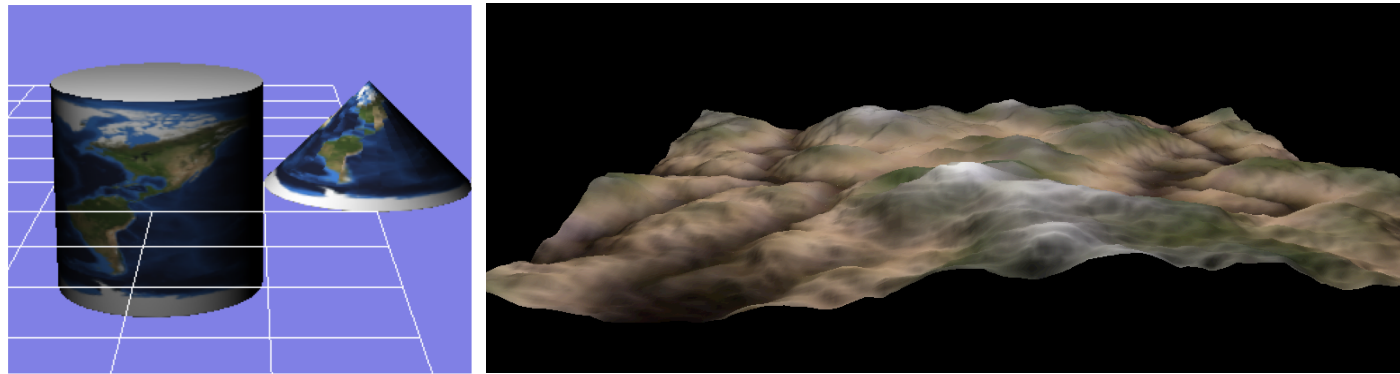
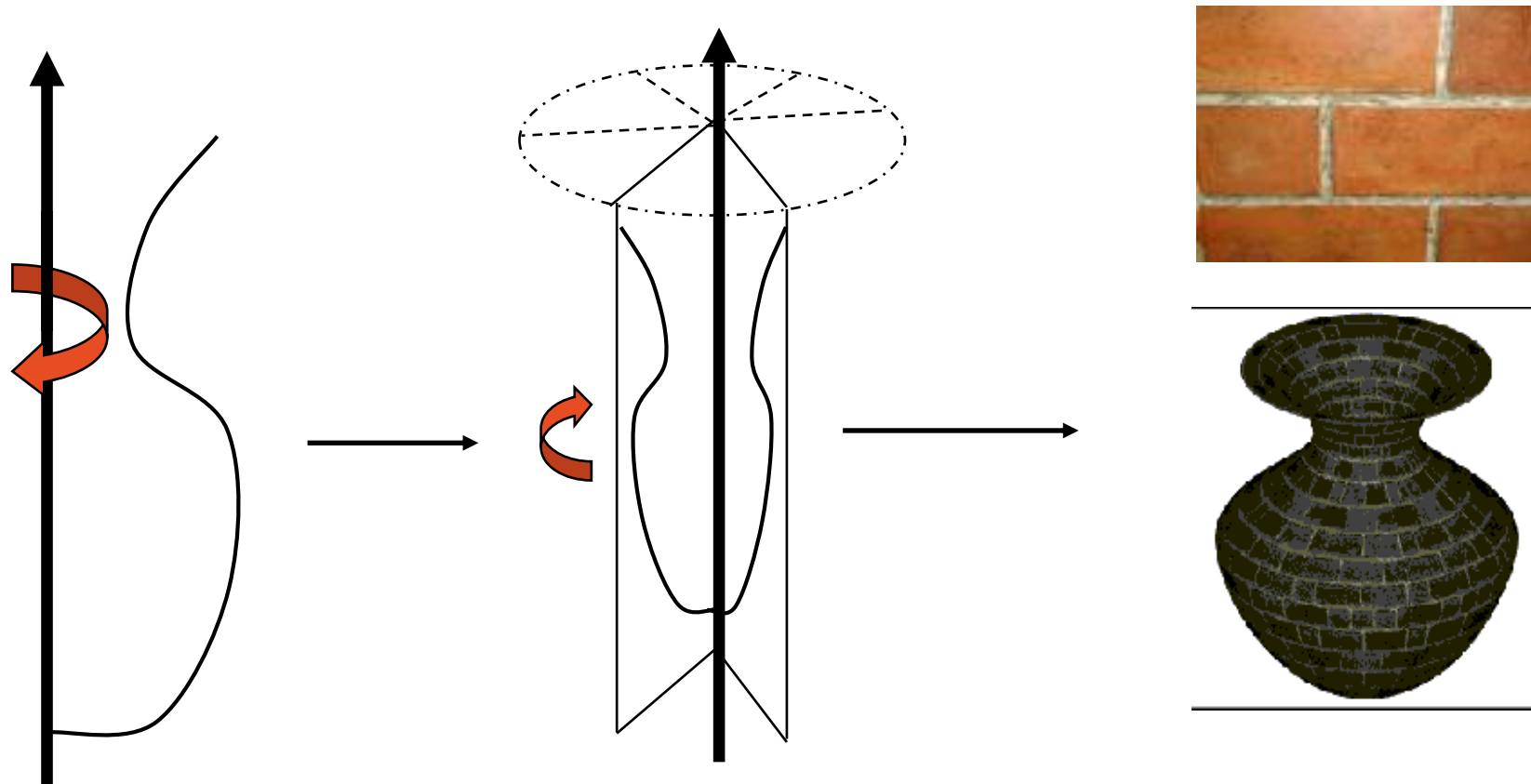

