

Construction d'un cube 3D avec PureBasic

par Comtois

Date de publication : 11/03/08

Dernière mise à jour :

Ce tutoriel est une introduction à la 3D avec PureBasic.

I - Introduction

II - Mesh

II-1 - Position

II-2 - Normale

II-3 - Couleur

II-4 - Coordonnées uv

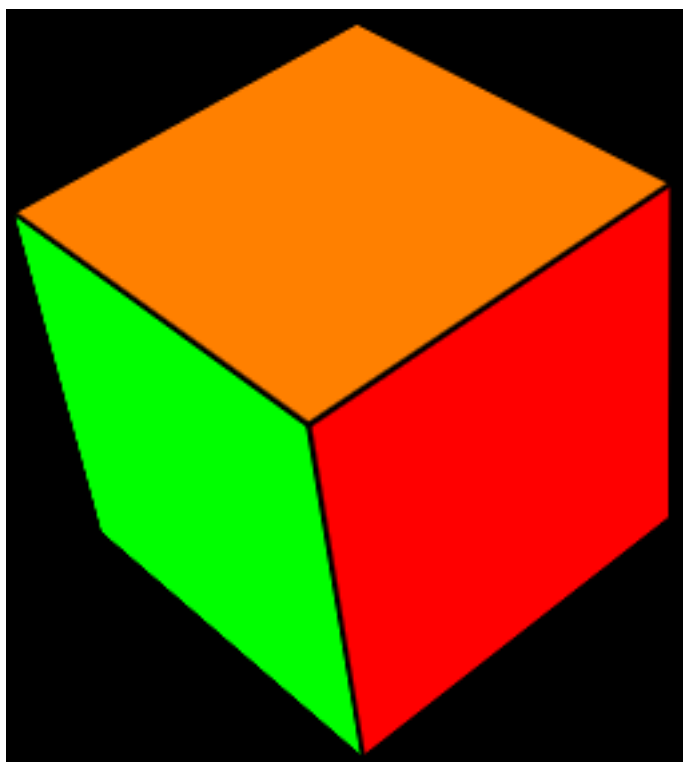
III - Sources

IV - Remerciements

I - Introduction

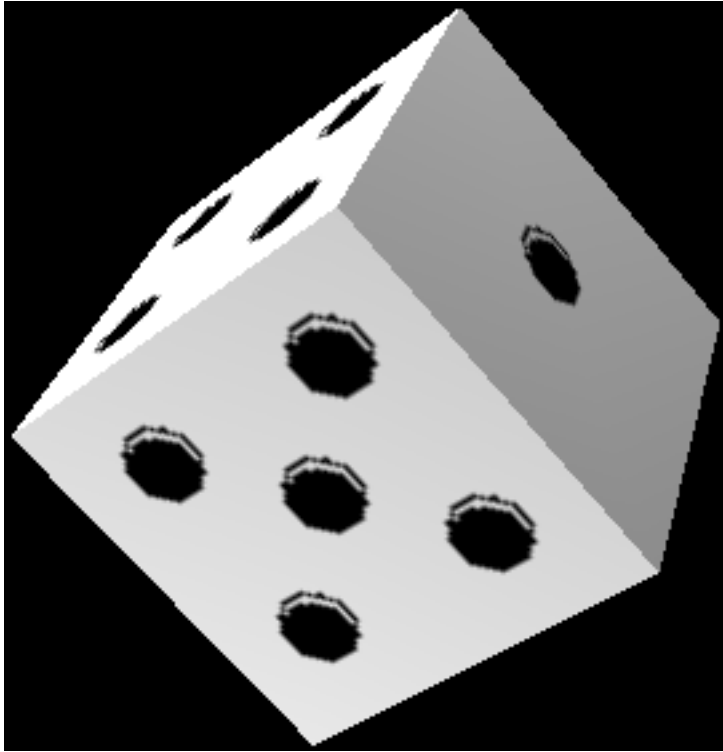
Dans ce tutoriel nous allons aborder la création d'un cube en définissant la position, la normale, la couleur et les coordonnées uv des sommets.

Ce cube servira ensuite pour d'autres tutoriels, notamment pour le prochain tutoriel dans lequel nous apprendrons à déplacer un cube et à gérer la caméra.



Cube 3D

Dans un second exemple, nous afficherons un dé en créant une texture et en adaptant les valeurs des coordonnées uv de la texture.



Dé 3D

 Reportez vous également à l'article  [construction d'un triangle 3D](#).

II - Mesh

Nous allons construire notre mesh en utilisant tout ce qui peut le caractériser , à savoir pour chaque sommet (vertex) :

- La position
- La normale
- La couleur
- Les coordonnées uv de la texture

Nous pourrons changer la couleur de chaque face et nous pourrons aussi définir une texture de façon à créer un dé.

II-1 - Position

Huit sommets pourraient suffire pour créer un cube. Vous trouverez un exemple de cube construit avec 8 sommets dans votre répertoire PureBasic, puis **Exemples\Sources\MeshManual.pb**.

Dans ce tutoriel notre cube sera composé de 4 sommets par face, soit 24 sommets.

