
Objektno programiranje (C++)

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

Datum objave: petak, 25. travnja 2025. od 20:00h	Rok za predaju preko Merlina: ponedjeljak, 5. svibnja 2025. do 20:00h
Ukupan broj zadataka: 1 zadatak	Ukupno moguće ostvariti bodova: 10 bodova

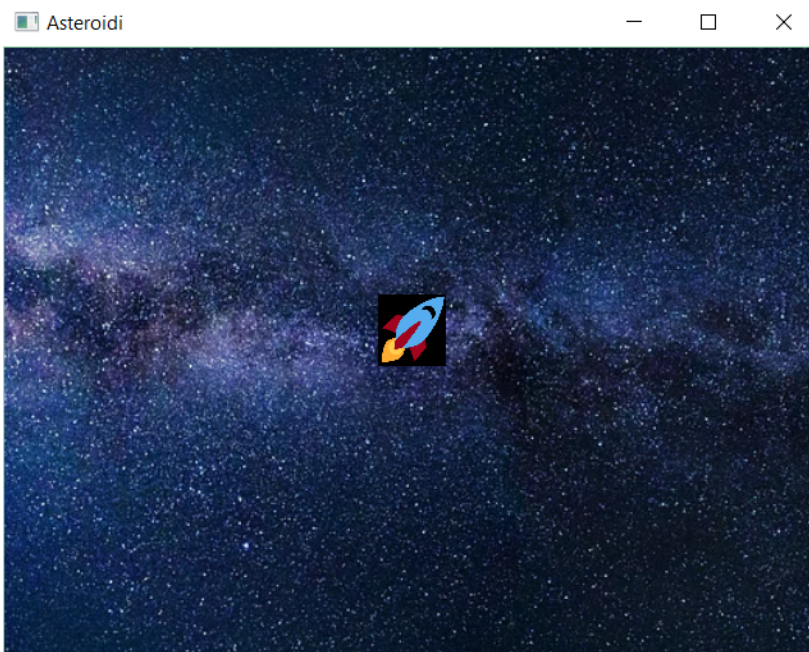
Zadatak. (10 bodova) Cilj zadatka je korištenjem SFML-a u programskom jeziku C++ implementirati pojednostavljenu verziju igre *Asteroids*¹ (s jednim igračem). Svrha zadatka je provjeriti znanje vezano uz upotrebu SFML-ovih klasa *Texture*, *Sprite*, *Clock* te upotrebu strukturiranja koda kakvo je prikazano na vježbama.

Za igru se koriste datoteke `main.cpp` i `Prozor.h` koje su dostupna na Merlinu uz ovaj zadatak. Sadržaj tih datoteka ispisan je na kraju ovog dokumenta s tekstom zadatka. Datoteke odgovaraju datotekama `main.cpp` i `Prozor.h` koje su napravljene na 6. vježbama iz ovog kolegija. Potrebno je prilagoditi i predati izmijenjenu datoteku `Igra.h` koja je također napisana na 6. vježbama, a sadrži sve podatke o igri. Zatim je potrebno napisati i predati datoteke `Brod.h` i `Asteroid.h` koje se brinu o brodu i (jednom!) asteroidu u igri koja je opisana u nastavku ovog zadatka. Također je potrebno predati sve slike (primjerice, s ekstenzijom `.png` ili drugačijeg tipa koji je podržan u SFML-u) koje se koriste za prikaz broda, asteroida i pozadine na ekranu tijekom igre (ukoliko niste autor upotrijebljenih slika, u komentaru na početku datoteke `Igra.h` navedite odakle su te slike preuzete).

Pravila pojednostavljene igre *Asteroids* (s jednim igračem):

Za igru se koristi prozor (tipa `sf::RenderWindow`) dimenzija 640×480 piksela. U naslovnoj traci prozora nalazi se tekst *Asteroidi*. Površinu prozora ispunjava pozadina koja se ne treba micati tijekom igre. Na njoj se početno na sredini ekrana nalazi svemirski brod. Jedan mogući izgled prozora prikazan je na slici 1. Brod se treba moći pomicati i rotirati upotrebom strelica na tipkovnici. Pritom brod ne smije moći niti jednim svojim dijelom izaći iz prozora.

