
Objektno programiranje (C++)

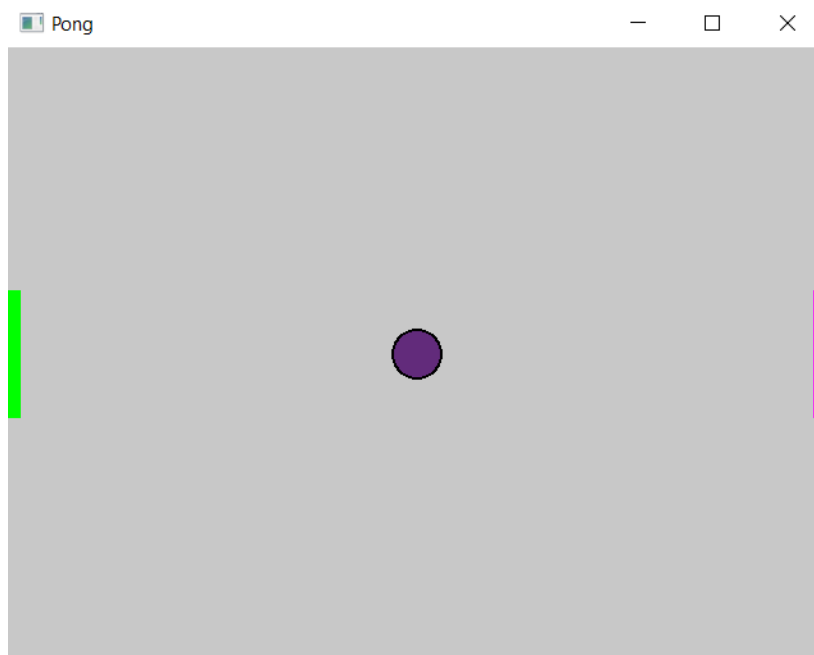
Druga domaća zadaća (ak. god. 2024./2025.)

Datum objave: petak, 4. travnja 2025. od 20:00h	Rok za predaju preko Merlina: ponedjeljak, 14. travnja 2025. do 20:00h
Ukupan broj zadataka: 1 zadatak	Ukupno moguće ostvariti bodova: 8 bodova

Zadatak. (6 bodova) Cilj zadatka je korištenjem SFML-a u programskom jeziku C++ implementirati igru *Pong* s više lopti za dva igrača. Svrha zadatka je provjeriti znanje vezano uz reagiranje na događaje, crtanje oblika, upotrebu (pseudo)slučajnih brojeva i pisanje vlastitih klasa. Prikaz igre čija pravila su opisana u nastavku (te kratki pregled datoteke `main.cpp`) možete vidjeti ovdje: www.youtube.com/watch?v=dmP4ayBQKO4 Za igru se koristi datoteka `main.cpp` koja je dostupna na Merlinu uz ovaj zadatak. Ta datoteka zahtijeva dva zaglavlja - zaglavlje `Lopta.h` i zaglavlje `Ploce.h`. Potrebno je napisati i preko Merlina predati ta dva zaglavlje (svako zaglavlje treba sadržavati implementaciju istoimene klase).

Pravila igre *Pong* s više lopti (za dva igrača):

Za igru se koristi prozor (tipa `sf::RenderWindow`) dimenzija 640×480 piksela. Na prozoru se uz lijevi i desni rub nalaze ploče različitih boja. Svaka od ploča ima dimenzije 10×100 piksela. Na prozoru se nalazi i 5 lopti. Svaka lopta ima radijus 20 piksela. Svih 5 lopti se početno nalazi na sredini prozora. Početni izgled prozora prikazan je na slici 1.

