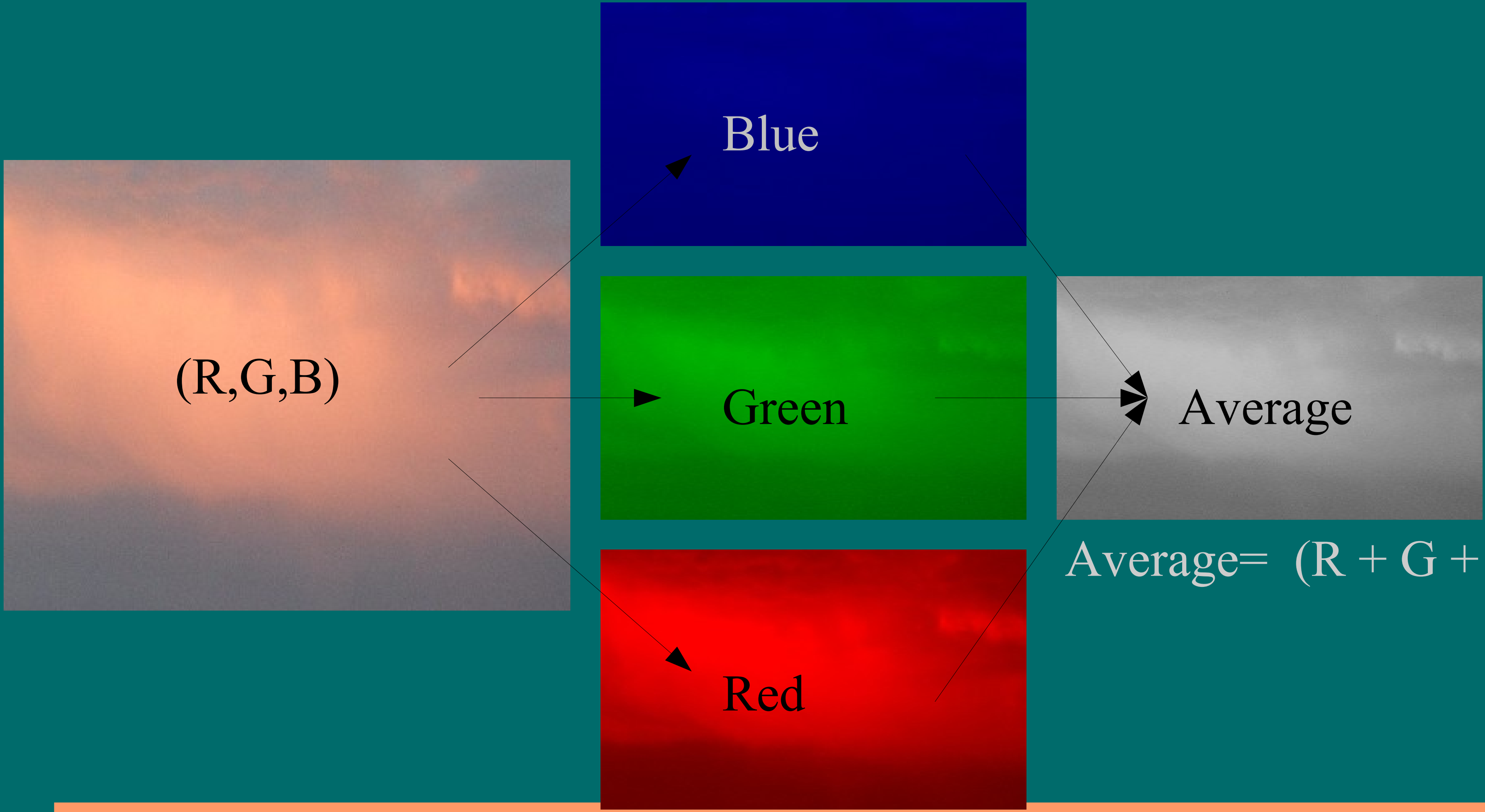
Image Mixer – Channels Introduction

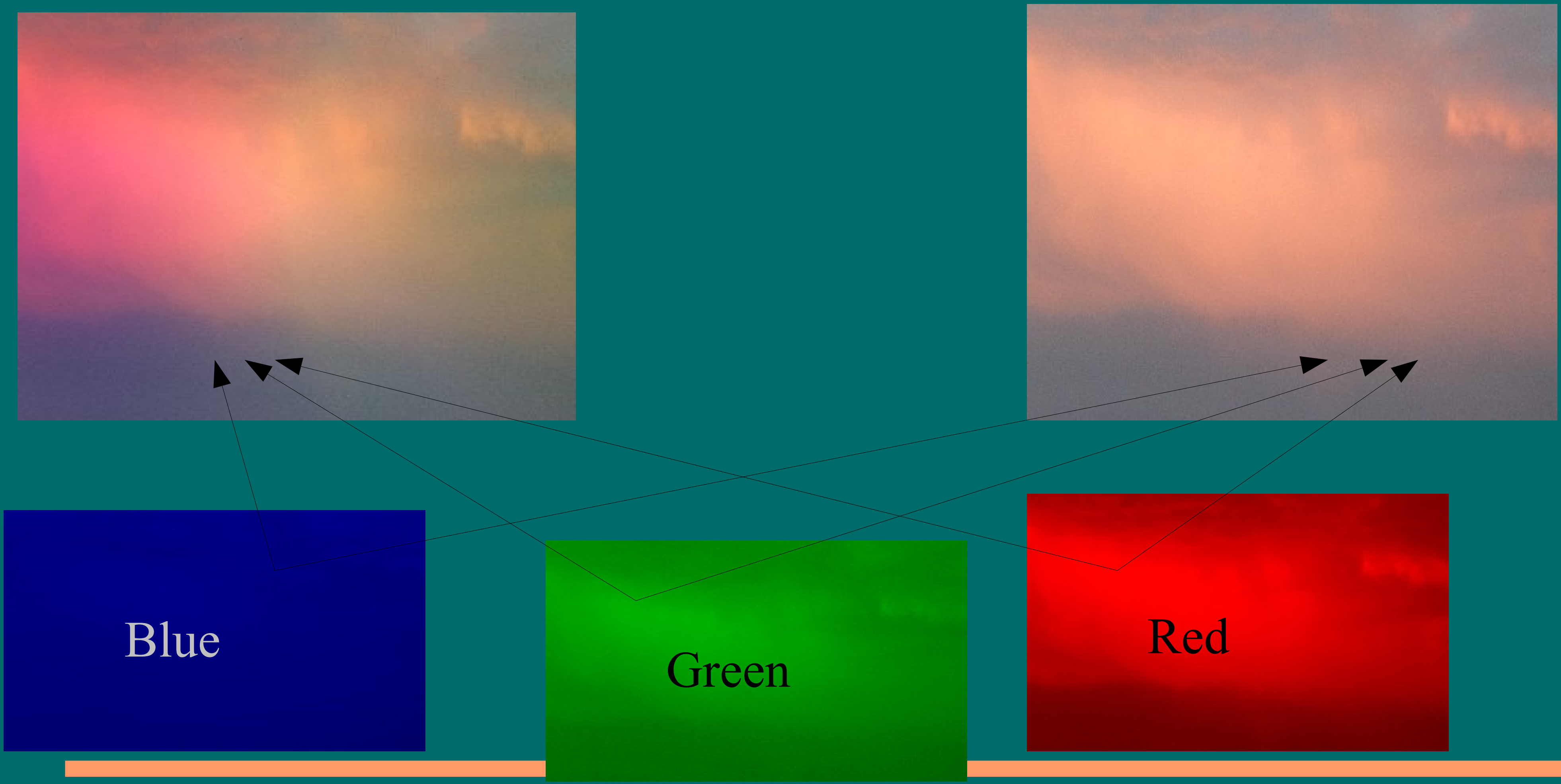
What channels are? Channels are just Info about your image. A Image is just made up of 3 colors (wich is what human eyes + brain can see)



