Projekat: Double Snake
Studenti: Enver Suljić 18376
Berina Suljić 18385
Datum : 30. 06. 2020.

DOUBLE SNAKE

- razrada projektnog zadatka sa zaduženjima članova tima -

Za svaki ekran ćemo imati jedan mode i iskoristit ćemo enum:

```
enum Mode {
        MAIN_MENU,
        COLOR_CHOOSE,
        GAME,
        END_GAME
}
```

Instanca Mode će označavati u kojem se mode-u trenutno igrica nalazi.

Funkcija **postaviEkran()** ispisuje na displej potreban UI na osnovu varijable Mode. Ona će se pozivati svaki poziv tikera mainTicker.

```
void postaviEkran(){
  switch (mode) {
    case MAIN_MENU:
      postaviMainMenu();
      break;
    case COLOR_CHOOSE:
      postaviColorChoose();
      break;
    case GAME:
      postaviGame();
      break;
    case END_GAME:
      postaviEndGame();
      break;
  }
}
```

Osim toga, inicijalizirat ćemo Interrupte za Buttone.

Trebamo napomenuti da je jedini dio koji ćemo uraditi u while petlji bez prekida u mainu (jer smo ustanovili da prekidima i ne radi baš očekivano), dio u kojem ćemo čitati vrijednosti sa potenciometara (simulacija joysticka za prvog igrača).

Akcije korisnika (pritisak buttona lijevo, desno, mijenjanje vrijednosti na potenciometrima i slično) ćemo razmatrati u funkcijama prviKontroler() i drugiKontroler(), za prvog i drugog igrača respektivno. U tim funkcijama će se u zavisnosti od mode-a u kojem je igrica trenutno, izvršavati određene akcije.

Imat ćemo 4 funkcije koje će prikazivati svaki od ekrana na displej:

- postaviMainMenu()
- postaviColorChoose()
- postaviGame()
- postaviEndGame()

U ovim funkcijama ćemo koristiti funkcije displeja kako bi iscrtavali željenje oblike i ispisivali, te dobili željeni izgled na displeju koji smo pokazali u specifikacijama.

Imat ćemo varijable koje označavaju boje zmija, smjerove zmija, njihov score, dužine, kao i niz pozicija svakog dijela svake zmije.Glava zmije će biti zadnja pozicija u tom nizu.

U mainu ćemo inicijalizirati Ticker koji će pozivati funkciju **gameTick()** i u slučaju da je trenutni mode GAME, ova funkcija realizuje cijelu logiku naše igrice. Možemo je nazvati „mozgom“ našeg programa. U njoj se realizuje pomijeranje zmija za jedno polje unaprijed u zavisnosti od smjera zmije. Također, ukoliko zmija pojede metu, njena dužina se produžava za jedan, a povećava se i njen score. Međutim, ukoliko jedna zmija udari u drugu zmiju, ona gubi, te je druga zmija pobjednik, a mode se mijenja na END_GAME i pojavljuje se ekran za game over. Sve će to biti realizovano u ovoj funkciji.

U nastavku ćemo postaviti isječke kodova za neke od funkcionalnosti koje su obrađene u gameTick() funkciji:

- Na samom početku ako mode nije GAME, funkcija se odmah završava

```
if(mode != GAME) return;
```

- Pomijeranje zmije svakih 0.3s vremena (period se naravno može i mijenjati, bit će definisan na početku programa)
  Ovdje je prikazano pomijeranje za samo prvu zmiju, dok će za drugu zmiju pomijeranje biti identično.

```
switch(smjerPrve){
    case UP:
      prvaY[duzinaPrve - 1] = prvaY[duzinaPrve - 1] - 1;
      if(prvaY[duzinaPrve - 1] == 0)
        prvaY[duzinaPrve - 1] = MAX_Y - 1;
      break;
    case DOWN:
      prvaY[duzinaPrve - 1] = (prvaY[duzinaPrve - 1] + 1) % MAX_Y;
      if(prvaY[duzinaPrve - 1] == 0) prvaY[duzinaPrve - 1] = 1;
      break;
    case LEFT:
      prvaX[duzinaPrve - 1] = prvaX[duzinaPrve - 1] - 1;
      if(prvaX[duzinaPrve - 1] == 0)
        prvaX[duzinaPrve - 1] = MAX_X - 1;
      break;
    case RIGHT:
      prvaX[duzinaPrve - 1] = (prvaX[duzinaPrve - 1] + 1) % MAX_X;
      if(prvaX[duzinaPrve - 1] == 0) prvaX[duzinaPrve - 1] = 1;
      break;
}
```

- Produžavanje prve zmije kada ona pojede metu (za drugu zmiju će biti identično)

```
if(prvaX[duzinaPrve - 1] == metaX && prvaY[duzinaPrve - 1] == metaY){
      scorePrve++;
      duzinaPrve++;
      prvaX[duzinaPrve - 1] = prvaX[duzinaPrve - 2];
      prvaY[duzinaPrve - 1] = prvaY[duzinaPrve - 2];
      postaviMetu();
   }else
      for(int i = 0; i < duzinaPrve - 2; i++){
         prvaX[i] = prvaX[i+1];
         prvaY[i] = prvaY[i+1];
      }
   prvaX[duzinaPrve - 2] = prvaGlavaX;
   prvaY[duzinaPrve - 2] = prvaGlavaY;
```

- Postavljanje pobjednika, ukoliko neka zmija udari u drugu, te prelazak na ekran game overa. Također ako zmija udari u samu sebe, ona gubi. U narednom kodu se prolazi kroz svaki dio prve zmije i provjerava se da li se neki dio poklapa sa drugim dijelom te zmije, ili se neki dio poklapa se dijelom druge zmije. (za drugu zmiju će biti realizovano identično)

```
for(int i = 0; i < duzinaPrve; i++){.
      if(prvaX[duzinaPrve - 1] == prvaX[i] && prvaY[duzinaPrve - 1] == prvaY[i] && i != duzinaPrve - 1){
         pobjednikPrvi = false;
         mode = END_GAME;
         break;
      }
      if(drugaX[duzinaDruge - 1] == prvaX[i] && drugaY[duzinaDruge - 1] == prvaY[i]){
         pobjednikPrvi = true;
         mode = END_GAME;
         break;
      }
}
```

Metu ćemo postaviti koristeći funkciju **postaviMetu()** u kojoj ćemo pozivati rand() funkciju kako bi koordinata mete uvijek bila nasumična:

```
void postaviMetu(){
   bool postavljena = true;
   do{
      postavljena = true;

      metaX = (rand() % (MAX_X - 2)) + 1;
      metaY = (rand() % (MAX_Y - 2)) + 1;

      for(int i = 0; i < duzinaPrve; i++)
         if(prvaX[i] == metaX && prvaY[i] == metaY)
            postavljena = false;

      for(int i = 0; i < duzinaDruge; i++)
         if(drugaX[i] == metaX && drugaY[i] == metaY)
            postavljena = false;
   }while(!postavljena);
}
```

Moramo napomenuti da nismo pisali koje dijelove će koji član tima uraditi, jer smo zaista sve zadatke zajedno radili putem Google Meeta, Team Viewera i međusobno nadopunjavali jedno drugo, pri čemu nam je trebalo otprilike 3 dana po 3 sata vremena.