TD3 - Textures :

vase, formes de base, terrain



Eric Galin (semestre printemps)
Florence Zara (semestre automne)
LIRIS-ORIGAMI, Université Lyon 1

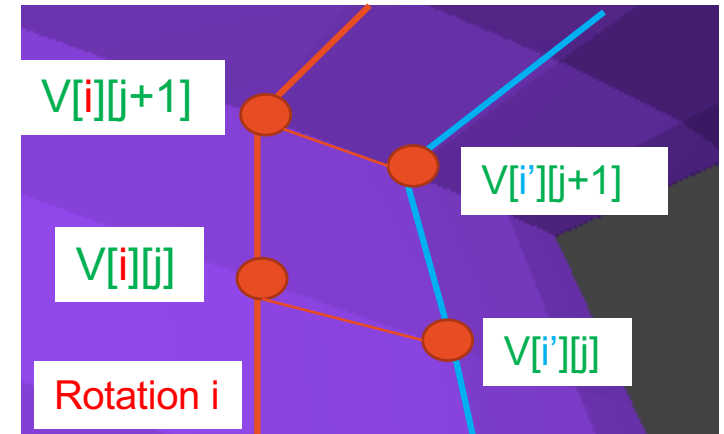
Exercice 1 - Texture sur le vase créé par révolution



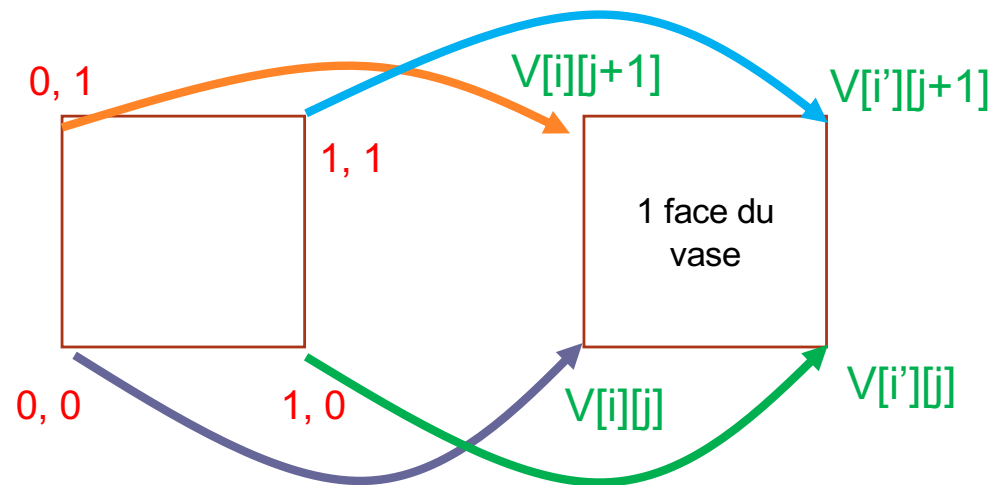
Exercice : définir (u,v) pour chaque sommet