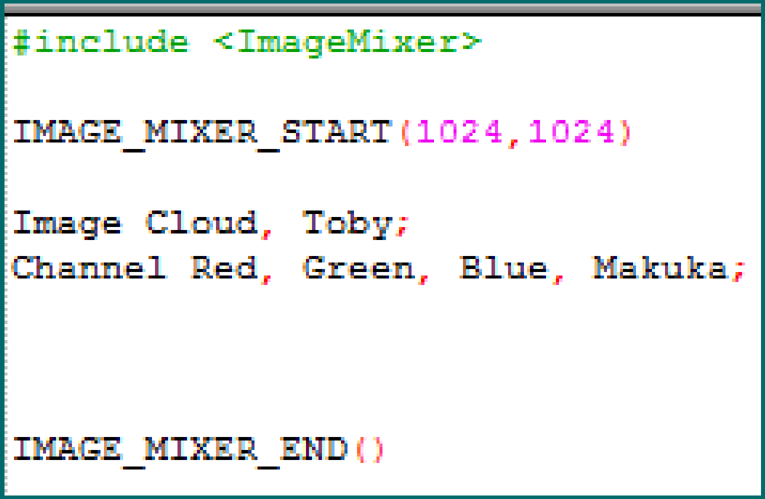
Average= (R + G + B) / 3

ImageMixer – Channel Introduction

- In Image Mixer You can use all the 3 channels as separated entities. That's mean that you can just mixup the Red of one image with the Blue of another image!