Ainsi chaque face du cube est indépendante, en effet aucune n'a de sommet en commun avec une autre. Il est alors possible d'affecter une couleur par face ou de calculer très simplement les coordonnées uv pour chaque sommet.

La position des sommets est déterminée de façon à centrer le cube sur ses axes de rotation et de façon à ce que les côtés du cube aient une taille de 1.

Prenons par exemple la face du haut, elle est définie par les sommets numérotés de 0 à 3 :

```
;Face du dessus avec les sommets 0 à 3
;Sommet 0
Data.f -0.5,0.5,-0.5 ; x,y et z
;Sommet 1
Data.f 0.5,0.5,-0.5 ; x,y et z
;Sommet 2
Data.f 0.5,0.5,0.5 ; x,y et z
;Sommet 3
Data.f -0.5,0.5,0.5 ; x,y et z
```

Ce qui nous permettra par la suite de mieux maîtriser les dimensions de nos entités. Pour obtenir une entité de forme rectangulaire de 100 de largeur, 30 de hauteur et 10 de profondeur il suffira d'écrire :

```
ScaleEntity(#Entity, 100, 30, 10)
```

II-2 - Normale

Pour que les effets de la lumière soient visibles, il faut ajouter le vecteur normal à chaque sommet constituant le mesh.

on peut l'obtenir simplement par le produit vectoriel de deux vecteurs directeurs du plan.

Le produit vectoriel peut se calculer ainsi :

```

Structure s_Vecteur
  x.f
  y.f
  z.f
EndStructure

Procédure ProduitVectoriel(*n.s_Vecteur, *v1.s_Vecteur, *v2.s_Vecteur)
  *n\x = (*v1\y * *v2\z) - (*v1\z * *v2\y)
  *n\y = (*v1\z * *v2\x) - (*v1\x * *v2\z)
  *n\z = (*v1\x * *v2\y) - (*v1\y * *v2\x)
  ProcédureReturn *n
EndProcédure
    
```

II-3 - Couleur

Voici la liste des procédures qui permettent de changer la couleur de chaque face du dé:

- **UpColorCube(*Mesh.s_Mesh, Couleur)** : Change la couleur de la face en haut.
- **DownColorCube(*Mesh.s_Mesh, Couleur)** : Change la couleur de la face en dessous.
- **FrontColorCube(*Mesh.s_Mesh, Couleur)**: Change la couleur de la face avant.
- **BackColorCube(*Mesh.s_Mesh, Couleur)** : Change la couleur de la face arrière.
- **LeftColorCube(*Mesh.s_Mesh, Couleur)** : Change la couleur de la face gauche.
- **RightColorCube(*Mesh.s_Mesh, Couleur)**: Change la couleur de la face droite.
- **UpdateCube(*Mesh.s_Mesh)** : Le changement de couleur est effectif lorsque cette commande est appelée.

Nous allons décrire ici le fonctionnement de la procédure **DownColorCube()**, le principe est identique pour les autres procédures.

```

Procédure DownColorCube(*Mesh.s_Mesh, Couleur)
  *Mem.s_Sommet = *Mesh\VBuffer + 4 * SizeOf(s_Sommet)
  For i = 0 To 3
    *Mem\co=Couleur
    *Mem + SizeOf(s_Sommet)
  Next i
EndProcédure
    
```

Cette procédure permet de modifier la couleur des sommets composant une face du cube, pour ce faire nous avons besoin de connaître l'adresse de ces sommets.

```
*Mem.s_Sommet = *Mesh\VBuffer + 4 * SizeOf(s_Sommet)
```

On ajoute $4 * \text{SizeOf}(s_Sommet)$ à l'adresse initiale ***MeshVBuffer**, car c'est la deuxième face dans les datas.

Ensuite il ne reste plus qu'à modifier la couleur des 4 sommets de la face :

```

For i = 0 To 3
  *Mem\co=Couleur
  *Mem + SizeOf(s_Sommet) ; Pointe sur le sommet suivant
Next i

