

```
////////////////////////////////////
//
// Copyright (C) 2014 Maximilian Wagenbach (aka. Foaly) (foaly.f@web.de)
//
// This software is provided 'as-is', without any express or implied warranty.
// In no event will the authors be held liable for any damages arising from the use of this software.
//
// Permission is granted to anyone to use this software for any purpose,
// including commercial applications, and to alter it and redistribute it freely,
// subject to the following restrictions:
//
// 1. The origin of this software must not be misrepresented;
// you must not claim that you wrote the original software.
// If you use this software in a product, an acknowledgment
// in the product documentation would be appreciated but is not required.
//
// 2. Altered source versions must be plainly marked as such,
// and must not be misrepresented as being the original software.
//
// 3. This notice may not be removed or altered from any source distribution.
//
////////////////////////////////////

#ifndef ANIMATEDSPRITE_INCLUDE
#define ANIMATEDSPRITE_INCLUDE

#include <SFML/Graphics/RenderTarget.hpp>
#include <SFML/System/Time.hpp>
#include <SFML/Graphics/Drawable.hpp>
#include <SFML/Graphics/Transformable.hpp>
#include <SFML/System/Vector2.hpp>

#include "Animation.hpp"

class AnimatedSprite : public sf::Drawable, public sf::Transformable
{
public:
    explicit AnimatedSprite(sf::Time frameTime = sf::seconds(0.2f), bool paused = false, bool looped = true);

    void update(sf::Time deltaTime);
    void setAnimation(const Animation& animation);
    void setFrameTime(sf::Time time);
    void play();
    void play(const Animation& animation);
    void pause();
    void stop();
    void setLooped(bool looped);
    void setColor(const sf::Color& color);
    const Animation* getAnimation() const;
    sf::FloatRect getLocalBounds() const;
    sf::FloatRect getGlobalBounds() const;
    bool isLooped() const;
    bool isPlaying() const;
    sf::Time getFrameTime() const;
    void setFrame(std::size_t newFrame, bool resetTime = true);

    //new home made shit
    const int getCurrentFrame() const;

private:
    const Animation* m_animation;
    sf::Time m_frameTime;
    sf::Time m_currentTime;
    std::size_t m_currentFrame;
    bool m_isPaused;
    bool m_isLooped;
    const sf::Texture* m_texture;
    sf::Vertex m_vertices[4];

    virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const;
};
```

```
#endif // ANIMATEDSPRITE_INCLUDE
```