

SFML - Tekst

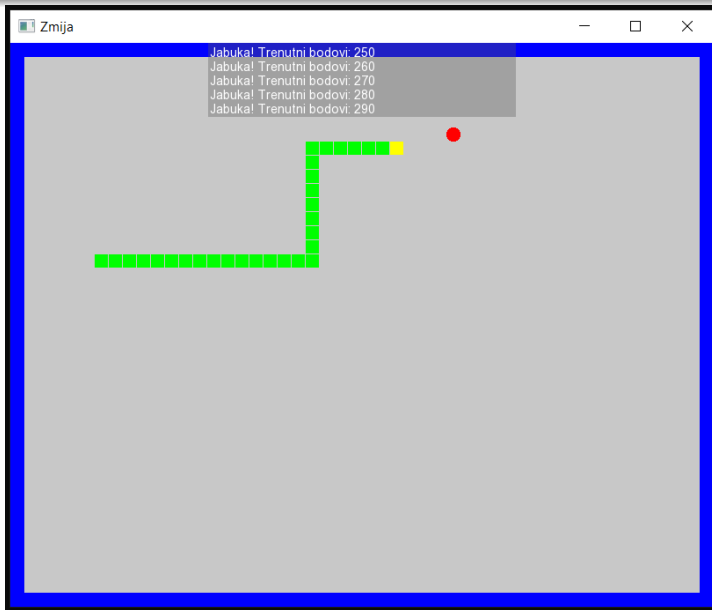
Objektno programiranje - 8. vježbe (1. dio)

Marko Živković

Prirodoslovno-matematički fakultet,
Sveučilište u Zagrebu

15. svibnja 2026. godine

Želimo dodati *textbox* s ispisom informacija o igri



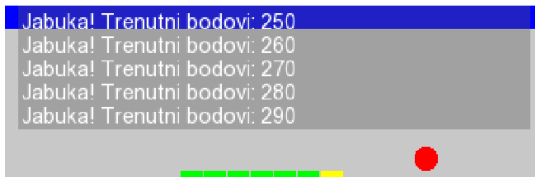
- dodamo u projekt novu datoteku **Textbox.h** početnog sadržaja:

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

class Textbox {
public:
private:
};
```

Pamćenje poruka za prikaz

```
...  
#include <string>  
#include <deque>  
  
class Textbox {  
    ...  
private:  
    int brLinija;  
    std::deque<std::string> poruke;  
    sf::RectangleShape pozadina; //polutransp. prav.  
};
```



- najviše je `brLinija` stringova u deque-u - novi se dodaje na kraj (ako ih je više od `brLinija` poruka, izbacujemo s početka)

Jednostavne funkcije te klase

- funkcija za **čišćenje spremnika** poruka za prikaz i funkcija za **dodavanje nove poruke** (i možda uklanjanje viška starih poruka)

```
class Textbox {
public:
    void Ocisti();
    void Dodaj(std::string);
    ...
};

void Textbox::Ocisti() {
    poruke.clear();
}

void Textbox::Dodaj(std::string nova) {
    poruke.push_back(nova);
    if (poruke.size() >= 6)
        poruke.pop_front();
}
```

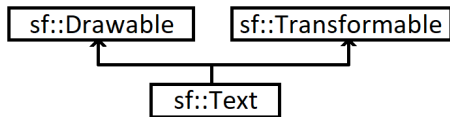
- prije **crtanja** teksta trebamo **font** (klasa **sf::Font**)
- učitavanje fonta iz datoteke (SFML ne učitava font automatski!):

```
sf::Font font;
if (!font.loadFromFile("arial.ttf")) {
    cout << "Font nije ucitan!" << endl;
}
```

- datoteka `arial.ttf` može se naći guglanjem i preuzeti - kao i pri učitavanju slika, datoteku stavimo u istu mapu s kodom

Klasa `sf::Text` - za crtanje teksta

Dijagram nasljeđivanja:



- prethodno učitani font postavljamo sa `setFont`
- string za ispis postavljamo sa `setString`
- veličinu znakova (u pikselima) postavljamo sa `setString`
- sa `setFillColor` postavljamo boju ispunje teksta (za boju i debljinu ruba: `setOutlineColor` i `setOutlineThickness`)

Primjer.

