Advanced Physics
Assignment 2
Shader programming

Using minimally one of these browsers:
- Firefox 17
- Chrome 23
- Internet Explorer 10
    - Install http://iewebgl.com/
- Safari 6
    - Turn on WebGL: Preferences > Advanced > Show Develop menu
    - Develop menu > Enable WebGL.

Visit the ShaderToy website: http://www.shadertoy.com/
Create an account if you want to easily save your efforts

List of variables and functions for this assignment
`float`, `vec2`, `vec3`, `vec4` - floating point data types
`aVector.x`, `aVector.y`, `aVector.z`, `aVector.w` - components
`aVector.xy`, `aVector.zw`, etc. - take lower dimensional vector from vector
`texture2D(channel, vec2 uv)` - samples input texture at coords u,v (range 0-1)
`gl_FragColor` - output (vec4 rgba)
`gl_FragCoord.xy` - output uv in pixels (texture coordinates)
`iResolution.xy` - viewport resolution
`vec2(float, float)` - 2-dimensional vector constructor
`sin(float)` - sine function
`iGlobalTime` - running time in seconds

A tutorial on GLSL: http://www.lighthouse3d.com/tutorials/glsl-tutorial/

- Create a new shader (this is a GLSL fragment shader)
- Set `iChannel0` to webcam input (use the 2nd video if no webcam)

For each of the following numbered exercises, explain what you did to come to the solution, and paste the resulting code and a screenshot into your report:

1. Make the shader display your webcam picture[1] by calculating fragment uv coordinates (from pixel range to 0-1 range) and setting `gl_FragColor` to sample the channel texture
2. Make the shader output only the red component of the video
3. Mirror the image horizontally, so that you can use your webcam like a mirror (as most webcam programs do)
4. Invert the colors

[1] If you don't have a working webcam on your system: use the music video recording with the green background, supplied with ShaderToy

5. Now invert only the green channel
6. Revert color changes; scale the picture by 0.5 horizontally, center the scaled picture, and let the area outside the picture be black
7. Create a horizontal blur effect by sampling not only the current pixel from the texel, but the neighboring pixels as well, weighing 0.4 for the pixel itself, 0.2 for the pixels next to it, and 0.1 for the pixels at a distance of 2.
8. Expand the blur filter to also weigh vertically adjacent pixels, like so:

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

9. Distort the image using a sine wave, then use iGlobalTime to make the sine wave move over time
10. Combine multiple horizontal and vertical sine waves with different wavelengths and speeds in order to achieve something that looks like water ripples
11. Analyse the way 3D vertices are projected in the shader at: https://www.shadertoy.com/view/XdlGzn

Deadline: 20-2-2014 23:59
PDF, via VLO Dropbox -> Stephan van der Feest