2D Graphics with SFML

Simple and Fast Multimedia Library

Meeting C++ 2018 | Lukas Dürrenberger | @DarkCisum


REMNANTS OF
NAEZITH


Castle Clamber


CENDRIC
AN RPG-PLATFORMER


AWAY TEAM


ZERAN'S
FOLLY


CUIT


GEMSTONE KEEPER
Gamepopper
Alpha Build
Start Game
Score Attack
Daily Run
Boss Rush
Exit


OCTO
NAUT
ODYSSEY


TURNOVER


MOONQUEST


DISPERSIO


DoodleBob
Virtual Companion


CREA

Getting Started

- `git clone https://github.com/SFML/SFML.git`
  `cmake . [options]`
  `cmake --build . --target install [options]`
- `conan install sfml/2.5.1@bincrafters/stable`
- `vcpkg install sfml`
- `apt-get install libsfml-dev`
- `pacman -S sfml`
- `brew install sfml`
- …

Main Loop

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

int main()
{
    sf::RenderWindow window{ { 800, 800 }, "Main Loop - Meeting C++ 2018" };
    window.setFramerateLimit(60);









}
```


Main Loop

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

int main()
{
    sf::RenderWindow window{ { 800, 800 }, "Main Loop - Meeting C++ 2018" };
    window.setFramerateLimit(60);

    while (window.isOpen())
    {









    }
}
```