```
sf::Text tekst;
tekst.setFont(font);
tekst.setString("Poruka");
tekst.setCharacterSize(20);
tekst.setFillColor(sf::Color::Red);
```

Postavljanje stila teksta

```
void sf::Text::setStyle(uint32 style)
```

- može se koristiti kombinacija više stilova:

Primjer. Podebljan i potcrtan tekst:

```
tekst.setStyle(sf::Text::Bold  
              | sf::Text::Underlined);
```

- | - bitwise OR, koristi se da kombinira stilove

Enumeracija koja se koristi za stilove:

Regular	bez posebnog stila	0000
Bold	podebljano	0001
Italic	ukošeno	0010
Underlined	potcrtano	0100
StrikeThrough	precrtano	1000

Povratak kodu za naš *textbox*

```
class Textbox {  
    public:  
        void Postavi(int, int, float, sf::Vector2f);  
        ...  
    private:  
        sf::Font font;  
        sf::Text tekst;  
        ...  
};
```

- funkcija **Postavi** postaviti će:
 - maksimalan broj poruka u spremniku
 - veličinu znakova (broj piksela)
 - širinu tog prozora s tekstom
 - poziciju na kojoj se taj prozor s tekstom crta

```
void Textbox::Postavi(int brL, int velZn,
    float sirinaPro, sf::Vector2f pozPro) {
    brLinija = brL;
    font.loadFromFile("arial.ttf");
    tekst.setFont(font);
    tekst.setString("");
    tekst.setCharacterSize(velZn);
    tekst.setFillColor(sf::Color::White);
    tekst.setPosition(pozPro
        + sf::Vector2f(2.0f, 2.0f));
    pozadina.setSize(sf::Vector2f(sirinaPro,
        (brL * (velZn * 1.2f))));
    pozadina.setFillColor(sf::Color(90, 90, 90, 90));
    pozadina.setPosition(pozPro);
}
```

Konstruktori i destruktori

```
class Textbox {
    public:
        Textbox();
        Textbox(int, int, float, sf::Vector2f);
        ~Textbox();
        ...
};

Textbox::Textbox() {
    Postavi(5, 9, 200, sf::Vector2f(0,0));
}

Textbox::Textbox(int brL, int velZn,
float sirinaPro, sf::Vector2f pozPro) {
    Postavi(brL, velZn, sirinaPro, pozPro);
}

Textbox::~~Textbox() {
    Ocisti();
}
```

Funkcija Renderiraj

```
class Textbox {
    public:
        void Renderiraj(sf::RenderWindow&);
        ...
};

void Textbox::Renderiraj(sf::RenderWindow& p) {
    std::string sadrzaj;
    for (auto& it : poruke)
        sadrzaj.append(it + "\n");
    if (sadrzaj != "") {
        tekst.setString(sadrzaj);
        p.draw(pozadina);
        p.draw(tekst);
    }
}
```

Upotreba našeg *textboxa* - u datoteci *Zmija.h*

```
...
#include <string>
...
class Zmija {
public:
    Zmija(int, Textbox*);
    ...
    void Izgubio() {
        textbox->Dodaj("Igra gotova. Ukupni bodovi: "
            + std::to_string(DohvatiBodove()));
        izgubio = true;
    }
    ...
private:
    Textbox* textbox;
    ...
};
```

Promjena konstruktora za zmiju i poruka za živote

```
Zmija::Zmija(int v, Textbox* t) : velBloka(v),
    textbox(t) {
    ...
}
...
void Zmija::Odrezi(size_t i) { ...
    textbox->Dodaj("Izgubio zivot! Preostalo zivota: "
    + std::to_string(DohvatiZivote()));
    ...
}
```

Promjene u klasi Svijet

```
class Svijet {  
    public:  
        Svijet(int, sf::Vector2u, Textbox*);  
        ...  
    private:  
        Textbox* ptextbox;  
        ...  
};
```

```
Svijet::Svijet(int vBloka, sf::Vector2u vProzora,  
    Textbox* t) : velicinaBloka(vBloka),  
    velicinaProzora(vProzora),  
    ptextbox(t) { ... }
```

Upotreba našeg *textboxa* - u datoteci *Igra.h*

```
...
#include "Textbox.h"
...
class Igra {
...
private:
    Textbox textbox; //u Igra.h nije pokazivac!
    ...
};
...
void Igra::renderiraj() {
    p.ocisti();
    ...
    textbox.Renderiraj(p);
    p.prikazi();
}
```

Promjena funkcija Svijet::Update

```
void Svijet::Update(Zmija& igrac) {  
    if (igrac.KoordinateGlave() == jabukaKoord) {  
        igrac.Produlji();  
        igrac.PovecajBodove();  
        ptextbox->Dodaj(std::string  
            ("Jabuka! Trenutni bodovi: ")  
            + std::to_string(igrac.DohvatiBodove()));  
        ...  
    }
```

Promjena konstruktora za klasu Igra

```
Igra::Igra() : p("Zmija", sf::Vector2u(800, 640)),  
             svijet(16,p.dohvatiVelicinu(), &textbox),  
             zmija(svijet.dohvatiVBloka(), &textbox) {  
    textbox.Postavi(5,14,350,sf::Vector2f(225,0));  
    textbox.Dodaj("Dobro dosli u igru Zmija!");  
}
```