

# SFML

Biblioteka programistyczna

Patryk Sroczyński

Politechnika Śląska  
Gliwice

# Czym jest SFML?



**SFML** to akronim **Simple and Fast Multimedia Library**

Jest to wieloplatformowa biblioteka programistyczna ułatwiająca tworzenie gier oraz programów multimedialnych.

Autorem jest Laurent Gomila

# Główne właściwości



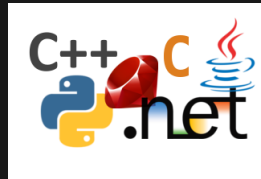
## Multi-media

Zawiera 5  
modułów: system,  
window, graphics,  
audio i network



## Multi-platform

Można uruchomić  
oraz skompilować  
na Windowsie,  
Linuxie, macOS



## Multi-language

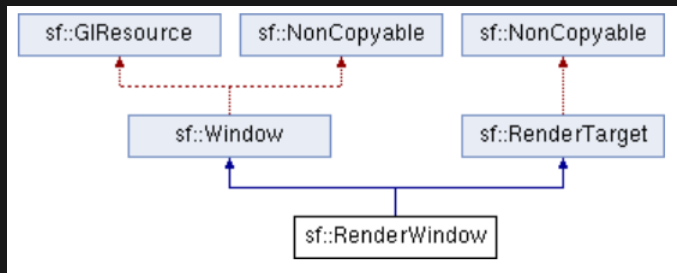
Głównym językiem  
jest C++, ale  
można również  
zbindować np.:  
Java, Ruby,  
Python

# Moduły

1. *System* – obsługuje czas i wątki
2. *Window* – obsługuje okna i interakcję z użytkownikiem
3. *Graphics* – umożliwia renderowanie grafiki
4. *Audio* – dostarcza interfejsu do odtwarzania muzyki i dźwięków
5. *Network* – odpowiedzialny za komunikację sieciową

# Wybrane metody klasy sf::RenderWindow

Klasa sf::RenderWindow jest najważniejsza, ponieważ dzięki niej możemy stworzyć okno, a następnie wyświetlać zawartość






# Wybrane metody klasy `sf::RenderWindow`

1. konstruktor: `sf::RenderWindow::RenderWindow(VideoMode, const String, Uint32, const ContextSettings )`
2. `void clear(const Color color=Color(0,0,0,255))`
3. `void draw(const Drawable drawable, const RenderStates states=RenderStates::Default)`
4. `void display()`
5. `void setVerticalSyncEnabled(bool enabled)`
6. `void setFrameRateLimit(unsigned int limit)`

# Program tworzący puste okno

```
1  #include <SFML/Graphics.hpp>
2
3  int main() {
4      sf::RenderWindow window(sf::VideoMode(800,
5      600), "Tytul");
6
7      while (window.isOpen()) {
8          sf::Event event;
9          while (window.pollEvent(event)) {
10             if (event.type == sf::Event::Closed)
11                 window.close();
12             }
13             window.clear();
14             window.display();
15         }
16     }
```

# Tworzenie prostych kształtów

Kształt	Wygląd	Kod	Rysowanie
○		<pre>sf::CircleShape circle(50.f);</pre>	<pre>window.draw(circle);</pre>
□		<pre>sf::RectangleShape rect(sf::Vector2f(50.f, 50.f));</pre>	<pre>window.draw(rect);</pre>
△		<pre>sf::CircleShape trian- gle(80, 3);</pre>	<pre>window.draw(triangle);</pre>



# Prosta gra 1/4

```
1 #include<iostream>
2 #include<SFML/Graphics.hpp>
3 #include<SFML/Window.hpp>
4 #include<SFML/System.hpp>
5 #include<cstdlib>
6 #include<vector>
7
8 int main()
9 {
10     srand(time(NULL));
11     sf::RenderWindow window(sf::VideoMode(640, 480),
12         "Ball shooter!");
13     window.setFramerateLimit(60);
14
15     //Balls
16     sf::CircleShape projectile;
17     projectile.setFillColor(sf::Color::Red);
18     projectile.setRadius(5.f);
19
20     sf::RectangleShape enemy;
21     enemy.setFillColor(sf::Color::Magenta);
22     enemy.setSize(sf::Vector2f(50.f, 50.f));
23     sf::CircleShape player;
24     player.setFillColor(sf::Color::White);
25     player.setRadius(50.f);
26     player.setPosition(window.getSize().x / 2 - player.getRadius(),
27         window.getSize().y - player.getRadius() * 2 - 10.f);
28     sf::Vector2f playerCenter;
29     int shootTimer{};
```

