



Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku



Dokumentacija implementacije

Ugradbeni sistemi

Ime i prezime:

**Semina Muratović
Jalal Romano
Nejla Bečirspahić**

Kako bismo implementirali našu igricu "Stoni tenis" na Banggood displeju, prije svega je bilo potrebno importovati biblioteku "[SPI_TFT_ILI9341.h](#)" u naš projekat. Ova biblioteka nudi razne funkcije za manipulaciju izgleda displeja, a koje su bile od velike koristi za naš projekat.

Uveli smo također i biblioteke "[Arial12x12.h](#)", "[Arial24x23.h](#)", "[Arial28x28.h](#)", koje su nam poslužile za formatiranje slova na displeju.

Radi mogućnosti ponovne upotrebe određenih dijelova koda i lakše čitljivosti, kod smo podijelili u sljedeće funkcije, a neke od njih ćemo u nastavku detaljno obrazložiti:

- winner () : void
- gameover () : void
- dajUgao(donjaGranica : int, gornjaGranica : int) : int
- pozicionirajReket () : void
- postaviVrijednostIgre(brzina : float, duzina1 : int, duzina2 : int, koef1 : int, koef2 : int) : void
- vratiNaSredinu () : void
- igra (mode : int) : void
- letiLoptica () : void
- welcome () : void
- azurirajPlasman () : void
- multiplayerIgra () : void
- soloplayerIgra () : void
- main () : int

Prva funkcija koja se poziva iz glavne funkcije, jeste `welcome () : void`. To je jedna veoma lagana funkcija za implementaciju, koja samo poziva određene funkcije iz “`SPI_TFT_ILI9341.h`” biblioteke i diktira izgled početnog, “welcome”, fragmenta našeg projekta. U funkciji `welcome` smo koristili `Ticker`, koji nam je služio za poziv funkcije `letiLoptica()` svakih 30ms.

Prije početka same igre, na korisniku je odluka da li želi igrati u `one-player mode-u`, ili u `two-players mode-u`. Naša prvobitna zamisao jeste bila da pritiskom određenih tastera, korisnik pokreće jedan od navedenih mode-ova. Međutim, obzirom da u fakultetskoj laboratoriji nije bilo adekvatnih tastera koje bismo spojili na naš sistem, mi smo se odlučili za nešto drugačiju opciju.

Naime, ukoliko korisnik okrene potencijometar, koji se nalazi na pinu `PTB1`, na vrijednosti između 0.1 i 0.9, tada on prelazi u `two-players mode`, a u koliko okrene potencijometar na vrijednost manju ili jednaku 0.1, ili veću ili jednaku 0.9 tada on započinje igru u `one-player mode-u`.

```
243 int main() {
244     srand(static_cast<unsigned int>(time(0)));
245     welcome();
246     uLCD.set_font((unsigned char *)Arial28x28);
247     uLCD.foreground(Magenta);
248     uLCD.background(Green);
249     uLCD.set_orientation(1);
250     uLCD.cls();
251     ugao = rand() % 360;
252
253     //kad se drugi potenc zavrne do kraja u nekom smjeru, kreće multiplayer
254     if(a2<0.9 && a2>0.1) multiplayerIgra();
255     else soloplayerIgra();
256
257 }
```

Slika 1: Dio koda koji se odnosi na pozivanje funkcije `welcome` i izbora mode-a igrice

Funkcija `multiplayerIgra() : void`, služi samo za uzastopni poziv funkcije `igra(mode : int) : void` ali sa različitim vrijednostima parametra koji se prosljeđuje. Ovom funkcijom smo osigurali da two-players mode ima 3 levela, koji se razlikuju u brzini loptice i veličini reketa. Nakon svakog levela dolazi do ažuriranja plasmana, a na kraju se poziva i funkcija `winner`, koja služi za prikazivanje imena pobjednika na ekranu (*Fragment "winner" je ustvari i indikacija kraja igre za two-players mode, nakon čega se igrice mora restartovati, kako bi se ponovno mogla igrati.*).

```
228 void multiplayerIgra(){
229     igra(1);
230     azurirajPlasman();
231     igra(2);
232     azurirajPlasman();
233     igra(3);
234     azurirajPlasman();
235     winner();
236 }
```

Slika 2: Izgled funkcije `multiplayerIgra()`

Funkcija `gameover()`, jeste slična funkciji `winner()`, jer ona predstavlja kraj igre za one-player mode.

Sada ćemo preći na analizu najkompleksnije funkcije u našem projektu, a to je funkcija `igra(mode : int) : void`.

Ova funkcija predstavlja samu esenciju našeg projekta. A zbog njene kompleksnosti, osvrnut ćemo se samo na najvažnije dijelove koda.

U zavisnosti od toga da li je parametar koji se prosljeđuje ovoj funkciji 0, 1, 2 ili 3, imat ćemo različite dužine reketa, brzinu same loptice i različite mode-ove u kojim se igrice igra (mode=0 => single-player mode, mode!=0 => two-player mode).