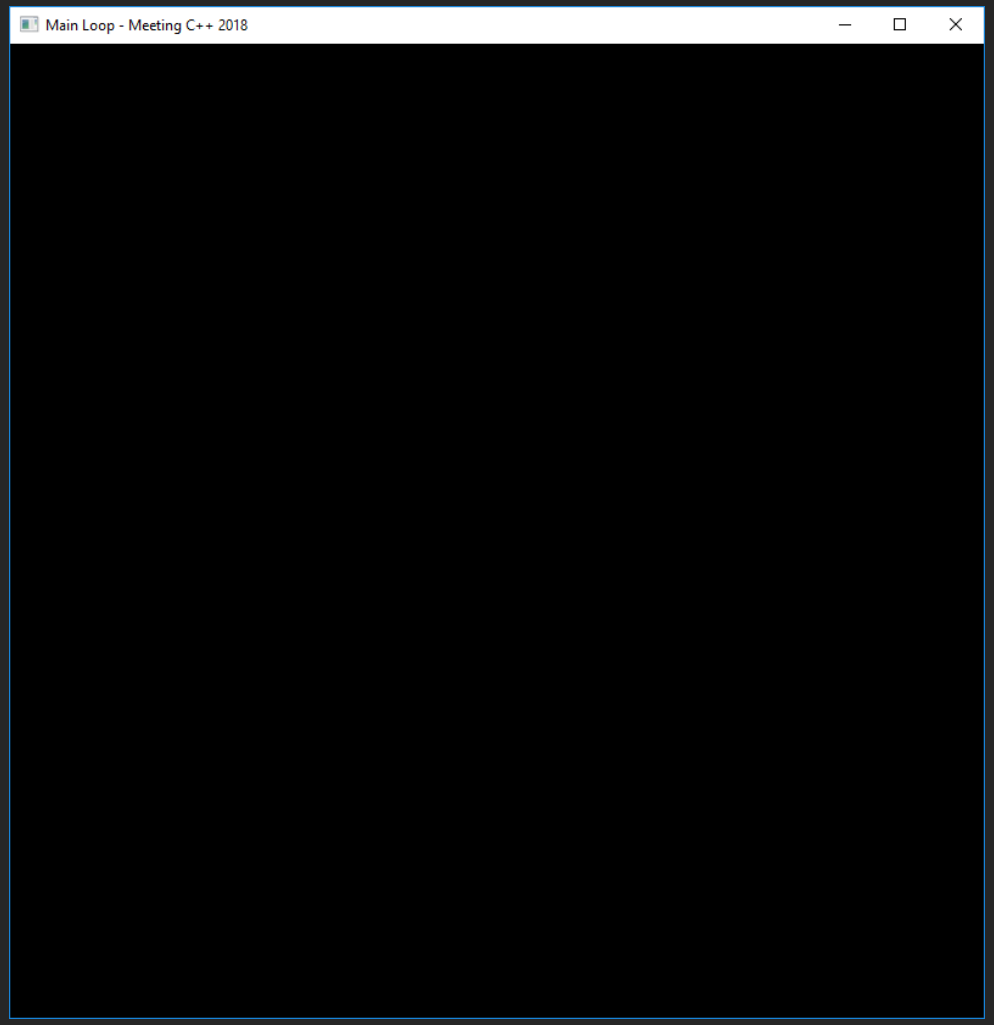

Main Loop

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

int main()
{
    sf::RenderWindow window{ { 800, 800 }, "Main Loop - Meeting C++ 2018" };
    window.setFramerateLimit(60);

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



    }
}
```


Main Loop

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

int main()
{
    sf::RenderWindow window{ { 800, 800 }, "Main Loop - Meeting C++ 2018" };
    window.setFramerateLimit(60);

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

        window.clear();
        window.display();
    }
}
```