# Prosta gra 2/4

```
30     std::vector<sf::CircleShape> projectiles;
31     projectiles.push_back(sf::CircleShape(projectile));
32
33     std::vector<sf::RectangleShape> enemies;
34     enemies.push_back(sf::RectangleShape(enemy));
35     int enemySpawnTimer = 0;
36     while (window.isOpen()) {
37         sf::Event event;
38         while (window.pollEvent(event)) {
39             if (event.type == sf::Event::Closed)
40                 window.close();
41             if (event.KeyPressed && event.key.code == sf::Keyboard::Escape)
42                 window.close();
43         }
44         playerCenter = sf::Vector2f(player.getPosition().x + player.getRadius(),
45                                     player.getPosition().y + player.getRadius());
46         player.setPosition(sf::Mouse::getPosition(window).x, player.getPosition().y);
47
48         if (shootTimer < 5)
49             shootTimer++;
50
51         if (sf::Mouse::isButtonPressed(sf::Mouse::Left) && shootTimer >= 5){
52             projectile.setPosition(playerCenter);
53             projectiles.push_back(sf::CircleShape(projectile));
54             shootTimer = 0;
55         }
56
57         for (size_t i = 0; i < projectiles.size(); i++){
58             projectiles[i].move(0.f, -10.f);
59
60             if (projectiles[i].getPosition().y <= 0)
61                 projectiles.erase(projectiles.begin() + i);
62         }
```

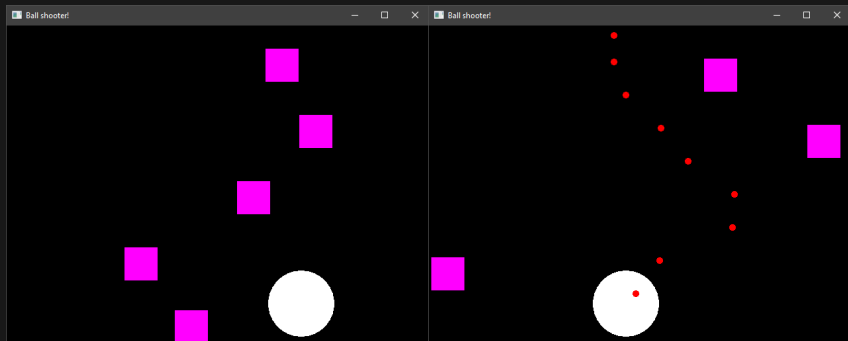
# Prosta gra 3/4

```
63 if (enemySpawnTimer < 20)
64     enemySpawnTimer++;
65
66 if (enemySpawnTimer >= 20) {
67     enemy.setPosition((rand() % int(window.getSize().x
68 - enemy.getSize().x)), 0.f);
69     enemies.push_back(sf::RectangleShape(enemy));
70     enemySpawnTimer = 0;
71 }
72
73 for (size_t i = 0; i < enemies.size(); i++) {
74     enemies[i].move(0.f, 5.f);
75
76     if (enemies[i].getPosition().y > window.getSize().y)
77         enemies.erase(enemies.begin() + i);
78 }
79
80
81 //Collision
82 for (size_t i = 0; i < projectiles.size(); i++) {
83     for (size_t j = 0; j < enemies.size(); j++) {
84         if (projectiles[i].getGlobalBounds().
85             intersects(enemies[j].getGlobalBounds())) {
86             projectiles.erase(projectiles.begin() + i);
87             enemies.erase(enemies.begin() + j);
88             break;
89         }
90     }
91 }
```

# Prosta gra 4/4

```
90         //Draw
91         window.clear();
92
93         window.draw(player);
94
95         for (size_t i = 0; i < enemies.size(); i++)
96             window.draw(enemies[i]);
97
98         for (size_t i = 0; i < projectiles.size(); i++)
99             window.draw(projectiles[i]);
100
101         window.display();
102     }
103     return 0;
104 }
```

# Ekran gry



# Dziękuję za uwagę!

Bibliografia:

<https://www.sfml-dev.org/>

[https://en.wikipedia.org/wiki/Simple\\_and\\_Fast\\_Multimedia\\_Library](https://en.wikipedia.org/wiki/Simple_and_Fast_Multimedia_Library)