

# Projektni zadatak

iz predmeta Ugradbeni sistemi

**Tema: Igrica Pong**

Grupa: 1

Naziv tima:

Član tima: Paldum Medin

Br. indexa: 18154

- Za naš projektni zadatak, igricu Pong, razvit ćemo određeni broj klasa, kako bi implementacijom raznovrsnih metoda olakšali kompletnu implementaciju. Projekat će da sadži 4 osnovne klase: *Lopta*, *Reket*, *Ekran* i *Pong*, čiji interfejs slijedi:

**Lopta:** (x: int, y: int, radius: int, delta\_x: int, delta\_y: int)

[Jusuf - termin 2]

```
// Konstruktor
Lopta(int x, int y, int radius, int delta_x, int delta_y);
// Isertava loptu sa centrom u (x,y) uz pomjernje lopte za  $\Delta x/\Delta y$ 
void pomjeri(N5110 display);
// Isertava loptu sa centrom u (x,y) bez pomjeranja
void nacrtaj(N5110 display);
// Vraća veličinu lopte
int velicinaLopte ();
// Vraća logičku vrijednost da li je lopta udarila granicu ekrana x/y
bool isUdarilaGranicuX();
bool isUdarilaGranicuY();
// Da li je loptu udario reket (desni ili lijevi)
bool isUdarilaDesniReket(Reket &desniReket);
bool isUdarilaLijeviReket(Reket &lijeviReket);
// Metode za obrtanje smjera lopte
void obrniX();
void obrniY();
// Da li je lopta udarila gornji/donji okvir ekrana
bool isUdarilaGornjiOkvir();
bool isUdarilaDonjiOkvir();
// Ažuriranje kretnje lopte u zavisnosti od lokacije udara u reket
void promijeniKretanje(Reket &lijeviReket, Reket &desniReket);
// Vraća ugao kojim se lopta vraća u zavisnosti od udara
int dajUgao(int y_reket, int visina);
```

**Reket:** (x: int, y: int, h: int, d: int, bodovi: int)

[Jusuf - termin 2]

```
//Konstruktor
Reket(int x, int y, int h, int d);
// Isertava reket sa centrom u (x,y) uz pomjeranje reketa za vrijednost  $\Delta$ 
void pomjeri(N5110 display, int delta);
// Isertava loptu sa centrom u (x,y) bez pomjeranja
void nacrtaj(N5110 display);
// Da li je reket osvojio poen (desni ili lijevi)
```

```

bool isOsvojioPoenDesniReket(Lopta & lopta, Reket &desniReket);
bool isOsvojioPoenLijeviReket(Lopta & lopta, Reket &lijeviReket);
// Ažuriramo bodove reketa koji je osvojio poen
void dodajPoen();

```

**Ekran: (display: N5110)**

[Medin - termin 2]

```

// Konstruktor
Ekran (N5110 display);
// Crta mrežu koja dijeli polja igrača
void nacrtajMrežu();
// Crta ekran prilikom igre
void Gameplay (Reket &lijeviReket, Reket &desniReket, Lopta &lopta);
// Crta startni meni (Start/Exit)
void StartMenu(bool opcija);
// Crta inicijalni ekran sa imenima članova tima
void InitialScreen();
// Crta ekran za odabir težine igre
void SelectDifficulty (int opcija);
// Crta ekran za odabir modula igre SP/MP
void SelectMode (int opcija);
// Crta ekran završetka igre
void GameOver (int mode, int winner);

```

**Pong:(1...7 N5110\_dipslay: Pin, joyY1: Pin , joyY2: Pin, sw1 Pin, sw2: Pin, tokIgre: int, mode: int, tezina: int, zapocetiIgru: int, pobjednik: int, lopta: Lopta, lijeviReket: Reket, desniReket: Reket)**

[Medin - termin 3]

```

// Konstruktor
Pong: (Pin 1...7 N5110_dipslay, Pin joyY1, Pin joyY2, Pin sw1, Pin sw2, int tokIgre, Lopta &lopta, Reket & desniReket, Reket &lijeviReket);
// Inicijalno stanje pri paljenju igrice
void inicijalnoStanje();
// Stanje Start/Exit
void startGameStanje(int odabir);
// Stanje biranja modula igre SP/MP
void modeStanje(int mode);
// Stanje biranja težine igre

```

```
void tezinaStanje();  
// Stanje igre  
void igraStanje();  
// Stanje završetka igre  
void gameOverStanje();  
// Stanje izlaska iz igrice  
void izadjiStanje();
```

**Main.cpp** će sadržavati samo funkciju *main()* u kojem će se, u ovisnosti od stanja u kojem se igra nalazi, pozivati odgovarajuća metoda klase *Pong*. Realizacija će biti izvršena korištenjem *switch-case* bloka.

[Jusuf i Medin - termin 2]