

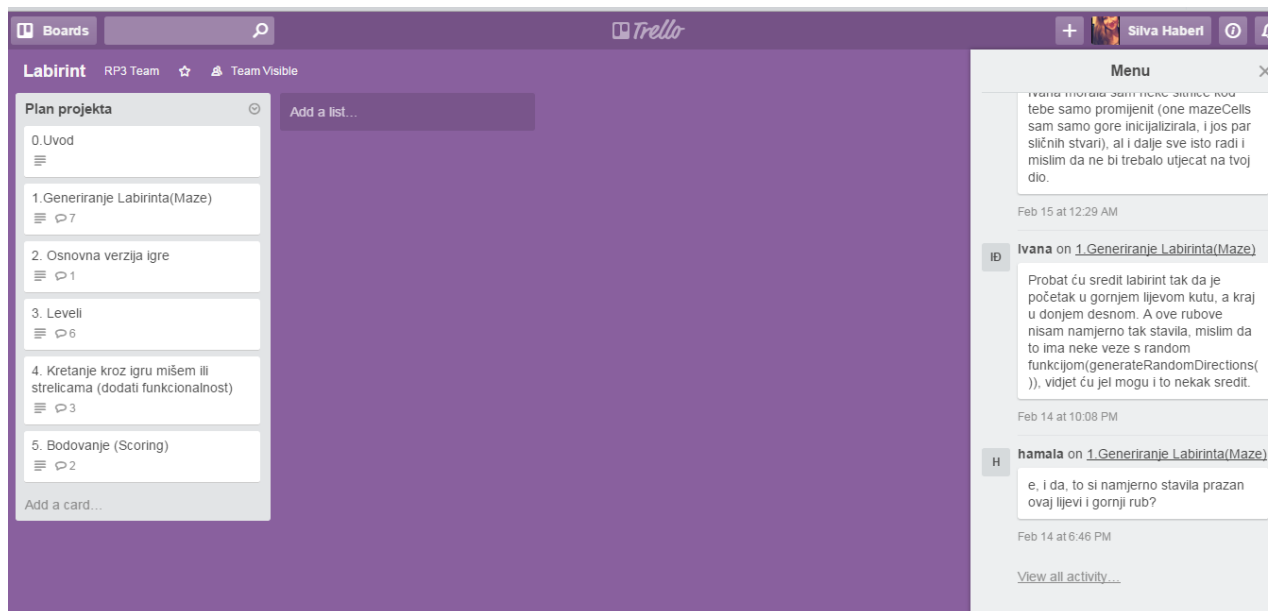
# Igra "Labirint"

## 0. Uvod

Rad na projektu u timu može biti jednostavniji korištenjem repozitorijem koda (Version Control System , Git). Koristile smo ga kako bi pratile sve izmjene na datotekama – tko ih je napravio, kada ih je napravio, što je promijenio. Najbolje od svega je što smo u svakom trenutku datoteku mogle vratiti na jedno od stanja u prošlosti i imamo cijelu povijest izmjena. Na otvorenom računu na GitHubu, projekt je rađen u repozitoriju "WindowsForms-Labirint" danog na url-u: <https://github.com/silvahaberl/WindowsForms-Labirint>. Sva tri člana tima dodana su kao suradnici (contributors: silvahaberl, idurdevic, mateaaa)

Također, koristile smo i organizacijski alat Trello.

Pokazao se kao vrlo koristan dodatak svakodnevnom razvoju uz pomoć kojega smo lako mogle pratiti najnovije zadatke, tko je za njih zadužen, u kojoj su fazi i u kojem roku ih moraju završiti.



Podijelile smo projekt na 5. podzadataka. Svaki član tima uzeo je svoj dio, i naknadno pomagao drugom članu tima, u eventualnim greškama u kodu i idejama kako riješiti problem u igri. Podjela zadataka je bila: Generiranje Labirinta(1., Ivana), Osnovna Verzija igre(2.Matea,Ivana), Leveli igre (3.Ivana,Silva), Kretanje kroz igru strelicama ili mišem (4.,Matea), Bodovanje (5.Silva, Matea).

## 1. Generiranje labirinta

Za generiranje smo koristile rekurzivni DFS (depth-first search) algoritam. Nalazi se u klasi `GenForm.cs`. Na početku je vrijednost svih članova polja `mazeCells` jednaka 0, tj. svugdje su zidovi. Algoritam funkcionira tako da na početku postavi `mazeCells[0,0]` na 1 (tj. prolaz) i potom rekurzivno ispituje sva četiri smjera (gore, dolje, lijevo, desno) za trenutno promatranu ćeliju (element polja `mazeCells[r,c]`). Labirint je zapravo specijalan oblik razapinjućeg stabla (graf bez ciklusa) pa ovaj algoritam funkcionira tako da u svakom koraku pazi da ne napravi ciklus.

```
public int[,] generateMaze(int sizeMX, int sizeMY, int sizeRX, int sizeRY)
```

