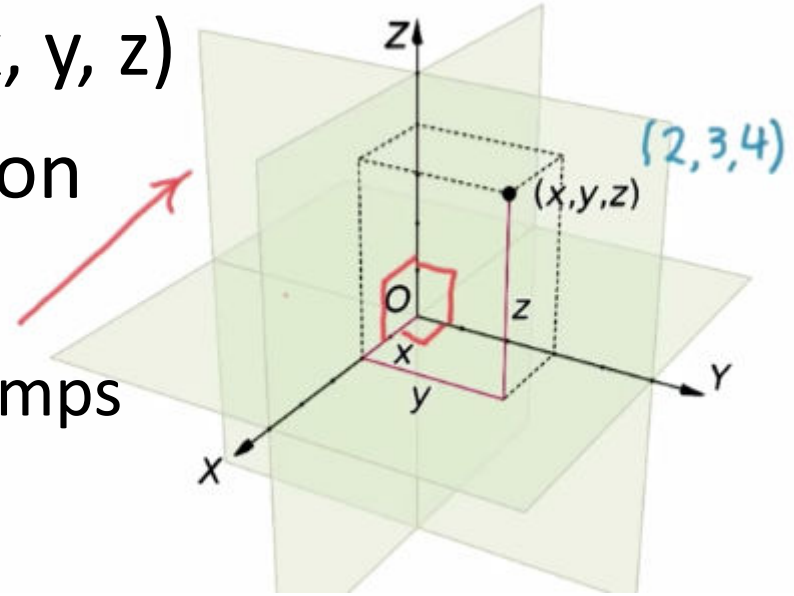


Point, Vecteurs et Maillages

Chapitre 2

Systeme de coordonnees

- On utilise les points et les vecteurs
- On a toujours besoin d'une reference: repere
- Coordonnees cartesiennes (perpendiculaires)
- Point P de coordonnees (x, y, z)
- Un vecteur est une direction
 - Sans points d'application
 - Similaire a une duree de temps



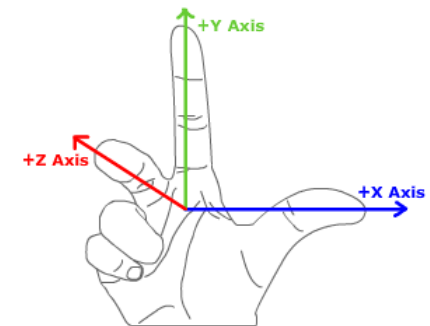
Quiz: Point ou vecteur?

Point	Vecteur	
<input type="checkbox"/>	<input type="checkbox"/>	Je te donne rendez-vous demain à 8 heure, si cela vous arrange.
<input type="checkbox"/>	<input type="checkbox"/>	Le diamètre de la terre est en général pas supérieur que 150 mètres en dessus du niveau de la mer.
<input type="checkbox"/>	<input type="checkbox"/>	Ce film est à peu pré de durée 3 heures.
<input type="checkbox"/>	<input type="checkbox"/>	La station de pétrole est à 200 mètres depuis cette route.

- Le vecteur est une durée
- Le point est une indication précise

Main droite/ main gauche

- Nous regardons selon le système de coordonnées de la main droite
- Le lapin creuse selon le système de coordonnées de la main gauche
- Passer d'un repère à l'autre est par inversion (négation) de l'axe Z
- Exemple:
 - aller vers la droite augmente la longitude
 - aller vers le haut augmente la latitude
 - le système (longitude/Latitude/Altitude) est Right-handed

