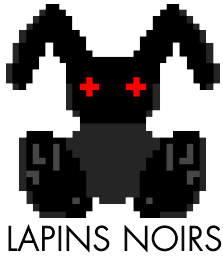




Skill Test 7 : Tracé de lignes : Mystify Your Mind



Skill Test 7

Tracé de lignes

Sujet : Mystify Your Mind

Le Laboratoire aux Lapins Noirs
lapinsnoirs@epitech.eu

Ce document contient un sujet d'examen associé au module d'infographie

Nom du dépôt de rendu : mystify_promotion (Exemple mystify_2042)



Skill Test 7 : Tracé de lignes : Mystify Your Mind

Index

1 – Consignes

2 – Sujet



1 – Consignes

L'examen doit être réalisé avec la LibLapin comme unique outil. Les seules fonctions autorisées sont celles précisées dans la section sujet.

La compilation sera effectuée avec les flags -Wall -Wextra.

Si votre rendu comporte un fichier binaire, un fichier .o ou un fichier tampon type « `###` » ou « `*~` », vous obtiendrez la note de 1,5.

Si votre fonction ne s'appelle pas de la bonne façon, le programme de correction ne pourra pas la trouver et vous obtiendrez la note de 1,5.

Si votre programme est trop lent (>2 secondes), boucle à l'infinie, reçoit un signal SIGSEV, SIGFPE ou SIGPIPE, vous obtiendrez la note de 1,5.

La réussite de la compilation de votre programme avec la moulinette vous apporte la note de 2. Cette note évolue en fonction des résultats obtenus aux exercices.

Votre rendu ne doit pas comporter de main. Nous compilerons l'intégralité des fichiers .c rendus avec la moulinette. Les fichiers .h seront pris en compte si situé à la racine de votre rendu.



2 – Sujet

Preliminaire :

1 points

```
void                tekpixel(t_bunny_pixelarray    *pix,  
                           t_bunny_position        *pos,  
                           unsigned int           color) ;
```

Écrivez la fonction suivante, qui dessine un pixel de la couleur color à la position pos dans pix.



Skill Test 7 : Tracé de lignes : Mystify Your Mind

Preliminaire :

2 points

```
void                                tekline(t_bunny_pixelarray    *pix,  
                                             t_bunny_position        *pos,  
                                             unsigned int            *color) ;
```

Écrivez la fonction suivante, qui dessine une ligne depuis la position `pos[0]` jusqu'à la position `pos[1]`, de couleur `color[0]` à `color[1]`.

Le dégradé de couleur sur la ligne ne sera pas testé. Il est néanmoins nécessaire à l'exercice 4.



Skill Test 7 : Tracé de lignes : Mystify Your Mind

Écrivez la fonction suivante :

5 points

```
void broken_lines(t_bunny_pixelarray *pix,
                  t_bunny_broken_point *pts,
                  size_t len) ;

typedef struct s_bunny_broken_point
{
    double x;
    double y;
    double x_speed;
    double y_speed;
} t_bunny_broken_point;
```

La fonction `broken_lines` commence par passer tout l'écran en noir.

La fonction `broken_lines` affiche dans `pix` une ligne brisée dont les `len` points sont donnés dans `pts`. `broken_lines` affiche des lignes blanches. Le dernier point est également relié au premier par une ligne.

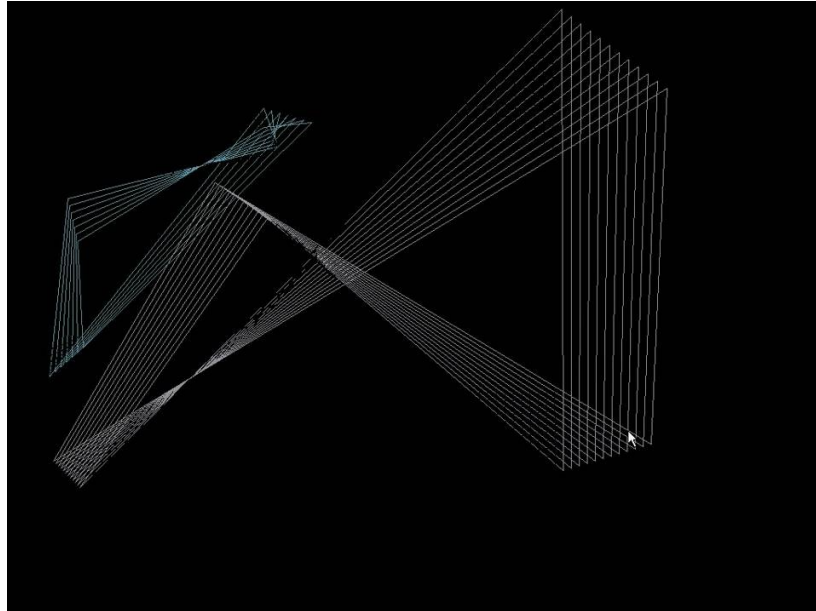
La fonction `broken_lines` va **ensuite** ajouter `x_speed` à `x` et `y_speed` à `y`. Si `x` ou `y` sortent de l'écran si ils se voient ajouté la vitesse, inversez la vitesse à la place et ne touchez pas à leur position.



Skill Test 7 : Tracé de lignes : Mystify Your Mind

Écrivez la fonction suivante :

10 points



L'authentique « Mystify Your Mind » dont nous allons faire un petit clone plus joli, avec une seule ligne et une seule forme.

```
void                                mystify_your_mind(t_bunny_pixelarray    *pix,
                                                    t_bunny_mystic_point    *pts,
                                                    size_t                len) ;

typedef struct                      s_bunny_mystic_point
{
    double    x;
    double    y;
    double    x_speed;
    double    y_speed;
    t_color    color;
    t_color    color_speed;
}
t_bunny_mystic_point;
```

La fonction `mystify_your_mind` a un fonctionnement similaire à la fonction `broken_lines` à l'exception des éléments suivants :

- L'écran n'est pas passé en noir mais en noir transparent avec 32 en alpha.
- Les lignes présentent des dégradés de couleur, la couleur est dans le champ `color`.
- A chaque appel à la fonction après avoir dessiné, chaque composante de `color_speed` (attribut mod) est ajoutée à `color`. Si une composante est susceptible de sortir des valeurs représentable par un unsigned char, au lieu d'ajouter la composante, inversez la vitesse du composant concerné.