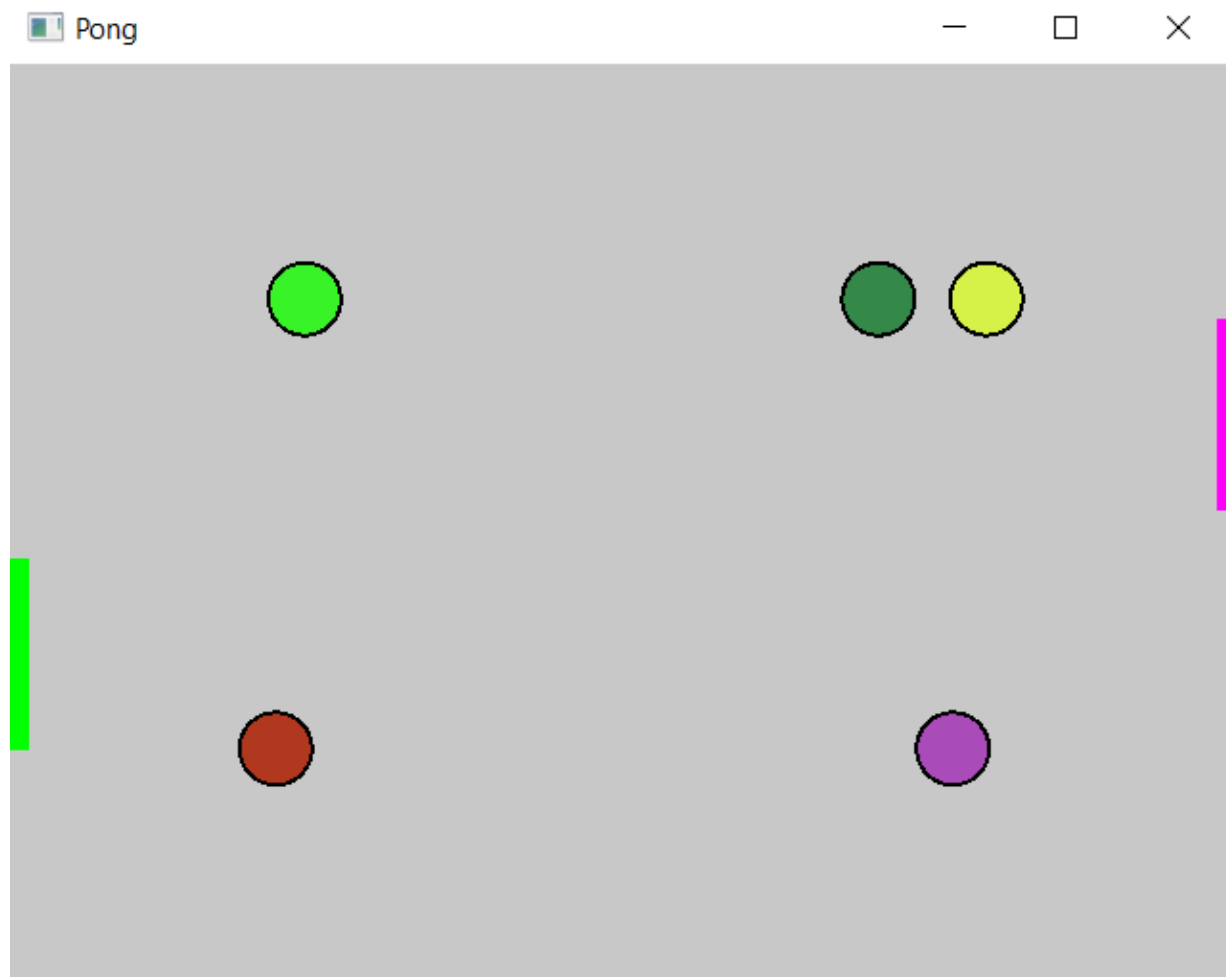


Slika 1. Prikaz na početku igre. Na slici se ne vidi svih 5 lopti jer su početno sve na istim koordinatama (na sredini prozora).

Svaka od ploča može se pomicati gore ili dolje (pri čemu niti jedna od ploča ne može izaći izvan prozora!). Prvi igrač pomiče lijevu ploču prema gore tako da drži pritisnutu tipu **W** te prema dolje tako da drži pritisnutu tipku **S**. Drugi igrač pomiče desnu ploču na isti način, ali sada pomoću tipaka strelice gore (za pomicanje prema gore) i strelice dolje (za pomicanje prema dolje) na tipkovnici.

Svaka od lopti ima na slučajan način određenu boju. Pritom svaka lopta ima i crni obrub debljine 2 piksela (ta dva piksela uračunata su u radijus od 20 piksela). Obrub je takav kako bi se svaka lopta vidjela na prozoru (čak i ako je slučajno odabrana boja lopte gotovo ista kao i boja pozadine prozora). Svaka od lopti ima početno odabran smjer kretanja na slučajan način. Pritom je važno da svaka lopta ima istu brzinu te da se niti jedna lopta ne kreće samo vertikalno (jer tada ne bi nikada došla do lijevog ili desnog ruba prozora).

Primjer izgleda prozora ubrzo nakon pokretanja igre prikazan je na slici 2.



Slika 2. Ubrzo nakon pokretanja igre, prikaz prozora bi mogao ovako izgledati (imati na umu da su boje i smjerovi kretanja lopti slučajno odabrani).

Svaka od lopti odbija se od gornjeg i donjeg ruba prozora te od ploče prvog te ploče drugog igrača (pritom se lopta odbija jednako, neovisno o tome u koji dio ploče je udarila). Ako lopta dodirne lijevi ili desni rub, ta lopta nestaje te se ponovo pojavljuje na sredini prozora (pri čemu joj je ponovo na slučajan način određena boja te smjer kretanja).

Kad pojedina lopta dodirne lijevi ili desni rub prozora te se ponovo pojavi na sredini, netko od igrača dobiva 1 bod (početno svaki od igrača ima 0 bodova). Dodjela bodova se odvija na sljedeći način:

- ako je neka lopta dodirnula lijevi rub prozora, tada drugi igrač dobiva 1 bod,
- ako je neka lopta dodirnula desni rub prozora, tada prvi igrač dobiva 1 bod.

Pri svakoj dodjeli nekog boda, potrebno je na (standardni) ekran (pomoću cout) ispisati trenutni rezultat (u zasebnom retku) u obliku:

bodovi prvog igrača : bodovi drugog igrača

Čim neki igrač skupi 10 bodova, kažemo da je taj igrač pobijedio. Tada je samo potrebno zatvoriti prozor s igrom.

