Zvonimir Ivković

Zivkov00@fesb.hr

Seminar iz kolegija Računalna grafika (250)

U Visual Studio Code razvojnom okruženju napisana u Javascript programskom jeziku

A close-up of a ring

Description automatically generated with low confidence

Seminar iz kolegija Računalna grafika (250)

Split, Siječanj 2022.

Sadržaj

[1. Uvod 1](#_Toc92903378)

[1.1. Opis Seminara 1](#_Toc92903379)

[1.2. Visual Studio Code 1](#_Toc92903380)

[1.3. JavaScript 2](#_Toc92903381)

[2. Opis igre 5](#_Toc92903382)

[2.1. Funkcionalni zahtjevi 5](#_Toc92903383)

[2.2. Use-case dijagram 5](#_Toc92903384)

[2.3. Stanja igre 5](#_Toc92903385)

[2.4. Slike objekata 5](#_Toc92903386)

[3. Praktični dio seminara 7](#_Toc92903387)

[3.1. Kratak opis igre 7](#_Toc92903388)

[3.2. Strukturna organizacija igre 7](#_Toc92903389)

[3.3. Main.js 7](#_Toc92903390)

[3.4. Display.js 11](#_Toc92903391)

[Popis Literature 12](#_Toc92903392)

# Uvod

U prvom poglavlju predstavit će se opće informacije vezan uz ovi seminar, 2d igru koja će se izradit, te će se ukratko opisati jezik i okruženje koje će se koristiti u svrhu realizacije seminara.

U drugom poglavlju će se opisati igra, njeni funkcionalni zahtjevi, te će se prikazati i opisati sva stanja i objekti korišteni u realizaciji igre.

Treće i zadnje poglavlje sadrži praktični dio igre, odnosno detaljan uvid i opis programskog koda, povezivanje dijelova igre i njeno funkcioniranje.

## Opis Seminara

Za potrebe realizacije ovoga seminara kreirana je igra Rabbit Ttrap namijenjena operacijskom sustavu Windows pomoću Visual Studio Code okruženja. Kod igre je pisan u Javascript programskom jeziku pomoću kojeg su kreirani i međusobno povezani objekti nužni za realizaciju igre. Također pošto se igra pokreće unutar web preglednika, napravljene su potrebne css i html datoteke.

## Visual Studio Code

Visual Studio Code je uređivač izvornog koda koji je napravio Microsoft za Windows, Linux i macOS. Sadrži sljedeće značajke[1]:

* Može se koristiti s raznim programskim jezicima, uključujući Java, JavaScript, Go, Node.js, Python i C++
* Ugrađena podrška za Git
* Mogućnost otklanjanja pogreški (engl. debugging)
* Isticanje sintakse koje omogućuje lakše utvrđivanje pogrešno napisane sintakse
* Inteligentno dovršavanje koda (engl. IntelliSense) pomoću kojeg se brže i točnije piše kod
* Podrška za kod isječke (engl. code snippets)
* Podrška za refaktoriranje koda
* Mogućnost nadogradnje sučelja pomoću ekstenzija

Visual Studio Code se pokazao kao favorit kod programera. U anketi Developers Survey 2021. Visual Studio Code je rangiran kao broj 1, a koristi ga 70% od 82.000 ispitanika. Pokazao se kao favorit radi svoga elegantnog, jednostavnog sučelja i svojih bogatih mogućnosti. Također promovira dobru praksu pisanja urednog i čitljivog koda pomoću automatskog uvlačenja, zatvaranja zagrada, itd. Također omogućuje suradnju članova tima preko ugrađene Git podrške, te također pomoću LiveShare ekstenzije omogućuje mijenjane i otklanjanja pogreški unutar koda u stvarnom vremenu[2].



Slika 1.1 Logo Visual Studio Code-a

## JavaScript

JavaScript je jednostavan, interpretiran programski jezik namijenjen ponajprije razvoju interaktivnih HTML stranica. Jezgra JavaScript-a uključena je u većinu današnjih preglednika[3]. JavaScript je jezik visoke razine. Ima dinamičko tipkanje, objektnu orijentaciju temeljenu na prototipu i prvorazredne funkcije. Podržava programiranje orijentirano na događaje (engl. event-driven programing), funkcionalno programiranje, i imperativno programiranje. Ima aplikacijsko sučelje (engl. application programing interface, API) za rad s tekstom, datumima, regularnim izrazima, standardnim strukturama podataka, i modelom objekta dokumenta (engl. Document Object Model).