Ajout des textures sur le vase

1 texture par face carrée du vase



Rotation $i' = (i+1) \% \text{nbrot}$



Ajout des textures sur le vase

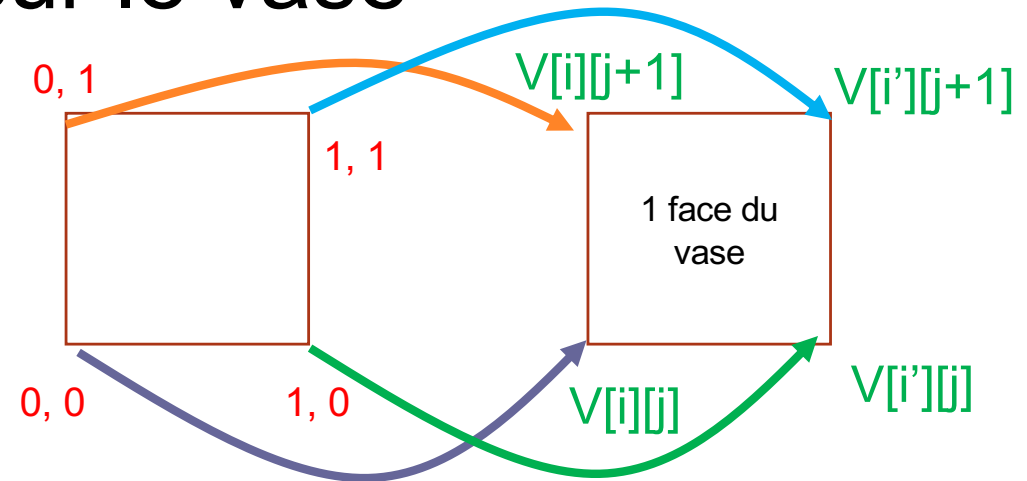
```
void ViewerEtudiant::init_vase()
{
    m_vase = Mesh(GL_TRIANGLES);
    m_vase.color(1.0, 1.0, 1.0);
    for(int i=0; i<vase_NBR0T; i++){
        // Attention boucle de 0 à vase_NBPT-2 (car j+1)
        for(int j=0; j<vase_NBPT-1; j++){
```

```
        // Premier triangle
```

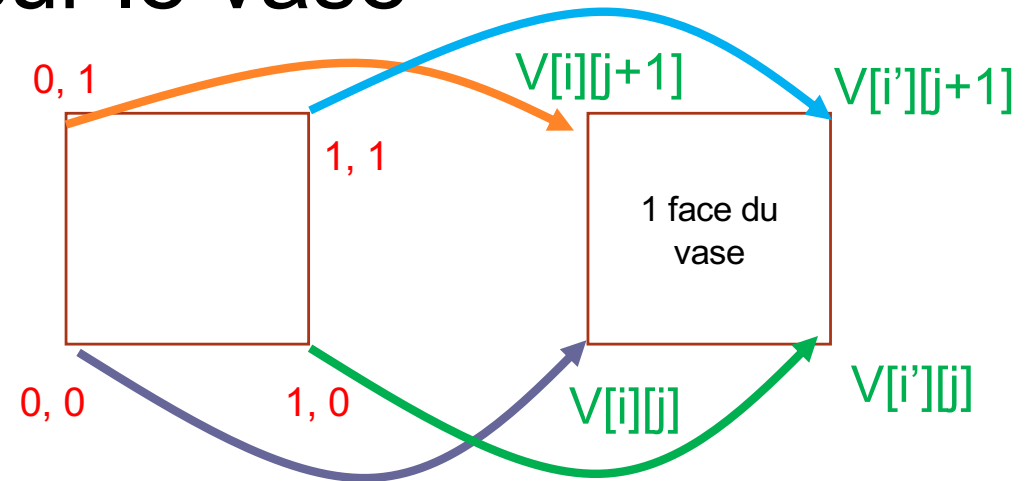
```
        m_vase.texcoord(0, 0);
        m_vase.normal(vase_vn[i][j]);
        m_vase.vertex(vase_v[i][j]);
```

```
        m_vase.texcoord(1, 1);
        m_vase.normal(vase_vn[(i+1) % vase_NBR0T][j+1]);
        m_vase.vertex(vase_v[(i+1) % vase_NBR0T][j+1]);
```

```
        m_vase.texcoord(1, 0);
        m_vase.normal(vase_vn[(i+1) % vase_NBR0T][j]);
        m_vase.vertex(vase_v[(i+1) % vase_NBR0T][j]);
```



Ajout des textures sur le vase



```
// Second triangle
```

```
m_vase.texcoord(0, 0);  
m_vase.normal(vase_vn[i][j]);  
m_vase.vertex(vase_v[i][j]);
```

```
m_vase.texcoord(0, 1);  
m_vase.normal(vase_vn[i][j+1]);  
m_vase.vertex(vase_v[i][j+1]);
```

```
m_vase.texcoord(1, 1);  
m_vase.normal(vase_vn[(i+1) % vase_NBROT][j+1]);  
m_vase.vertex(vase_v[(i+1) % vase_NBROT][j+1]);
```

```
}//for_j
```