Main Loop

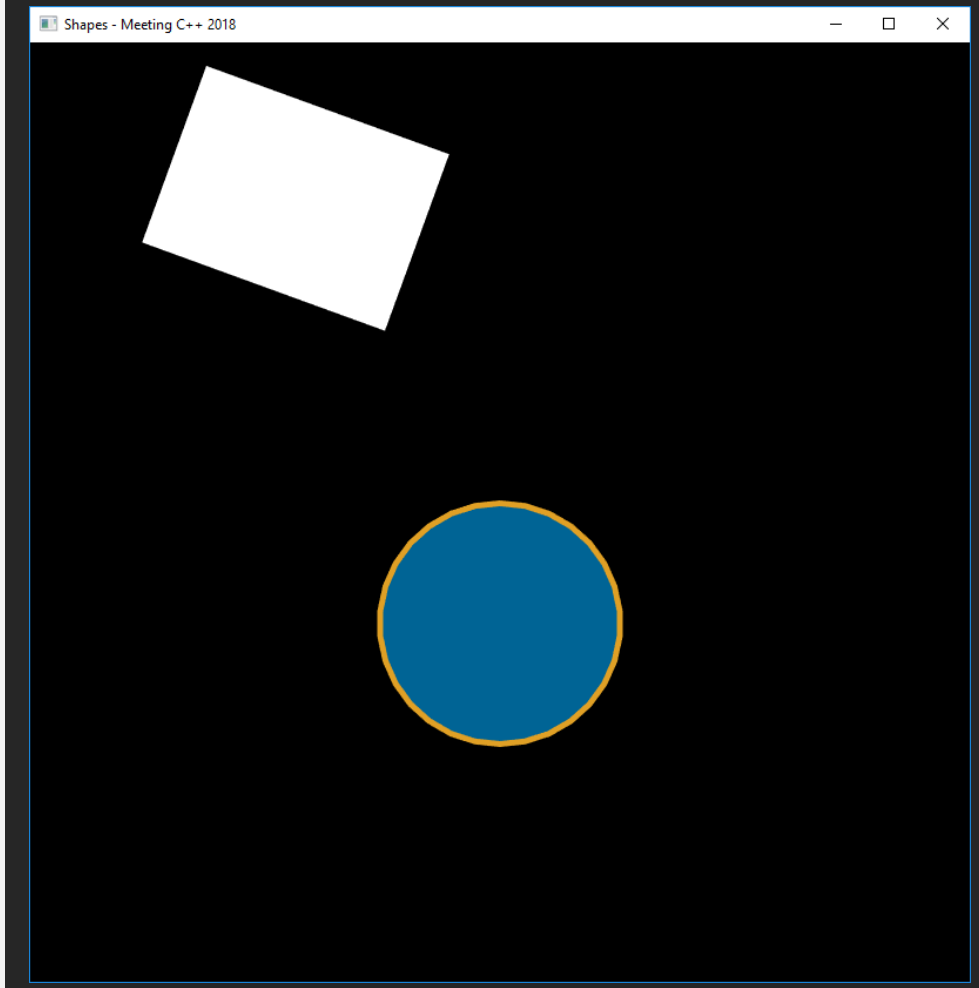


Shapes

```
sf::RectangleShape rectangle{ { 220.f, 160.f } };
rectangle.setFillColor(sf::Color::White);
rectangle.setPosition({ 150.f, 20.f });
rectangle.rotate(20.f);

// ...

window.clear();
window.draw(rectangle);
window.display();
```

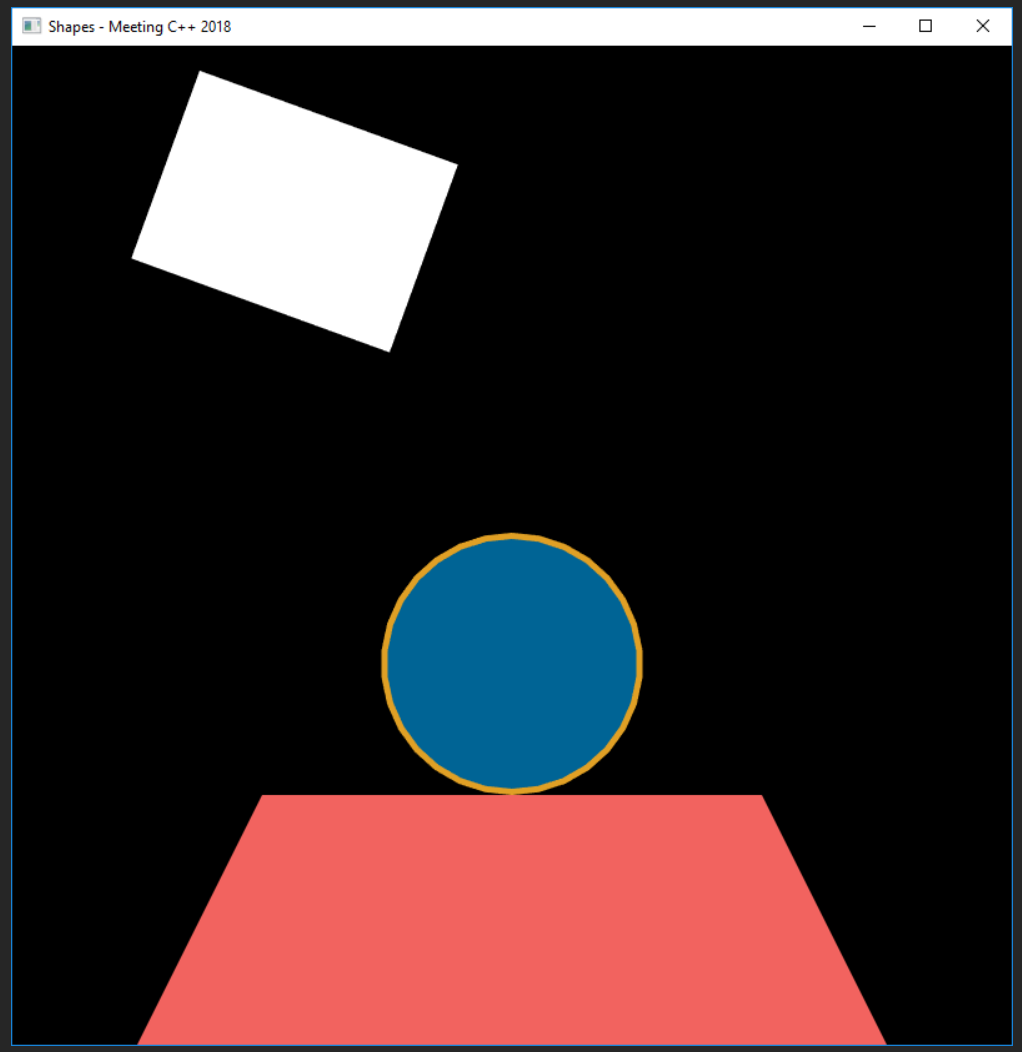
Shapes

```
sf::CircleShape circle{ 100.f };
circle.setFillColor(sf::Color{ 0x006495FF });
circle.setOutlineColor(sf::Color{ 224, 160, 37, 255 });
circle.setOutlineThickness(5.f);
circle.setPosition({ 300.f, 395.f });

// ...

window.clear();
window.draw(ractangle);
window.draw(circle);
window.display();
```