Pregled main.cpp datoteke uz koju se mora ispravno izvršavati predani kod:

Priložena main.cpp datoteka na Merlinu ima sljedeći sadržaj:

```
#include <iostream>
#include <SFML/Graphics.hpp>
#include <vector>

#include "Lopta.h"
#include "Ploce.h"

using namespace std;

sf::Vector2u rezultat(0, 0);

int main(void) {
    sf::RenderWindow prozor(sf::VideoMode(640,480), "Pong");
    sf::Event event;

    vector<Lopta> lopte(5);
    Ploce ploce(prozor.getSize());

    while (prozor.isOpen()) {
        while(prozor.pollEvent(event))
            if (event.type == sf::Event::Closed)
                prozor.close();

        prozor.clear(sf::Color(200,200,200));

        ploce.azuriraj();
```

```

        for (auto& lopta : lopte) {
            lopta.azuriraj(ploce, rezultat);
            if (rezultat.x == 10 || rezultat.y == 10)
                break;
        }

        if (rezultat.x == 10 || rezultat.y == 10) {
            prozor.close();
            break;
        }

        for (auto& lopta : lopte)
            lopta.prikaz(prozor);

        ploce.prikaz(prozor);

        prozor.display();
    }

    return 0;
}

```

- U globalnom vektoru `rezultat` pamtimo trenutni rezultat prvog igrača (kao `rezultat.x`) te trenutni rezultat drugog igrača (kao `rezultat.y`).
- Koristimo vektor s 5 lopti (uočiti da se tu poziva konstruktor koji treba napraviti potrebne početne postavke za pojedinu loptu tipa `Lopta`).
- Koristimo i objekt `ploce` tipa `Ploce`. On predstavlja obje ploče na prozoru (tj. i lijevu ploču prvog igrača i desnu ploču drugog igrača). Kako bi konstruktor znao ispravno odrediti početne pozicije ploča na prozoru (uz lijevi ili desni rub prozora, te vertikalno na sredini prozora), potrebno mu je poslati dimenzije otvorenog prozora (tj. vektor `prozor.getSize()`).
- Unutar glavne petlje (uz standardnu provjeru zahtjeva korisnika za zatvaranjem prozora, čišćenjem prozora pomoću `clear` metode i prikaza pomoću `display`) događa se i sljedeće:
 - Ažurira se objekt `ploce` - potrebno je provjeriti drži li koji od igrača pritisnutu neku od relevantnih tipaka za pomicanje ploča, te ovisno o tome pomaknuti pojedinu ploču.
 - Ažurirati svaku loptu `lopta` iz vektora `lopte` - odrediti gdje će se pomaknuti lopta te ju pomaknuti. Tu treba paziti da se lopta odbija od gornjeg ili donjeg ruba prozora te od pojedine ploče igrača (zato bi `azuriraj` metoda trebala

primiti referencu na ploce kako bi upotrebom omeđujućih okvira lopte i pojedine ploče vidjeli treba li doći do odbijanja lopte od te ploče). Ako je lopta došla do lijevog ili desnog ruba prozora, lopta se vraća na sredinu prozora (te ima novu, slučajno odabranu boju te novi, slučajno odabrani smjer kretanja). Također, ažuriraj prima i referencu na rezultat, kako bi se rezultat mogao promijeniti (i ako se promijenio treba doći do ispisa trenutnog rezultata na ekran).

- Ako je u nekom trenutku bilo koji igrač skupio 10 bodova, nije više potrebno raditi ažuriranja lopti te se prozor zatvara i program završava.
- Nakon ažuriranja, potrebno je prikazati sve lopte i ploče na prozoru (zato sve prikaz metode primaju referencu na otvoreni prozor).

VAŽNO:

1. **Preko Merlina predajete samo datoteke `Lopta.h` i `Ploce.h`, koje sadrže implementacije istoimenih klasa.** Te klase moraju imati metode koje su ovdje navedene u zadatku te se moraju moći kompajlirati s danom datotekom `main.cpp`. U datotekama `Lopta.h` i `Ploce.h` možete dodati sve potrebne `include`-ove, ali **ne smijete** u te datoteke dodati

```
using namespace std;
```

Također, osim navedenih metoda, možete dodavati u klase vlastite metode (i ostale dijelove) za koje smatrate da su Vam potrebni za rješavanje ovog zadatka.

2. Ukoliko niste implementirali sve traženo u zadatku, možete napisati napomenu na Merlinu i/ili u komentaru na početku predane datoteke s kodom kako bi mogli ostvariti barem dio bodova predviđenih na ovoj zadaći.
3. Za dobivanje pozitivnog broja bodova, Vaš kod mora se kompilirati. Ukoliko Vaš kod sadrži problematične dijelove, tj. dijelove koji se ne kompiliraju, zakomentirajte ih i napišite uz njih napomenu u predanom kodu (kao komentar).
4. Traženu datoteku treba predati do roka navedenog na Merlinu (ponedjeljak, 14. travnja 2025. do 20:00). Naknadne predaje ove zadaće nisu moguće.