



RAČUNARSKA ELEKTRONIKA

Projektni zadaci

2019/20

22.03.2020.

Za svaki projekat potrebno je dostaviti:

- **Projektni izveštaj sa naslovom *REx_Izveštaj* u **PDF formatu****
Izveštaj sadrži opis vašeg projekta koje korisniku treba da omogući jasan i nedvosmislen uvid u realizaciju vašeg projektnog zadatka. U projektnom izveštaju **nije potrebno dodavati ceo kod** projektnog zadatka već samo delove koda ukoliko smatrate da je to neophodno u cilju pojašnjenja određenih funkcionalnosti. Projektni izveštaj treba da sadrži naslovnu stranu, sadržaj i treba da bude formatiran po sekcijama.
- **Folder sa naslovom *REx_Kod***
Kôd projektnog zadatka treba da bude modularan, čitljiv i razumljiv. Svaka logička celina koda treba da bude praćena odgovarajućim komentarima koji na nedvosmislen način upućuju na suštinu algoritma koji taj kod realizuje. Ovaj folder ne treba da sadrži objektne, izvrše i druge fajlove već samo *.asm* i *.inc* fajlove neophodne za pokretanje projekta.
- **Izvršni (EXE) fajl projekta**
Nakon uspešno odrađenog projektnog zadatka, kao rezultat kompajliranja dobićete odgovarajući *.exe* fajl koji takođe treba poslati sa projektom.

Izveštaj, folder sa *source* i *include* fajlovima kao i izvršni fajl sačuvati u folderu sa imenom **RE_X**, gde je **X** broj projekta. Folder je potrebno zapakovati u *.zip* format i poslati na email adresu haris@etf.rs. U *Subject* sekciju email-a staviti **OE3RE_Projekat_BBBB_GGGG** gde je **BBBB** broj indeksa a **GGGG** godina upisa.

Odbrana projekata će se organizovati po rasporedu koji će biti naknadno objavljen na sajtu predmeta. Projekti označeni * se mogu predati **prvi dan** ispitnog roka: *junskog, julskog, septembarskog ili oktobarskog*. Ostali projekti se predaju do *prvog dana junskog i julskog ispitnog roka*. Odbrana projekata će biti organizovana u danima pomenutog ispitnog roka po dogovoru.

Projekti se rade u grupama i svaka grupa treba da realizuje jedinstveno rešenje. Prepisivanje i kopiranje tuđih rešenja biće najstrožije sankcionisano!

Pregled projekata

| | |
|---|----|
| 1. Asteroids* | 4 |
| 2. 2048 * | 4 |
| 3. Simulacija raskrsnice | 5 |
| 4. Pronađi otpornik | 6 |
| 5. Simulacija led reklame* | 6 |
| 6. Network protocol | 7 |
| 7. Tekstualni editor I* | 9 |
| 8. Tekstualni editor II* | 10 |
| 9. Patch maker | 11 |
| 10. Apply patch | 12 |
| 11. Osmosmerka* | 12 |
| 12. 2D Racing game* | 13 |
| 13. Simulacija štoperice i tajmera | 13 |
| 14. Skočko* | 14 |
| 15. Testiranje brzine prekucavanja teksta | 14 |
| 16. Bubble trouble* | 14 |
| 17. Whack-a-Mole game* | 15 |
| 18. Moj broj | 15 |
| 19. Igra "Mice" za dva igrača* | 16 |
| 20. Black jack* | 16 |
| 21. Kovertor Bin-to-hex | 16 |
| 22. Loto | 17 |
| 23. Poligon za gađanje* | 17 |
| 24. Divided attention* | 17 |
| 25. Water glass* | 18 |
| 26. Repeat pattern game* | 18 |
| 27. Other hello game* | 19 |
| 28. Slide puzzle game* | 19 |
| 29. Huffman coding | 20 |
| 30. Mirror photo | 20 |
| 31. Parsiranje tekstualnog fajla | 20 |

Projektni zadaci

Ukoliko neki detalji realizacije nisu navedeni tekstem projektnog zadatka mogu se usvojiti razumne pretpostavke. Ukoliko postoje veće nedoumice oko realizacije projektnih zadataka javiti se mail-om na haris@etf.rs

1. Asteroids*

Potrebno je realizovati igricu Asteroids (*pogledati: <http://www.freeasteroids.org/>*). Na početku igrice na sredini ekrana se nalazi svemirski brod veličine 4 prazna bloka. Svake sekunde se pojavljuje po jedan asteroid sastavljen od 8 praznih blokova na proizvoljnim pozicijama. Svemirski brod pritiskom tastera SPACE ispaljuje mine oblika jednog praznog bloka. Mina se kreće unapred zdatom brzinom, a ako se i svemirski brod kreće onda je brzina mine jednaka: $vM = vM, const + 3 vB$, gde je vM brzina mine, $vM, const$ unapred zadata konstantna brzina mine i vB brzina broda. Brod može drugačije da se orijentiše pritiskom tastera levo i desno na tastaturi gde se svakim pristkom obće u smeru kazaljke na satu (odnosno obrnutom smeru od kazaljke na satu) za približno 10 stepeni . Pritiskom kursora na gore se brod ubrzava tako što se dodaje inkrement na njegovu brzinu, a taster na dole ga usporava. U toku kretanja u svemiru postoji i trenje koje se ogleda u tome što se na svakih nekoliko polja smanjuje brzina broda koji staje ukoliko nije pritisnut taster na gore (*pogledati video: [Asteroids.mp4](#)*). Kada mina udari u asteroid, asteroid biva raznet i nestaje, a broj bodova osvojen u igrici se uvećava za 100. Ukoliko svemirski brod udari u asteroid ili se novi asteroid pojavi na svemirskom brodu, onda se igrica završava i ispisuje se ukupan broj osvojenih bodova. Igrica se završava i pritiskom tastera ESC.

2. 2048 *

Napraviti igricu 2048 (*[https://en.wikipedia.org/wiki/2048_\(video_game\)](https://en.wikipedia.org/wiki/2048_(video_game))*). Igrica se igra na 4×4 mreži kao na slici 1. Na početku igrice na mreži se nasumično popunjava jedno polje kvadratom na kome je ispisan broj koji predstavlja stepen dvojke. Svi stepeni dvojke imaju odgovarajuću boju polja. Pritiskom kursora na tastaturi se popunjena polja pomeraju skroz levo, desno, gore ili dole. Ukoliko se pomeranjem dodirnu dva polja sa istim stepenom dvojke, ona se zamenjuju poljem čija je vrednost jednaka zbiru ova dva polja, a ukupan broj poena se inkrementira za tu istu vrednost. U slučaju ako se desi da postoje 3 susedna polja koja imaju istu vrednost, onda se dva koja su bliža onoj granici mreže na koju ukazuje pritisnuti kursor zamenjuju jednim poljem sa duplo većom vrednošću. Svaki put kada se pritisne neki od kursora na tastaturi, nakon pomeranja svih polja u smeru tog kursora, na neko od praznih polja se nasumično generiše novi element. Vrednost na polju se generiše nasumično i to 2 sa verovatnoćom $5/8$, 4 sa verovatnoćom $2/8$ i 8 sa verovatnoćom $1/8$. Igrica se završava pritiskom tastera ESC ili kada na mreži ne postoji više nijedno prazno polje, a susedni elementi ne mogu pomeranjem da daju jedno polje.