JavaScript sadrži standardnu ​​biblioteku objekata, kao što su **Array**, **Date** i **Math**, te osnovni skup jezičnih elemenata kao što su operatori, kontrolne strukture i iskazi. Jezgra JavaScripta može se proširiti za razne svrhe dopunjavajući ga dodatnim objektima; na primjer[4]:

* *JavaScript na strani klijenta* proširuje osnovni jezik opskrbljujući objekte za kontrolu preglednika i njegovog *modela objekta dokumenta* (DOM). Na primjer, proširenja na strani klijenta omogućuju aplikaciji da postavi elemente na HTML obrazac i odgovori na korisničke događaje kao što su klikovi mišem, unos obrasca i navigacija stranicama.
* *JavaScript na strani poslužitelja* proširuje temeljni jezik isporukom objekata relevantnih za pokretanje JavaScripta na poslužitelju. Na primjer, proširenja na strani poslužitelja omogućuju aplikaciji komunikaciju s bazom podataka, osiguravaju kontinuitet informacija od jednog pozivanja do drugog aplikacije ili obavljaju manipulacije datotekama na poslužitelju.

JavaScript omogućava izvršavanje radnji u inače statičnim HTML dokumentima, npr. omogućava interakciju s korisnikom, promjenu svojstava prozora preglednika, ili dinamičko stvaranje HTML sadržaja.

Logo

Description automatically generated

Slika 1.2 Neslužbeni logo JavaScript

Program koji obrađuje i izvršava skripte zove se interpreter. Interpreter čita kod i prevodi ga u strojni jezik svakog puta kada se pokrene skripta. Svaki jezik koji se interpretira, tj. koji izvršava interpreter, naziva se skriptni jezik. Interpreter za JavaScript ugrađen je u većinu današnjih web preglednika.

Razvoj programa je znatno jednostavniji uz skriptne jezike, zato što se kod skriptnog jezika ne treba prevoditi u strojni jezik.

Koraci kod skriptnih jezika su sljedeći:

1. Napisati ili popraviti skriptu
2. Pokrenuti interpreter
3. Za potrebne popravke ponovno krenuti od 1. koraka

HTML je jezik koji se koristi za opis dokumenata i nema dinamičnih elemenata. Ukazala se potreba za uvođenje dinamičnog načina stvaranja HTML elemenata i stvaranje interaktivnog sadržaja u HTML-u. Danas za to postoji mnogo tehnologija, no u ovome radu je najinteresantnija skupina takozvanih klijentskih jezika.

Kod klijentskih jezika se interpretira na klijentskog strani u web pregledniku. Glavni predstavnik klijentskih jezika je JavaScript. Danas je podrška za JavaScript izvrsna u svim preglednicima, tako da se danas autori preglednika natječu u brzini izvođenja određenih algoritama, dok su se prije natjecali u podršci.

Document Object Model (DOM) je model za prikaz i interakciju s objektima u HTML dokumentu. Omogućava jednoznačan i jednostavan pristup dijelovima HTML dokumenta te rukovanje njegovim dijelovima, npr. elementi u HTML dokumentu. JavaScript definira svoj DOM u obliku hijerarhijske strukture prikazane na

Diagram

Description automatically generated

Slika 1.3 JavaScript DOM

Svakom objektu ili svojstvu pristupa se kroz taj model, tj. document je osnovni objekt preko kojeeg se pristupa svim drugim objektima dokumenta.

# Opis igre

U ovome poglavlju je detaljno opisana igra, te su dane informacije o funkcionalnim zahtjevima igre te su prikazana stanja i slike objekata koje su korištene u realizaciji igre.

## Funkcionalni zahtjevi

Unutar tablice prikazani su svi funkcionalni zahtjevi postavljeni u dvodimenzionalnoj Rabbit Trap igri.

Tablica 2.1 Funkcionalni zahtjevi

|  |  |  |
| --- | --- | --- |
| ID Zahtjeva | Naziv Zahtjeva | Detaljan opis |
| Z-1 | Pokretanje igre | Igra počinje kada korisnik u svoj web preglednik unese IP adresu servera na kojem je pokrenuta igra |
| Z-2 | Igranje igre | Korisnik koristi tipke na tipkovnici kako bi igrao igru, te igrica se prekida tek kada korisnik dođe predodređenog završetka igre |
| Z-3 | Jednostavnost | Igra treba biti dovoljno jednostavna kako bi je što veći broj korisnika mogao zaigrati, te kako bi što veći broj korisnika mogao doći do završetka igre |
| Z-4 | Uvid u broj sakupljenih objekata | Za vrijeme trajanja igre korisnik ima uvid u trenutni broj objekata koje je prikupio unutar igre |

## Use-case dijagram

Na dijagramu prikazan je use-case dijagram korisnika s radnjama koje može obavljati unutar aplikacije.