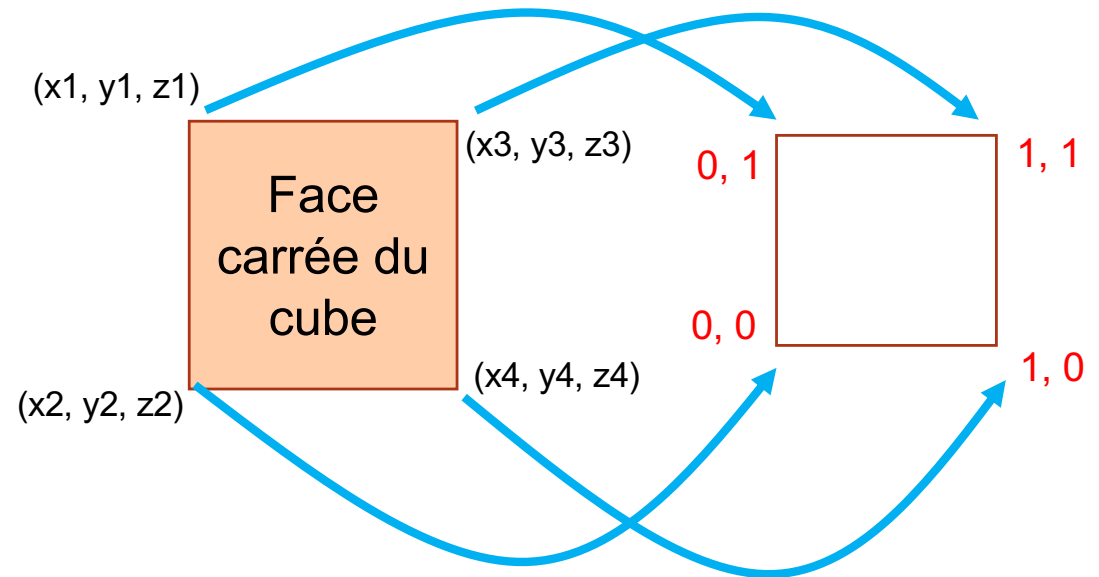
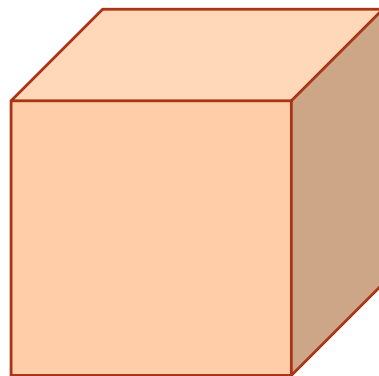
```
}for_i
```

```
//void
```

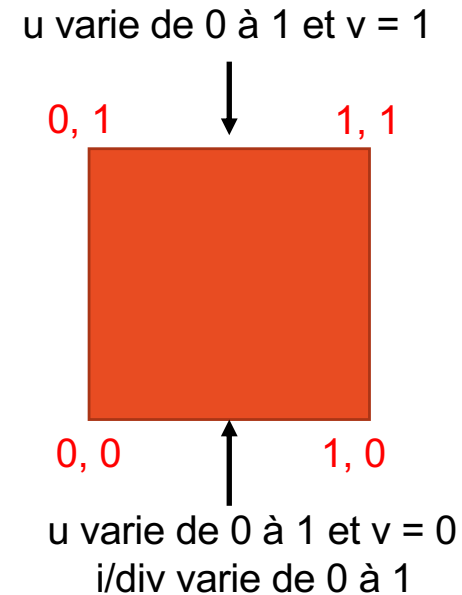
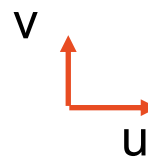
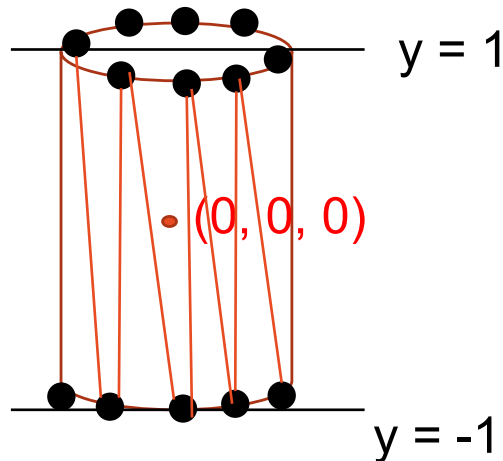
Exercice 2 - Texture sur les formes de base

- Rajouter les coordonnées de texture aux formes de base
 - Cube : 1 texture par face carrée
 - Cylindre : 1 texture (sans considérer le dessus/dessous)
 - Cône : 1 texture (sans considérer le dessous)
 - Sphère : 1 texture

Forme de base : cube



Forme de base : cylindre



Boucle sur i pour faire varier l'angle de 0 à 2π avec div le nombre de divisions
 i/div varie de 0 à 1 → permet d'avoir une variation entre 0 et 1