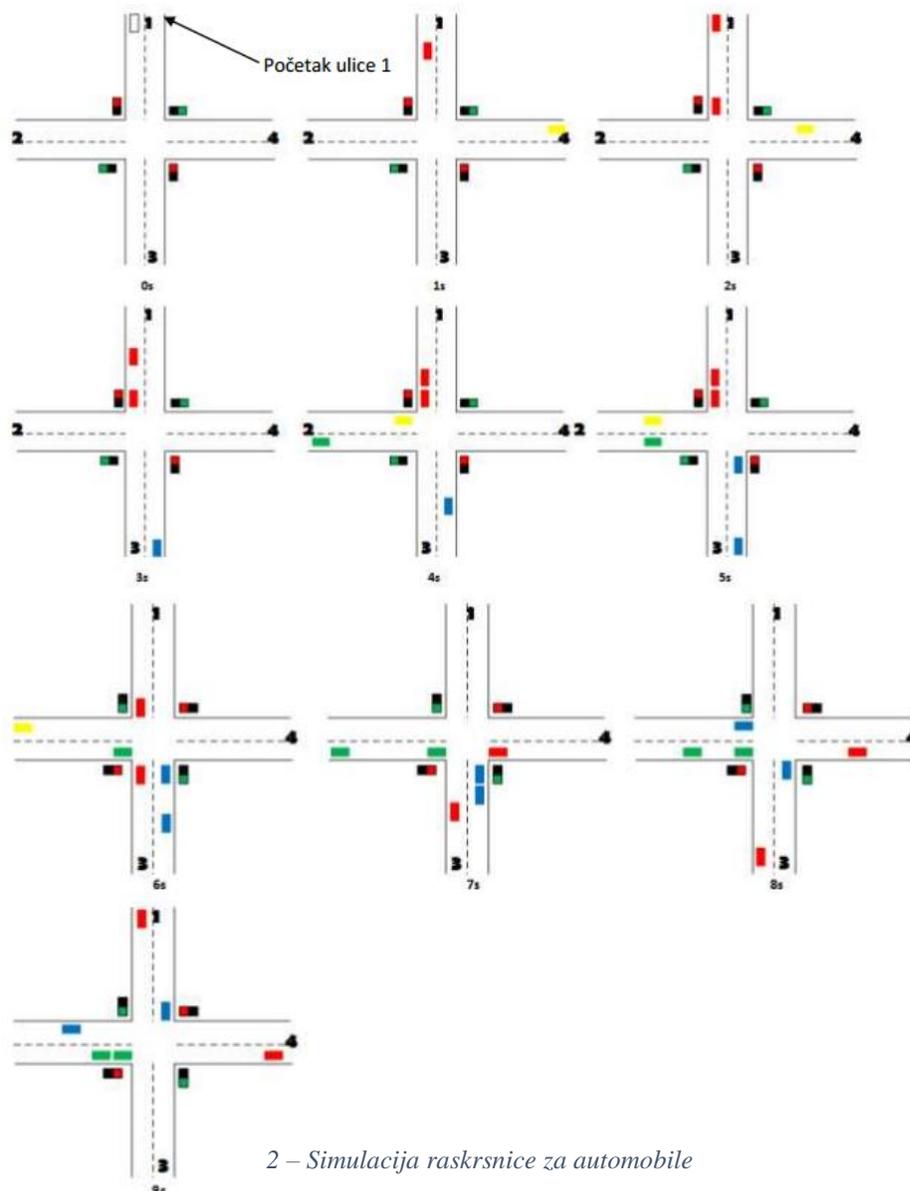
| | | | |
|--|---|----|----|
| | | 2 | 4 |
| | | 4 | 8 |
| | 2 | 16 | 32 |
| | 2 | 2 | 16 |

1 - Slika uz projekat 2.

3. Simulacija raskrsnice

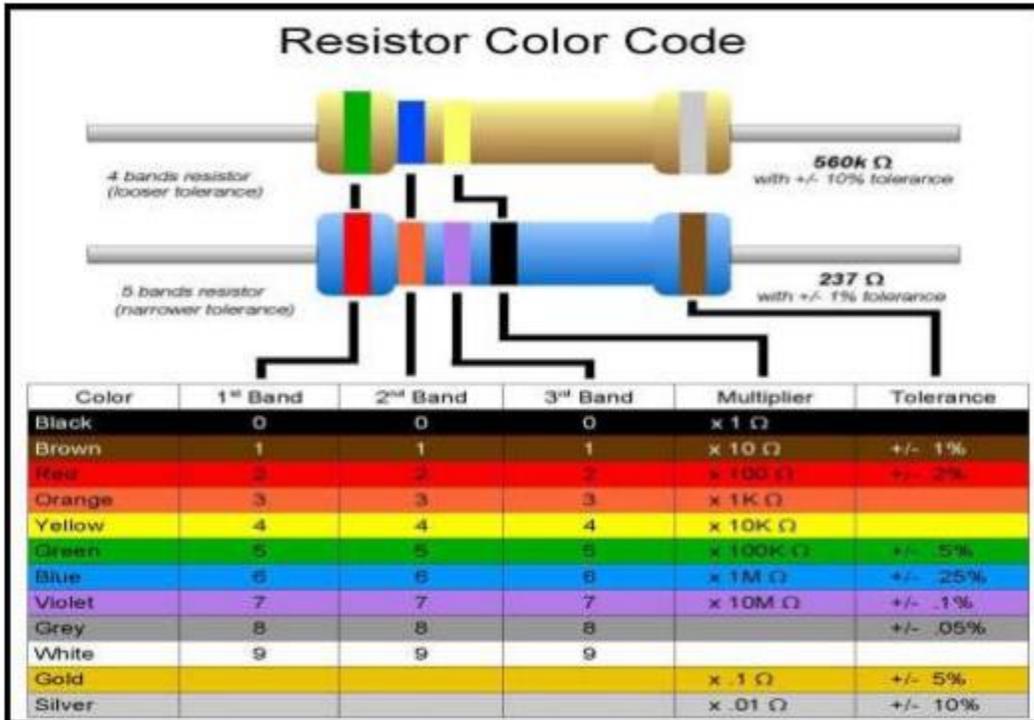
Na raskrsnici se ukrštaju 4 ulice (označene brojevima 1-4). U svakoj ulici se proizvoljno, sa periodom od 1s, generiše po jedan automobil. Automobil se uvek generiše na početku jedne od 4 ulice (Početak ulice – deo ulice koji se nalazi na krajevima konzole). U jednoj ulici se maksimalno mogu nalaziti dva automobila dok simulator može maksimalno prikazivati šest automobila na sve četiri ulice. Svakom od automobila treba 2s da stigne od mesta na kome je generisan do semafora raskrsnice. Semafori menjaju stanje sa periodom od 6s i rade u parovima 1-3, 2-4.

Potrebno je napisati program koji simulira kretanje automobila na raskrsnici. Ukoliko je na semaforu crveno svetlo za automobile koji dolaze iz neke od 4 ulice, ti automobili čekaju sve dok se ne uključi zeleno svetlo. Ukoliko je na semaforu zeleno svetlo automobili ulaze na raskrsnicu pod uslovom da neki od automobila iz suprotne ulice , u kojoj je takođe zeleno svetlo, ne želi da preseče put nekom od automobila. U ovoj situaciji se poštuje pravilo desne strane koje prednost daje onom automobilu koji dolazi sa vozačeve desne strane. Princip generisanja automobila, kretanja automobila i funkcionisanja saobraćaja dat je na slici 2.



4. Pronađi otpornik

Potrebo je realizovati program koji će studentima u lab. 39 😊 pomoći da na laboratorijskim vežbama brže nađu vrednost otpornika. Student unosi celobrojnu vrednost koja predstavlja željenu vrednost otpornika a program treba u konzoli da iscrta četiri pravougaonika čije boje predstavljaju tu vrednost (pravougaonik koji odgovara toleranciji se izostavlja). Program treba pretragom tabele otpornosti, koja je data na slici 4 , da nađe boje koje odgovaraju unetoj vrednosti. Ukoliko korisnik unese otpornost koja ne postoji u tabeli neophodno je obavestiti korisnika o tome a zatim naći vrednost u tabeli koja je najbliža unetoj vrednosti i za nju iscrtaati pravougaonike u odgovarajućim bojama.



Resistor Color Code

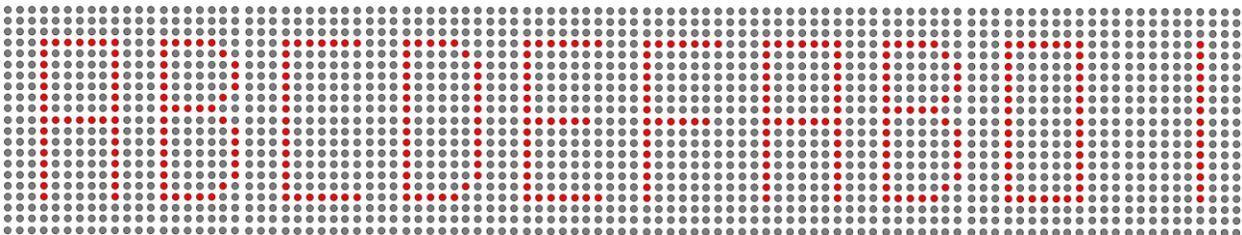
4 bands resistor (looser tolerance) **560k Ω** with +/- 10% tolerance

5 bands resistor (narrower tolerance) **237 Ω** with +/- 1% tolerance

| Color | 1 st Band | 2 nd Band | 3 rd Band | Multiplier | Tolerance |
|--------|----------------------|----------------------|----------------------|-----------------|-----------|
| Black | 0 | 0 | 0 | x 1 Ω | |
| Brown | 1 | 1 | 1 | x 10 Ω | +/- 1% |
| Red | 2 | 2 | 2 | x 100 Ω | +/- 2% |
| Orange | 3 | 3 | 3 | x 1K Ω | |
| Yellow | 4 | 4 | 4 | x 10K Ω | |
| Green | 5 | 5 | 5 | x 100K Ω | +/- 5% |
| Blue | 6 | 6 | 6 | x 1M Ω | +/- 25% |
| Violet | 7 | 7 | 7 | x 10M Ω | +/- 1% |
| Grey | 8 | 8 | 8 | | +/- .05% |
| White | 9 | 9 | 9 | | |
| Gold | | | | x .1 Ω | +/- 5% |
| Silver | | | | x .01 Ω | +/- 10% |