- `sizeMX, sizeMY` – širina, odnosno visina labirinta
- `sizeRX, sizeRY` – širina, odnosno visina pravokutnika kojim ćemo crtati zidove i prolaze

```
private void recursion(int r, int c)
```

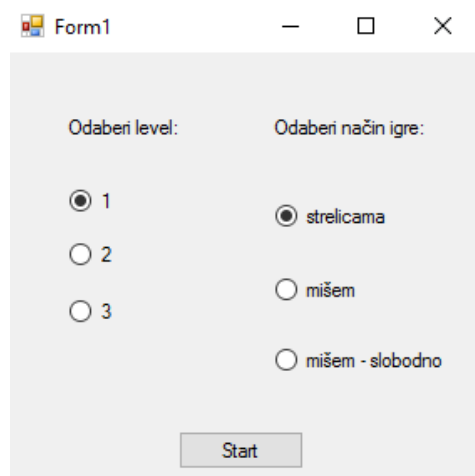
- `r, c` – redak, odnosno stupac promatrane ćelije (`mazeCells`)
- glavni dio implementacije DFS algoritma

```
private int[] generateRandomDirections()
```

- funkcija koja nasumično ispremiješa brojeve od 1 do 4
- služi za odabir redoslijeda smjerova (tako da ne ispituje uvijek prvo smjer gore pa desno pa dolje i na kraju lijevo, nego je taj redoslijed nasumičan) – tako da dobijemo što bolje “razgranat” labirint

## 2. Osnovna verzija igre

Pri pokretanju programa, prikazuje se forma s mogućnostima odabira levela i načina igre.



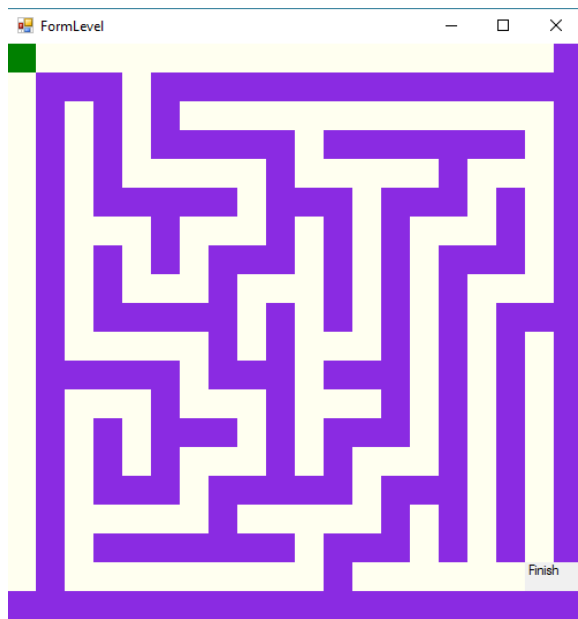
The screenshot shows a Windows application window titled "Form1". It contains two columns of radio button options. The first column is labeled "Odaberi level:" and has three options: "1", "2", and "3". The second column is labeled "Odaberi način igre:" and has three options: "strelicama", "mišem", and "mišem - slobodno". At the bottom center of the form is a button labeled "Start".

### 3.Leveli igre

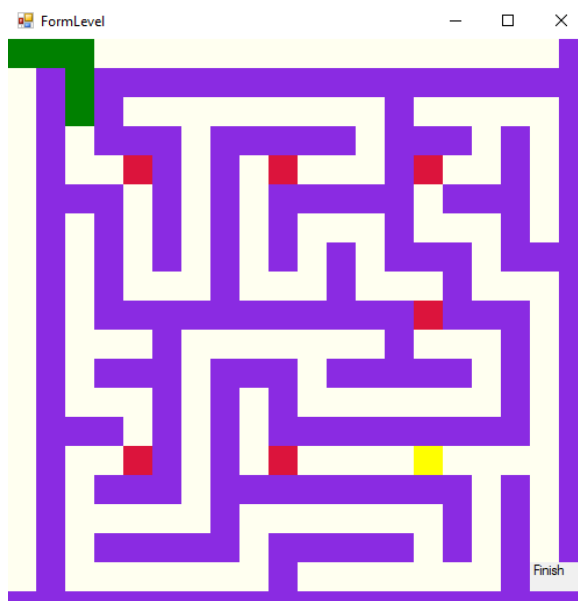
#### 3.1. Leveli 1,2,3

Imamo tri levela igre.

U prvom je cilj doći od gornjeg lijevog kuta (zeleni kvadratić) do donjeg desnog (label s oznakom "Finish").



U drugom i trećem imamo plijenove koje treba sakupiti da bi se zaradilo više bodova. U trećem levelu je još postavljeno i ograničenje od 50 s na duljinu igranja (ako igrač ne dođe do kraja u tom vremenu, igra je gotova).



