Assignment 01: 2D coordinates

The purpose of Assignment 01 is twofold: first let you set up the development environment, and second let you experience with *normalized screen coordinates*.

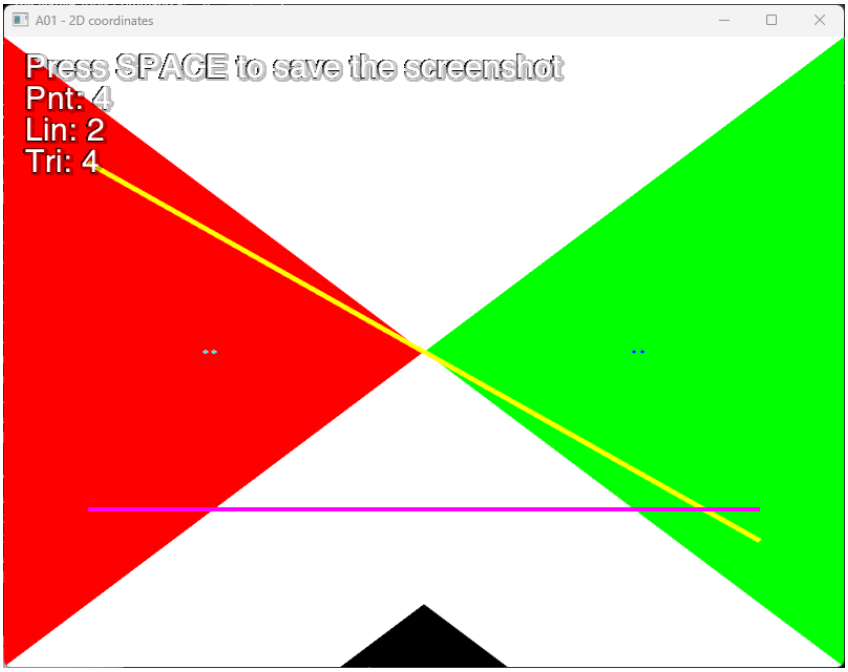
Part 1: Set up the Development Environment

Following the tutorial on:

https://vulkan-tutorial.com/Development_environment

Install your own Vulkan development environment. Use it to create your first project, and use it to compile the simple application contained in file **`A01.cpp`** . Setting up the environment so that the compiler finds all the required library and header files is not easy, and strongly dependent on your system configuration and on your development choices. I personally use a very simple command-line based environment, which is extremely easy to set-up in Linux, not too difficult on MacOS, and not so simple in Windows. I have included the simple scripts file I use to compile my solutions in the three main O.S. in file **`CommandLineCompile.zip`** , available in the same folder on WebEx where you found this archive. I personally do not encourage you using that solution, unless you are really comfortable with that like myself, and I suggest you to look for an IDE based environment. Nevertheless these small files might give you some hints on which options you should look for when setting up your environment, so please have a look at them if you experience some difficulties.

If you manage to compile and run your application, you should a window appear, showing the following picture:



When you are able to finally compile and run the code, just press ESC to quit, and go on with the next step!

Part 2: 2D coordinates

Using the three commands below, draw a simple picture of your choice. Only modify file **`figures.hpp`** , which is included by the main file **`A01.cpp`** that you have compiled in Part 1.

Function:

```
A->Plot(float x1, float y1, float r, float g, float b)
```

Draws a single point at (x1,y1), colored with the (r,g,b) specified color. The point is addressed in Normalized screen coordinates, (i.e. in range [-1,+1]), following the Vulkan convention. The red, green and blue (r,g,b) values are in the range [0,1].

Function:

```
A->Line(float x1, float y1, float x2, float y2, float r, float g, float b)
```

Draws a segment from point (x1,y1) to point (x2, y2), colored with the (r,g,b) specified color. Starting and ending points are expressed in Normalized screen coordinates, (i.e. in range [-1,+1]), following the Vulkan convention. The red, green and blue (r,g,b) values are in the range [0,1].

Function:

```
A->Triangle(float x1, float y1, float x2, float y2, float x3, float y3,
            float r,  float g,  float b)
```

Draws a triangle connecting points (x1,y1) (x2, y2) and (x3, y3), colored with the (r,g,b) specified color. Vertex are expressed in Normalized screen coordinates, (i.e. in range [-1,+1]), following the Vulkan convention. The red, green and blue (r,g,b) values are in the range [0,1].

First remove the ten functions already in file **`figures.hpp`** (given only as an example), and replace them with your own drawing code. Please note that there is an upper limit in the number of primitives that can be used (89998). This number is however huge, and it should not pose any limit, not even for the most complex drawings. Once you are satisfied with your picture, press SPACE: this will save a screenshot of your window in file **`A01.png`** . Please check that its content matches your picture, as such file will be an important part of the final delivery of this assignment. Also remember that all the assignments will be delivered in a single upload, which will be made available near the end of the course, and the deadline is about a year from now. You are however encouraged to complete this assignment as soon as possible, as its purpose is to let you experiment on a specific topic, and help you to proceed in the program.

WARNING!

Since it is a C program, you can use for loops and functions if you think they can be helpful in your solution. However, please include all your code in this file, since it will be put in an automatic correction process for the final evaluation. Please also be cautious when using standard libraries and symbols, since they might not be available in all the development environments (especially, they might not be available in the final evaluation environment, preventing your code from compiling). This WARNING will be valid for ***ALL THE ASSIGNMENTs***, but it will not be repeated in the following texts, so please remember these advices carefully!