4 - Tabela otpornosti za projekat 4

5. Simulacija led reklame*



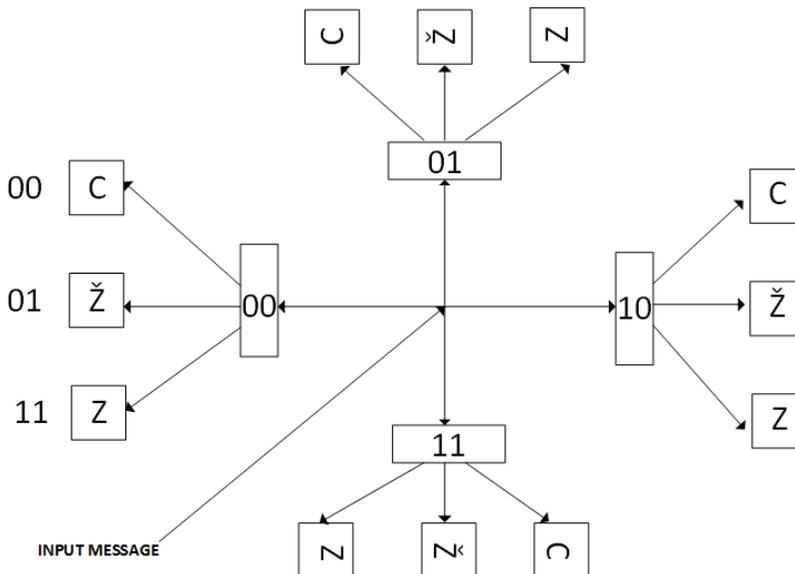
3 - LED Displej

Potrebno je napraviti program koji će u konzoli simulirati rad led displeja. Led displej je širine 100x21 piksela. Jedna dioda na led displeju predstavlja piksel. Jedno slovo na displeju se predstavlja grupom isključenih ili uključenih led dioda koje formiraju izgled nekog slova. Svako slovo zauzima površinu 15x7 piksela. Tekst koji možemo ispisati je maksimalne dužine 50 karaktera. Razmak između karaktera je 3 px. Izgled dela led displeja prikazan je na slici 3 . Ukoliko je dužina unetog teksta veća od 10 karaktera, koliko je moguće prikazati odjednom na displeju , potrebno je rotirati tekst na displeju tako da se omogući prikaz svih unetih ASCII karaktera.

Na početku programa se od korisnika traži da unese sekvencu ASCII karaktera . Zatim se uneta sekvencu ispisuje na ekranu na prethodno opisani način. Brzina rotacije se povećava pritiskom na taster UP(strelica na gore) a smanjuje pritiskom na taster DOWN (strelica na dole).

6. Network protocol

Na slici je prikazana topologija jedne mreže. Cilj ovog projekta je simulacija protoka poruka kroz mrežu i reagovanje uređaja (00 - 11) na odgovarajuće poruke. Preko mreže se može upravljati svim uređajima odjednom ili svakim uređajem ponaosob. Da bi se omogućilo upravljanje svakim uređajem ponaosob, svaki uređaj ima svoju jedinstvenu adresu (00,01,10,11).



5 - Topologija mreže

Svaki uređaj preko mreže dobija odgovarajuće instrukcije koje treba da prepozna i da ih izvrši. Instrukcije koje se mogu poslati preko mreže su :

- Uključi diodu
- Isključi diodu
- IDLE

Instrukcija „Uključi diodu“ uključuje diodu odgovarajuće boje dok instrukcija „Isključi diodu“ isključuje diodu odgovarajuće boje. Koju diodu treba isključiti ili uključiti se takođe šalje preko mreže. Instrukcija „IDLE“ ne sprovodi nikakvu akciju već samo uređaj prepoznaje da je bio adresiran. Poruke se „izbacuju“ na mrežu sa periodom T koju je potrebno konfigurisati.

Poruke koje se šalju na mrežu se čitaju iz fajla „*protocol.txt*“ koji u jednom redu ima osmobiitnu poruku. Poruke se iz fajla čitaju sa periodom T i šalju se ka svim uređajima. Struktura osmobiitnih poruka je prikazana na slici

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|------|----|------|----|-------|----|
| C | B | ADDR | | FUNC | | COLOR | |

C = 1 – Sistemska poruka
 C = 0 – Druge poruke
 B = 1 – Broadcast poruka
 B = 0 – Druge poruke
 ADDR = 00 – Uređaj 0
 ADDR = 01 – Uređaj 1
 ADDR = 10 – Uređaj 2
 ADDR = 11 – Uređaj 3
 FUNC = 00 – NOPE instrukcija
 FUNC = 01 – Ukljuci diodu zadate boje
 FUNC = 10 – Iskljuci diodu zadate boje

Bit D7 osmobaritne poruke nosi informaciju da li je u pitanju sistemska poruka. Sistemska poruka služi za startovanje prenosa, zaustavljanje prenosa i podešavanje periode slanja i primanja poruka po magistrali. Sistemske poruke su sledeće sadržine:

- 10000001 – Perioda čitanja 1s
- 10000010 – Perioda čitanja 0.5s
- 10000000 – Zaustavi prenos
- 11111111 – Započni prenos

Ukoliko je bit D7 setovan , u pitanju je sistemska poruka u suprotnom se posmatraju niži biti.

Ukoliko je setovan BIT D6 poruka koja se šalje je BROADCAST poruka. Na BROADCAST poruku reaguju svi uređaji na mreži. U slučaju BROADCAST poruke ne posmatraju se biti D5 i D4 ali se posmatraju biti D3-D0 koji nose informaciju o tome koju akciju treba sprovesti na uređajima. Ukoliko bit D6 nije setovan posmatraju se niži biti.

Bitima D5 i D4 određena je adresa uređaja u slučaju da je bit D6 jednak 0. To znači da na željenu instrukciju koja se šalje bitima D3 i D2 odgovara jedino uređaj čija adresa se poklapa sa vrednošću grupe bita D5 i D4.

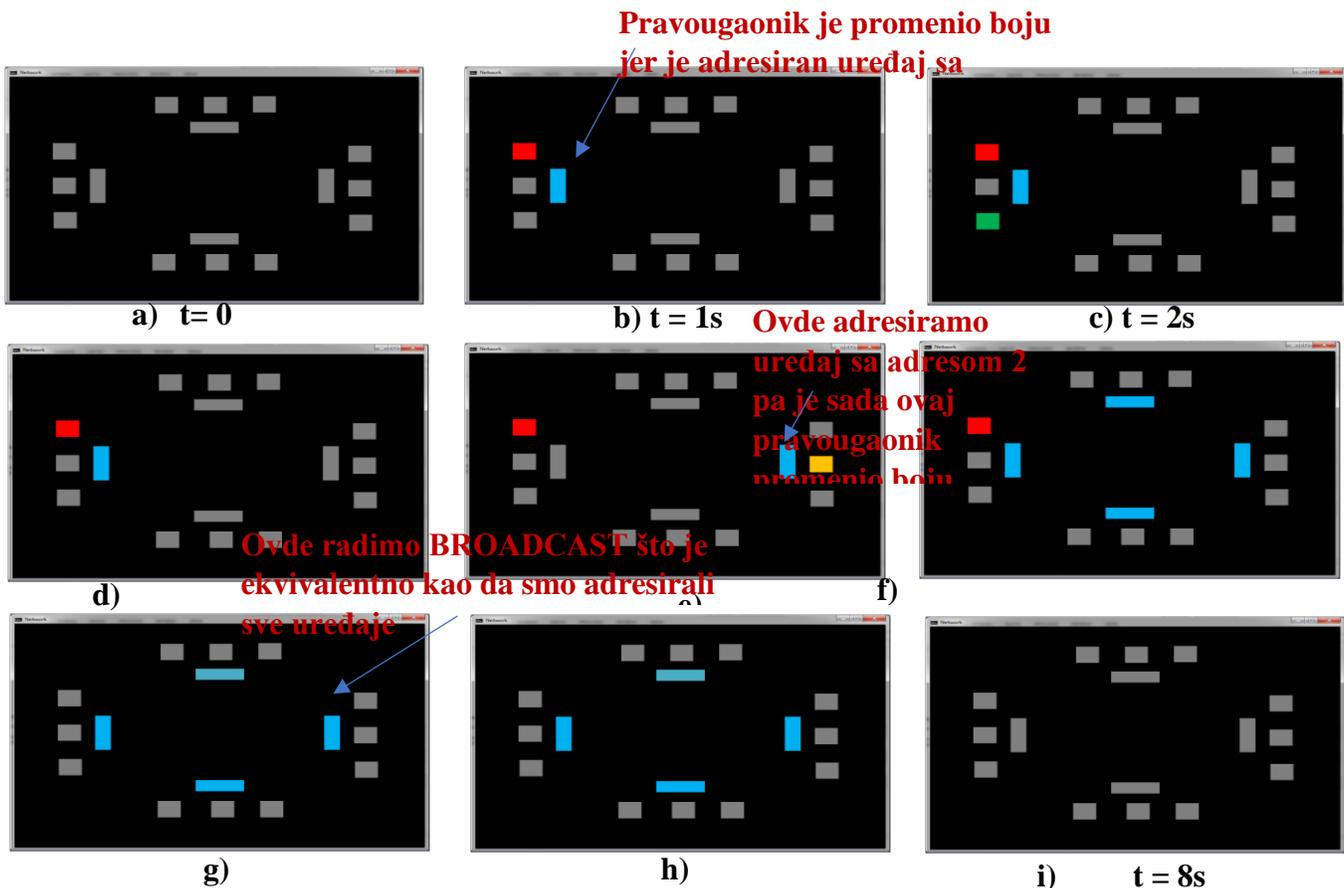
Biti D3 i D2 nose informaciju o instrukciji koja se šalje uređajima. Biti D1 i D0 nose informaciju o diodi koju treba uključiti ili isključiti.

