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

```
enum Mode {
        MAIN_MENU,
        INSTRUCTIONS,
        GAME,
        END_GAME
};
```

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

Funkcija display

```
void display(){
   switch (mode) {
      case MAIN_MENU:
         mainMenu();
         break;
      case INSTRUCTIONS:
         uputstva();
         break;
      case GAME:
         igra();
         break;
      case END_GAME:
         krajIgre();
         break;
   }
}
```

Funkcija display
Ona će pozivati svaki poziv tikera mainTicker.

Akcije korisnika (pritisak buttona lijevo, desno, mijenjanje vrijednosti na potenciometrima i slično) ćemo razmatrati u funkcijama player1() i player2(), 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:

- **mainmenu()**
- **upustva()**
- **igra()**
- **krajIgre()**

Dokumentacija implementacije
Predmet: Ugradbeni sistemi
Studenti: Masleša Emina 18741 i Herak Sanina 18607

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, glava zmije kao i niz pozicija svakog dijela svake zmije.

Glavna funkcija naše igrice je **gameTick()**.
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.
Na samom početku ako mode nije GAME, funkcija se odmah završava

→ if(mode != GAME) return;

*** Pomijeranje zmije:**

```
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 zmije kada ona pojede metu**

```
 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 drawTarget() 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);
}
```