SFML - Događaji

Objektno programiranje - 4. vježbe (2. dio)

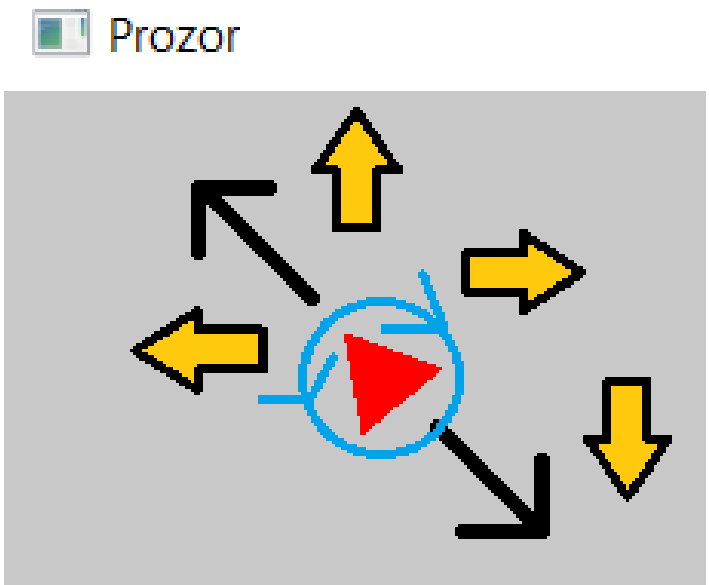
dr. sc. Sebastijan Horvat

Prirodoslovno-matematički fakultet,
Sveučilište u Zagrebu

2. travnja 2025. godine

Zadatak

Zadatak. Nacrtati crveni jednakostranični trokut (radijus opisane kružnice je 20 piksela) koji se početno nalazi na sredini 640×480 sivog prozora te koji se može pomicati pritiskanjem strelica na tipkovnici (kao na slici ispod: strelica gore je naprijed, strelica dolje je natrag, strelica lijevo je rotacija u smjeru obrnutom od smjera kazaljke na satu, a strelica desno je rotacija u smjeru kazaljke na satu).



`sf::Event` klasa

- **unija** $\Rightarrow$ samo jedan njegov član valjan (onaj koji odgovara tipu događaja)
- samo funkcije `pollEvent` i `waitEvent` daju valjane događaje

Tipična petlja za događaje:

```
sf::Event event;
while (prozor.pollEvent(event)) {
    switch (event.type) {
        case sf::Event::Closed:
            prozor.close();
            break;
        case sf::Event::KeyPressed:
            ...
            break;
        default:  //ako nas ne zanimaju ostali
            break;
        }
}
```

Događaj promjene veličine prozora

- pri promjeni veličine prozora (ili korisnik ručno ili u programu pozivom `prozor.setSize(...)`
- omogućuje promjenu postavki renderiranja nakon događaja

Primjer. Ispis dimenzija prozora pri promjeni veličine prozora:

```
while(prozor.isOpen()) {
  sf::Event event;
  while (prozor.pollEvent(event)) {
      switch (event.type) {
          case sf::Event::Closed:
              prozor.close();
              break;
          case sf::Event::Resized:
              cout << "(" << prozor.getSize().x << ","
                   << prozor.getSize().y << ")" << endl;
              break;
      }
  }
}
```

Događaji dobivanja i gubljenja fokusa

- promjena fokusa događa se kad promijeni trenutno aktivni prozor
- prozor koji nije u fokusu ne dobiva ulaz s tipkovnice
- primjer upotrebe: pauziranje igre kad prozor nije u fokusu

Primjer.

```
...
case sf::Event::GainedFocus:
    cout << "Dobio fokus!" << endl;
    break;
case sf::Event::LostFocus:
    cout << "Izgubio fokus!" << endl;
    break;
...
```

Ulazak miša u prozor i izlazak miša iz prozora

- `sf::Event::MouseEntered` - prilikom ulaska pokazivača miša u prozor
- `sf::Event::MouseLeft` - prilikom izlaska pokazivača miša iz prozor

Primjer.

```
...
case sf::Event::MouseEntered:
    cout << "Usao u prozor!" << endl;
    break;
case sf::Event::MouseLeft:
    cout << "Napustio prozor!" << endl;
    break;
...
```

Pomak miša

- **`sf::Event::MouseMoved`** - pri pomicanju miša **unutar prozora** (ne računa se naslovna traka niti rub)
- radi čak i ako prozor nije u fokusu
- dobivanje koordinata pokazivača miša (unutar prozora!):

(**`mouseMove.x`**, **`mouseMove.y`**)

Primjer.

```
...
case sf::Event::MouseMoved:
    cout << "(" << event.mouseMove.x << ","
         << event.mouseMove.y << ")" << endl;
    break;
...
```

Pritisak i otpuštanje tipke miša

- `sf::Event::MouseButtonPressed` - pritisak tipke miša
- `sf::Event::MouseButtonReleased` - otpuštanje tipke miša
- koordinate pokazivača dobivamo pomoću `mouseButton`
- tipke miša koje SFML podržava: `Left` (lijevo), `Right` (desno), `Middle` (kotačić), `XButton1`, `XButton2` (tipke sa strane)

