Keyboard, Mouse

The tutorial explains how to acess global input devices: keyboard, mouse. This must not be confused with events. Real-time input allows you to query the global state of keyboard, mouse and joysticks at any time **is this button currently pressed?", "where is the mouse currently?"** while events notify you when something happens **this button was pressed", "the mouse has moved**.

Keyboard

The class that provide acess to the keyboard state is sf::Keyboard. It contain only one function isKeyPressed which checks the current state of a key (pressed or released). It is a static function so you don’t need to instantiate sf::Keyboard to use it.

This function directly reads the keyboard state, ignoring the focus state of your window. This means that `isKeyPressed` may return true even if your window is inactive.

```
if (sf::Keyboard::isKeyPressed(sf::Keyboard::Left))
{
    // left key is pressed: move our character
    character.move(1, 0);
}
```

Key codes are defined in the `sf::Keyboard::Key` enum.

Mouse

The class that provides access to the mouse state is `sf::Mouse`. Like its friend `sf::Keyboard`, `sf::Mouse` only contains static functions and is not meant to be instantiated (SFML only handles a single mouse for the time being).

```
if (sf::Mouse::isButtonPressed(sf::Mouse::Left))
{
    // left mouse button is pressed: shoot
    gun.fire();
}
```

Mouse button codes are defined in the `sf::Mouse::Button` enum. SFML supports up to 5 buttons: left, right, middle (wheel), and two additional buttons whatever they may be.

You can also get and set the current position of the mouse, either relative to the desktop or to a window :

```
// get the global mouse position (relative to the desktop)
sf::Vector2i globalPosition = sf::Mouse::getPosition();

// get the local mouse position (relative to a window)
sf::Vector2i localPosition = sf::Mouse::getPosition(window); // window is a sf::Window
```

No function for mouse wheel

There is no function for reading the current state of the mouse wheel. Since the wheel can only be moved relatively, it has no absolute state that can be queried. By looking at a key you can tell whether it's pressed or released. By looking at the mouse cursor you can tell where it is located on the screen. However, looking at the mouse wheel doesn't tell you which "tick" it is on. You can only be notified when it moves (`MouseWheelScrolled`event).