## Stanja igre

Igrica posjeduje samo jedno stanje, stanje Game. Kada se igra pokrene, korisnik dobije kratke upute kako igrati igricu. Korisnik se nalazi u Game stanju sve dok ne sakupi sve potrebne mrkve, ili dok igricu ne ugasi. Kada korisnik sakupi sve mrkve, dobije kratku obavijest da je uspješno sakupio sve mrkve i završio igricu, te ako pritisne na tipku OK, ponovno se pokreće stanje Game od početka. Iz igrice se izlazi tako da se ili ugasi web browser u kojem je pokrenuta igra, ili ako se ugasi web kartica u kojoj je igrica pokrenuta.

## Slike objekata

Slika potrebna za realizaciju Rabbit Trap igrice preuzeta je sa stranice [5]. Svi elementi potrebni za realizaciju igrice nalaze se unutar iste .png dadoteke. Na slici 2.1 prikazan je sadržaj rabbit-trap.png datoteke. Dimenzije rabit-trap.png slike su 128 x 128. Slika unutar sebe ima osam stupaca, te će se dijelovi slike „isjeći“ u male blokove veličine 16 x 16 piksela. Blokovi će se zatim slagati jedan pokraj drugoga kako bi se formirao svijet igre. Također će se brzom izmjenom ovih isječaka postići animacija igračevog lika, elemenata koje igrač treba skupiti, te animaciju trave koja se nalazi na platformama svijeta.

A picture containing text, electronics, circuit

Description automatically generated

Slika 2.1 Sadržaj datoteke rabbit-trap.png

# Praktični dio seminara

U ovom poglavlju su detaljno opisani, s obzirom na ovaj kolegij, najinteresantniji i najbitniji dijelovi koda platformerske igre u JavaScript-u.

## Kratak opis igre

## Strukturna organizacija igre

## Main.js

Unutar main.js funkcije nalazi se klasa AssetManager, koja ima ulogu učitavanja i spremanja rabbit-trap.png slike, i JSON datoteka pomoću kojih će se učitavati nivoi unutar igrice. Ova klasa unutar sebe ima funkcije requestJSON i requestImage. Koristimo JavaScript callback() funkciju kako bi se funkcije za kreiranje svijeta i pokretanje igre tek pokrenule nakon što se uspješno učita JSON i konvertira u JavaScript objekt, i nakon što se rabbit-trap,png uspješno učita. Pomoću eventListener procedure, gore navedene funkcije se tek pokreću kada se datoteke krenu učitavati, te pošto učitavanje traje, s pomoć callback() funkcije smo osigurali da se funkcije izvrše tek kada to učitavanje bude gotovo.

Text

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

Resize funkcija unutar main poziva funkciju resize koja se nalazi unutar Display.js. Pokreće se samo kada se izvrši potrebni događaj (engl. event) u ovom slučaju kada se promjeni veličina prozor web preglednika. Unutar nje se mijenja veličina platna, te time i svih elemenata koji se nalaze unutar platna (mrkve, igrač, platforme, itd.). Također se mijenja i veličina brojača sakupljenih mrkva.

Text

Description automatically generated

A screenshot of a video game

Description automatically generated

Slika 3.1 Igrač bez vertikalnog pomaka

Text

Description automatically generated

Funkcija render zadužena je za pozivanje funkcija definiranih unutar Display.js datoteke. U ovoj funkciji se šalju podatci učitani iz rabbit-trap.png i iz JSON objekata kako bi se generirao svijet. Obje for petlje prolaze kroz JSON datoteku koja definira zonu u kojoj se korisnik trenutno nalazi, čitaju njen sadržaj, te taj sadržaj prosljeđuje funkcijama za crtanje. Može se primijetiti da se svaki element pomiče za neki „offset“. To se radi kako bi elementi prirodnije i bolje pristajali unutar blokova svijeta. Za primjer na slici 3.1 možemo pogledati šta se dogodi ako izbrišemo vertikalni pomak igračevog objekta. Vidimo da igrač „propada“ kroz svijet igre. To se događa zato što smo iz rabbit-trap.png isjekli pravokutne sličice za igrača, te one nisu same prilagođene da savršeno pristaju generiranom svijetu. Zato ih je potrebno pomaknuti za nekoliko piksela.

