Ambiant : sert à simuler les ombres/lumières. On rajoute à nos objets un peu de lumière

Diffuse : ce qui réagit en fonction de la position de la lumière. Ex. : le matériau qui n’a que de la lumière diffuse : le plâtre.

Spéculaire : c’est le reflet, ça prend en compte la position de la caméra. Ex. : le miroir est le matériau spéculaire par excellence.

Mode forward : il calcule la lumière au pixel près. Sous Unity, ils ont fait une bidouille :il considère que des lumières sont prioritaires par rapport à d’autres.

Deffered : quand on fait de la 3D, y’a deux manières de calculer les choses. On rend toute la scène

Render Modes :

* RGB : couleurs normales
* Alpha : définit la transparence. Plus l’alpha est grand, moins c’est transparent.
* Vue Mipmaps : c’est pour montrer si la texture est trop détaillée ou pas. Le rouge : ça va, bleu : trop détaillé.

Les manip’ :

* Déplacer la vue : clic droit : rotation vue
* Alt + clic gauche : rotation centrée sur l’objet
* Bouton milieu souris : translation vue
* Molette : zoom
* Gizmo : se ramener à une vue de base
* Double clic dans hierarchy window : centre la caméra sur l’objet.

Les objets : Unity c’est le seul à prendre en compte tous les rendus.

Sujet : faire un L.D. de progression

Principe du material :

Un material est une aggregation de propriétés de rendu d’une surface. Définit avec les données et le calcul du rendu du material. Un material peut utiliser des textures, il définit l’interaction avec la lumière (diffuse, spectaculaire). Il est lié à un programme shader (qui définit le calcul).

Le bump mapping : technique pour faire croire qu’il y a des reliefs

Le shader : manière dont le pixel est illuminé, c’est un programme.

Trigger : déclenche un événement.

Les Rigidbodies : pour qu’un objet soit soumis à la physique :

* Lui appliquer un rigidbody (c’est un component)
* Lui appliquer un collider s’il n’en a pas

Manipuler un objet physique : on ne lui applique pas de transformation :

* On lui applique une force
* Le moteur physique calcule sa nouvelle position
* (prise en compte de la masse, des collisions etc.)
* Le moteur de rendu affiche la nouvelle position/orientation

//Applique une force de 10 vers les y positifs :

Au niveau du code, ça donne : rigidbody.AddForce (0,10,10) ;

Static collider : c’est un objet qui a juste un collider et il ne bougera pas.

Système de particules : point d’émission qui émet plein de petites particules.

Les animations : se trouve dans « Windows => animation »

Le scripts : c’est un composant, un monobehavior.

Pour créer un script : windows => create => [choisir langage]. Trois langages : Javascript, Csharp et Boo. Utiliser le CSharp. On clique sur son script et on peut éditer pour le modifier.

Point d’arrêt = dire au code lors de l’utilisaion de s’arrête rà une ligne précise.

Update : appelé à chaque calcul de frame. Il mettra à jour tous les objets.