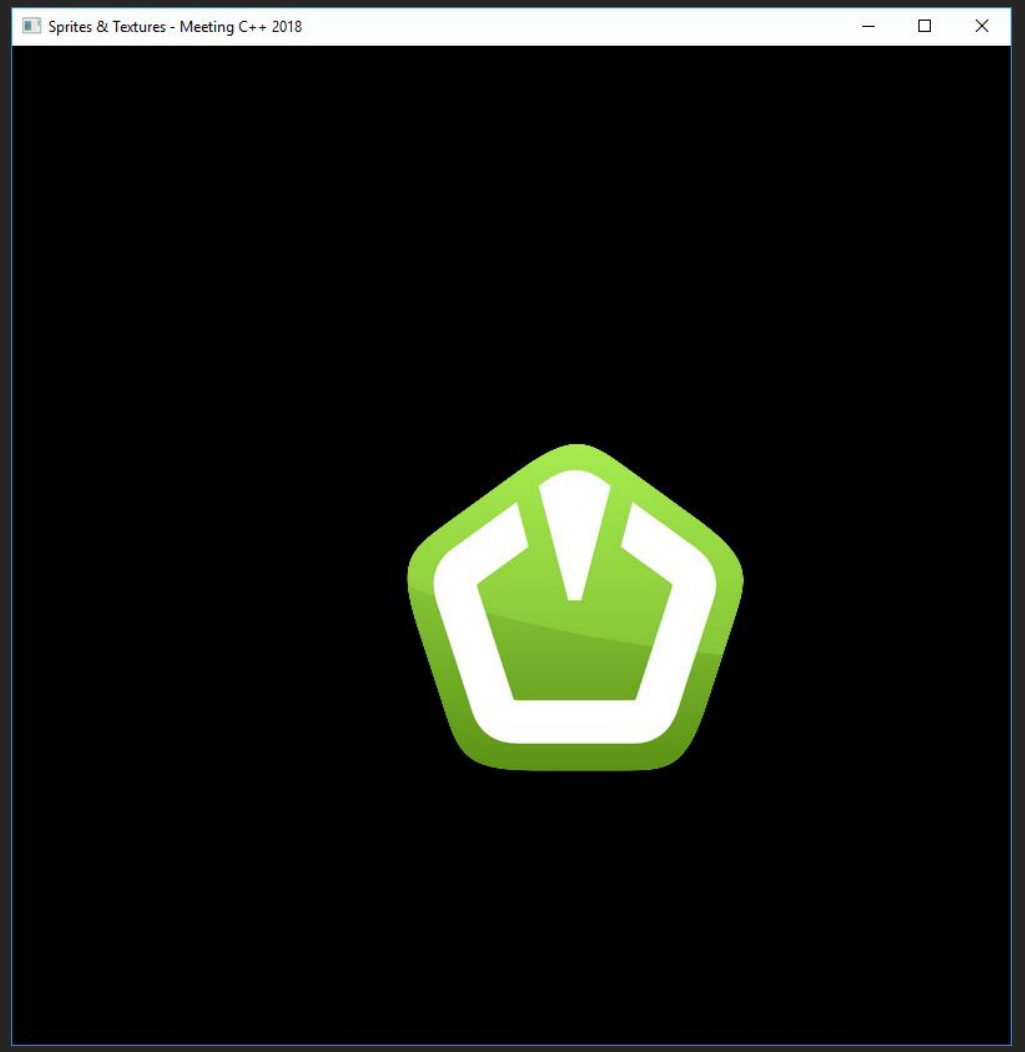
Shapes

```
sf::ConvexShape trapezoid{ 4 };
trapezoid.setPoint(0, { 100.f, 0.f });
trapezoid.setPoint(1, { 0.f, 200.f });
trapezoid.setPoint(2, { 600.f, 200.f });
trapezoid.setPoint(3, { 500.f, 0.f });
trapezoid.setFillColor(sf::Color{ 0xF2635FFF });
trapezoid.setPosition({ 100.f, 600.f });
// ...
window.clear();
window.draw(ractangle);
window.draw(circle);
window.draw(trapezoid);
window.display();
```



