



Skill Test 3B : Lancer de rayons : Flashlight



Skill Test 3B

Lancer de rayons

Sujet : Flashlight

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 : flashlight_promotion (Exemple flashlight_2042)



Skill Test 3B : Lancer de rayons : Flashlight

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.



Skill Test 3B : Lancer de rayons : Flashlight

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 3B : Lancer de rayons : Flashlight

Écrivez la fonction suivante :

2 points

```
typedef struct          s_bunny_3d_space
{
    t_color             *buffer ;           /* size is width * height * depth */
    int                 width ;             /* width in number of cube */
    int                 height ;            /* height in number of cube */
    int                 depth ;             /* depth in number of cube */
    int                 cube_size ;         /* the size of one cube */
    t_bunny_3d_space ;
}

t_color                *tekgetvertex(t_bunny_3d_space    *buf,
                                     int                  x,
                                     int                  y,
                                     int                  z) ;
```

La fonction `tekgetvertex` retourne un pointeur vers la couleur situé dans l'espace 3D `buf`, à la position `(x, y, z)`.

N'oubliez pas de prendre en compte la taille des cubes !

Par exemple, si chaque cube fait 10 de côté, alors un point situé à 3, 3, 3 est sur le cube en 0, 0, 0 !

La fonction retourne NULL si `(x, y, z)` est en dehors de l'espace.

...n'oubliez pas de déclarer la structure dans votre fichier...

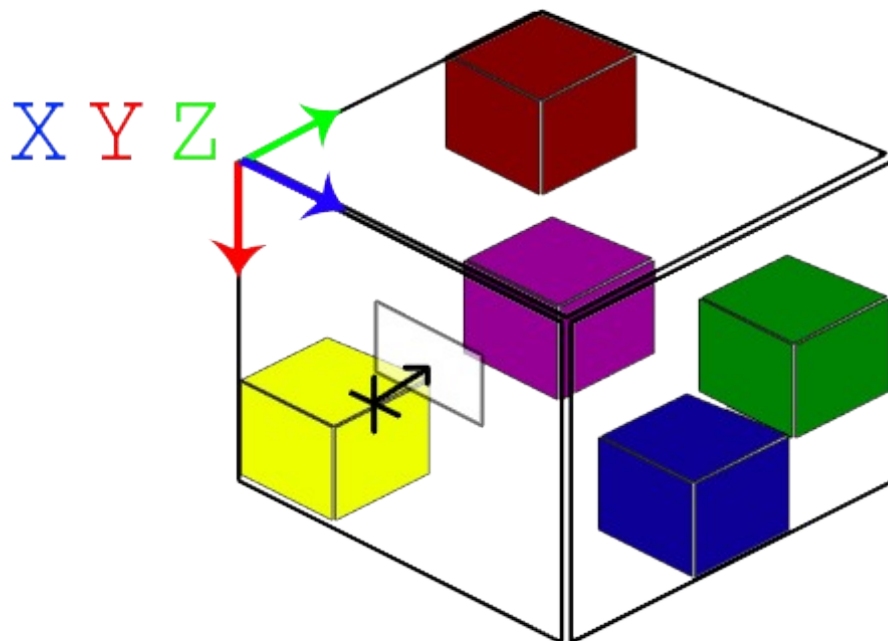


Skill Test 3B : Lancer de rayons : Flashlight

Écrivez la fonction suivante :

15 points

```
void flashlight(t_bunny_pixelarray  
               int  
               const t_bunny_position  
               const t_bunny_3d_space  
               *pix,  
               eye_screen_dist,  
               *pos,  
               *buf) ;
```



La fonction flashlight dessine dans pix la vision d'un écran de même taille que pix. Le point de vision est en pos->x en X, en pos->y en Y et en 0 en Z.

La distance entre l'écran et l'oeil est de eye_screen_dist.

Vous devez lancer des rayons dans l'espace buf.

Chaque cube dont la transparence n'est pas 255 est un cube qui ne doit pas être affiché. Les autres cubes doivent figurer à l'écran et être de la bonne couleur.

Attention, chacune de leur composante de couleur doit être diminuée de 1 par unité de distance entre le point d'impact et l'œil, hors transparence.

Si vous sortez de l'espace virtuel, la couleur que vous devez afficher à l'écran est noire.

Vous avez le droit à la fonction sqrt.