

```
////////////////////////////////////
//
// 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.
//
////////////////////////////////////

//Some changes to class
//perhaps not necessary

#ifndef ANIMATION_INCLUDE
#define ANIMATION_INCLUDE

#include <vector>
#include <SFML/Graphics/Rect.hpp>
#include <SFML/Graphics/Texture.hpp>

class Animation
{
public:
    Animation();
    ~Animation();

    void setActiveAnimation(std::string ani);
    void addFrame(std::vector<std::pair< sf::IntRect, sf::Time>> rect, const std::string animation);
    void setSpriteSheet(const sf::Texture& texture);
    const sf::Texture* getSpriteSheet() const;
    std::size_t getSize() const;
    const std::pair< sf::IntRect, sf::Time>& getFrame(std::size_t n) const;

    const std::vector<std::string>* GetNames();
    const std::string getActiveAnimation();

private:
    std::map<std::string, std::vector<std::pair< sf::IntRect, sf::Time>>> m_animations;
    std::vector<std::pair< sf::IntRect, sf::Time>> m_frames;
    const sf::Texture* m_texture;

    int m_active;
    std::vector<std::string> m_aniName;
};

#endif // ANIMATION_INCLUDE
```