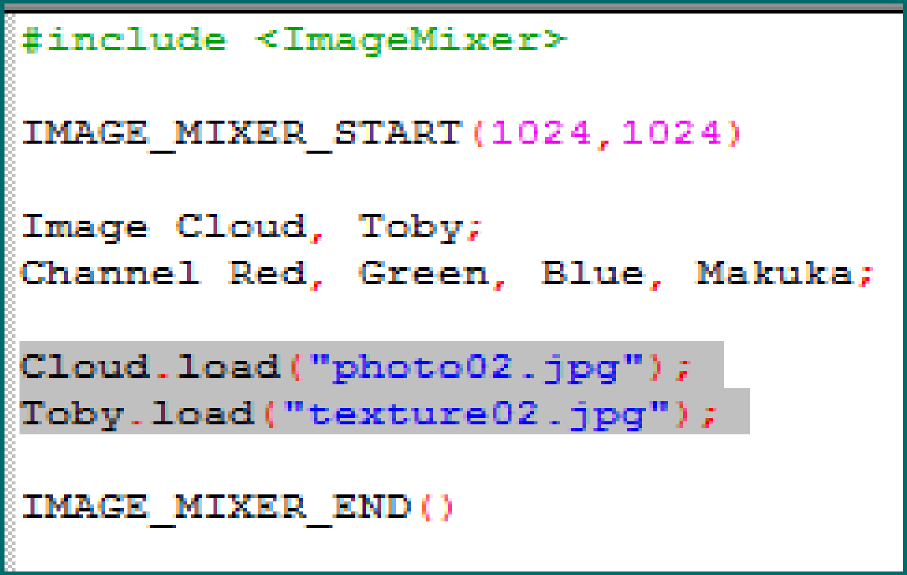
ImageMixer – Let's look again to code.. :)



```
#include <ImageMixer>

IMAGE_MIXER_START(1024,1024)

Image Cloud, Toby;
Channel Red, Green, Blue, Makuka;



IMAGE_MIXER_END()
```

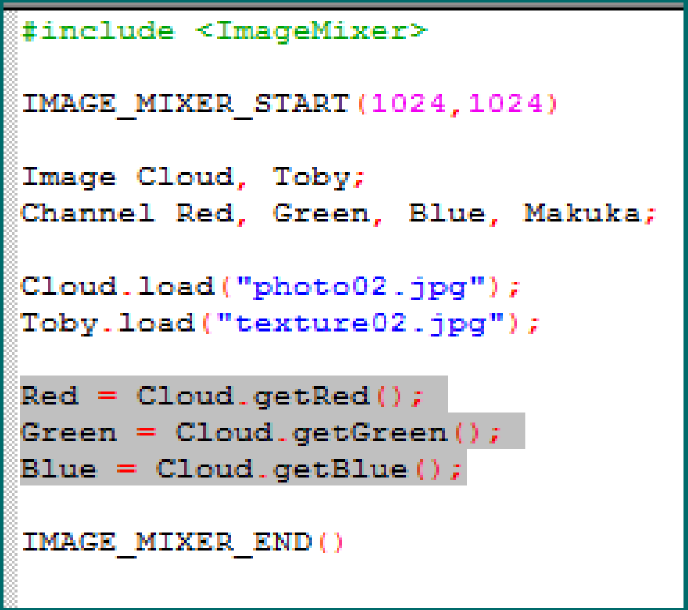
- <--- Look at te code example. Now you have new items in our code.. the Channels! Of course I called 3 of them “Red, Green and Blue”.. but you can call them as you wish.. for example “Makuka”. You can still have also images along with channels (Cloud and Toby are images!).

- As done in the previouse chapter you must start by loading some nice images. Without some pre-existing images, is hard doing something interesting.



