

HMIN323: Informatique graphique

Rendu TP : PBR

Tianning MA

M2 IMAGINA

09/11/2020

Table de matière

1. Introduction	2
2. Rendu et explication des exercices	2
2.1 Ajout des PBR paramètres et la texture	2
2.2 Changement de modèle (de plus de triangles)	4
2.3 Découvert et application d'un Modèle de PBR	6

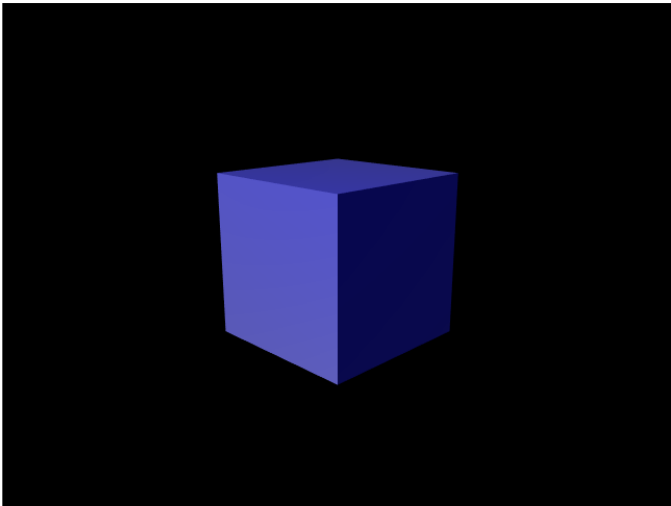
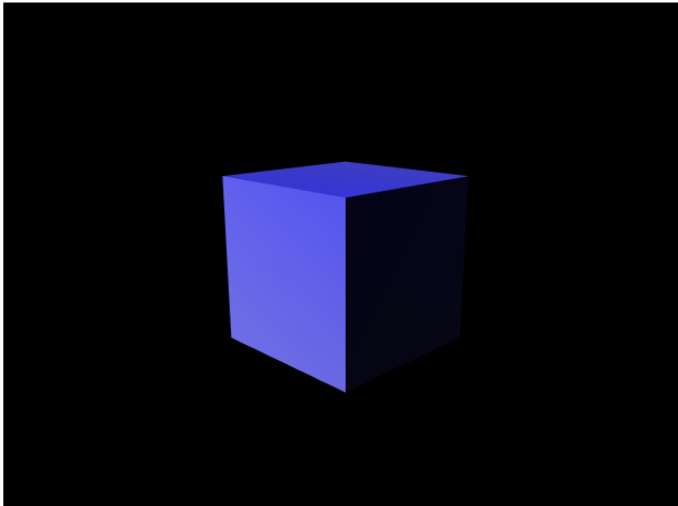
1. Introduction

Ce compte rendu est dédié au TP PBR en WebGL et basé sur le code de M. Beugnon. Vous pouvez trouver le code dans mon espace git : <https://github.com/matianning/InformatiqueGraphique/tree/main/PBR>

2. Rendu et explication des exercices

Remarque : j'ai bien constaté que le roughness doit être un floatant de $[0,1]$, cependant les captures d'écran montrent c'est un chiffre de $[0,100]$. Je suis conscient de ce problème, mais au moment de capturer (certaines) les images, je l'ai pas encore modifié, cela reste à corriger.