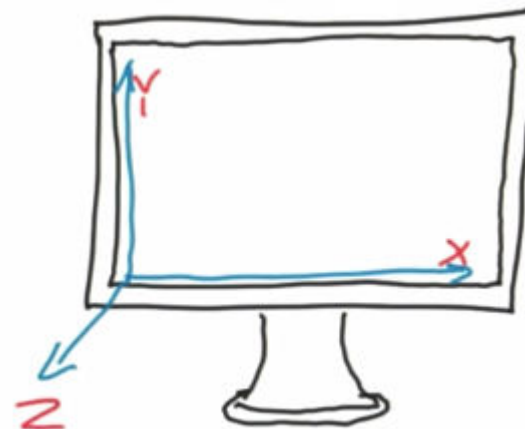
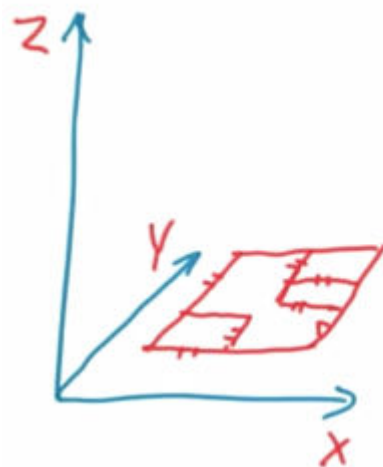


Quiz: Coordonnée main gauche/droite

- Configuration 1: (Coord. MainG□/mainD□)
 - Axe X pointe vers le sud
 - Axe Z pointe vers l'ouest
 - Axe Y pointe vers le haut
- Configuration 2: (Coord. MainG□/mainD□)
 - Axe X pointe vers l'est
 - Axe Y pointe vers le sud
 - Axe Z pointe vers le haut

Meilleur système de coordonnées?

- Selon notre choix:
 - Si on travaille sur l'écran
 - Si on travaille sur sur une tablette graphique, etc.
- Les deux sont en main droite
- Avec une rotation de 90° selon l'axe X

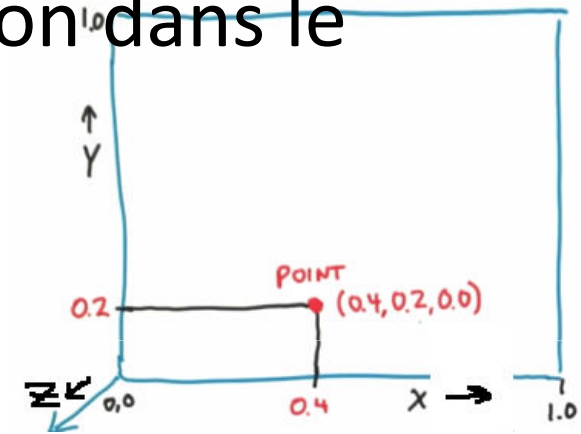


Transformation de coordonnées

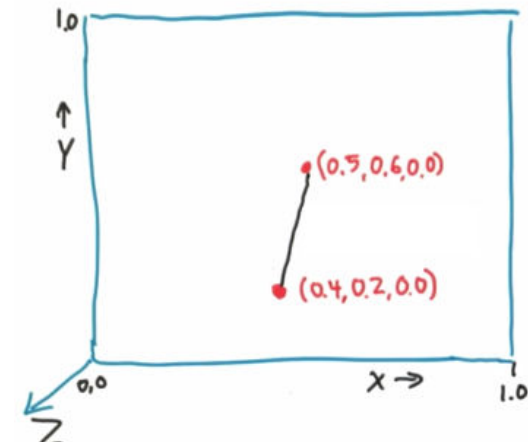
- Pour un architecte:
 - X vers l'est, Y vers le nord, Z vers le haut
 - Point P de coordonnée (10, 24, 13) selon l'architecte
- Pour le Dessinateur 2D, sont repère préféré est:
 - X vers l'est, Y vers la haut et Z vers le sud
 - Le même point est alors de coordonnées:
 - $X = \square$, $Y = \square$, $Z = \square$