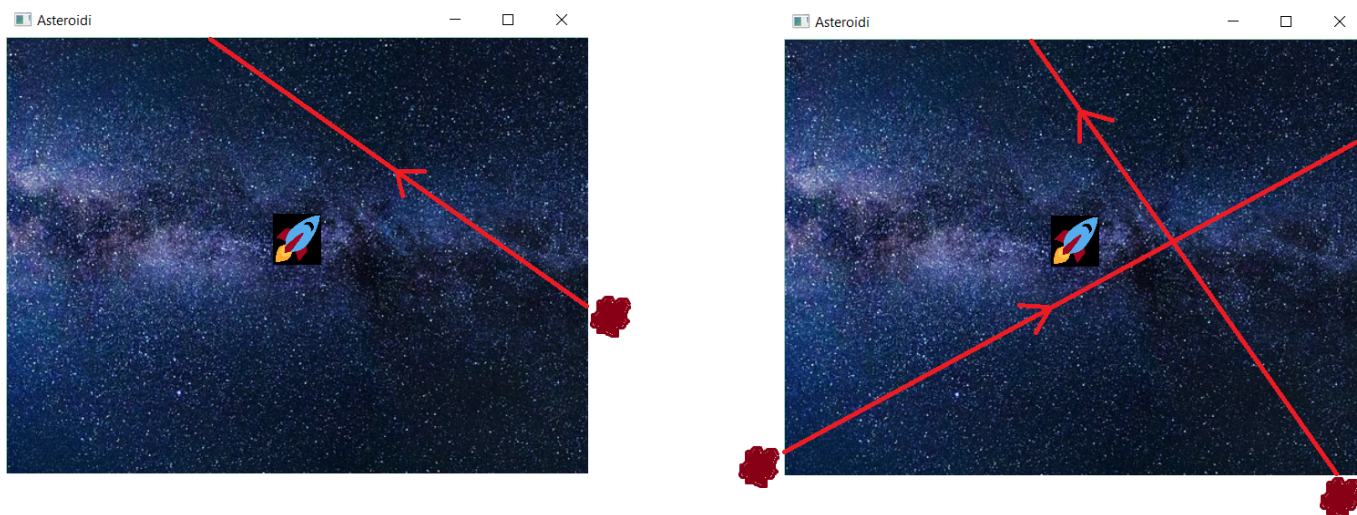
¹ www.youtube.com/watch?v=BgloG8yt-jA



Slika 1. Prikaz na početku igre. Prikazana je pozadina i na njoj (u sredini prozora) svemirski brod.

Početno postoji samo jedan asteroid koji nije prikazan na prozoru, nego je pozicioniran s vanjske strane do ruba prozora (vidi lijevi dio slike 2.). Zatim se asteroid pravocrtno giba kroz prozor dok ne izađe u potpunosti iz njega. Čim se to dogodi, on se ponovo pojavljuje s vanjske strane uz rub prozora. Pritom je mjesto gdje se ponovo pojavljuje slučajno odabrano, te je smjer kojim se giba također slučajno odabran (ali takav da se asteroid ponovo kreće preko prozora!). Taj asteroid koji je početno jedini u igru, u nastavku teksta ćemo nazivati glavnim asteroidom.

Svaki put kad glavni asteroid napusti područje prozora (te se ponovo pojavljuje negdje uz vanjski rub prozora), stvara se još jedan asteroid (vidi desni dio slike 2.). Taj novi asteroid ima isto ponašanje kao i glavni asteroid (s razlikom da se novi asteroidi stvaraju samo kad glavni asteroid izađe izvan prozora!). Svi asteroidi imaju istu brzinu kao i igračev svemirski brod. Napomenimo da strelice koje na slici 2. prikazuju gibanje pojedinog asteroida ne treba crtati (one su prikazane samo na slici 2. radi ilustracije gibanja pojedinog prikazanog asteroida). Broj asteroida koji u nekom trenutku mogu postojati u igri nije unaprijed ograničen.



Slika 2. Lijevi dio prikazuje situaciju na početku igre u kojoj je samo jedan asteroid nastao na slučajno odabranom mjestu izvan prozora te je prikazano njegovo daljnje gibanje po prozoru. Čim on u cijelosti napusti prozor, ponovo se pojavljuje uz neki vanjski rub prozora te se ponovo kreće po ekranu. Pritom se, kao što je prikazano na desnom dijelu slike, stvara još jedan asteroid s istim ponašanjem.

Cilj igre je što dulje izbjegavati sudar svemirskog broda s bilo kojim asteroidom. Čim dođe do sudara između svemirskog broda i nekog asteroida, igra završava. Tada je potrebno zatvoriti prozor s igrom. Na standardnom izlazu treba (pomoću `std::cout`) biti ispisana poruka igraču o tome koliko je dugo izdržao u igri (tj. koliko je, u sekundama, prošlo od početka igre do završetka igre, tj. do sudara svemirskog broda s nekim asteroidom). U ovoj igri ne promatramo sudare između asteroida (tj. možete samo pustiti da prijeđu jedan preko drugoga).

