

May 31, 16 7:55

ClickableElement.cpp

Page 1/2

```
#include "ClickableElement.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
 * -----
 */

//=====
// Constructeur
//=====
ClickableElement::ClickableElement(const std::vector<sf::IntRect> &clipRects, sf
::Texture &image, int x, int y, int w, int h, int nbSpriteHorizontal, int nbSpri
teVertical, BUTTON id, sf::String buttonName, sf::Font *font, float taille, floa
t position)
    : GraphicElement(image, x, y, w, h, nbSpriteHorizontal, nbSpriteVertical)
    , _clip_rects{clipRects}
    , _id {id}
    , _current_clip_rect{0}
    , _checked{false}
{
    _buttonName.setFont(*font);
    _buttonName.setPosition(x + w/5, y + position);
    _buttonName.setColor(sf::Color(255, 255, 255));
    _buttonName.setString(buttonName);
    _buttonName.setCharacterSize(taille);
}

//=====
// Dessine le bouton selon le
// rectangle de lecture
//=====
void ClickableElement::draw(sf::RenderWindow *window){
    sf::IntRect rect = _clip_rects[_current_clip_rect];
    this->setTextureRect(rect);
    window->draw(*this);
}

//=====
// DÃ©tecte si le point de coordonnÃ©es x, y
// se trouve sur le clickableElement.
// Si c'est le cas, retourne son id, sinon,
// retourne -1.
//=====
BUTTON ClickableElement::over(int x, int y){
    sf::IntRect button(this->getX(), this->getY(), this->getW(), this->getH());
    if(button.contains(x, y))
        return this->_id;
    else
        return NONE;
}

//=====
// Change le rectangle de lecture Ã
// afficher.
//=====
void ClickableElement::ChangeSprite(BUTTON id){
    if(id != NONE)
        _current_clip_rect = 1;
    else
        _current_clip_rect = 0;
}
```

May 31, 16 7:55

ClickableElement.cpp

Page 2/2

```
//=====
// Permet de changer l'activation du son
//=====
void ClickableElement::InverseChecked() {
    _checked = !_checked;
}

//=====
// Accesseurs
//=====
sf::Text ClickableElement::getButtonName() const{
    return _buttonName;
}

BUTTON ClickableElement::getId() const { return _id; }

bool ClickableElement::getChecked() const { return _checked; }
```