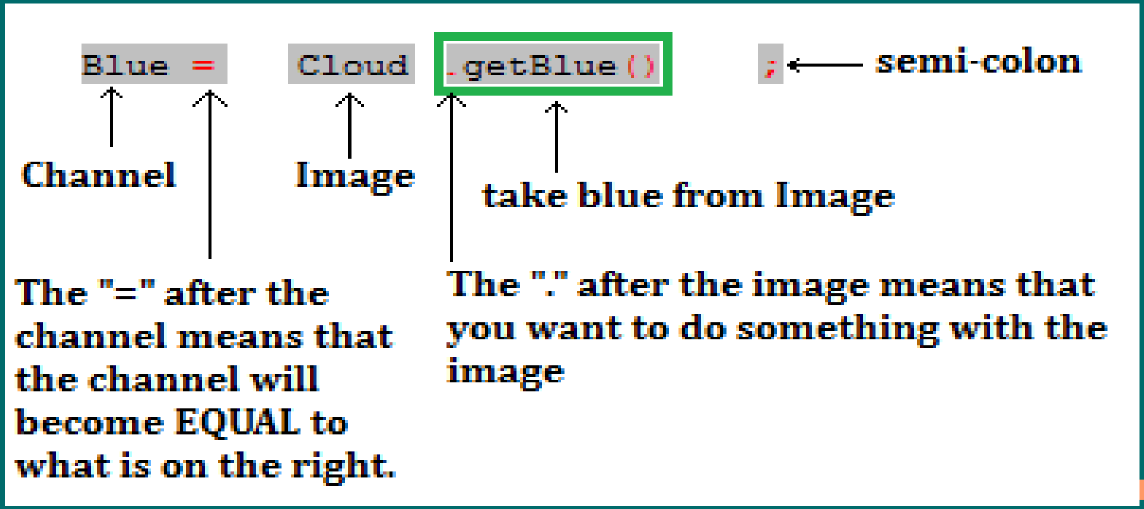
```
#include <ImageMixer>

IMAGE_MIXER_START(1024,1024)

Image Cloud, Toby;
Channel Red, Green, Blue, Makuka;

Cloud.load("photo02.jpg");
Toby.load("texture02.jpg");

IMAGE_MIXER_END()
```

Image Mixer – now we start doing nice things



```
#include <ImageMixer>

IMAGE_MIXER_START(1024,1024)

Image Cloud, Toby;
Channel Red, Green, Blue, Makuka;

Cloud.load("photo02.jpg");
Toby.load("texture02.jpg");

Red = Cloud.getRed();
Green = Cloud.getGreen();
Blue = Cloud.getBlue();

IMAGE_MIXER_END()
```

- Now you can start doing nice things. How can you use channels? That's very simple. First of all you have to extract channels from the images. So you have to tell to the image to take the blue. “getBlue()” will do this.
- After you have the blue you must store it somewhere. So you can just use the “Blue” Channel.



```
Blue = Cloud .getBlue() ;
```

semi-colon

Channel

Image

take blue from Image

The "=" after the channel means that the channel will become EQUAL to what is on the right.