Bodovanje rješenja:

- (1.) Ispunjavanje svih uvjeta koji su do sad navedeni o samoj igri donosi najviše 5 bodova.
- (2.) Za najviše 2 boda, brzina brodova i asteroida ne smije ovisiti o „jačini računala” (preciznije, broju FPS-a koje računalo može prikazati). To možete testirati naredbom `prozor.setFramerateLimit(X)`; (uz konkretan broj umjesto X) koja je trenutno zakomentirana u datoteci `Prozor.h`.
- (3.) Za najviše 3 boda, kod mora biti strukturiran tako da se klasa `Brod`, čija je sva implementacija u datoteci `Brod.h`, brine o svemirskom brodu (tj. ima konstruktor, metodu za pomicanje broda i dr.), a klasa `Asteroid`, čija je sva implementacija u datoteci `Asteroid.h`, brine o (jednom!) asteroidu (tj. ima konstruktor, metodu za ponovno postavljanje asteroida nakon što izađe izvan ekrana i dr.). Pritom se klasa `Asteroid` brine samo o jednom asteroidu, dok u klasi `igra` (u datoteci `Igra.h`) imamo neki spremnik (vektor, listu ili drugi koji odaberete) u koji spremamo sve asteroide tipa `Asteroid` koji trenutno postoje u igri. Također se klase `Brod` i `Asteroid` moraju brinuti o pripadnim *spriteovima* koji se prikazuju kao svemirski brod ili asteroid (za *sprite* pozadine je zadužena klasa `igra`).

VAŽNO:

1. **Preko Merlina predajete samo datoteke Brod.h, Asteroid.h i Igra.h koje sadrže implementacije istoimenih klasa, te sve slike koje ste koristili za *spriteove* u igri.** Navedene klase moraju ispunjavati sve uvjete iz teksta zadatka te se moraju moći kompajlirati s danim datotekama main.cpp i Prozor.h. U datotekama Brod.h, Asteroid.h i Igra.h **ne smijete** napisati

`using namespace std;`

Također je preporučljivo na početku svake .h datoteke napisati: `#pragma once`

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. Tražene datoteke treba predati do roka navedenog na Merlinu (ponedjeljak, 5. svibnja 2025. do 20:00). Naknadne predaje ove zadaće nisu moguće.

Pregled datoteka main.cpp i Prozor.h uz koje se mora ispravno izvršavati predani kod:

Priložena main.cpp datoteka na Merlinu ima sljedeći sadržaj:

```
#include <SFML/Graphics.hpp>
#include <iostream>
#include "Igra.h"

using namespace std;

int main()
{
    Igra igra;
    while (!igra.gotovo()) {
        igra.obradiUlaz();
        igra.update();
        igra.renderiraj();
        igra.restartSata();
    }

    return 0;
}
```

Priložena Prozor.h datoteka na Merlinu ima sljedeći sadržaj:

```
#pragma once

#include <SFML/Graphics.hpp>
#include <string>

class Prozor {
private:
    sf::RenderWindow prozor;
    sf::Vector2u velicina;
    std::string naslov;
    bool gotov;
    bool cijeliZaslona;
    void Stvori();
    void Unisti();
    void Postavi(const std::string&, const sf::Vector2u&);
public:
    Prozor();
    Prozor(const std::string&, const sf::Vector2u&);
    ~Prozor();
}
```

```

        void prebaciNaCijeli();
        void ocisti();
        void prikazi();
        void update();
        void crtaj(sf::Drawable&);
        bool jelGotov() {
            return gotov;
        }
        bool jelCijeli() {
            return cijeliZaslon;
        }
        sf::Vector2u dohvatiVelicinu() {
            return velicina;
        }
};

void Prozor::crtaj(sf::Drawable& d) {
    prozor.draw(d);
}

void Prozor::update() {
    sf::Event event;
    while (prozor.pollEvent(event)) {
        if (event.type == sf::Event::Closed)
            gotov = true;
        else if (event.type == sf::Event::KeyPressed
            && event.key.code == sf::Keyboard::F5) {
            prebaciNaCijeli();
        }
    }
}

void Prozor::ocisti() {
    prozor.clear(sf::Color(200,200,200));
}

void Prozor::prikazi() {
    prozor.display();
}

Prozor::~~Prozor() {
    Unisti();
}

```

```

void Prozor::prebaciNaCijeli() {
    cijeliZaslon = !cijeliZaslon;
    Unisti();
    Stvori();
}

void Prozor::Postavi(const std::string& s,
                    const sf::Vector2u& u)
{
    naslov = s;
    velicina = u;
    gotov = false;
    cijeliZaslon = false; // vazno postaviti prije Stvori()
    Stvori();
}

Prozor::Prozor() {
    Postavi("Prozor", sf::Vector2u(640,480));
}

Prozor::Prozor(const std::string& s, const sf::Vector2u& u) {
    Postavi(s, u);
}

void Prozor::Stvori() {

    auto stil = (cijeliZaslon) ? sf::Style::Fullscreen
                             : sf::Style::Default;
    prozor.create(sf::VideoMode(velicina.x, velicina.y, 32),
                 naslov, stil);
    // sljedece se moze otkomentirati pri testiranju
    // (uz promjenu iznosa FPS-a):
    // prozor.setFramerateLimit(20);
}

void Prozor::Unisti() {
    prozor.close();
}

```