Primer jednog *protocol.txt* fajla kao i željena funkcionalnost u konzoli dati su u sledećem primeru:

```
11111111 - Započni prenos (a)
10000001 - Perioda čitanja iz fajla je 1s (a)
00000100 - Uključi crvenu diodu na uređaju čija je adresa 0 (b)
00000111 - Uključi zelenu diodu na uređaju čija je adresa 0 (c)
00001011 - Isključi zelenu diodu na uređaju čija je adresa 0 (d)
00100101 - Uključi žutu diodu na uređaju čija je adresa 2 (e)
01101001 - Na svim uređajima isključi žutu diodu (f)
01101000 - Na svim uređajima isključi crvenu diodu (g)
01100000 - Svi uređaji dobili Nope Instrukciju , zadržavaju
. . . . .
```

Pravougaonik plave boje se aktivira uvek kada je neki uređaj adresiran. Pravougaonik menja boju sa periodom jednakoj periodi čitanja fajla.

Potrebno je potvrditi funkcionalnost projekta na barem 3 različita fajla *protocol.txt* .



6 - ilustracija funkcionalnosti mreže iz projekta 6

7. Tekstualni editor I*

U konzoli je potrebno napraviti tekstualni editor koji ima mogućnost da kreira, edituje i uklanja fajlove sa ekstenzijom `.txt`. Sva podešavanja tekstualnog editora nalaze se u fajlu `config.txt`. Podešavanja koja podržava tekstualni editor su:

- Boja slova
- Boja pozadine
- Putanja u kojoj se nalazi radni direktorijum (*Direktorijum koji sadrži tekstualne fajlove koji se edituju*)
- Naziv tekstualnog fajla (na primer `index.txt`), smeštenog u random direktorijumu, koji sadrži nazive svih tekstualnih fajlova koje je moguće editovati u radnom direktorijumu

Nakon pokretanja programa korisniku se otvara glavni meni koji nudi izbor između sledećih opcija :

1. Kreiranja novog tekstualnog dokumenta
2. Editovanja već postojećeg tekstualnog dokumenta
3. Brisanja postojećeg dokumenta
4. Pregled postojećih dokumenata

Ukoliko korisnik odabere kreiranje novog tekstualnog dokumenta, najpre se od korisnika traži da unese naziv tog dokumenta a nakon toga se uneti naziv upisuje u fajl `index.txt`. Nakon unosa naziva fajla,

u random direktoriju se kreira tekstualni fajl sa unetim nazivom a zatim se korisniku daje mogućnost da unese sadržaj fajla. Unos sadržaja fajla se vrši putem tastature. Kretanje kroz uneti tekst vrši se pomoću navigacionih tastera (strelica na tastaturi). Uklanjanje karaktera vrši se pritiskom na taster DELETE. Kraj unosa se identifikuje pritiskom na kontrolni taster (CTRL) nakon čega se na dnu konzole ispisuje poruka “Da li želite da sačuvate fajl?(Y-Da, N-Ne)”. Korisnik potvrđuje naredbu pritiskom na taster Y dok se pritiskom na slovo N poništava prethodno započeo proces čuvanja fajla i vraća na editovanje teksta.

Ukoliko korisnik odabere opciju editovanja već postojećeg tekstualnog fajla, od korisnika se najpre zahteva unos naziva postojećeg fajla. Ukoliko uneti naziv fajla postoji u *index.txt* tada se korisniku otvara konzola za editovanje teksta koja ima iste osobine kao konzola za unos teksta (opcija 1.) izuzev što se u ovom slučaju otvara konzola na kojoj je ispisan sadržaj tekstualnog fajla. Ukoliko fajl sa unetim nazivom ne postoji u *index.txt* datoteci potrebno je ignorisati unos i vratiti se u glavni meni.

Brisanje postojećeg dokumenta podrazumeva da korisnik najpre unese naziv tekstualnog fajla koji želi da ukloni iz radnog direktorijuma. Ukoliko tekstualni fajl sa unetim nazivom ne postoji potrebno je ignorisati unos i vratiti se u glavni meni. Ukoliko za uneti naziv fajla postoji evidencija u *index.txt* fajlu potrebno je izvršiti uklanjanje naziva fajla iz *index.txt*.

Ukoliko korisnik odabere pregled postojećih fajlova neophodno je izlistati sadržaj fajla *index.txt* u kome se nalaze svih postojećih fajlova.

Struktura *config.txt* i *index.txt* fajlova je data u okviru teksta projektnog zadatka

```
Red
Green
E:\RE_Project\
index.txt
```

config.txt

```
filename1.txt
filename2.txt
filename3.txt
```

index.txt

8. Tekstualni editor II*

U konzoli je potrebno napraviti tekstualni editor koji ima mogućnost da kreira, edituje i uklanja tekstualni sadržaj. Svi tekstualni sadržaji kreirani posredstvom ovog tekstualnog editora čuvaju se u okviru jednog fajla sa ekstenzijom.txt. Takođe, svaki tekstualni sadržaj ima jedinstveni naziv na osnovu koga ga je moguće pronaći u zajedničkom fajlu.

Sva podešavanja tekstualnog editora nalaze se u fajlu *config.txt*. Podešavanja koja podržava tekstualni editor su:

- Boja slova
- Boja pozadine
- Naziv tekstualnog fajla koji se koristi za čuvanje sadržaja svih tekstualnih sadržaja kreiranih posredstvom ovog editora (na primer *contents.txt*)
- Naziv tekstualnog fajla (na primer *index.txt*) koji sadrži nazive svih tekstualnih sadržaja kao i informacije potrebne za ekstrakciju tekstualnog sadržaja iz fajla *contents.txt*

Nakon pokretanja programa korisniku se otvara glavni meni koji nudi izbor između sledećih opcija :

1. Kreiranje novog tekstualnog sadržaja
2. Editovanje već postojećeg tekstualnog sadržaja
3. Brisanje postojećeg tekstualnog sadržaja
4. Pregled postojećih tekstualnih sadržaja

Ukoliko korisnik odabere kreiranje novog tekstualnog sadržaja, najpre se od korisnika traži da unese naziv tog sadržaja a nakon toga se uneti naziv upisuje u fajl *index.txt*. Nakon unosa naziva tekstualnog sadržaja, u random direktoriju se kreira tekstualni sadržaj sa unetim nazivom a zatim se korisniku daje

mogućnost da unese tekst. Unos teksta se vrši putem tastature. Kretanje kroz uneti tekst vrši se pomoću navigacionih tastera (strelica na tastaturi). Uklanjanje karaktera iz teksta vrši se pritiskom na taster DELETE. Kraj unosa se identifikuje pritiskom na kontrolni taster (CTRL) nakon čega se na dnu konzole ispisuje poruka “Da li želite da sačuvate tekstualni sadržaj?(Y-Da, N-Ne)”. Korisnik potvrđuje naredbu pritiskom na taster Y dok se pritiskom na slovo N poništava prethodno započet process čuvanja fajla i vraća na editovanje teksta.

Ukoliko korisnik odabere opciju editovanja već postojećeg tekstualnog sadržaja, od korisnika se najpre zahteva unos naziva postojećeg tekstualnog sadržaja. Ukoliko uneti naziv tekstualnog sadržaja postoji u *index.txt* tada se korisniku otvara konzola za editovanje teksta koja ima iste osobine kao konzola za unos teksta (opcija 1) izuzev što se u ovom slučaju otvara konzola na kojoj je ispisan tekst koji odgovara zadatom nazivu. Ukoliko uneti naziv ne postoji u *index.txt* datoteci potrebno je ignorisati unos i vratiti se u glavni meni.

Brisanje postojećeg tekstualnog sadržaja podrazumeva da korisnik najpre unese naziv teksta koji želi da ukloni iz fajla *contents.txt*. Ukoliko tekstualni fajl sa unetim nazivom ne postoji potrebno je ignorisati unos i vratiti se u glavni meni. Ukoliko za uneti naziv fajla postoji evidencija u *index.txt* fajlu potrebno je izvršiti uklanjanje naziva fajla iz *index.txt* i uklanjanje sadržaja iz *contents.txt* tekstualnog fajla.

