



Particules

- Le Laboratoire aux Lapins Noirs -
pedagogie@ecole-89.com

Dans ce TP, vous allez créer une fontaine de particule qui tourne sur elle-même.

Ce document est strictement personnel et ne doit en aucun cas être diffusé.



01 – Rotation

Vous allez écrire la fonction suivante :

```
t_bunny_zposition    e89_rotation(t_bunny_zposition    target,  
                                t_bunny_zposition    rotation) ;
```

La structure `t_bunny_zposition` contient trois float, `x`, `y` et `z`.

La fonction `rotate` effectue une rotation d'axe `rotation.x`, `rotation.y`, `rotation.z` des coordonnées situés dans `target` avant de les renvoyer.

Ci-dessous, les matrices de rotation.

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}, \quad R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}, \quad R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Chaque ligne de la matrice représente une somme devant être assignée à) une dimension `X`, `Y` ou `Z` de l'élément à modifier.

Les fonctions `cos` et `sin` sont autorisées, bien entendu.



02 – Fontaine

Vous allez écrire la fonction suivante :

```
typedef struct s_bunny_particle
{
    t_bunny_zposition position ;
    t_bunny_zposition speed ;
    unsigned int color ;
} t_bunny_particle ;

void e89_particles(t_bunny_pixelarray *pix,
                  t_bunny_particule *pos,
                  size_t len) ;
```

Cette fonction dessine l'ensemble des points contenu par pos. Il y a len point dans pos.

L'origine de l'axe X est situé à la moitié de la largeur de pix au lieu d'être en haut à gauche. Cela veut dire que si un vertex est à 0 en X, il faut la dessiner au milieu de pix en X.

L'axe Y est inversé et commence en bas de l'écran. Cela veut dire qu'une particule située à 0 en Y est située sur la dernière ligne de pix, soit sa hauteur - 1.

L'axe Z est la profondeur. Lors du dessin à l'écran, la coordonnée Z est ignorée.

La couleur de la particule est l'attribut color de la structure.



Ecrivez maintenant :

```
void e89_physics(t_bunny_particule  
                size_t  
                *pos,  
                len) ;
```

La fonction physics ajoute speed.x, speed.y et speed.z aux attributs position.x, position.y et position.z. **Ensuite**, si la position Y est strictement inférieure à 0, la particule doit subir le traitement suivant :

- Remise à zéro de ses coordonnées.
- Récupération d'un nombre aléatoire via rand() (Sans modification!)
- Assignment à la vitesse en X de cos de ce nombre aléatoire.
- Assignment à la vitesse en Z de sin de ce nombre aléatoire.
- Assignment à la vitesse en Y de rand() % 5 + 1.

Ou, si la vitesse en Y est inférieure à zéro, alors, augmentez-la d'un coefficient de 1,05.

Ou sinon, multipliez la vitesse en Y par un coefficient de 0,95. Si ensuite, elle est strictement inférieure à 0,5, alors multipliez-là par -1.