heig-vd

Mini OpenGL en C++

heig-vd

Objectifs

- Ecrire un Renderer en quelques centaines de lignes de C++
- Purement software, mais en n’oubliant pas qu’en pratique il tournerait sur la carte graphique
- Pas à pas …
- Idéalement jusqu’à arriver à l’image ci-contre

heig-vd

Format d'affichage

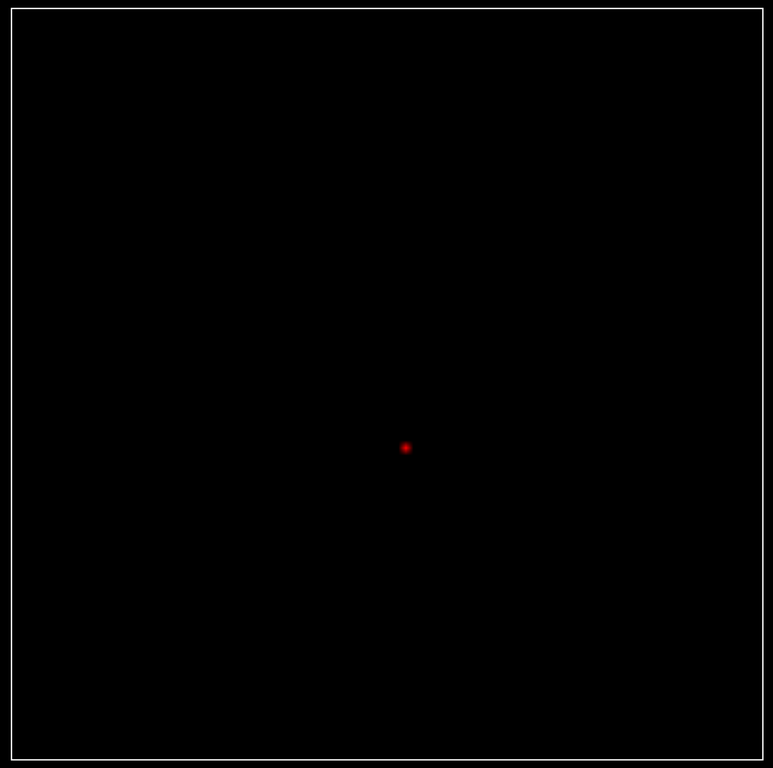
- Images au format TGA
  - Simple
  - Pixels au formats BW, RGB ou RGBA
- Je vous fourni une classe TGAImage minimaliste.
  - Constructeur définit la taille
  - La méthode set permet d'écrire dans un pixel

```
class TGAImage {
protected:
    unsigned char* data;
    int width;
    int height;
    int bytespp;

    bool   load_rle_data(std::ifstream &in);
    bool unload_rle_data(std::ofstream &out);
public:
    enum Format {
        GRAYSCALE=1, RGB=3, RGBA=4
    };

    TGAImage();
    TGAImage(int w, int h, int bpp);
    TGAImage(const TGAImage &img);
    bool read_tga_file(const char *filename);
    bool write_tga_file(const char *filename, bool rle=true);
    bool flip_horizontally();
    bool flip_vertically();
    bool scale(int w, int h);
    TGAColor get(int x, int y);
    bool set(int x, int y, TGAColor c);
    ~TGAImage();
    TGAImage & operator =(const TGAImage &img);
    int get_width();
    int get_height();
    int get_bytespp();
    unsigned char *buffer();
    void clear();
};
```

heig-vd

Exemple

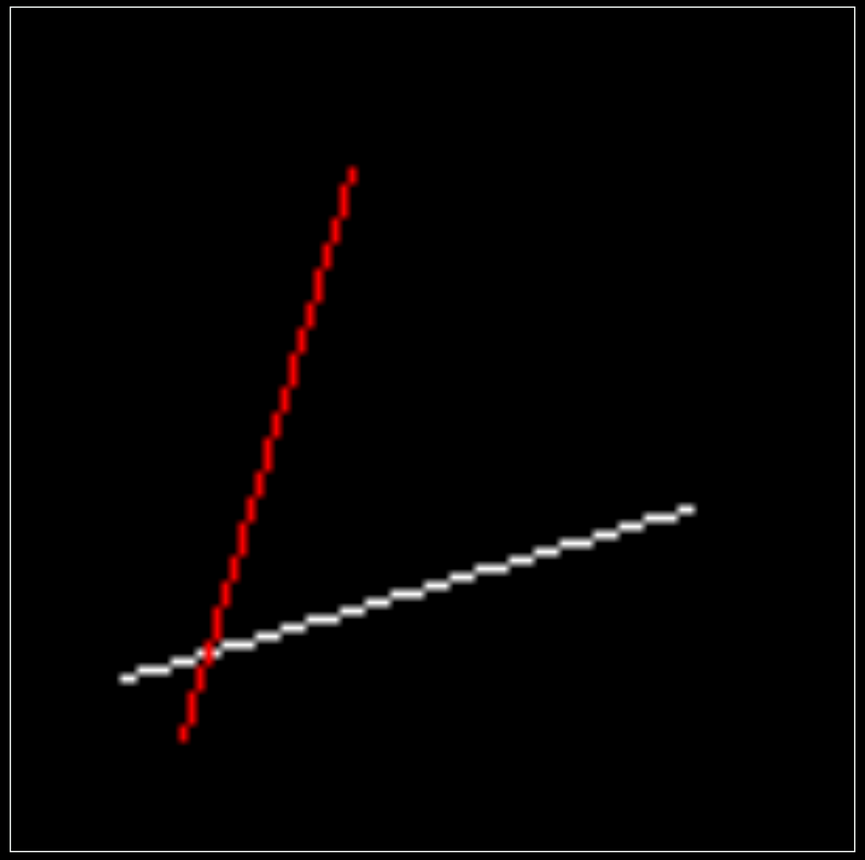
- Crée une image 100x100 noire
- Colorie le pixel 52,41 en rouge
- Sous le résultat dans output.tga



```cpp
#include "tgaimage.h"

const TGAColor white = TGAColor(255, 255, 255, 255);
const TGAColor red   = TGAColor(255, 0,   0,   255);

int main(int argc, char** argv) {

    TGAImage image(100, 100, TGAImage::RGB);
    image.set(52, 41, red);
    image.flip_vertically();
    image.write_tga_file("output.tga");
    return 0;

}
```

A vous de jouer…

- A vous d'écrire la fonction line pour que ce programme produise l'image ci-dessous…



```
#include "tgaimage.h"

const TGAColor white = TGAColor(255, 255, 255, 255);
const TGAColor red   = TGAColor(255, 0,   0,   255);

void line(int x0, int y0, int x1, int y1,
          TGAImage &image, TGAColor color);

int main(int argc, char** argv) {
    TGAImage image(100, 100, TGAImage::RGB);
    line(13, 20, 80, 40, image, white);
    line(20, 13, 40, 80, image, red);
    image.flip_vertically();
    image.write_tga_file("output.tga");
    return 0;
}
```