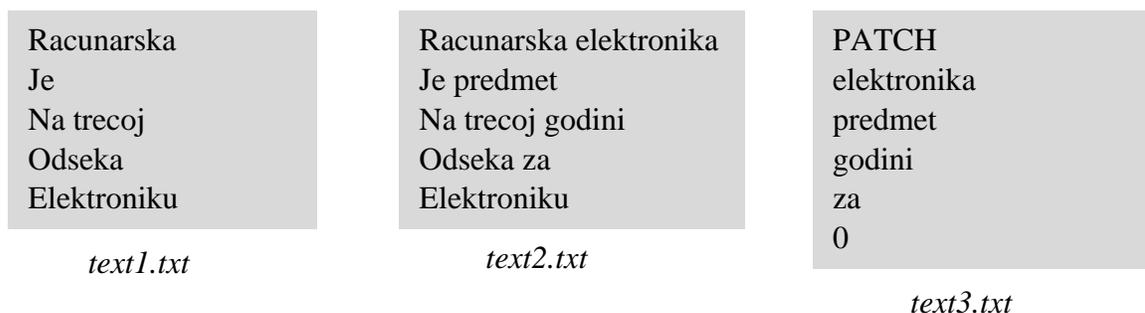
Ukoliko korisnik odabere pregled postojećih naslova neophodno je izlistati sadržaj fajla *index.txt* u kome se nalazi spisak svih postojećih naslova.

Struktura *config.txt*, *index.txt* i *contents.txt* fajla data je u nastavku



9. Patch maker

Potrebno je napraviti program koji pronalazi razliku između dva tekstualna fajla i tu razliku (*patch*) smešta u treći tekstualni fajl. Ovi fajlovi u prvom redu fajla sadrže reč PATCH ispod koje slede reči koje predstavljaju razliku između dva fajla na odgovarajućim linijama. Ukoliko razlika između sadržaja dve linije fajla ne postoji na to mesto se u patch fajlu stavlja 0. Način kreiranja jednog *patch* fajla dat na sledećem primeru:



Nakon pokretanja programa od korisnika se zahteva unos naziva dva tekstualna fajla od kojih treba kreirati *patch* fajl (u primeru *text1.txt* i *text2.txt*), a zatim se od njega zahteva unos naziva patch fajla (u primeru *text3.txt*).

Potrebno je potvrditi funkcionalnost projekta na barem 3 različita primera.

10. Apply patch

Potrebno je napraviti program koji na neki tekstualni fajl primenjuje patch fajl i rezultat smešta u treći tekstualni fajl (više o patch fajlovima videte u primeru 9). Dodavanje patch fajla na neki tekstualni fajl predstavlja dopunjavanje redova nekog tekstualnog fajla. Način generisanja rezultatnog fajla *text3.txt* dat je na sledećem primeru

| | | |
|---|--|---|
| Racunarska Je Na trecjoj Odseka Elektroniku | PATCH elektronika predmet godini za 0 | Racunarska elektronika Je predmet Na trecjoj godini Odseka za Elektroniku |
| <i>text1.txt</i> | <i>text2.txt</i> | <i>text3.txt</i> |

Nakon pokretanja programa od korisnika se zahteva unos naziva fajla na koji se dodaje patch file (u primeru *text1.txt*). Nakon toga se od korisnika zahteva unos naziva patch fajla (u primeru *text2.txt*) a nakon toga se vrši provera da li fajl sa navedenim nazivom predstavlja patch fajl (zapravo se proverava da li prva linija fajla sadrži reč PATCH). Ukoliko je navedeni patch fajl validan vrši se generisanje rezultatnog fajla (u primeru *text3.txt*).

11. Osmosmerka*

Potrebno je realizovati igru “Osmosmerka”, gde je u tabeli od 10x10 slova potrebno pronaći i obeležiti 10 različitih reči (imenica u nominativu jednine od najmanje četiri slova). Potrebno je smisliti 10 reči, smestiti ih unutar tabele u nekom od 8 različitih smerova (4 dijagonalna, 2 horizontalna i 2 vertikalna). Dozvoljeno je preklapanje pojedinačnih slova iz dve različite reči. Ostala polja tabele popuniti nasumično generisanim slovima iz srpske azbuke. Tabelu od 10x10 polja, gde je jedno polje veličine 2x2 i na kome

| | | | | | | | | | | | | | | |
|----|---|---|---|---|----|---|---|---|---|----|---|----|----|----|
| O | G | S | E | E | NJ | A | V | I | Đ | A | R | G | A | N |
| P | R | O | F | I | T | Z | A | R | P | M | C | T | F | U |
| L | U | G | Z | U | M | A | S | D | D | O | H | I | R | P |
| A | P | J | A | D | O | R | T | N | O | K | K | M | E | R |
| N | A | L | Č | N | T | A | S | D | F | G | R | NJ | ĐŽ | A |
| I | A | N | I | R | I | Č | L | V | B | C | E | U | A | V |
| R | U | S | V | T | V | Z | A | S | O | Đ | D | L | N | LJ |
| A | O | K | O | Z | I | A | O | M | O | D | I | A | E | A |
| NJ | P | Ć | K | U | S | R | R | V | Č | V | L | U | M | NJ |
| E | V | C | Š | I | A | A | F | T | A | P | U | U | Đ | E |
| M | G | H | O | O | NJ | D | O | R | E | NJ | A | Ž | K | T |
| N | U | I | R | P | E | A | P | A | H | L | E | A | V | A |
| B | V | S | T | R | A | T | E | G | I | J | A | J | B | E |

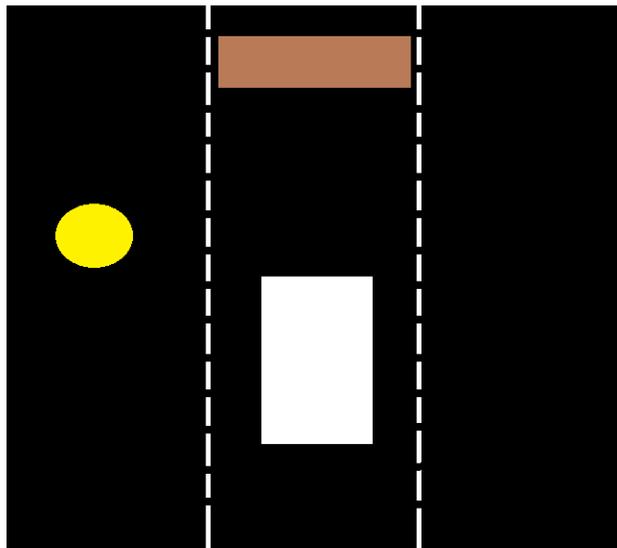
7 - Slika uz projekat 11

je ispisano odgovarajuće slovo, potrebno je iscrutati u konzoli pri inicijalizaciji programa. Omogućiti kretanje jasno označenog kursora kroz tabelu u svih osam smerova pritiskom na odgovarajuće tastere na

tastaturi, gde pritiskom na taster “space”, počinje se i završava sa obeležavanjem potencijalne reči i ukoliko je reč zaista pogodena, ona ostaje obeležena do kraja igre, dok ukoliko nije, sva slova iz tog ciklusa obeležavanja ostaju neobeležena. Nakon što se počne sa obeležavanjem reči i odabere se smer “prostiranja” te reči (tj. obeleži se i drugo slovo nakon početnog), omogućiti kretanje samo u tom smeru, sve do kraja tog ciklusa obeležavanja. Igra se završava kada se svih 10 reči uspešno pronađu i obeleže.

12.2D Racing game*

Potrebno je realizovati igricu u kojoj se vozilo kreće po putu od tri trake i u kome izbegava prepreke i skuplja novčiće. U konzoli je potrebno iscrtati tri trake međusobno razdvojene vertikalnom isprekidanom linijom i vozilo koje se na početku nalazi na sredini srednje trake. Pritiskom na odgovarajuće tastere, vozilo prelazi u levu ili desnu susednu traku. Na putu se u određenim vremenskim trenucima, u nasumično odabranoj traci pojavljuju prepreke, koje je potrebno izbegavati, i novčići, koje je potrebno skupljati. Vremenom je potrebno povećavati brzinu vozila i učestanost pojavljivanja prepreka i novčića. Prepreka je udarena i novčić je skupljen ukoliko se vozilo nalazi u istoj traci gde i predmet, a vertikalne coordinate vozila i predmeta se makar nekim delom preklapaju. Igrač ima 3 “života”, što znači da kada se treći put udari prepreka, igra se završava. Potrebno je u konzoli prikazivati broj preostalih života. Takođe, potrebno je brojati skupljene novčiće i taj broj takođe prikazivati u konzoli.



8- Slika uz projekat 12

Napomena: studentima se ostavlja da sami osmisle dizajn vozila, novčića i prepreka.

