Lab 4 – Sprite class

1. Firstly, we need to create our sprite class. Create two new files Sprite.h and Sprite.cpp and use the following code to create the Sprite class.

   Sprite.h

```
#pragma once
#include "SFML\Graphics.hpp"

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

        virtual void update(float dt)=0;
        void setVelocity(sf::Vector2f vel);
        void setVelocity(float vx, float vy);
        sf::Vector2f getVelocity();

protected:
        sf::Vector2f velocity;


};
```

   Sprite.cpp

```
#include "Sprite.h"

Sprite::~Sprite()
{}

Sprite::Sprite(const sf::Vector2f & size) : RectangleShape(size)
{}

void Sprite::setVelocity(sf::Vector2f vel)
{
        velocity = vel;
}
void Sprite::setVelocity(float vx, float vy)
{
        velocity.x = vx;
        velocity.y = vy;
}

sf::Vector2f Sprite::getVelocity()
{
        return velocity;
}
```

2. Download the “Lab 4 gfx.zip” from Blackboard and extract the gfx folder into the project directory. The gfx folder contains a few test images for testing with Sprite work. Create an Enemy class by inheriting from Sprite. Load in the goomba.png provided and use the enemy sprite class to render it.

   a. Add logic to the enemy sprite class (in the update function) so that the enemy sprite bounces around the window (similar to last week). This logic should be contained within the enemy class.

3. Create a Player class by inheriting from Sprite. This class will represent the player and will need an update function. This player class will also need access to the Input class, so add a function to the player class to pass in a reference to the input class. Have the player class us "Mushroom.png" as the sprite image. Add logic to the Player class (in the update function) that moves the player class sprite around the window (up, down, left and right) similar to last week.
4. Create a Cursor class, this class will be used to replace the cursor. Use the icon.png provided and have the sprite always be at the mouse's location.
   a. To hide the cursor you can use the following line of code

   ```
   window->setMouseCursorVisible(false);
   ```

5. Using the sprite sources provided in the lecture, or google, find or create your own image to be used as a sprite along with your own sprite class.
   a. http://www.spriters-resource.com/
   b. http://opengameart.org/art-search-advanced?keys=&field_art_type_tid%5B%5D=9
6. Make a background class. This should load in any image and be used as a background for your window.
7. Create an NPC sprite class. Use your own image for this sprite. The NPC class should constantly move towards the player class, but at half the speed of the player.