Samu simulaciju kretanja reketa smo postigli tako što postoje dva pravougaonika čija je boja ista kao i boja pozadine, a čija y-koordinate se mijenjaju tako da "prekriju" ostatke pravougaonika crvene boje (reketa), koje nam više nisu potrebne. Također simulaciju kretanja loptice postizemo na sličan način, odnosno postoji loptica koja je iste boje kao i boja pozadine, a koja se postavlja na prethodne koordinate naše loptice.

```
//prikaz reketa i loptice sa novim pozicijama
uLCD.fillcircle(loptaX, loptaY, 5, Black);
uLCD.fillrect(reket1x1, reket1y1, reket1x2, reket1y2, Red);
uLCD.fillrect(reket2x1, reket2y1, reket2x2, reket2y2, Red);

//brisanje prethodne pozicije oba reketa, odnosno vraćanje zelene boje
uLCD.fillrect(0, 30, 5, reket1y1 - 1, Green);
uLCD.fillrect(0, reket1y2 + 1, 5, 240, Green);
uLCD.fillrect(315, 30, 320, reket2y1 - 1, Green);
uLCD.fillrect(315, reket2y2 + 1, 320, 240, Green);
```

Slika 3: Simulacija kretanja reketa

Ono što nam je također predstavljalo poteškoće, jeste i sam ugao pod kojim će se loptica kretati nakon odbijanja od krajeve terena ili reket.

Za ovu svrhu smo uveli biblioteku `rand()` iz "cmath" biblioteke, koja služi za vraćanje random integer-a.

U određenim slučajevima, koje smo naveli u kodu, ugao kretanja loptice mora biti u određenom opsegu. Dobivanje random ugla iz određenog opsega smo osigurali funkcijom `dajUgao(donjaGranica : int, gornjaGranica : int) : int`, čija implementacija je data na sljedećoj slici:

```
61  ✓ int dajUgao(int donjaGranica, int gornjaGranica) {  
62      int ugao;  
63      return donjaGranica + (rand() % (gornjaGranica - donjaGranica + 1));  
64  }
```

Slika 4

U slučajevima kada nije potrebno vratiti ugao iz određenog opsega, koristili smo da je `ugao=rand()%360`, a ovim zapisom smo osigurali da će vrijednost našeg ugla uvijek biti u segmentu `[0, 359]`.

```
101  if (loptaX == 310 && loptaY >= reket2y1 && loptaY <= reket2y2) ugao = dajUgao(91, 269);  
102  else if (loptaX == 10 && loptaY >= reket1y1 && loptaY <= reket1y2) ugao = dajUgao(271, 359);  
103  ✓ else if (loptaX >= 310 && (loptaY < reket2y1 || loptaY > reket2y2)) {  
104      poeniIgrac1++;  
105      vratiNaSredinu();  
106  }  
107  ✓ else if (loptaX <= 10 && (loptaY < reket1y1 || loptaY > reket1y2)) {  
108      poeniIgrac2++;  
109      vratiNaSredinu();  
110  }  
111  ✓ else if (loptaY >= 235) {  
112      if (ugao >= 180 && ugao <= 270) ugao = dajUgao(90, 180);  
113      if (ugao > 270 && ugao <= 360) ugao = dajUgao(0, 90);  
114  }  
115  ✓ else if (loptaY <= 35) {  
116      if (ugao >= 0 && ugao <= 90) ugao = dajUgao(270, 360);  
117      if (ugao > 90 && ugao <= 180) ugao = dajUgao(180, 270);  
118  }
```

Slika 5 : Postavljanje ugla kretanja loptice za različite slučajeve

Ispis bodova zavisi od toga da li je mode=0, ili je mode!=0. Kada je mode=0, nalazimo se u single-player mode-u i tada su poeni predstavljeni u obliku loptica. Na početku player ima 11 poena, svaki put kada ne uspije odbraniti lopticu, broj poena se smanjuje za 1. Ukoliko je mode!=0 tada smo u two-players mode-u i prikazuju se ostvareni poeni kako za prvog playera, tako i za drugog. Ukoliko player1 ne uspije odbraniti lopticu, player-u 2 se broj poena uvećava za 1 i obrnuto.

```
169  if(mod!=0) {
170      uLCD._printf("P1:", 4);
171      uLCD._printf(p1, 3);
172      uLCD.locate(210, 5);
173      uLCD._printf("P2:", 4);
174      uLCD._printf(p2, 3);
175  }
176  else{
177      //preostali zivoti u obliku loptica u soloplayer modu
178      int bodovi = 11 - poeniIgrac2;
179      int poluprecnik = 0;
180      uLCD.fillcircle(starix, 15, 10, Green);
181      while(bodovi!=0) {
182          uLCD.fillcircle(15+poluprecnik,15,10, Magenta);
183          poluprecnik = poluprecnik + 25;
184          bodovi--;
185      }
186      starix = poluprecnik - 10;
187  }
```

Slika 6: Ispis ostvarenih/preostalih poena