2.1 Ajout des PBR paramètres et la texture

Visualisation des différents résultats selon les paramètres de roughness et metalness																																																																	
<div><p>60.0</p></div> <div><h3>Parameters</h3><table><tr><td>viewPosition[0]</td><td>5</td><td>[1] 2</td><td>[2] 5</td></tr><tr><td>lightPosition[0]</td><td>0</td><td>[1] 10</td><td>[2] 5</td></tr><tr><td>lightColor[0]</td><td>1</td><td>[1] 1</td><td>[2] 1</td></tr><tr><td>objectColor[0]</td><td>1</td><td>[1] 0</td><td>[2] 0</td></tr></table><h3>PBR parameters (TO DO)</h3><table><tr><td>albedo[0]</td><td>0</td><td>[1] 0</td><td>[2] 1</td></tr><tr><td>roughness</td><td>1</td><td></td><td></td></tr><tr><td>metalness</td><td>0.5</td><td></td><td></td></tr><tr><td>ao</td><td>0</td><td></td><td></td></tr></table></div>	viewPosition[0]	5	[1] 2	[2] 5	lightPosition[0]	0	[1] 10	[2] 5	lightColor[0]	1	[1] 1	[2] 1	objectColor[0]	1	[1] 0	[2] 0	albedo[0]	0	[1] 0	[2] 1	roughness	1			metalness	0.5			ao	0			<div><p>59.7</p></div> <div><h3>Parameters</h3><table><tr><td>viewPosition[0]</td><td>5</td><td>[1] 2</td><td>[2] 5</td></tr><tr><td>lightPosition[0]</td><td>0</td><td>[1] 10</td><td>[2] 5</td></tr><tr><td>lightColor[0]</td><td>1</td><td>[1] 1</td><td>[2] 1</td></tr><tr><td>objectColor[0]</td><td>1</td><td>[1] 0</td><td>[2] 0</td></tr></table><h3>PBR parameters (TO DO)</h3><table><tr><td>albedo[0]</td><td>0</td><td>[1] 0</td><td>[2] 1</td></tr><tr><td>roughness</td><td>2</td><td></td><td></td></tr><tr><td>metalness</td><td>1</td><td></td><td></td></tr><tr><td>ao</td><td>0</td><td></td><td></td></tr></table></div>	viewPosition[0]	5	[1] 2	[2] 5	lightPosition[0]	0	[1] 10	[2] 5	lightColor[0]	1	[1] 1	[2] 1	objectColor[0]	1	[1] 0	[2] 0	albedo[0]	0	[1] 0	[2] 1	roughness	2			metalness	1			ao	0		
viewPosition[0]	5	[1] 2	[2] 5																																																														
lightPosition[0]	0	[1] 10	[2] 5																																																														
lightColor[0]	1	[1] 1	[2] 1																																																														
objectColor[0]	1	[1] 0	[2] 0																																																														
albedo[0]	0	[1] 0	[2] 1																																																														
roughness	1																																																																
metalness	0.5																																																																
ao	0																																																																
viewPosition[0]	5	[1] 2	[2] 5																																																														
lightPosition[0]	0	[1] 10	[2] 5																																																														
lightColor[0]	1	[1] 1	[2] 1																																																														
objectColor[0]	1	[1] 0	[2] 0																																																														
albedo[0]	0	[1] 0	[2] 1																																																														
roughness	2																																																																
metalness	1																																																																
ao	0																																																																

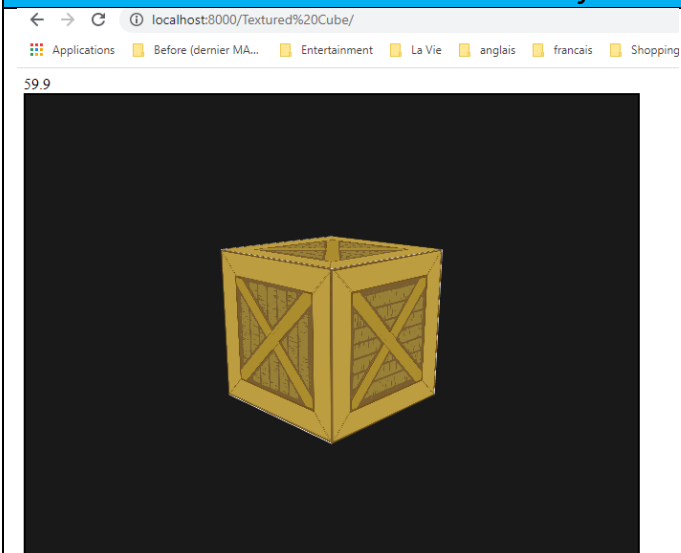
Ajout de la texture

Ajouter la texture simple sans lumières

J'ai essayé d'installer un server local chez moi, et tout d'abord appliquer la texture sur la cube sans prenant compte de la lumière. Cela donne le résultat comme l'image ci-contre.