Text

Description automatically generated

Update funkcija vrši ažuriranje igre. Unutar nje se poziva funkcija za pokretanje novog nivoa ako igrač prođe kroz vrata koja ga vode na drugi nivo. Vidi se da se igrica na tren zaustavi kada igrač prođe kroz vrata. Zaustavi se kako bi igrica imala vremena da uspješno učita JSON datoteku o drugom nivou, te uspješno nacrta sljedeći nivo. Nakon toga se igrica ponovno pokreće. Također se vidi da se igrica ažurira kada korisnik pritisne tipke za pomicanje lika.

## Display.js

Unutar Display.js datoteke se nalaze funkcije za crtanje svijeta i objekata u njemu, te funkcija za promjenu veličine.

Resize funkcija je dosta jednostavna. Svijet igre je smješten unutar platna (engl. canvas), te se veličina platna, a time i veličina svijeta igre, mijenja ovisno o veličini korisnikovog web preglednika. Kada je preglednik maksimalne veličine, platno je također svoje maksimalne veličine. Kako se veličina preglednika smanjuje, tako se smanjuje i veličina platna.

Text

Description automatically generated

Funkcije za crtanje objekata su kompleksnije. rabbit-trap.png ima unutar sebe 8 stupaca, a svijet igre ima 12 stupaca, s tim da je veličina svakog bloka unutar svijeta 16 x 16 piksela. source\_x i source\_y predstavljaju poziciju bloka unutar rabbit-trap.png kojeg ćemo isjeći. Ustvari namještamo x i y poziciju unutar rabbit-trap.png datoteke, te onda siječemo 16 piksela u širinu i 16 piksela u visinu kako bi formirali blok veličine 16 x 16, unutar kojeg se nalazi željeni dio slike. Zatim taj blok crtamo na željenu poziciju unutar svijeta igre. Isti postupak se vrši za svaki redak i stupac svijeta igre.

Text

Description automatically generated

## Game.js

Unutar datoteke Game.js se nalazi najveća količina najbitnijeg koda za kreaciju svijeta i animiranje objekata unutar svijeta. Većina klasa unutar ove datoteke su apstrahirane pomoću prefiksa Game., ustvari koristi se . sintaksa kako bi se te klase dodale unutar glavne Game klase. U principu ova apstrakcija je najbitnija za organizaciju koda, no korisna je i za odvajanje koda naše igrice od glovalno koda web stranice (na primjer ako naša igrica završi na web stranici preko koje je moguće pristupiti mnogim drugim igrama, želimo odvojiti naš kod od koda drugih igara).

Za početak najbolje je pogledati Frame i TileSet klasu.

Text

Description automatically generated

Frame klasa definira pravokutnu regiju iz koje ćemo isjeći sličicu iz rabbit-trap.png slike. Offset svojstva služe za pomicanje isjećka unutar našeg svijeta.

Graphical user interface

Description automatically generated with medium confidence

Unutar TileSet klase prati broj stupaca i veličinu svakog isjećka glavne slike. Animacije objekata unutar svijeta se postižu pomoću frames niza. Promotrimo prvi poziv Frame klase unutar niza. On predstavlja pravokutnik koji sjećemo iz slike rabbit-trap.png, te on predstavlja isjećak unutar kojeg se igračev lik, bijeli zec, nalazi u svojoj lijevoj mirnoj (engl. idle) poziciji. Grupa isjećaka u trećem i šestom redu niza predstavljaju sve isjećke koji formiraju lijevju i desnu animaciju kretnje. Same po sebi, ovaj niz neće animirati našeg igrača jer nije definirano kada se će se koji element niza pozvati. Unutar Player klase se definira kada će se pozvati koja sličica iz frames niza.

# Popis Literature

1. <https://en.wikipedia.org/wiki/Visual_Studio_Code>
2. <https://code.visualstudio.com/learn>
3. <https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf>
4. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
5. <https://github.com/pothonprogramming/pothonprogramming.github.io/tree/master/content/rabbit-trap>
6. <https://www.youtube.com/watch?v=w-OKdSHRlfA>