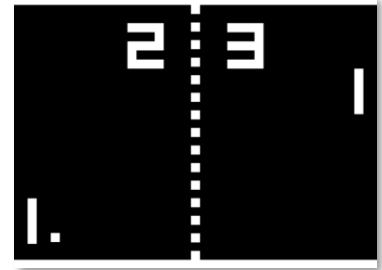


2018/2019, 4th quarter

INFOGR: Graphics

Practical 4: Retake

Author: Jacco Bikker



The assignment:

The short story: write a Pong-game in a post-processing shader.

In P3 you may already have worked on a post processing shader, e.g. for vignetting or depth of field. Although we normally use a post processing shader to slightly modify an existing image, we can also take it further. It is even possible to write a full game, (almost) completely in the shader. Details will follow, but first: formalities.

The following rules for submission apply:

- Your code must compile and run 'out-of-the-box' (exception: we will restore packages if necessary). You can reduce the risk that your code fails during assessment by testing it on someone else's machine. To be very safe, ask a TA to test it.
- Make sure you **clean** your solution before submitting (i.e. remove all the compiled files and intermediate output). This can easily be achieved by running clean.bat (included with the template). This will also kill the OpenTK package. Contrary to popular belief, this is OK as it significantly reduces the zipped size.
- For the retake assignment you must work alone, even if your P3 partner also needs to do the practical retake.

Grading:

If you produce a working game-in-a-shader, you get a 6. Implement additional features to obtain additional points (up to a 10). From the base grade of 6, we deduct points for a solution that was not cleaned, a solution that does not compile, or a solution that crashes (1 point for each problem).

Deliverables:

A ZIP-file containing the contents of your (cleaned) solution directory, and the read-me (in the .txt file format). The contents of the solution directory should contain your solution files (.sln, .vcxproj), all your source code and all your asset files (including shaders, models and textures).

Send your submission by e-mail to bikker.j@gmail.com.

Deadline: Friday, July 12, 2019, 11:59h (NOON!)

This is a hard deadline, and there is no late or very late deadline. Any materials handed in after the deadline will not be graded.

High-level Outline

For this assignment you will build a small game that handles all its rendering in a single shader that is applied to a full-screen quad (so, just like the post processing shaders). 'All its rendering' means: visualizing the pads and the ball, as well as any other elements you may require, such as score digits and playfield boundaries. It also means that there are no OpenGL commands that produce output, except for the commands that draw the full-screen quad. And finally, for clarity, it means that your 'post processing shader' is executed on a black image (or: overwrites whatever is there).

Some parts of your game still need to run on the CPU side. The GPU has no access to your mouse or keyboard, so that needs to happen in C# code. The GPU also cannot update uniforms from shader code, so keeping track of the ball position is best handled on the CPU.

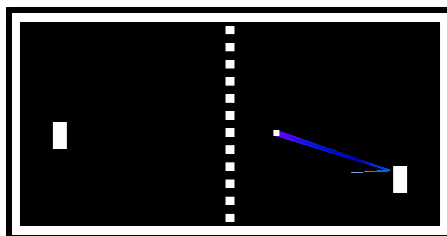
Template

Please use the P3 template for the retake assignment.

A Bit Extra

Meeting the minimum requirements earns you a 6 (assuming practical details are all in order). An additional four points can be earned by implementing optional features. An incomplete list of options, with an indication of the difficulty level:

- [EASY] Give the ball a trail. (1.0 pt)
- [EASY] Draw score digits. (1.0 pt)
- [EASY] Add a VHS tape simulation in a post process (1.0 pt)
- [EASY] Add a random bonus on the field that spawns a second ball. (1.0 pt)
- [.....] Add other creative things to the game.



And Finally...

Don't forget to have fun; make something beautiful!

May the Light be with you,

- Jacco.