```

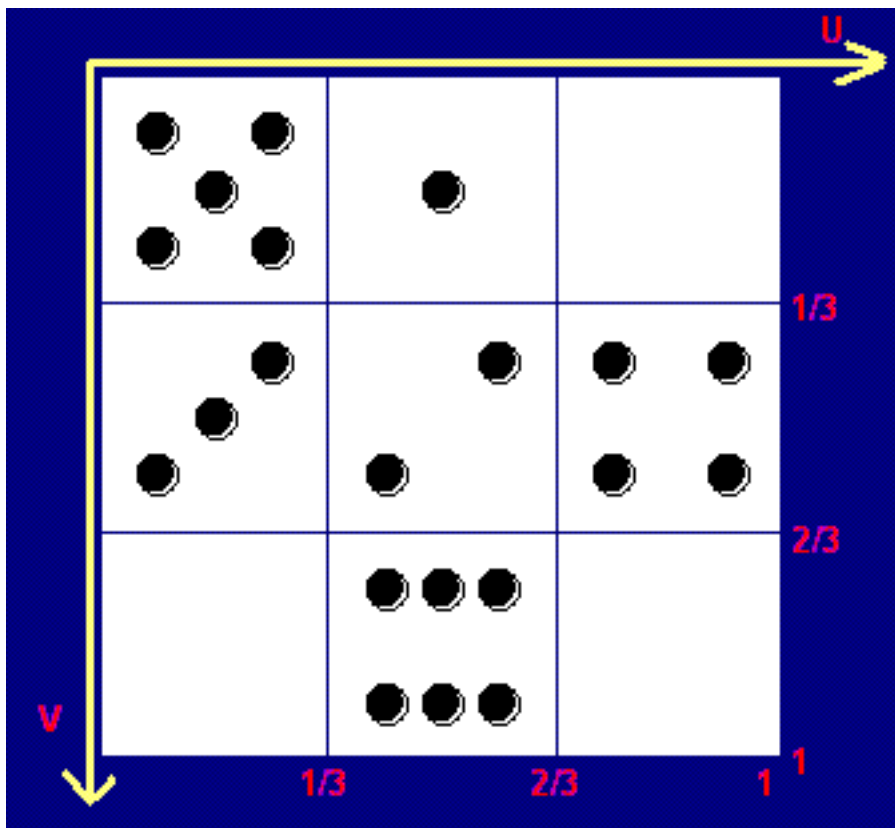
Pour rendre la modification effective, nous devons à nouveau construire le mesh en appelant la fonction **SetMeshData()**, c'est le rôle de la procédure **UpdateCube()**.

```

Procédure UpDateCube(*Mesh.s_Mesh)
  Flag = #PB_Mesh_Vertex | #PB_Mesh_Normal | #PB_Mesh_UVCoordinate | #PB_Mesh_Color
  SetMeshData(*Mesh\No, Flag , *Mesh\VBuffer, 24)
EndProcédure

```

II-4 - Coordonnées uv



Les coordonnées uv de la texture

Rappel : Une face d'un dé est composée de 4 sommets, à ces sommets il est possible d'associer des coordonnées u et v d'une texture.

Prenons l'exemple de la face du dessous du dé, j'ai décidé arbitrairement d'afficher le '2'.

Voici un extrait des datas correspondant à la face du dessous, je n'ai gardé que les coordonnées uv des sommets qui composent cette face :

```
#Tiers = 1.0/3.0

;Dessous 4 à 7
;Sommet en haut à gauche
Data.f #Tiers, #Tiers ; u = 1/3 & v = 1/3

;Sommet en haut à droite
Data.f #Tiers*2,#Tiers ; u = 2/3 & v = 1/3

;Sommet en bas à droite
Data.f #Tiers*2,#Tiers*2 ; u = 2/3 & v = 2/3

Sommet en bas à gauche
Data.f #Tiers,#Tiers*2 ; u = 1/3 & v = 2/3
```

on retrouve bien les coordonnées qui encadrent notre '2' dans l'image ci dessus.

Pour afficher la texture entière sur la face du dessous changez les coordonnées ainsi :

```
;Dessous 4 à 7
;Sommet en haut à gauche
Data.f 0, 0 ; u = 0 & v = 0

;Sommet en haut à droite
Data.f 1,0 ; u = 1 & v = 0

;Sommet en bas à droite
Data.f 1,1 ; u = 1 & v = 1

Sommet en bas à gauche
Data.f 0,1 ; u = 0 & v = 1
```

Pour créer une mosaïque avec la texture sur la face du dessous changez les coordonnées ainsi :

```
;Dessous 4 à 7
;Sommet en haut à gauche
Data.f 0, 0 ; u = 0 & v = 0

;Sommet en haut à droite
Data.f 2,0 ; u = 2 & v = 0

;Sommet en bas à droite
Data.f 2,2 ; u = 2 & v = 2

Sommet en bas à gauche
Data.f 0,2 ; u = 0 & v = 2
```

Avec le code ci dessus , la texture sera répétée 4 fois sur la face du dé. 2 fois en largeur et 2 fois en hauteur.

Il est possible de faire tourner la texture de la face en changeant les coordonnées :

```
#Tiers = 1.0/3.0

;Dessous 4 à 7
;Sommet en haut à gauche
```

```
Data.f #Tiers,#Tiers*2 ; u = 1/3 & v = 2/3

;Sommet en haut à droite
Data.f #Tiers, #Tiers ; u = 1/3 & v = 1/3

;Sommet en bas à droite
Data.f #Tiers*2,#Tiers ; u = 2/3 & v = 1/3

Sommet en bas à gauche
Data.f #Tiers*2,#Tiers*2 ; u = 2/3 & v = 2/3
```

III - Sources

La première source permet d'afficher un cube 3D et de changer la couleur de chaque face, la seconde source montre comment créer la texture d'un dé et comment l'appliquer sur le cube pour afficher un dé 3D.

Source [Construction d'un cube 3D.](#)

Source [Construction d'un dé 3D.](#)

IV - Remerciements

Je tiens à remercier tout particulièrement **olsimare** pour la relecture de l'article.