Points, lignes et triangles

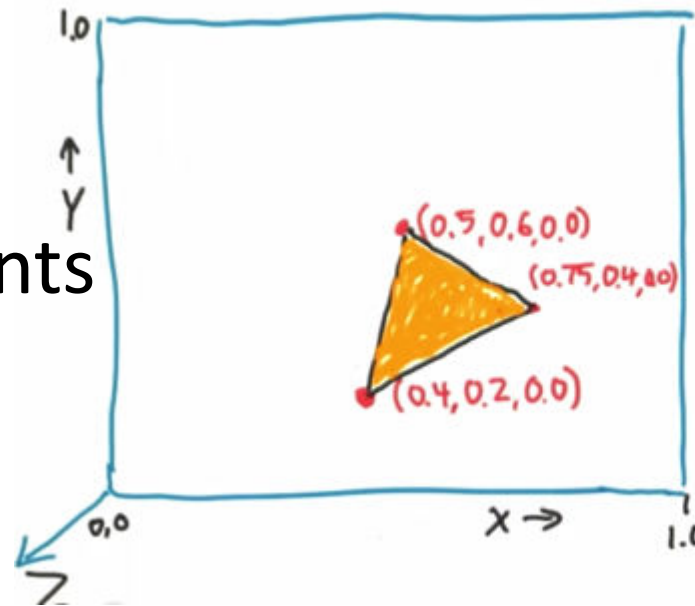
- Un point est donné par sa position dans le repère $P(x, y, z)$



- En définissant un 2^{ème} point, on peut créer un segment de ligne.
 - L'ordre n'est pas important



- Triangles
- Ensemble de 3 points
- Une simple facette
- En coloration fixe
- Code:



TRIANGLES

(0.4, 0.2, 0.0)
 (0.5, 0.6, 0.0)
 (0.75, 0.4, 0.0)

```
// defining geometry
var triangleGeometry = new THREE.Geometry();
// vertices
triangleGeometry.vertices.push( new THREE.Vector3( 1, 1, 0 ) );
triangleGeometry.vertices.push( new THREE.Vector3( 3, 1, 0 ) );
triangleGeometry.vertices.push( new THREE.Vector3( 3, 3, 0 ) );
// face
triangleGeometry.faces.push( new THREE.Face3( 0, 1, 2 ) );

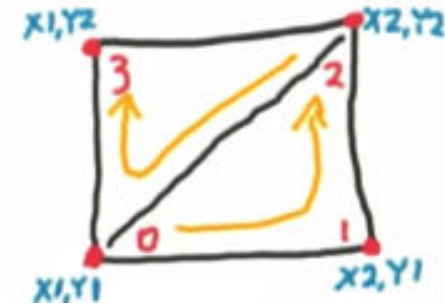
// putting it all together
var triangleMaterial = new THREE.MeshBasicMaterial({
  color: 0x2685AA,
  side: THREE.DoubleSide
});
var triangleMesh = new THREE.Mesh( triangleGeometry, triangleMaterial );
scene.add(triangleMesh);
```

Exercice

- On vous demande de compléter la fonction

Solution

- Plusieurs avec 2 facettes:
 - Brute par 6 points
 - Compacte par 4 points
- Possible aussi de faire appel à une commande



```
square.vertices.push( new THREE.Vector3( x1, y1, 0 ) ); // vertex 0
square.vertices.push( new THREE.Vector3( x2, y1, 0 ) ); // vertex 1
square.vertices.push( new THREE.Vector3( x2, y2, 0 ) ); // vertex 2
square.vertices.push( new THREE.Vector3( x1, y2, 0 ) ); // vertex 3
```

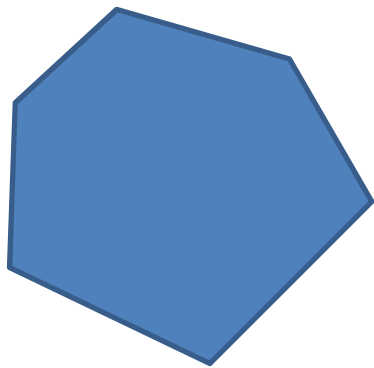
```
// using 4 vertices and THREE.Face3
square.faces.push( new THREE.Face3( 0, 1, 2 ) );
square.faces.push( new THREE.Face3( 2, 0, 3 ) );
```

```
// using THREE.Face4
square.faces.push( new THREE.Face4( 0, 1, 2, 3 ) );
```