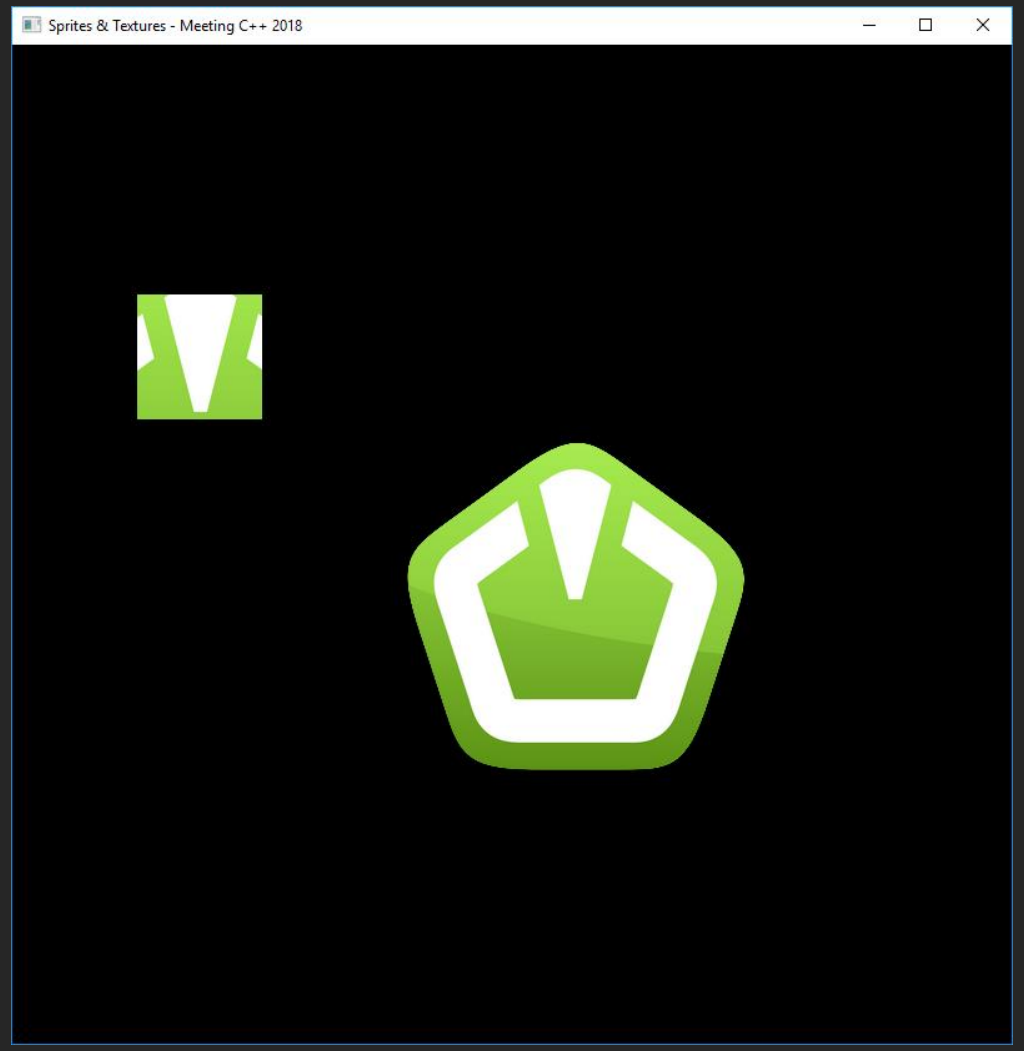
Sprites & Textures

```
sf::Texture texture;
if (!texture.loadFromFile("sfml-logo.png"))
{
    return -1;
}

sf::Sprite full_logo{ texture };
full_logo.setPosition({ 300.f, 300.f });

// ...

window.clear();
window.draw(full_logo);
window.display();
```