Sommets du haut

Coordonnées de texture : $u = i/\text{div}, v = 1$

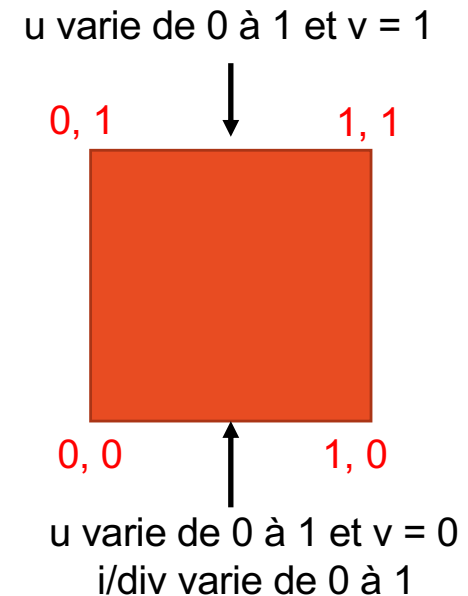
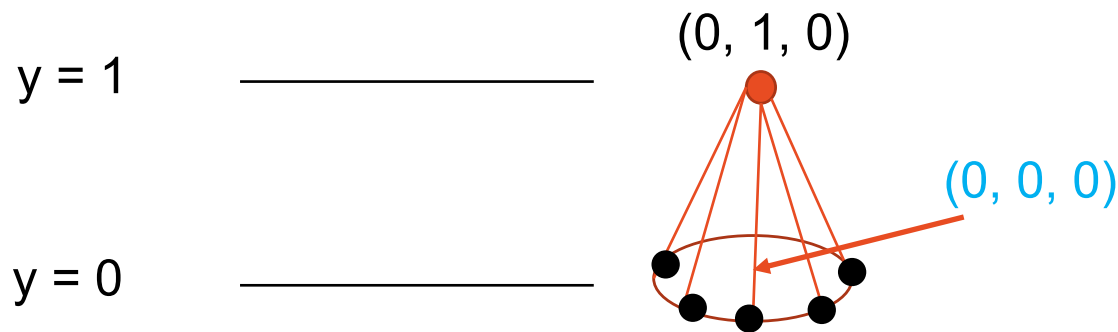
Coordonnées 3D : $x = \cos \alpha, y = 1, z = \sin \alpha$

Sommets du bas

Coordonnées de texture : $u = i/\text{div}, v = 0$

Coordonnées 3D : $x = \cos \alpha, y = -1, z = \sin \alpha$

Forme de base : cône



Boucle sur i pour faire varier l'angle de 0 à 2π avec div le nombre de divisions
i/div varie de 0 à 1 → permet d'avoir une variation entre 0 et 1