13. Simulacija štoperice i tajmera

Potrebno je realizovati program koji simulira funkcionalnost štoperice i tajmera. Nakon pokretanja programa, otvara se meni u kome se bira da li se prelazi u režim tajmera ili režim štoperice. U režimu tajmera, postoji dva polja u koje korisnik unosi broj minuta i broj sekundi. Nakon unosa, pritiskom na taster “enter”, pokreće se tajmer koji odbrojava unazad minute i sekunde unete od strane korisnika. Kada tajmer dođe do vremena 00:00, iskače “pop-up” poruka da je vreme tajmera isteklo. U režimu štoperice, pritiskom na taster “enter” započinje se odbrojavanje sekundi i minuta naviše počev od vremena 00:00. Omogućiti zaustavljanje i pauziranje brojanja, kao i resetovanje brojača pritiskom na određeni taster.

14. Skočko*

Potrebno je napraviti igricu “Skočko” iz popularnog kviza “Slagalice”. Postoji 6 različitih simbola i na početku svake igre generiše slučajni niz od 4 simbola (ne moraju da budu različiti simboli). Igrač ima šest pokušaja da pogodi kombinaciju, pa je potrebno u konzoli iscertati po 4 polja u 6 redova, gde jedan red predstavlja jedan pokušaj, gde se odgovarajućim tasterima na tastaturi bira simbol za polje na kom se nalazi kursor i iscertava se u njemu (dakle, potrebno je omogućiti “šetnju” kursora po poljima, ali samo u trenutnom redu tj. pokušaju). Pritiskom na taster “enter”, ukoliko su sva polja u aktuelnom redu popunjena, poredi se igračeva kombinacija sa tačnom I ispisuje se broj pogođenih simbola na odgovarajućem mestu I broj pogođenih simbola na pogrešnom mestu, a takođe i kursor prelazi u naredni red. Ukoliko je kombinacija pogođena, završava se igra i obaveštava se igrač o pogođenoj kombinaciji, dok ako ni nakon šestog pokušaja kombinacija nije pogođena, igra se takođe završava i igrač se obaveštava da je izgubio.

15. Testiranje brzine prekucavanja teksta

Potrebno je napraviti program kojim se proverava brzina prekucavanja teksta ispisano na ekranu. Potrebno je napraviti bazu od 50 smislenih reči iz srpskog jezika. Pri svakom pokretanju programa, potrebno je tih 50 reči poređati u nasumičnom redosledu i u tom redosledu ispisati ih u konzoli. Korisnik tada počinje da prekucava tekst koji se takođe ispisuje u konzoli. Obezbediti i merenje vremena prekucavanja koje je moguće videti sve vreme. Korisnik završava prekucavanje pritiskom na taster “enter” i tada se u konzoli ispisuje vreme koje je bilo potrebno korisniku, broj grešaka koje je korisnik napravio i konačan rezultat koji se dobija dodavanjem sekunde na vreme prekucavanja za svaku grešku koju je korisnik napravio.

16. Bubble trouble*

Potrebno je realizovati varijantu “Bubble Trouble” igrice (<https://www.miniclip.com/games/bubble-trouble/en/>). Igrač se kreće levo-desno pritiskom na strelice na tastaturi, dok strelu ispaljuje vertikalno naviše pritiskom na taster “space”. Trajektorija strele ostaje iscertana sve dok strela ne dosegne gornju ivicu prozora. U konzoli na početku postoji I blok veličine 8x8 koji se kreće odbijajući se o sve ivice konzole (zanemariti uticaj gravitacije). Cilj igre je uništiti blok time što će udariti u strelu koju je ispalio igrač, ili u vidljivu trajektoriju strele. Kada se desi neki od prethodna dva slučaja, blok se raspada na dva dela veličina 4x4 koja se odbijaju na dve različite strane nakon što nastanu. Potrebno je uništiti sve podblokove (od bloka 4x4 nastaju dva bloka od po 2x2, od bloka 2x2 nastaju dva bloka od po 1x1, blok 1x1 nestaje nakon što je pogođen), čime se završava igra. Ukoliko bilo koji od blokova pogodi igrača, igra se takođe završava i igrač gubi.



9 - Slika uz projekat 16

17. Whack-a-Mole game*

Potrebno je napraviti igricu Whack-a-Mole. U konzoli postoji tabla veličine 3x3. Na nasumično odabranom polju u određenim vremenskim trenucima se pojavljuje “krtica” koja “izviruje” na određeni broj sekundi koju je potrebno da igrač “udari” dok je ona vidljiva pritiskom na odgovarajući taster od 9 mogućih taster, gde svaki taster odgovara jednom od polja. Igrač ima 3 “života”, tj. igra se završava nakon što igrač i treći put propusti da udari krticu. Preostali broj života je potrebno sve vreme igre prikazivati na konzoli. Učestanost pojavljivanja krtica vremenom povećavati, a vreme za koje je krtica vidljiva vremenom smanjivati.



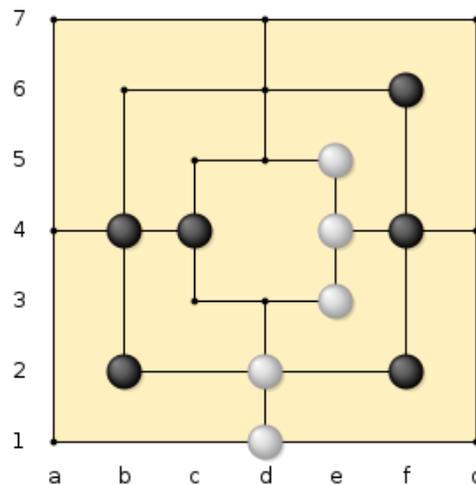
10 - Slika uz projekat 17

18. Moj broj

Potrebno je realizovati igru “Moj broj” iz kviza “Slagalice”. Kada se započne igra, generiše se nasumičan broj u opsegu 1-999, kao i četiri broja u opsegu 1-9, jedan broj iz skupa {10, 15, 20} i jedan broj iz skupa {25, 50, 75, 100}. Potrebno je dobiti prvi generisani broj koristeći se isključivo ostalim generisanim brojevima (svaki broj se sme koristiti samo jednom), osnovnim računskim operacijama (sabiranje, oduzimanje, množenje i deljenje) i zagradama. Potrebno je sve generisane brojeve (osim prvog) ispisati u zasebnim poljima i omogućiti kretanje kursora po njima i njihov izbor pritiskom na odgovarajuće tastere. Nakon što se broj izabere, onemogućiti novi izbor istog broja. Obezbediti zasebna polja i za svaku od operacija i zagrada. Pritiskom na taster „space“, bira se broj/operacija na kojoj se kursor nalazi. Jasno označiti polje na kojem se nalazi kursor. Potrebno je i ispisivati matematičku formulu koju igrač koristi za dobijanje traženog broja. Nakon što igrač završi sa ispisivanjem formule, pritiskom na taster „enter“ rezultat se izračunava i igraču se šalje poruka ukoliko je uspeo da pogodi tačan broj ili koliko mu fali do tačnog broja

19. Igra "Mice" za dva igrača*

Potrebno je realizovati jednostavniju verziju igre "Mice" za dva igrača. Na početku je u konzoli potrebno iscrtati tablu za igru koja se sastoji iz 24 polja (na slici). Potrebno je omogućiti kretanje kursora po poljima pritiskom odgovarajućih tastera. Svaki od igrača poseduje 12 svojih figurica. Igrači naizmenično postavljaju svoje figure na slobodna polja na table. Cilj igrača je da sklopi što više redova (horizontalnih, vertikalnih ili dijagonalnih) čija sva tri polja sadrže njegove figure. Polje koje je zauzeto od strane različitog igrača je potrebno obojiti različitom bojom. U svakom trenutku igre, potrebno je ispisivati koji igrač je na potezu. Jasno označiti polja na kojima se nalazi kursor. Pritiskom na taster "space" se zauzima slobodno polje. Potrebno je ne dozvoliti da se zauzme polje koje je već zauzeto. Nakon što se sva polja zauzmu, prebrojava se ko je skupio više redova i ispisuje se koji je igrač pobedio (ili da je nerešeno ako bude nerešeno).



