**Transforming Texture Arrays**

https://www.youtube.com/watch?v=LbfB198YF\_c&list=PLRtjMdoYXLf776y4K432eL\_qPw4na\_py3&index=32

#include "stdafx.h"

#include "SFML/Graphics.hpp"

#include <iostream>

#include <windows.h>

int main()

{

sf::RenderWindow window (sf::VideoMode(600, 600), "SFML works!");

sf::VertexArray quad(sf::Quads, 4);

quad[0].position = sf::Vector2f(5, 5);

quad[1].position = sf::Vector2f(200, 5);

quad[2].position = sf::Vector2f(200, 200);

quad[3].position = sf::Vector2f(5, 200);

quad[0].texCoords = sf::Vector2f(0, 0);

quad[1].texCoords = sf::Vector2f(100, 0);

quad[2].texCoords = sf::Vector2f(100, 100);

quad[3].texCoords = sf::Vector2f(0, 100);

sf::Texture texture;

texture.loadFromFile("texture2.jpg");

sf::Transform transform;

transform.rotate(45, sf::Vector2f(100, 100));

sf::RenderStates states;

states.transform = transform;

states.texture = &texture;

// states object saved both rotation and texture

while (window.isOpen()) {

sf::Event event;

while (window.pollEvent(event)) {

switch (event.type) {

case sf::Event::Closed:

window.close();

break;

}

}

window.clear();

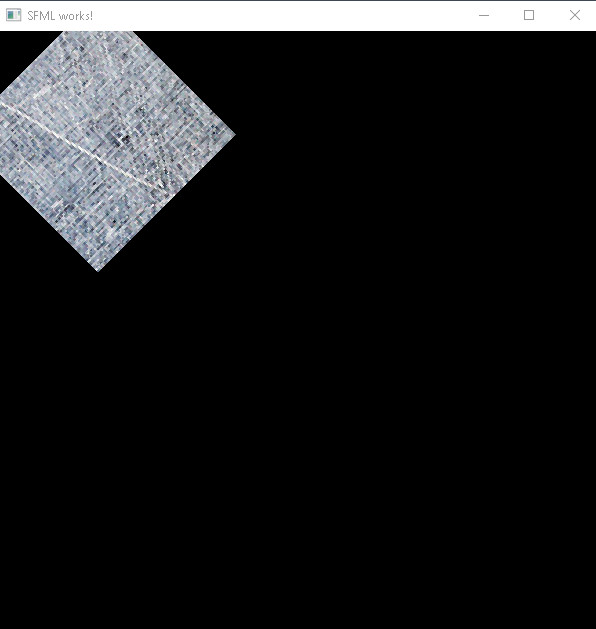
window.draw(quad, states);

window.display();

}

}

**Result**



**Important notes:**

* *RenderStates* object saves both texture and rotation. Basically, it saves the state of something, and applies that state to any sprites or vertex arrays in *window.draw;*