

Lab 5 - Animated Sprites

1. Using the code provide below create an AnimatedSprite class, this should extend the sprite class from last week.

AnimatedSprite.h

```
#pragma once
#include "Sprite.h"

class AnimatedSprite : public Sprite
{
public:
    AnimatedSprite(const sf::Vector2f &size = sf::Vector2f(0, 0));
    ~AnimatedSprite();

    void update(float dt);

    void setFrameSize(int width, int height);
    int getFrameWidth() { return frame.width; };
    int getFrameHeight() { return frame.height; };
    void setAnimationSpeed(float aspeed);

protected:
    // variable for controlling animation
    float elapsedTime;
    float animationSpeed;
    sf::IntRect frame;
};
```

AnimatedSprite.cpp

```
#include "AnimatedSprite.h"

AnimatedSprite::AnimatedSprite(const sf::Vector2f &size) : Sprite(size)
{
    // Configure default values
    elapsedTime = 0.f;
    animationSpeed = 1.0f;
    frame = sf::IntRect(0, 0, 0, 0);
}

AnimatedSprite::~AnimatedSprite()
{}

void AnimatedSprite::update(float dt)
{
    // increment time
    elapsedTime += dt;

    // if enough time has passed move onto next frame
    if (elapsedTime >= animationSpeed)
    {
        frame.left += frame.width;
        // check if we have passed last frame of animation
        if (frame.left > getTexture()->getSize().x - frame.width)
        {
            frame.left = 0;
        }
        setTextureRect(frame);
        // reset timer
        elapsedTime = 0;
    }
}

void AnimatedSprite::setAnimationSpeed(float speed)
{
    animationSpeed = speed;
}

void AnimatedSprite::setFrameSize(int width, int height)
{
    frame.width = width;
    frame.height = height;
    setTextureRect(frame);
}
```

- a. Test the Animated sprite class by building the walking zombie example from the lecture. The zombie should animate while standing still (for now). The zombie frame size is 55x108 pixels and a recommend animation speed is 0.1f.
2. Create a zombie race; multiple animated sprites with different movement speeds and animated speeds moving across the window (from left to right). Have 3 or 4 zombie sprites move across the screen at the same time.
3. Implement animation flipping so the zombie sprite can walk both directions. This requires changing the IntRect values so the frame is built backwards. The zombie should move under user control; the zombie should not animate when still and animate correctly when moving left/right based on user control (left and right arrow keys).

4. Create your own sprite sheet and create an animated sprite using it. Start with a single animation (like zombie in the lecture). This doesn't have to be some great piece of art work, a dancing stick man made in paint will suffice.
5. Create a new class for handling multiple animations from a single sprite sheet. Implement animation swapping, so when the character stands still an idle animation plays and when moving plays a walking animation. You can use your own sprite sheet for this, or create one from online resources.