Sommet du haut

Coordonnées de texture : $u = i/\text{div}, v = 1$

Coordonnées 3D : $x = 0, y = 1, z = 0$

Sommets du cercle de coordonnées

Coordonnées de texture : $u = i/\text{div}, v = 0$

Coordonnées 3D : $x = \cos \alpha, y = 0, z = \sin \alpha$

Forme de base : sphère

Sphère = superposition de cercles

Variation de l'angle α de $-\pi/2$ à $\pi/2$

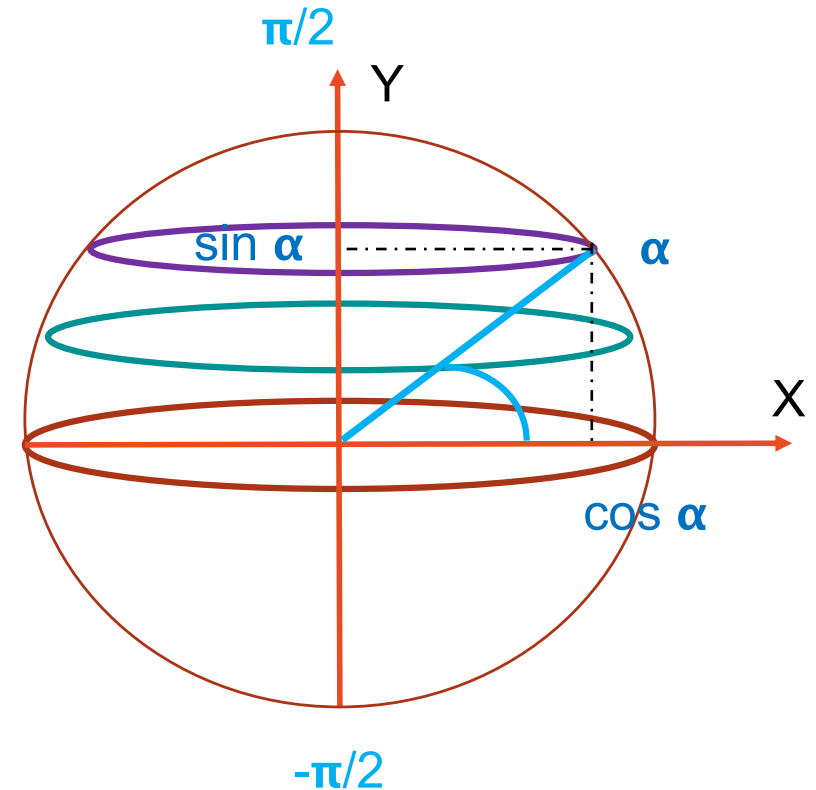
Variation de l'angle β de 0 à 2π

Sommets de la sphère de coordonnées :