Primjer.

```
...
case sf::Event::MouseButtonPressed:
    if(event.mouseButton.button == sf::Mouse::Left)
        cout << "Lijevi klik na ("
             << event.mouseButton.x << ","
             << event.mouseButton.y << ")" << endl;
    break;
...
```

Napomena. Ostali događaji (poput `MouseWheelScrolled`): link

Pritisak i otpuštanje tipke na tipkovnici

- događaji tipa **`KeyPressed`** i **`KeyReleased`**
- to koristimo za **reakciju** na pritisak tipke (npr. pomicanje lika u igri), a ne `TextEntered` događaje (za unos teksta - njih ćemo raditi kasnije)

Primjer.

```
...
case sf::Event::KeyPressed:
    cout << "Tipka!" << endl;
    break;
...
```

- probati gornji kod - pritisnuti i držati neku tipku
- generira se više `KeyPressed` događaja, s *defaultnim* **kašnjenjem** (isto kao pri pisanju u neki dokument)
- za onemogućavanje ponavljanja `KeyPressed` događaja pri držanju pritisnute tipke treba na prozoru izvršiti:

```
prozor.setKeyRepeatEnabled(false)
```

Kako provjeriti koja tipka je pritisnuta

- `key.code` daje kod pritisnute tipke
- kodovi - `sf::Keyboard::`
  - A, B, C, ..., X, Y, Z, Num0, Num1, ..., Num9
  - LControl, LShift, LAlt, LSystem, RControl, RShift, RAlt, RSystem
  - Escape, Menu, Pause
  - LBracket ([), RBracket (]), Semicolon (;), Comma (,), Period (.), Quote (‘), Slash (/), Backslash (\), Tilde (~), Equal (=), Hyphen (-)
  - Space, Enter, Backspace, Tab, PageUp, PageDown, End, Home, Insert, Delete
  - Add (+), Subtract (-), Multiply (*), Divide (/)
  - Left ($\leftarrow$), Right ($\rightarrow$), Up ($\uparrow$), Down ($\downarrow$)
  - Numpad0, Numpad1, ..., Numpad9
  - F1, F2, ..., F15
- sljedeći kodovi su zastarjeli (u zagradama je navedeno čime su zamijenjeni):
  - Dash (Hyphen)
  - BackSpace (Backspace)
  - BackSlash (Backslash)
  - SemiColon (Semicolon)
  - Return (Enter)

Još malo o tipkama

- funkcija za provjeru je li određena tipka trenutno pritisnuta:

```
static bool sf::Keyboard::isKeyPressed(Key key)
```

Napomena.

- neke tipke imaju posebno značenje (ovisno o operacijskom sustavu) pa to može dovesti do neočekivanog ponašanja
- primjerice, na Windowsu i Visual studiu tipka F12 pokreće *debugger*

Primjer

```
...
case sf::Event::KeyPressed:
    if(event.key.code == sf::Keyboard::Up)
        cout << "Tipka gore!" << endl;
    break;
...
```

Rješenje početnog zadatka

```
sf::Vector2f pocetni_pomak(0.f, -0.1f); //gore

int main() {
    sf::RenderWindow prozor(sf::VideoMode(640,480),
                            "Prozor");
    float radijus = 20.f;
    sf::CircleShape t(radijus,3);
    t.setFillColor(sf::Color::Red);
    t.setOrigin(radijus, radijus);
    t.setPosition(prozor.getSize().x / 2.f,
                  prozor.getSize().y / 2.f);
    sf::Vector2f pomak(pocetni_pomak);
    float kut = 0;
    while (prozor.isOpen()) {
        sf::Event event;
        ...sljedeći slajd...
    }
    return 0;
}
```

Nastavak rješenja (unutar `while` petlje)

```
while (prozor.pollEvent(event))
    if(event.type == sf::Event::Closed)
        prozor.close();
if(sf::Keyboard::isKeyPressed(sf::Keyboard::Up))
    t.move(pomak);
if(sf::Keyboard::isKeyPressed(sf::Keyboard::Down))
    t.move(-pomak);
if(sf::Keyboard::isKeyPressed(sf::Keyboard::Left)){
    t.setRotation(kut -= 0.1);
    updejt_pomaka(pomak, t.getRotation());
}
if(sf::Keyboard::isKeyPressed(sf::Keyboard::Right)){
    t.setRotation(kut += 0.1);
    updejt_pomaka(pomak, t.getRotation());
}
prozor.clear(sf::Color(200, 200, 200, 255));
prozor.draw(t);
prozor.display();
```

Pomoćna funkcija

```
void updejt_pomaka(sf::Vector2f& pomak, float kut){
    float rad = kut / 180 * 3.14;
    pomak.x = cos(rad) * pocetni_pomak.x
              - sin(rad) * pocetni_pomak.y;
    pomak.y = sin(rad) * pocetni_pomak.x
              + cos(rad) * pocetni_pomak.y;
}
```

- ovo zahtijeva `#include<cmath>`
- `kut` je u stupnjevima, a za funkcije `sin` i `cos` trebamo u radijanima - zato smo prvo pretvorili stupnjeve u radijane