Tessellation

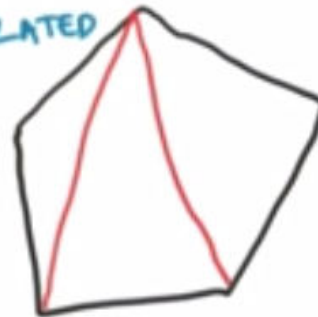
- Similaire à la triangulation
- C'est quoi le nombre minimal de triangles à utiliser?
- Avec N cotes

Besoin de $N-2$ triangles

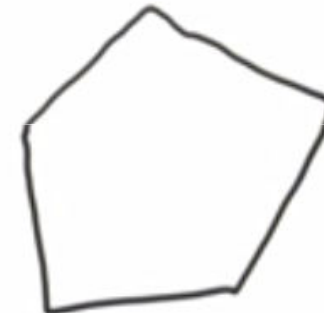
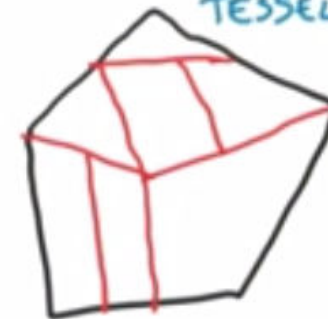


TESSELLATION
TRIANGULATION

TRIANGULATED

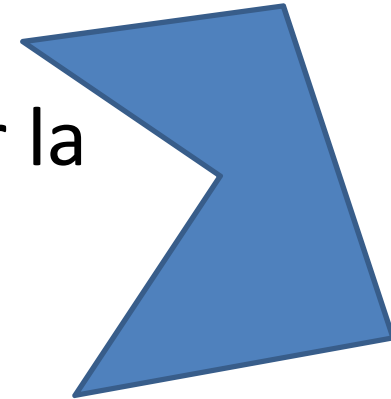


TESSELLATED



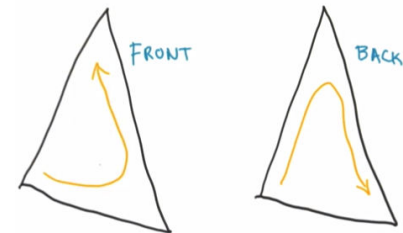
Backface culling-masquage de facettes

- Nombre de façons différentes pour la triangulation?



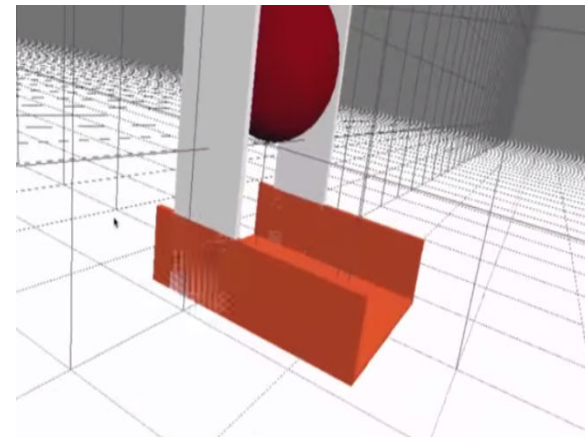
- Afin de gagner en performance
- On a besoin de déterminer si la facette est en tournée en avant ou vers l'arrière.
- L'ordre des sommets est important

```
geometry.faces.push( new THREE.Face3( 0, 1, 2 ) );  
// the order of the face is flipped (was 2, 0, 3):  
geometry.faces.push( new THREE.Face3( 2, 3, 0 ) );
```