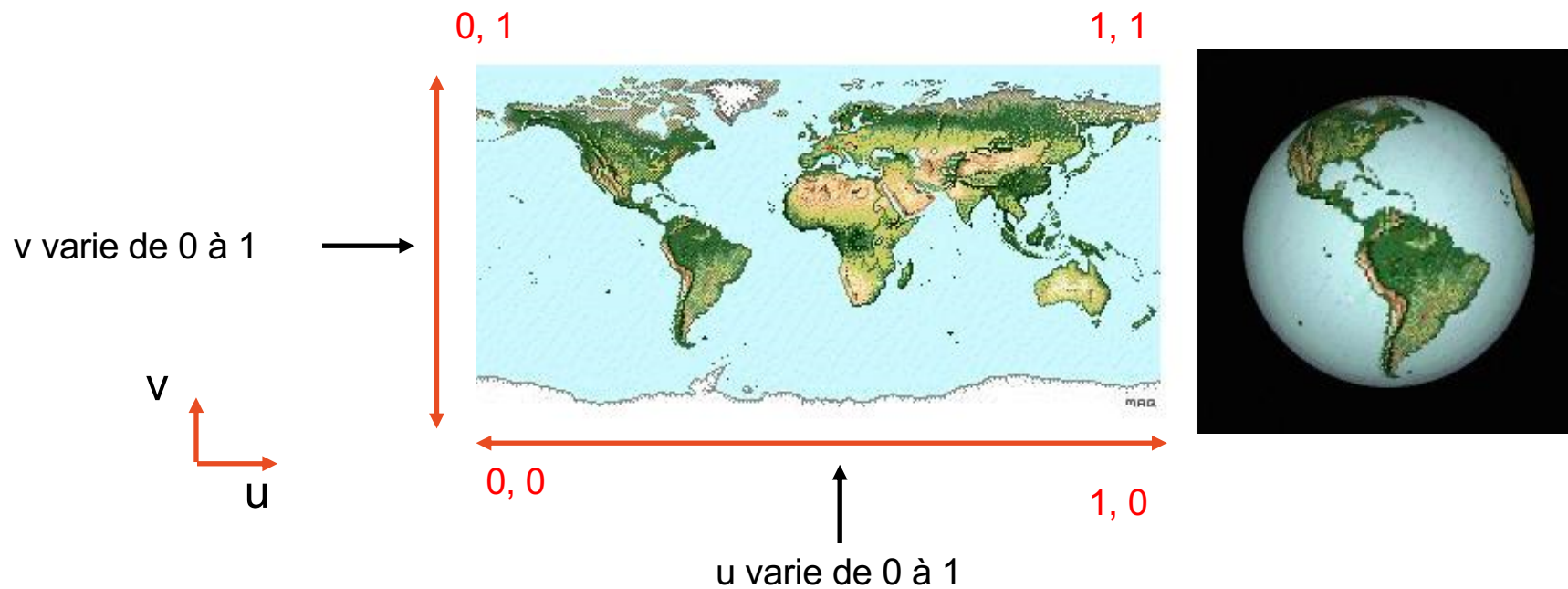
$$x = \cos \alpha * \cos \beta$$

$$y = \sin \alpha$$

$$z = \cos \alpha * \sin \beta$$



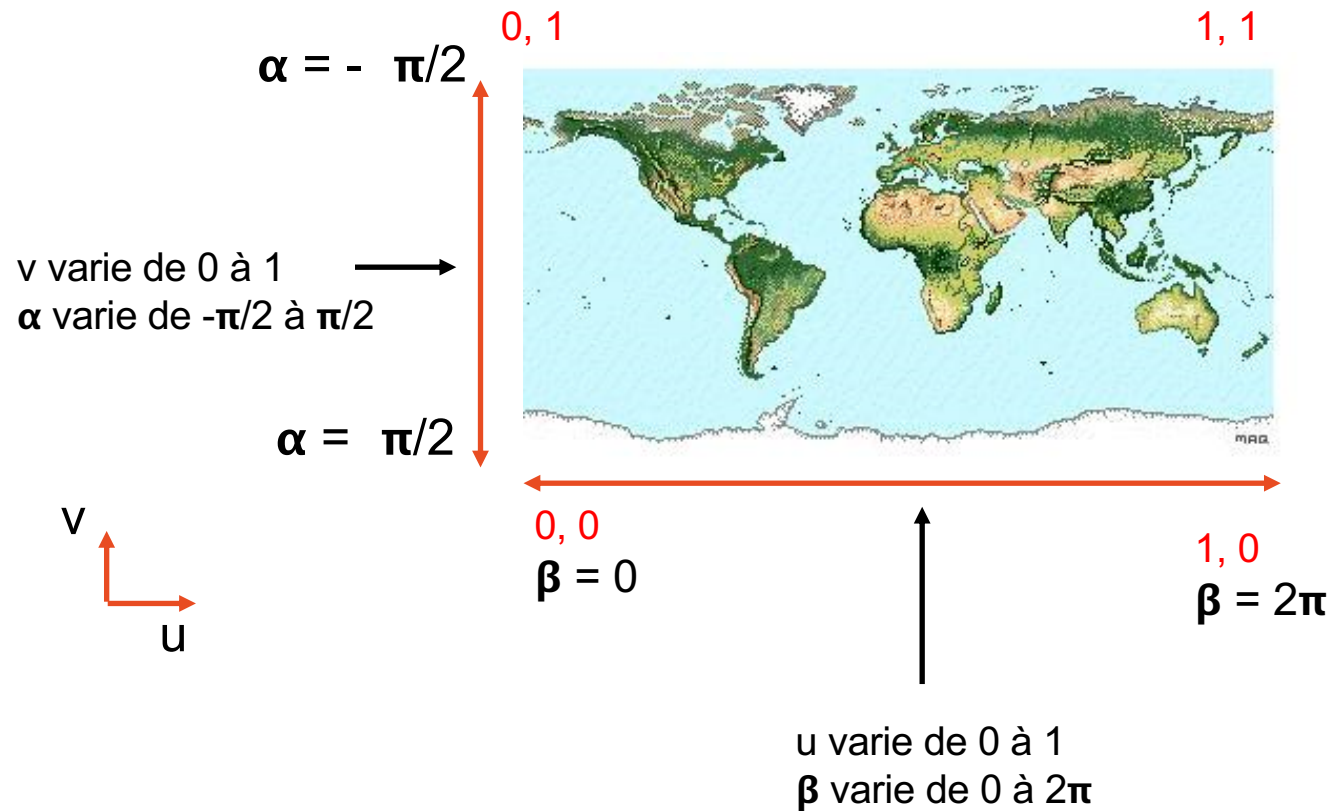
Forme de base : sphère



Coordonnées de textures (u,v) doivent varier de 0 à 1

Correspondance à faire entre la variation des angles (β, α) , et la variation de (u,v)

Forme de base : sphère

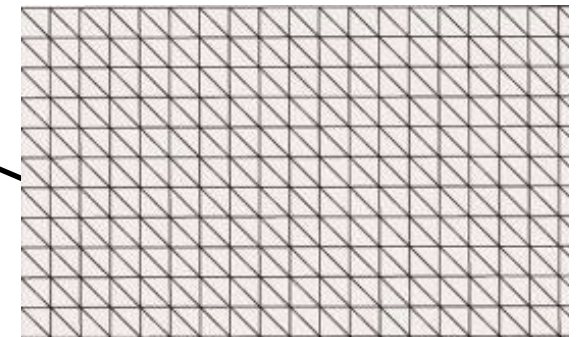
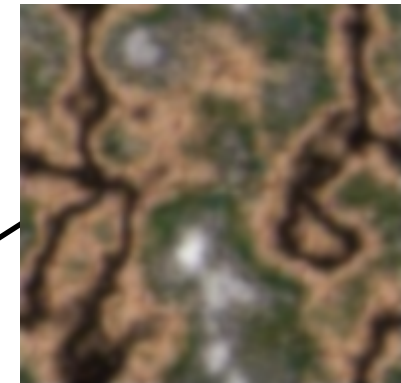
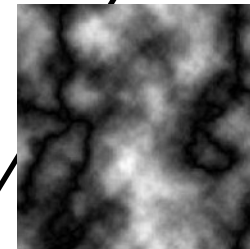
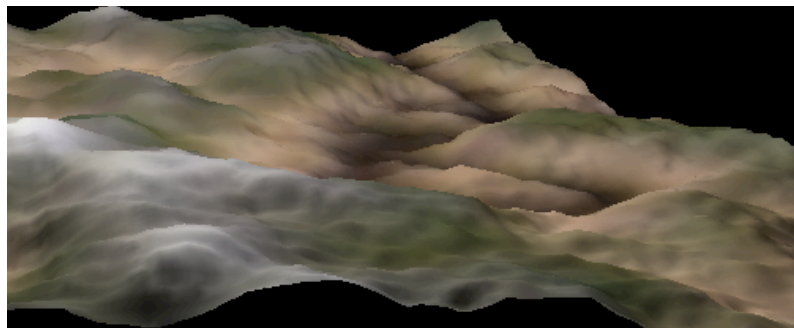