11 - Slika uz projekat 19

20. Black jack*

Potrebno je realizovati igru kartama "BlackJack" (poznatu i kao "Ajnc"). Igra se igra sa jednim špilom od 52 karte, bez džokera (nije potrebno voditi računa o boji karata, već samo o broju). U svakoj ruci, igraču i kompjuteru se dele po dve karte (nasumično generisane). Igrač ima uvid samo u svoje karte i one se ispisuju u konzoli. Svaka karta vredi određeni broj poena: 1-9 imaju svoju vrednost, dok štihovi nose po 10. Cilj igre je skupiti 21 poen u ruci, ili broj poena što bliži 21, ali ne i veći. Igrač koji to ostvari dobija bod na kraju svake ruke. Nakon što je igrač dobio svoje karte, može da bira da li će ih otkriti ili traži još karata. Ukoliko traži još, dodeljuje mu se još jedna karta. Igrač može tražiti još karata sve dok ne pređe 21 poen. U tom slučaju automatski gubi i bod se dodeljuje kompjuteru. Ukoliko igrač odluči da otkrije karte, ostvaren rezultat se poredi sa rezultatom kompjutera. Čiji rezultat je bliži 21, on odnosi bod (ukoliko je rezultat isti, bod odnosi igrač). Potrebno je realizovati da igrač bira da li hoće još karata ili da li hoće da otkrije karte pritiskom na neki od tastera. Takođe, potrebno je voditi evidenciju o ukupnom broju bodova svakog od igrača i to ispisivati u konzoli za sve vreme igre. Na kraju svake ruke, potrebno je ispisati ukupan broj poena u ruci svakog od igrača i koji igrač je pobedio i odneo bod. Po završetku tekuće ruke, pritiskom na taster "enter", započinje se sledeća.

Napomena: radi lakše realizacije, računati da se svaki put dele karte iz punog špila.

21. Kovertor Bin-to-hex

Potrebno je realizovati konvertor koji pretvara broj u nekom od formata binarni, decimalni, heksadecimalni, u broj u druga dva formata. Potrebno je u konzoli iscrtati 3 polja, po jedno za svaki od tri formata. Omogućiti pomeranje kursora po poljima pritiskom na odgovarajuće tastere. Pritiskom na taster "enter" dok je kursor na nekom od polja, vrši se izbor ulaznog formata i tada treba zabraniti pomeranje kursora. Novim pritiskom na "enter", ulazni format se deselektuje i opet je moguće pomeranje kursora.

Nakon što je format selektovan, unosi se ulazni broj u izabranom format. Pritiskom na taster “space”, vrši se konverzija i rezultati u izlaznim formatima se ispisuju u odgovarajućim poljima. Prilikom deselekcije formata novim pritiskom taster “enter”, ujedno se i brišu izračunati stari rezultati iz polja.

22. Loto

Potrebno je realizovati simulaciju “LOTO” igre na sreću. U konzoli je potrebno iscrtati 7 polja koja su predviđena da u njih korisnik unese broj u opsegu 1-39. Kursor se na početku nalazi na prvom polju. Unos broja u polje se završava pritiskom na taster “space”, čime se pored potvrde unosa, prenosi kursor na naredno polje. Potrebno je onemogućiti unos broja koji je unet u neko od prethodnih polja. Nakon što se potvrdi unos i na poslednjem polju, potrebno je generisati 7 različitih slučajnih brojeva iz opsega 1-39, ispisati ih u rastućem poretku, zacrveneti polja u kojima se nalaze brojevi koje je korisnik uneo, a ne nalaze se u nasumično generisanim brojevima i za zeleniti polja u kojima se nalaze brojevi koje je korisnik uneo, a nalaze se među slučajno generisanim brojevima. Nakon toga, potrebno je ispisati poruku korisniku u kojoj se navodi broj pogođenih brojeva.

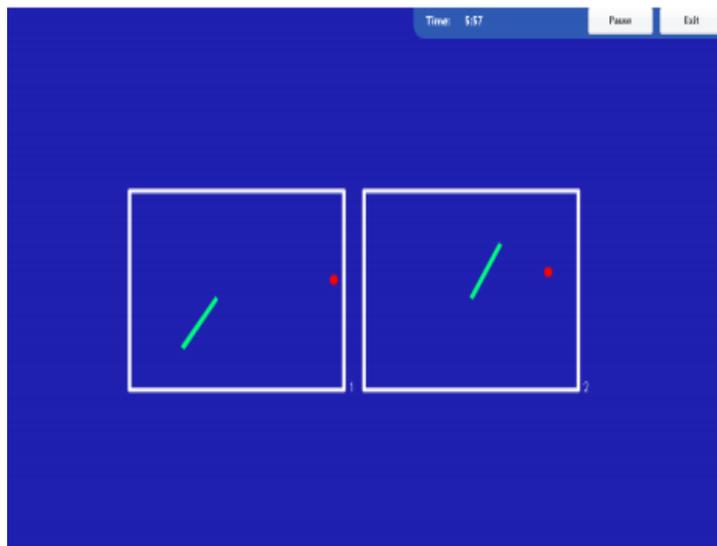
23. Poligon za gađanje*

Meta(recimo glineni golub) se sa sredine ekrana kreće odozdo na gore različitim brzinama generisanim pomoću generatora random brojeva. Sa leve strane ili desne se ispaljuje metak iz pištolja koji treba da pogodi objekat tako što će se ispaliti na vreme proporcionalno brzini kretanja objekta i susresti ga. Na vrhu ekrana se ispisuje broj pogođenih i promašenih objekata. Objekti se ispaljuju neprestano jedan za drugim, a brzina im se generiše random generatorom. Brzina metka je fiksna i proizvoljna. Na početku igrice se ispisuje tekst sa pravilima igrice I klikom na taster “space” igra se pokreće. Metak iz pištolja se takođe ispaljuje pomoću tastera “space”, dok se igra prekida pritiskom na taster “ESC”.

Napomena: parametre kao što su veličina metka i mete(u pikselima) kao i brzinu mete(piksel/vreme) usvojii proizvoljno tako da vizuelno sve bude lepo uočljivo I pregledno.

24. Divided attention*

Ekran se podeli na dva kvadrata (npr. dva oblasna prostora). U svakom od kvadrata(u granicama tog kvadrata) kreću se po dva objekta (npr. dva aviona), predstavljeni kao jedan krug i jedan pravougaonik (krug je veličine jednog bloka dok je pravougaonik veličine 4 bloka) po proizvoljnoj putanji i brzini s tim što se odbijaju od ivica kvadrata pod istim uglom pod kojim su udarili u ivice (na početku oba objekta kreću sa sredina kvadrata). Treba pratiti oba ekrana i predvideti u kom od kvadrata će se desiti sudar između objekata (kruga i pravougaonika) pritiskom na jedan od dva tastera na tastaturi (taster 1 i taster 2). Ukoliko se u predviđeno vreme reaguje na sudar (klikom na tastere) tj. u periodu kada su

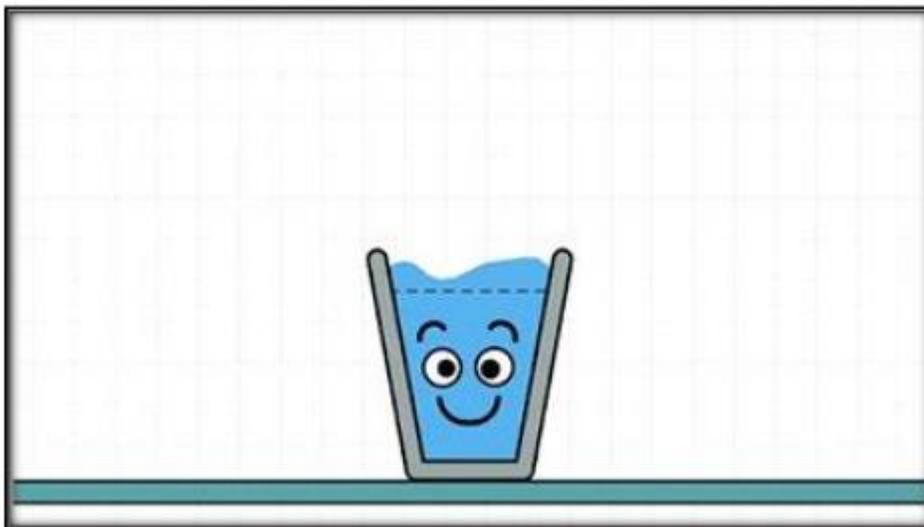


12 - Slika uz projekat24

objekti preklopljeni to je ispravno reagovanje i nosi jedan poen, ukoliko se pak ne reaguje, a objekti se preklope to nosi jedan negativan poen. Tokom izvršavanja programa na vrhu ekrana se ispisuje trenutni skor ispravnih i pogrešnih reagovanja na sudare. Na početku igrice se ispisuje tekst sa pravilima igrice i pritiskom na space dugme tastature igra se pokreće.