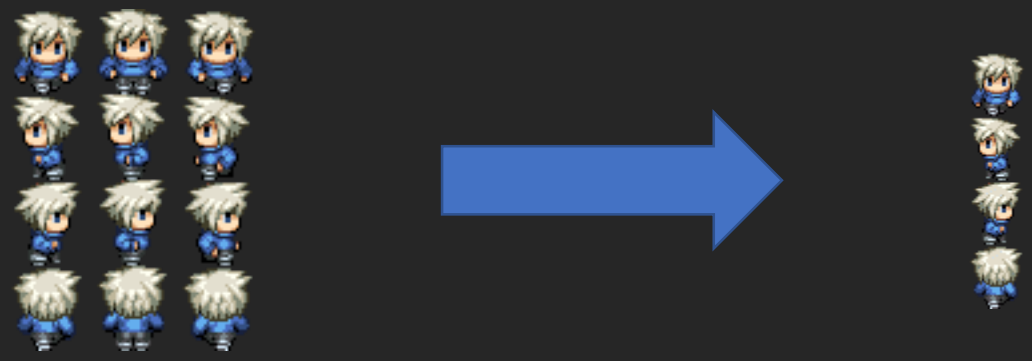
Sprites & Textures

```
sf::Texture texture;
if (!texture.loadFromFile("sfml-logo.png"))
{
    return -1;
}
// ...
sf::Sprite partial_logo{ texture };
partial_logo.setTextureRect({ 100, 50, 100, 100 });
partial_logo.setPosition({ 100.f, 200.f });
// ...
window.clear();
window.draw(full_logo);
window.draw(partial_logo);
window.display();
```