(Méthode utilisée : stocker l'image dans la balise img caché et identifié, puis récupérer l'élément par `GetElementById`)

On voit que la texture est bien appliqué sur la cube.

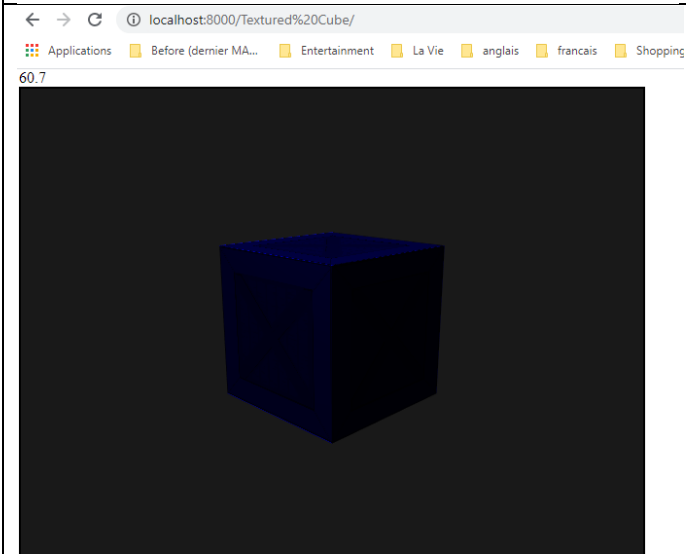


Parameters

viewPosition[0] 5 [1] 2 [2] 5
lightPosition[0] 0 [1] 10 [2] 5
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 0.5
metalness 0
ao 0

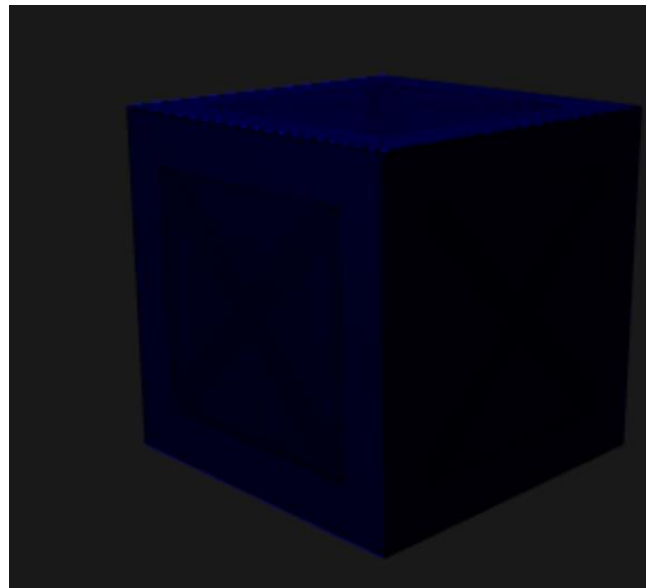


Parameters

viewPosition[0] 5 [1] 2 [2] 5
lightPosition[0] 0 [1] 10 [2] 5
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 0 [1] 0 [2] 1

PBR parameters (TO DO)

albedo[0] 0 [1] 0 [2] 1
roughness 2
metalness 0
ao 0



Ajout de la texture avec le modèle de **PHONG**

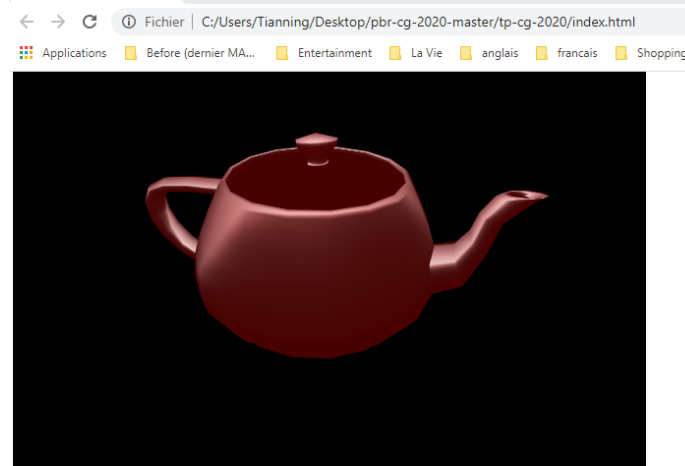
On voit la texture est bien appliquée, mais dans ce modèle, la couleur paraît très foncé, la texture n'est pas assez évident.