Combat sur la position Z

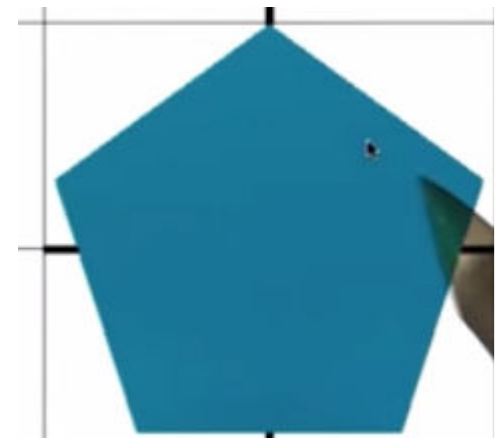
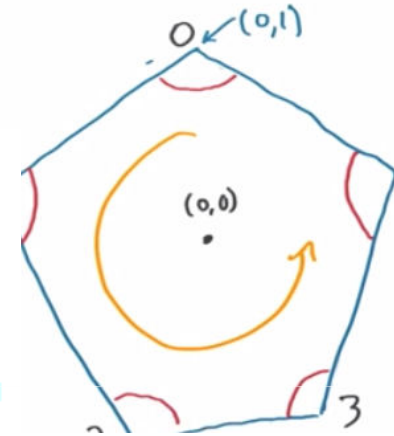
- Superposition de 2 objets de couleurs différentes sur la même profondeur
- Comment corriger cet effet?
 - Rendre chaque barre blanche moins épaisse
 - Utiliser un GPU avec un Z-buffer plus précis
 - Utiliser un autre algorithme de rendu tel que le lancer de rayons
 - Supprimer les parties basses des pieds



Générateur de polygones

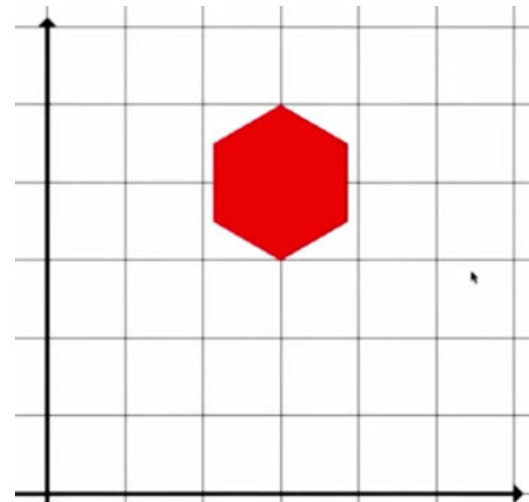
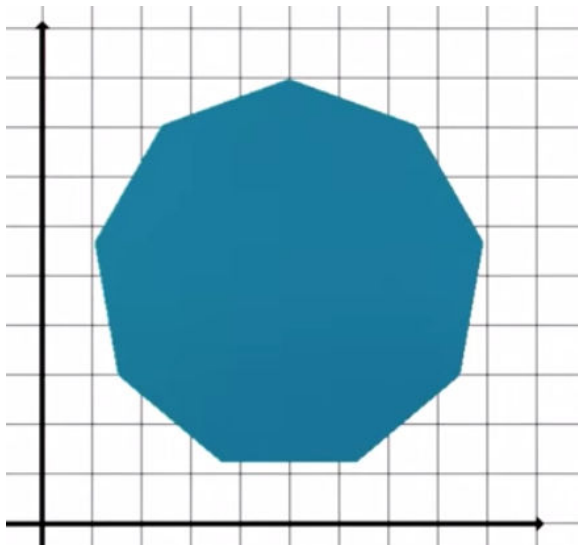
- Générer les facettes dans une boucle

```
function PolygonGeometry(sides) {  
  var geo = new THREE.Geometry();  
  
  // generate vertices  
  for ( var pt = 0 ; pt < sides; pt++ )  
  {  
    // Add 90 degrees so we start at +Y axis, rotate counterclockwise around  
    var angle = (Math.PI/2) + (pt / sides) * 2 * Math.PI;  
  
    var x = Math.cos( angle );  
    var y = Math.sin( angle );  
  
    // YOUR CODE HERE  
    // Save the vertex location - fill in the code  
  
  }  
  // YOUR CODE HERE  
  // Write the code to generate minimum number of faces for the polygon.  
  
  // Return the geometry object  
  return geo;  
}
```