25. Water glass*

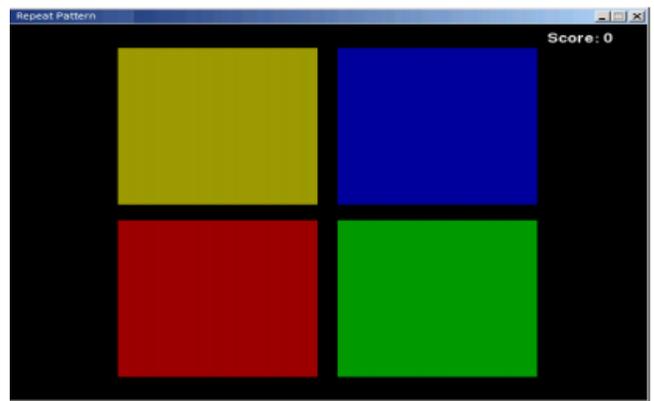
Časa pravougaonog oblika prizvoljne veličine (npr. 10x40 blokova) se puni vodom. Brzina punjenja se uzima u svakoj iteraciji proizvoljno na osnovu generatora random brojeva u određenom okviru (tako da brzina ne bude ni prevelika ni premala, npr. od 1 do 7 blokova u sekundi). Nivo vode se povećava tako što se popunjava red po red piksela u okviru čase. Na 35 blokova visine čase treba postaviti liniju širine jednog bloka koja će određivati nivo preko kog voda ne sme da pređe. Na početku voda kreće da puni od dna čaše različitim brzinama u svakoj od iteracija I kada voda dostigne određeni nivo treba readovati klikom na space taster. Za ispravno reagovanje koje nosi jedan pozitivan poen smatrati svako reagovanje u okviru od 0-2 piksela PRE nego voda dostigne nivo. Ukoliko voda pređe dati nivo, a ne bude registrovano reagovanje pritiskom tastera dobija se negativan poen. Tokom izvršavanja programa na vrhu ekrana se ispisuje trenutni skor pozitivnih I negativnih poena. Na početku igrice se ispisuje tekst sa pravilima igrice I klikom na space dugme tastature igra se pokreće, dok se igra prekida klikom na Esc dugme tastature.



13 - Slika uz projekat 25

26. Repeat pattern game*

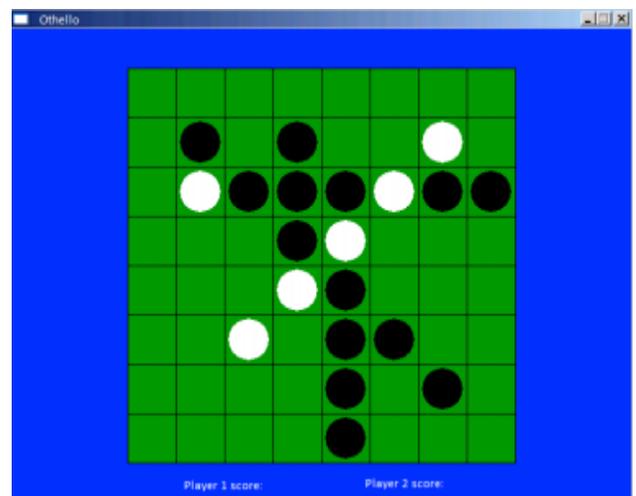
Data je tabela 2x2 polja (polja veličine 8x8) sa različitim bojama. Cilj igrice je da se svaki put pogodi odgovarajuća kombinacija boja koja se prikazuje u tabeli pritiskom na odgovarajuće tastere u odgovarajućem redosledu gde svakoj od boja koje se mogu pojaviti u tabeli odgovara jedan taster. Boje koje se mogu pojaviti su: plava, crvena, žuta i zelena. Bojama se na svakom početku igrice po random šablonu dodeljuju brojevi od 1 do 4 koji zapravo predstavljaju taster koji je neposredno dodeljen toj boji. Nakon ove dodele parovi boja-taster se prikazuju igraču sve dok on ne pritisne taster "space" na tastaturi kako bi zapamtio kombinaciju, odnosno kojoj boji odgovara koji taster(broj), a zatim se kreće u izvršavanje igrice. Boje se pojavljuju u blokovima po slučajnom redosledu i poenta igrice je da se pogodi odgovarajući redosled boja pritiskom na odgovarajuće tastere koji pripadaju datim bojama. Sve dok igrač ispravno pogađa boje, tabela se iznova generiše. Ukoliko igrač pogreši igra se prekida uz odgovarajuću poruku i mogućnost da se krene ispočetka.



14 - Slika uz projekat 26

27. Other hello game*

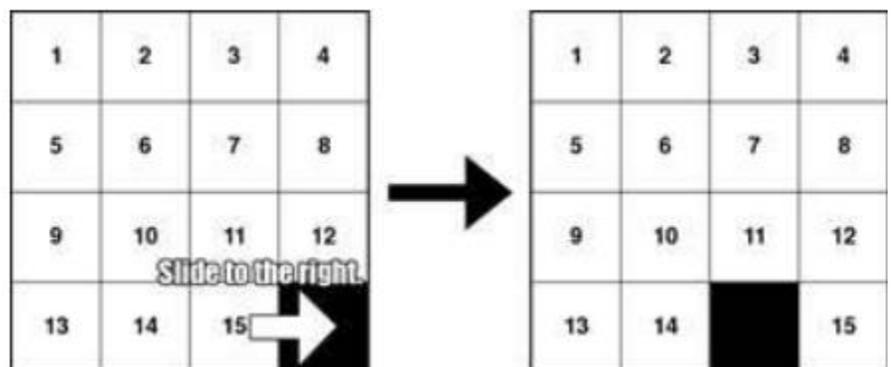
Tabela veličine 8x8 polja (polja veličine 2x2) koja su na početku igrice prazna. Dva igrača sa crnim (*Player 1*) i belim (*Player 2*) žetonima koji jedan za drugim postavljaju žetone na polja u tabeli. Ukoliko se tokom igrice beli žeton nađe između dva crna žetona taj beli žeton postaje crn i obrnuto. Pobednik igrice je onaj igrač koji na kraju, kada se popune sva 64 polja, ima više žetona u svojoj boji na tabeli. Tokom igrice prikazivati trenutno stanje boja belih i crnih žetona na tabeli. Detaljno objašnjenje pravila igrice možete naći na sledećem linku <https://www.youtube.com/watch?v=Puch83HSvZM>.



15 - Slika uz projekat 27

28. Slide puzzle game*

Tabela veličine 4x4 polja numerisana brojevima od 1 do 15 i jedno nenumerisano, prazno, polje kao što je prikazano na slici. Polja napraviti veličine 4x4. Numeracija polja kao i raspored (mesto) nenumerisanog polja u tabeli se random generišu na početku igrice. Poenta igrice je da se numerisana polja poređaju počevši od najmanjeg do



16 - Slika uz projekat 28

najvećeg premeštanjem polja. Numerisano polje se može pomerati samo na mesto nenumerisanog, praznog, polja. Kretanje kroz tabelu vršiti kursorima na tastaturi (na neki način označiti polje na kome se trenutno nalazi kursor), a pomeranje polja pomoću tastera space tako što će se polje na kome se trenutno nalazi kursor, a koje se nalazi pored praznog polja zameniti sa njim. Ukoliko se klikne dugme space za polje koje nije susedno praznom polju ignorisati. Kraj igrice se registruje odgovarajućom porukom na ekranu ukoliko su sva polja poređana u odgovarajućem redosledu.

29. Huffman coding

Potrebno je napisati program u MASM jeziku koji vrši Huffmanovo kodovanje simbola. Ovo kodovanje zasnovano je na verovatnoći pojavljivanja simbola gde se simboli sa najvećom verovatnoćom pojavljivanja koduju sa kodnom rečju najmanje dužine. Više o ovom mehanizmu kodovanja može se naći na linku <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/>

Nakon pokretanja programa od korisnika se traži da unese naziv tekstualne datoteke nad kojom želi da primeni mehanizam Huffman-ovog kodovanja. Kao rezultat kodovanja na konzoli se ispisuje verovatnoća pojavljivanja pojedinih simbola kao i kodne reči koje su dodeljene pojedinom simbolu.

30. Mirror photo

Napisati program koji učitava sliku u PGMA formatu i nakon učitavanja slike, pomocu četiri kursora na tastaturi kontrolise operacije nad učitanom slikom na sledeći način:

- Ukoliko se pritisne kursor strelica na gore nad učitanom slikom treba primeniti efekat ogledala u odnosu na gornju ivicu date slike.
- Ukoliko se pritisne kursor strelica na dole nad učitanom slikom treba primeniti efekat ogledala u odnosu na donju ivicu date slike.
- Ukoliko se pritisne kursor strelica na levo nad učitanom slikom treba primeniti efekat ogledala u odnosu na levu ivicu date slike.
- Ukoliko se pritisne kursor strelica na desno nad učitanom slikom treba primeniti efekat ogledala u odnosu na desnu ivicu date slike.

Rezultat operacije primenjene nad slikom se upisuje u izlazni fajl sa istom ekstenzijom. Više o radu sa PGMA formatom i načinom korišćenja ovog formata moguće je naći na http://tnt.etf.rs/~oe3re/pdf/Projekti/2018/Projekti_Postavka_2018.pdf (Uvodni deo)

31. Parsiranje tekstualnog fajla

Napisati program koji će analizirati ulazni fajl. Svaki red u ulaznom fajlu je sekvenca karaktera: $x_0 y_0 x_1 y_1$ boja, gde su x_0, y_0 koordinate gornjeg desnog ugla kvadrata, x_1, y_1 koordinate donjeg desnog ugla kvadrata, boja - boja kvadrata. Na osnovu sadržaja fajla iscrtava se odgovarajući broj kvadrata na ekranu.