Sprites & Textures

```
sprite.setTextureRect({ left, top, width, height });
```



Texts & Fonts

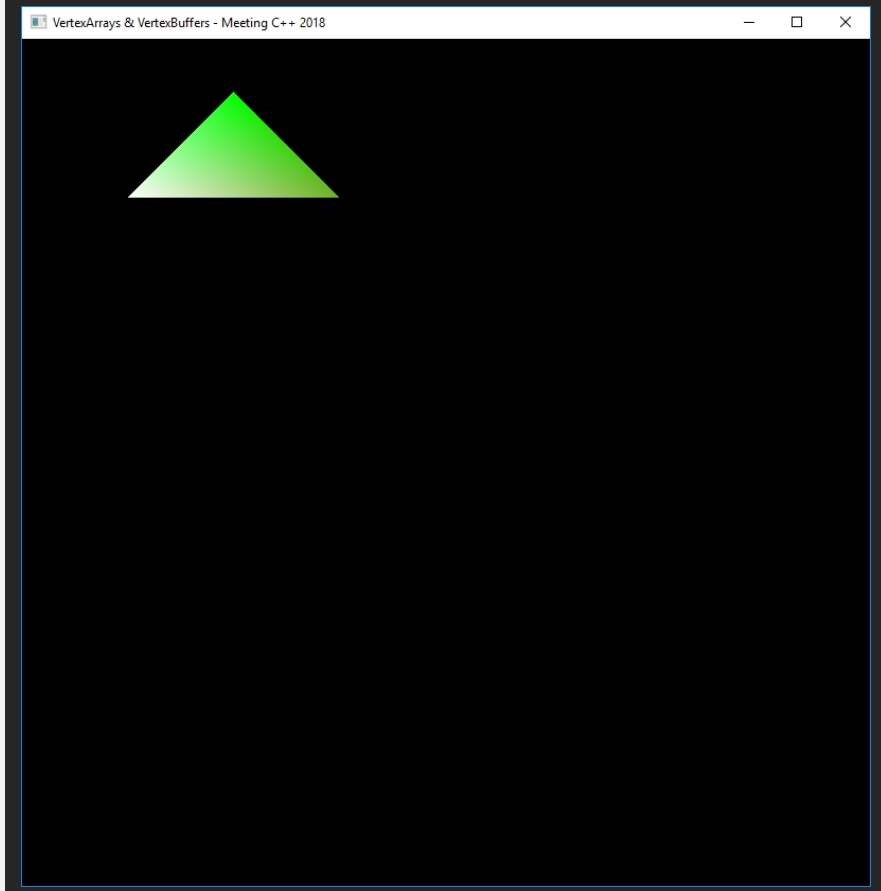
```
sf::Font font;
if (!font.loadFromFile("DejaVuSans.ttf")) {
    return -1;
}

sf::Text meeting_cpp{ "Meeting C++ 2018", font, 60 };
meeting_cpp.setPosition({ 100.f, 300.f });
meeting_cpp.setStyle(sf::Text::Bold);
meeting_cpp.rotate(20.f);

window.clear();
window.draw(meeting_cpp);
window.display();
```

Texts & Fonts

```
sf::Font font;
// ...

sf::Text sfml{ "SFML", font, 80 };
sfml.setPosition({ 300.f, 100.f });
sfml.setFillColor(sf::Color::White);
sfml.setOutlineColor(sf::Color{ 0x8ECF3CFF });
sfml.setOutlineThickness(5.f);
sfml.setLetterSpacing(1.5f);

window.clear();
window.draw(meeting_cpp);
window.draw(sfml);
window.display();
```

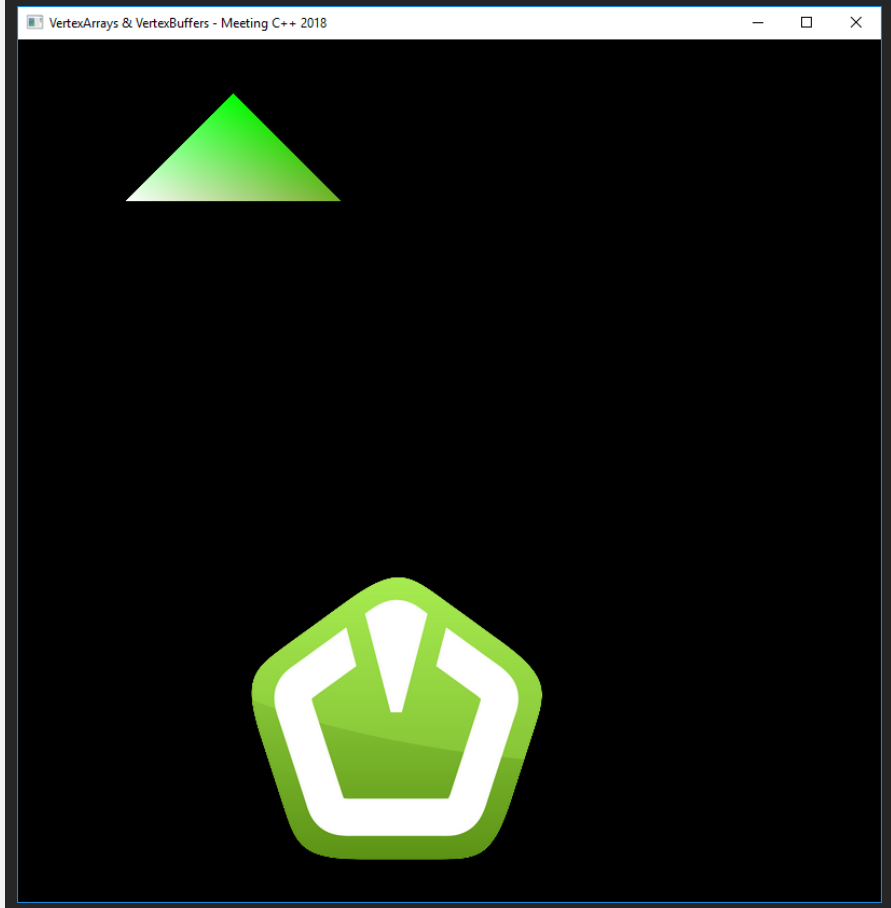
VertexArrays & VertexBuffers

```
sf::VertexArray triangle{ sf::PrimitiveType::Triangles, 3 };
triangle[0].position = { 200.f, 50.f };
triangle[0].color = sf::Color::Green;
triangle[1].position = { 100.f, 150.f };
triangle[1].color = sf::Color::White;
triangle[2].position = { 300.f, 150.f };
triangle[2].color = sf::Color{ 0x73AE27FF };

Window.clear();
window.draw(triangle);
Window.display();
```



