

May 31, 16 7:54

Animatedgraphicelement.cpp

Page 1/3

```
#include "Animatedgraphicelement.h"
#include "Model.h"
#include <iostream>

/*
 * -----
 * "THE BEER-WARE LICENSE" (Revision 42):
 * Antoine BOULANGÃM-^I & Pierre_Elliot CABRERA wrote this file. As long as you
 * retain this notice you
 * can do whatever you want with this stuff. If we meet some day, and you think
 * this stuff is worth it, you can buy me a beer in return
 * -----
 */

//=====
// Constructeurs
//=====
AnimatedGraphicElement::AnimatedGraphicElement(const std::vector<sf::IntRect> &c
lipRects, sf::Texture &image, int x, int y, int w, int h, int nbSpriteHorizontal
, int nbSpriteVertical, int counter)
: GraphicElement(image, x, y, w, h, nbSpriteHorizontal, nbSpriteVertical), _
clip_rects{clipRects}
{
    _current_clip_rect = 0;
    _current_step = 0;
    _nb_steps = counter;
    _left = false;
    _limit_destruction = nbSpriteHorizontal * _nb_steps;
    _counter_destruction = 0;
}

AnimatedGraphicElement::AnimatedGraphicElement(AnimatedGraphicElement *other, in
t nbSpriteHorizontal, int nbSpriteVertical)
: GraphicElement(*(other->getTexture()), other->getX(), other->getY(), other
->getWidth(), other->getHeight(), nbSpriteHorizontal, nbSpriteVertical)
, _clip_rects(other->getClipRects())
, _nb_steps(other->getNbSteps())
{
    _current_clip_rect = 0;
    _current_step = 0;
    _left = false;
    _limit_destruction = nbSpriteHorizontal * _nb_steps;
    _counter_destruction = 0;
}

//=====
// SÃ@lectionne le bon rectangle de lecture
// puis affiche l'AnimatedGraphicElement
//=====
void AnimatedGraphicElement::draw(sf::RenderWindow *window){
    sf::IntRect rect = _clip_rects[_current_clip_rect];
    this->setTextureRect(rect);
    if (_current_step == _nb_steps)
    {
        _current_clip_rect++;
        _current_step = 0;
    }
    if (_clip_rects.size() == _current_clip_rect)
        _current_clip_rect = 0;
    _current_step += 1;
    _counter_destruction += 1;
    window->draw(*this);
}

//=====
// SÃ@lectionne le bon rectangle de lecture
// puis affiche le ninja
//=====
```

Tuesday May 31, 2016

May 31, 16 7:54

Animatedgraphicelement.cpp

Page 2/3

```
void AnimatedGraphicElement::drawNinja(sf::RenderWindow *window, int mvtNinjaX,
NINJA_STATUS statusNinja, bool invincibility){
    sf::IntRect rect = _clip_rects[_current_clip_rect];
    switch(statusNinja){
        case WALKING :
            if (mvtNinjaX < 0){
                rect.top = 200;
                _left = true;
            }
            else if (mvtNinjaX > 0){
                rect.top = 0;
                _left = false;
            }
            else if (mvtNinjaX == 0){
                if (!_left)
                    rect.top = 400;
                else
                    rect.top = 600;
            }
            break;
        case JUMPING :
            if (!_left)
                rect.top = 800;
            else
                rect.top = 1000;
            break;
        default :
            if (mvtNinjaX < 0){
                rect.top = 1400;
                _left = true;
            }
            else if (mvtNinjaX > 0){
                rect.top = 1200;
                _left = false;
            }
            else if (mvtNinjaX == 0){
                if (_left)
                    rect.top = 1400;
                else
                    rect.top = 1200;
            }
            break;
    }
    if (_current_step == _nb_steps)
    {
        _current_clip_rect++;
        _current_step = 0;
    }
    if (_clip_rects.size() == _current_clip_rect)
        _current_clip_rect = 0;
    _current_step += 1;
    this->setTextureRect(rect);
    if(invincibility)
        this->setColor(sf::Color(rand()%256, rand()%256, rand()%256, 255));
    else
        this->setColor(sf::Color(255, 255, 255, 255));
    window->draw(*this);
}

//=====
// Accesseurs
//=====
std::vector<sf::IntRect> AnimatedGraphicElement::getClipRects() const{ return _c
lip_rects; }

int AnimatedGraphicElement::getNbSteps() const{ return _nb_steps; }

int AnimatedGraphicElement::getCurrentStep() const{ return _current_step; }
```

1/2

Animatedgraphicelement.cpp

May 31, 16 7:54

Animatedgraphicelement.cpp

Page 3/3

```
int AnimatedGraphicElement::getCounterDestruction() const{ return _counter_destr  
uction; }  
  
int AnimatedGraphicElement::getLimitDestruction() const{ return _limit_destructi  
on; }
```