The "." after the image means that you want to do something with the image

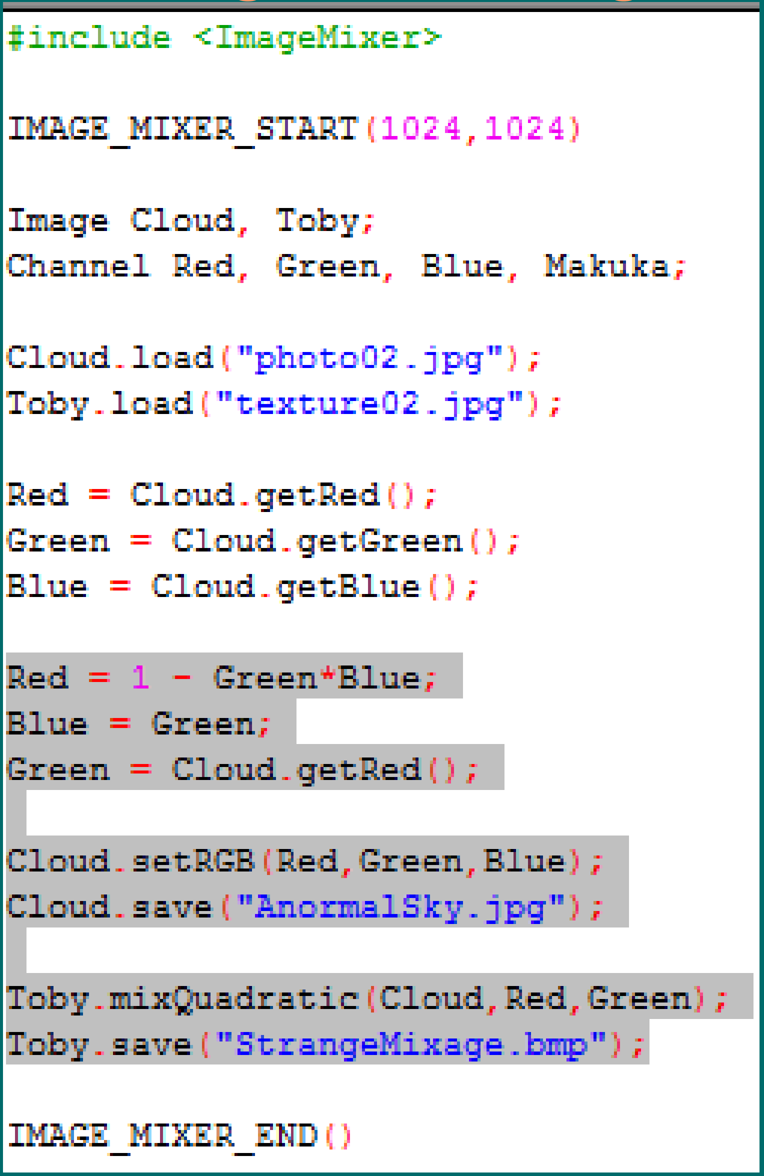
The channel named “Blue” will become equals to the blue of the image “Cloud”.

“Cloud.getBlue()” is a channel!

You can't write infact

Blue = Cloud ; ← ERROR!

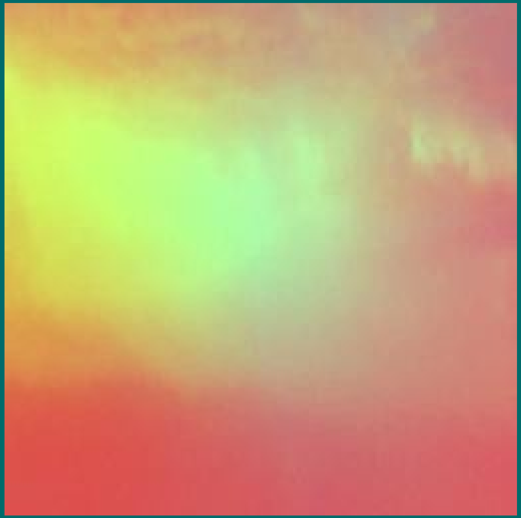
ImageMixer – Algebraic and mixages;



```
#include <ImageMixer>

IMAGE_MIXER_START(1024,1024)

Image Cloud, Toby;
Channel Red, Green, Blue, Makuka;

Cloud.load("photo02.jpg");
Toby.load("texture02.jpg");

Red = Cloud.getRed();
Green = Cloud.getGreen();
Blue = Cloud.getBlue();

Red = 1 - Green*Blue;
Blue = Green;
Green = Cloud.getRed();

Cloud.setRGB(Red,Green,Blue);
Cloud.save("AnormalSky.jpg");

Toby.mixQuadratic(Cloud,Red,Green);
Toby.save("StrangeMixage.bmp");

IMAGE_MIXER_END()
```

- Now let's try something harder. As you can see you can do Algebraic with Channels. So you can do “1 - Green*Blue”
- More attention needs to be payed here: *“Green = Cloud.getRed();”* why we do that? Well. We cannot just simply write “Green = Red” because “Red” is already equal to *“1- Green*Blue”*. So if you want the old Red you have to take it again from “Cloud”. If you like to take “1-Green*Blue” instead of “Red” you can do that. But in this case I wanted just the old Red.
- Before saving “Cloud” we change all its channel. You have just to call “setRGB (chan1 , chan2 , chan3)” where chan1 will be saved as Red, chan2 will be saved as Green and chan3 will be saved as Blue.
- Before saving “ Toby” I'll mix it. This time instead of using “mix(image,chan)” i will use something more advanced “mixQuadratic(image, image, chan)”. As you can see “image” cane be replaced by “chan”. So is not possible doing
- Red = Cloud .. but **is possible doing Cloud = Red!**

Image Mixer – finally some nice image!



- AnormalSky.jpg

- StrangeMixage.bmp