VertexArrays & VertexBuffers

```
sf::Texture texture;
// ...
sf::VertexArray sfml_logo{ sf::PrimitiveType::Triangles, 6};
sfml_logo[0].position = { 200.f, 480.f };
sfml_logo[1].position = { 200.f, 780.f };
sfml_logo[2].position = { 500.f, 780.f };
sfml_logo[3].position = { 500.f, 780.f };
sfml_logo[4].position = { 500.f, 480.f };
sfml_logo[5].position = { 200.f, 480.f };
sfml_logo[0].texCoords = { 0.f, 0.f };
sfml_logo[1].texCoords = { 0.f, 300.f };
sfml_logo[2].texCoords = { 300.f, 300.f };
sfml_logo[3].texCoords = { 300.f, 300.f };
sfml_logo[4].texCoords = { 300.f, 0.f };
sfml_logo[5].texCoords = { 0.f, 0.f };

window.clear();
window.draw(triangle);
window.draw(sfml_logo, &texture);
window.display();
```



VertexArrays & VertexBuffers

```
std::vector<sf::Vertex> m_vertices;
sf::VertexBuffer m_vertex_buffer;
sf::Texture& m_tileset;
// ...
m_vertex_buffer.setUsage(sf::VertexBuffer::Usage::Static);
m_vertex_buffer.setPrimitiveType(sf::Quads);
m_vertex_buffer.create(tiles.size() * 4);
// ...
m_vertex_buffer.update(m_vertices.data());
// ...
target.draw(m_vertex_buffer, &m_tileset);
```


VertexArrays & VertexBuffers

```
std::vector<int> level{
    0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0,
    1, 1, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 3,
    0, 1, 0, 0, 2, 0, 3, 3, 3, 0, 1, 1, 1, 0, 0, 0,
    0, 1, 1, 0, 3, 3, 3, 0, 0, 0, 1, 1, 1, 2, 0, 0,
    0, 0, 1, 0, 3, 0, 2, 2, 0, 0, 1, 1, 1, 1, 2, 0,
    2, 0, 1, 0, 3, 0, 2, 2, 2, 0, 1, 1, 1, 1, 1, 1,
    0, 0, 1, 0, 3, 2, 2, 2, 0, 0, 0, 0, 1, 1, 1, 1,
};
```

A few more things…

- Shaders (GLSL)
- Render Texture
- Views
- OpenGL Context
- Windowing
- Keyboard Input
- Mouse Input
- Joystick/Controller Input
- Sound
- Music
- 3D Spatial Audio
- TCP & UDP Sockets
- UTF-32/16/8 Conversion
- Windows, Linux & macOS
- Android & iOS
- …

Thanks!

https://www.sfml-dev.org

https://github.com/eXpl0it3r/Talks

@DarkCisum

Stickers! Anyone?