Générateur de polygones sur mesure

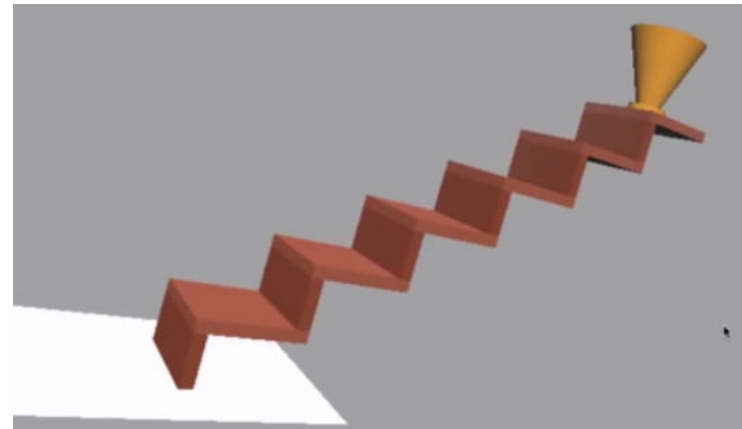
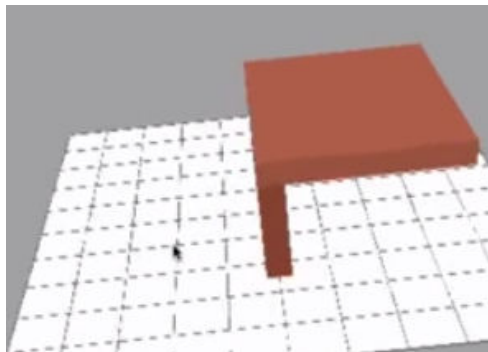
- Gestion de la position comme étant un vecteur 3D
- Gestion du rayon du polygone



Création d'un escalier

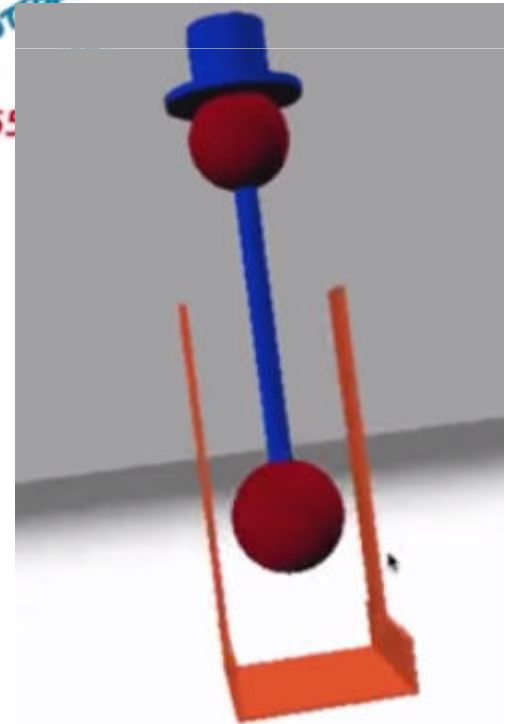
- Débuter avec le modele de base
- Réaliser une boucle de pas

```
function createStairs() {  
  
    // +Y direction is up  
    // Define the two pieces of the step, vertical and horizontal  
    // THREE.CubeGeometry takes (width, height, depth)  
    var stepVertical = new THREE.CubeGeometry(stepWidth, verticalStepHeight, stepThickness);  
    var stepHorizontal = new THREE.CubeGeometry(stepWidth, stepThickness, horizontalStepDepth);
```



Création du oiseux buveur

```
boxGeometry = new THREE.CubeGeometry( 125.6, 389.8, 202.1 );  
sphereGeometry = new THREE.SphereGeometry( 202.1, 32, 16 );  
cylinderGeometry = new THREE.CylinderGeometry( 29.4, 202.1, 55
```



Tracking camera / objets 3D

- <http://www.hollywoodcamerawork.us/trackingplates.html>
- Width 6, focus ~5, scale 5m