J'ai essayé importer le modèle de utah teapot pour visualiser le changement de effet lumière sur un maillage plus complexe (plus de triangles). Donc, quand on est sur le modèle de phong, cela nous donne le résultat comme l'image ci-contre.

On s'aperçoit que le maillage est bien importé et modèle de phong est appliqué correctement.

L'image de droite est celle de texturée, on voit bien que la texture est bien appliquée sur le maillage avec le calcul de lumière.

PBR basique

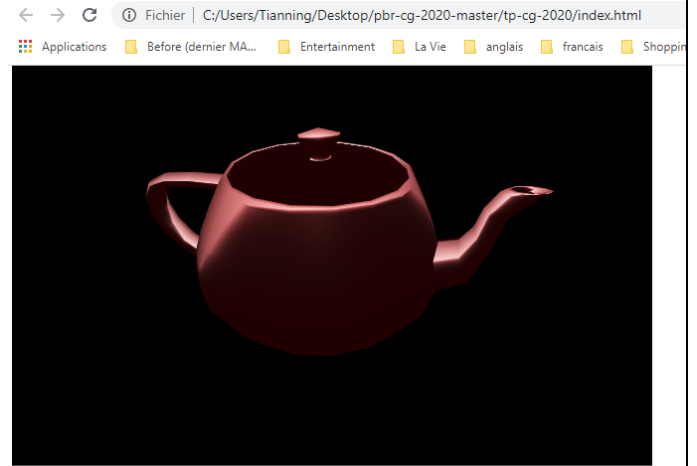


Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 10
metalness 0.5
ao 0



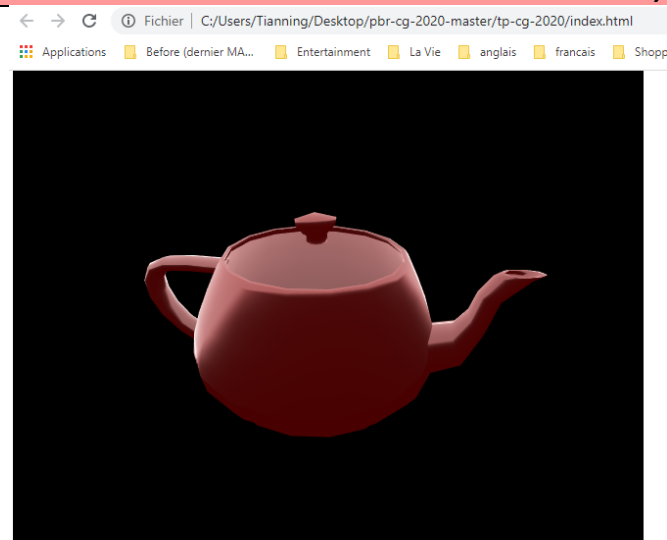
Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 20
metalness 0.8
ao 0

Amélioration avec Geometry Smith (au lieu d'utiliser Schlick)

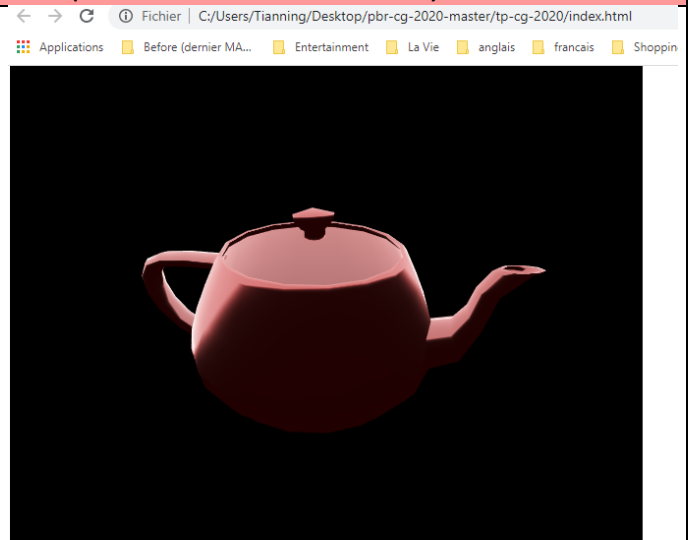


Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 10
metalness 0.5
ao 0



Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)


