svaki kut slike popunjen sa bojom kako bi se lakše detektirala kolizija. Posljednja uočena greška jest sa pohranom podataka u bazu ako su podaci svi isti. Dodavanjem upita u metodu *storeData()* da ako baza sadrži objekt sa istim „name“ i „score“ vrijednostima s objektom koji se želi pohraniti, onda se prekida izvođenje funkcije. Ovim rješenjem je uklonjena i posljednja uočena greška tijekom testiranja.

6. ZAKLJUČAK

Ovim završnim radom opisan je postupak stvaranja web aplikacije korištenjem poglavito skriptne tehnologije. Za stvaranje web stranice korišteni su razni alati poput HTML-a i CSS-a koji su služili za određivanje poretka svih elemenata web stranice te njihov odgovarajući dizajn. Glavni dio završnog rada jest detaljan opis svih funkcija unutar svih JavaScript datoteka kako bi se pojasnio čitav postupak izrađivanja web aplikacije. Uz opis svake funkcije pojedinačno, također su opisani i drugi alati i tehnologije koje su bile značajne u izradi stranice i aplikacije.

Web aplikacija služi za jednostavno i kompetitivno iskustvo što nudi na način da zahtjeva preciznost i veliku koncentraciju pri igranju. Također omogućuje uspoređivanje rezultata sa drugim korisnicima te na taj način stvaranja navike da korisnik postane bolji, a da se to očituje u rezultatima. Korisnik može često posjećivati stranicu iz dosade ili zbog velikog natjecateljskog duha.

Web aplikacija je naravno po uzoru na igru „Flappy Bird“ te je cijeli završni rad inspiriran ovom igrom jer je ostavila velik učinak u to doba dok je postojala te neočekivano uklonjena.

LITERATURA

[1] flappybird.io, dostupno na: https://flappybird.io/ [5.9.2022.]

[2] Playcanv.as, dostupno na: https://playcanv.as/p/2OlkUaxF/ [5.9.2022.]

[3] Flappy Bird, Unity WebGL Player, dostupno na: http://personal.denison.edu/~lalla/314f20/ladi/index.html [5.9.2022.]

[4] Flappy Bird – proširenje za Mozilla FireFox pretraživač, dostupno na: https://addons.mozilla.org/hr/firefox/addon/flappy-bird/ [5.9.2022.]

[5] Flappy Bird Multiplayer – Gameforge.com, dostupno na: https://gameforge.com/hr-HR/littlegames/flappy-bird-multiplayer/ [5.9.2022.]

[6] W3Schools, dostupno na: https://www.w3schools.com/ [6.9.2022.]

[7] FireBase, dostupno na: https://firebase.google.com/ [7.9.2022.]

SAŽETAK

U uvodnom dijelu rada dane su web aplikacije također napravljene po uzoru na originalnu „Flappy Bird“ igru, po čijem uzoru je napravljen završni rad. Uzimajući u obzir sve prednosti i nedostatke mogućih sličnih rješenja ovaj završni rad je težio eliminirati sve nedostatke sličnih rješenja, ali prilagoditi i sve prednosti. Cijela web aplikacija se temelji na isključivo JavaScript tehnologiji, te su omogućene sve funkcionalnosti kao što ima originalna igra. Osim skriptne tehnologije za izradu i dizajniranje igre su također korištene HTML i CSS datoteke koje su povezivale izgled cijele web aplikacije u jednu komponiranu cjelinu. Dizajn svih elemenata igre je ostvaren pomoću Photoshop aplikacije, ali svi prizori i slike su nalik sličnim rješenjima web aplikacije. Korištenjem Firebase stranice omogućeno je lako upravljanje bazom podataka u stvarnom vremenu, a postavljanje iste je jednostavno kroz upute na Firebase stranici. U konačnici igra je napravljena povezivanjem svih funkcija i datoteka u jednu cjelinu.

Ključne riječi: „Flappy Bird“, igra, „Paper Planes“ , platformer, skriptna tehnologija

ABSTRACT

Application of client script technology in creating a computer game

In the introductory part of the paper, web applications were given which were also modelled after the original "Flappy Bird" game as was this paper. Considering all the advantages and disadvantages of possible similar solutions, this paper tried to eliminate all the disadvantages of similar solutions, but also adapt all the advantages. The entire web application is based on exclusively JavaScript technology, and all the functionalities like the original game are embedded. In addition to script technology, HTML and CSS files were also used to create and design the game, which connect the appearance of the entire web application into one balanced ensemble. The design of all game elements was realized using the Photoshop application, but all images are similar to the other web application solutions. By using the Firebase page, it is possible to easily manage the real-time database, and setting it up is simple through the instructions on the Firebase page. Ultimately, the game is made by connecting all the functions and files into one whole

Key words: „Flappy Bird “, game, „Paper Planes “, platformer, script technology

ŽIVOTOPIS

Krešimir Matić rođen je 14.06.2000. godine u Zagrebu, s prebivalištem u općini Vitez, Bosna i Hercegovina. Osnovnu i srednju školu je završio u Vitezu, te 2019. godine se upisuje na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku na preddiplomski sveučilišni studij Računarstvo.

Tijekom osnovnoškolskog i srednjoškolskog obrazovanja sudjelovao je na brojnim natjecanjima iz matematike, te je u 4. razredu srednje škole osvojio drugo mjesto na školskom natjecanju u Mostaru. Učio je i usavršio sporazumijevanje na engleskom jeziku tijekom cijelog školovanja.

_____

Potpis autora

PRILOZI

1. „Primjena klijentske skriptne tehnologije u izradi računalne igre“ u .docx formatu
2. „Primjena klijentske skriptne tehnologije u izradi računalne igre“ u .pdf formatu
3. Izvorni kod web aplikacije