Coordonnées de texture : $u = \beta / 2\pi$, $v = 0.5 + \alpha / \pi$

Coordonnées 3D : $x = \cos \alpha * \cos \beta$, $y = \sin \alpha$, $z = \cos \alpha * \sin \beta$

Exercice 3 - Texture sur le terrain

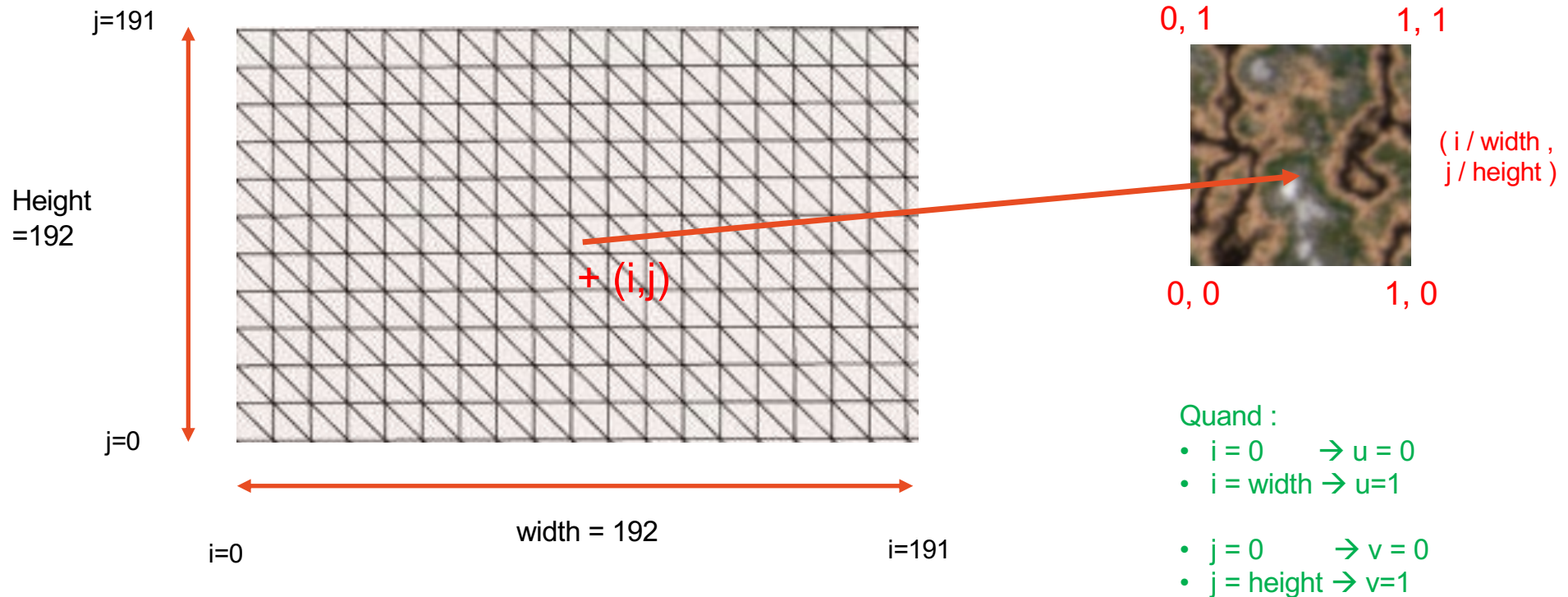
- Représentation d'un terrain
 - Carte de hauteur (niveau de gris)
 - Texture pour la couleur
 - Triangulation de la carte de hauteur



Exercice : définir (u,v) pour chaque sommet du terrain

Ajout de la texture sur le terrain

Il faut avoir une variation entre 0 et 1 des coordonnées de texture



Boucle sur la dimension en i

Boucle sur la dimension en j

```
m_terrain.textcoord( i / width, j / height );  
m_terrain.vertex (i, valeur de l'image en (i, j), j);
```