albedo[0] 1 [1] 0 [2] 0
roughness 20
metalness 0.8
ao 0

PBR standard en ajoutant la texture

localhost:8000/tp-cg-2020/

Applications Before (dernier MA... Entertainment La Vie anglais francais Shopping

60.1



Parameters

viewPosition[0]	10	[1] 20	[2] 40
lightPosition[0]	0	[1] 40	[2] 20
lightColor[0]	1	[1] 1	[2] 1
objectColor[0]	1	[1] 0	[2] 0


PBR parameters (TO DO)

albedo[0]	1	[1] 0	[2] 0
roughness	10		
metalness	0.5		
ao	0		

localhost:8000/tp-cg-2020/

Applications Before (dernier MA... Entertainment La Vie anglais francais Shopping

60.8



Parameters

viewPosition[0]	10	[1] 20	[2] 40
lightPosition[0]	0	[1] 40	[2] 20
lightColor[0]	1	[1] 1	[2] 1
objectColor[0]	1	[1] 0	[2] 0

PBR parameters (TO DO)

albedo[0]	1	[1] 0	[2] 0
roughness	20		
metalness	0.8		
ao	0		

2.3 Découvert et application d'un Modèle de PBR

Modèle PBR choisi : **Filament**

Filament est un moteur de PBR pour android est qui a pour but de faciliter les développeurs à créer les rendus de haute qualité en 2D ou 3D. Le travail suivant est de découvrir les équation et les théories utilisés dans filament et de choisir un modèle pour visualiser le résultat.

Comme vu dans le cours,

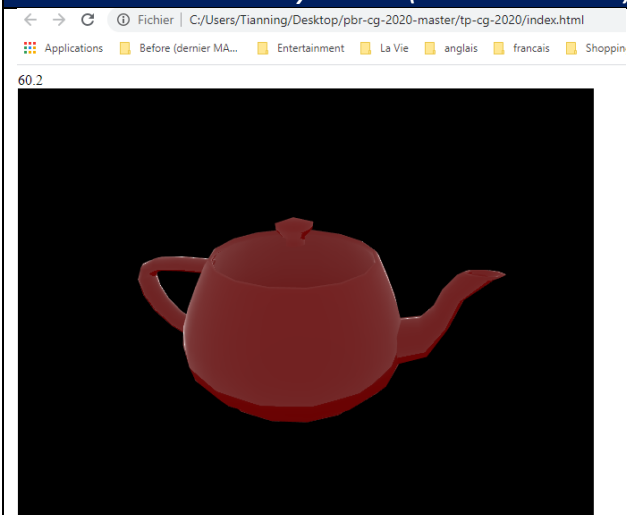
Pour calculer le *Specular BRDF*, on a besoin D (distribution), G(geometry), F(fresnel). Ici, pour préciser, dans la partie Geometrique, j'ai utilisé la fonction géométrique shadowing de Smith (Smith-GGX).

Pour la partie diffuse BRDF, j'utilise la diffuse Lambertian BRDF ($1/\pi$)

Clear coat model : specular BRDF

Le BRDF precedent est plutôt pour les objets avec des surface isotropique (constitué d'un seul couche.) il y a surement des autres matériaux avec plusieurs couches. Clear Coat est un modèle qui ajoute une couche spéculaire de plus pour présenter les objets plus réfléchissants.