### **3.2 Plijenovi**

klasa Plijen

```
public Plijen (int xos,int yos, int sirina, int  
velicina)
```

- konstruktor plijena
- pravokutnik, početne točke (x,y) ,visine i širine 25x25 kao i veličina igrača
- lista objekata sviPlijenovi

labela Kraja "Finish"

```
private void FormLevel_Load(object sender, EventArgs e)
```

- funkcija, event
- generiranje labirinta
- crtanje labirinta
- generiranje plijenova, ako smo u levelu 2 ili 3 , 20 komada, od točke (100,100) pa desni dio forme , do kraja, dodajemo plijenove koristeći funkciju Add() u listu sviPlijenovi, objekte tipa Plijen

```
private void FormLevel_Paint(object sender,  
PaintEventArgs e)
```

- iscrtavanje forme plijenovima (pravokutnicima u tri boje)
- ako je varijabla paint == true, iscrtaj pravokutnike i ispunj bojom

```
private void timer2_Tick(object sender, EventArgs e)
```

- brisanje plijenova- ispunjavanje bijelom bojom, ako je varijabla paint ==false

mjesto gdje se nalaze plijenovi su u polju mazecells[,] (gdje je vrijednost elementa jednaka 3 )

#### **4.Kretanje kroz igru strelicama ili mišem**

```
public void drawPoint (object sender, EventArgs e)
```

- Služi za crtanje zelenih polja, tj. polja u labirintu koja smo već prošli.

```
public void makeRect()
```

- Crtanje bijelih kvadrata.

mazeCells[,]=0 – znači da je to polje ljubičasto, tj. po tome se ne smijemo kretati

mazeCells[,]=1 – polje je prazno

mazeCells[,]=2 – polje je zeleno, tamo smo već bili

mazeCells[,]=3 – na tom polju je plijen

```
private void FormLevel_KeyDown (object sender, EventArgs e)
```

- Omogućava kretanje labirintom pomoću tipkovnice.
- Imamo 4 osnovna slučaja, ovisno o tome koja je strelica pritisnuta.
- U svakom slučaju se provjerava i da li bi pomakom u tom smjeru izašli izvan labirinta.
- Svaki od slučaja ima i 2 podslučaja: u 1. se provjerava provjerava da li se vraćamo unatrag od kud smo došli, i tada se tim mjestima crta bijeli kvadrat; 2. se događa ako smo naišli na prazno polje ili na plijen, tada se na tom mjestu crta novi zeleni kvadrat te se ako smo pojeli plijen dodaju bodovi. U oba podslučaja se polja mazeCells[,] postavljaju na odgovarajući broj.
- Ukoliko dodemo do kraja labirinta ispisuje se odgovarajuća poruka.

```
private void FormLevel_MouseDown (object sender, EventArgs e)
```

- Detektira pritisak lijeve tipke na mišu te računa na kojem se mjestu to dogodilo. U oba načina kretanja mišem mora se početi iz gornjeg lijevog označenog kuta labirinta.

```
private void FormLevel_MouseMove (object sender, MouseEventArgs e)
```

- Omogućava kretanje mišem (simpleMouseMove, freehandMouseMove)

```
private void again(int i)
```

- Pomoćna funkcija koja se koristi kod slobodnog kretanja mišem. Sva polja s vrijednosti i boja bijelom bojom. Također se koristi i kod brisanja plijenova iz polja.

```
private void freehandMouseMove (MouseEventargs e)
```

- Za slobodno kretanje mišem.
- Pomoću dvije for petlje obilazimo cijelu ploču, pomoću varijabla boxXstart, boxXend, boxYstart, boxYend znamo koordinate polja na ploči. Provjeravamo u kojem se od tih polja trenutno nalazi miš i ukoliko se nalazi iznad praznog polja ili iznad polja u kojemu je plijen iscrtava se linija prođena mišem. Ako dodemo s mišem na ljubičasto polje, tj. na pregradu labirinta moramo krenuti ispočetka, tj. poziva se funkcija `again (int i)`. Isto se događa i ako u nekom trenutku pustimo tipku miša.
- Također ako se s mišem nađemo iznad polja koje predstavlja završetak labirinta, ispisuje se odgovarajuća poruka.

```
private void simpleMouseMove (MouseEventargs e)
```

- Obično kretanje mišem.
- U svakom trenutku gledamo gdje se nalazi miš te se prema tome slično kao i u slučaju tipkovnice krećemo po ploči.

## **5.Bodovanje**

```
klasa Bodovanje
```

```
private bodovi
```

- broj bodova skupljen tokom igre

```
static public int broj_bodova
```

- dohvaća vrijednost varijable bodovi

```
static public void skupljenPlijen
```

- za svaki skupljen plijen dobivamo 3 boda