Avec GeometrySmith (PBR standard)



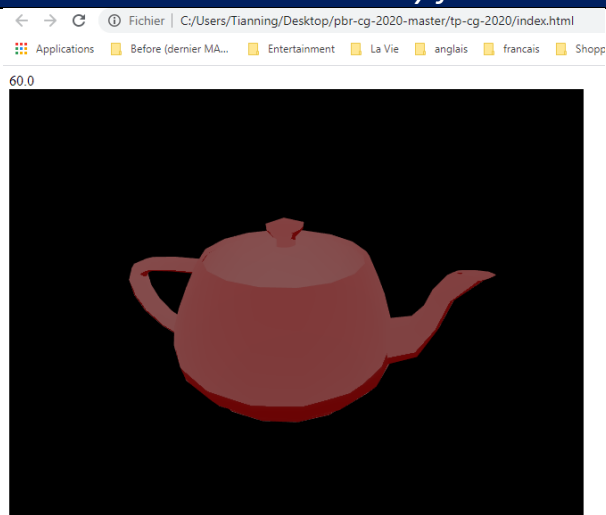
Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 1
metalness 0.2
ao 0

Avec Kelemen visibility function



Parameters

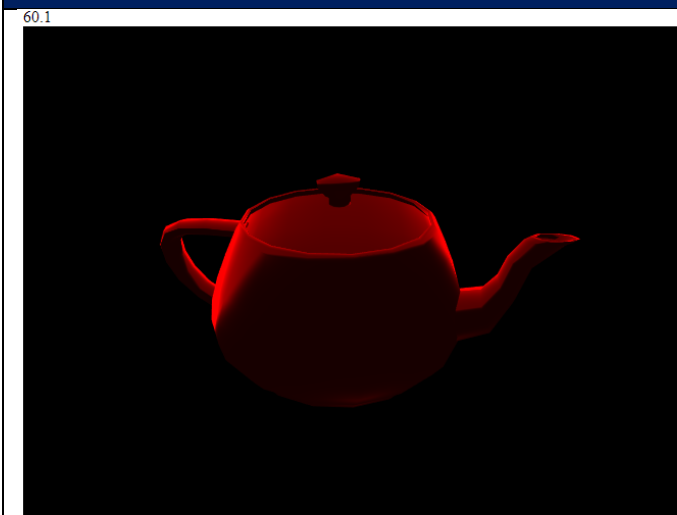
viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 1
metalness 0.2
ao 0

Selon les images ci-dessus, on voit que avec la fonction visibilité de Kelemen, la couche spéculaire est plus évident, les couleurs sont plus éclairée. Cela pourrait nous aider à réaliser ce modele de clear coat.

Amélioration en recalculant et en ajoutant la couche spéculaire

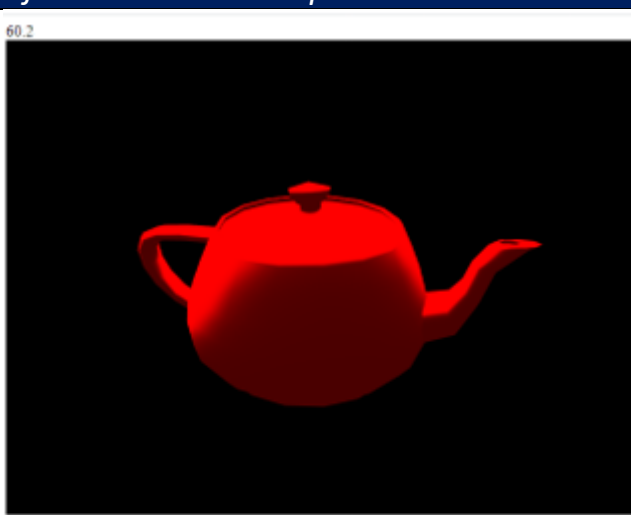


Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 10
metalness 0.5
ao 0



Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 10
metalness 0.5
ao 0

On constate que les parties réfléchissantes sont les mêmes (par rapport à PBR standard). Mais sur l'image, on s'aperçoit que les couleurs des parties diffuses et surtout celles de spéculaires sont renforcées. Cela rend le théière plus réfléchissant. (Image ci-dessus (droite) : Re-appliquer tonemapping et gamma-correction image droite)

59.7



Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

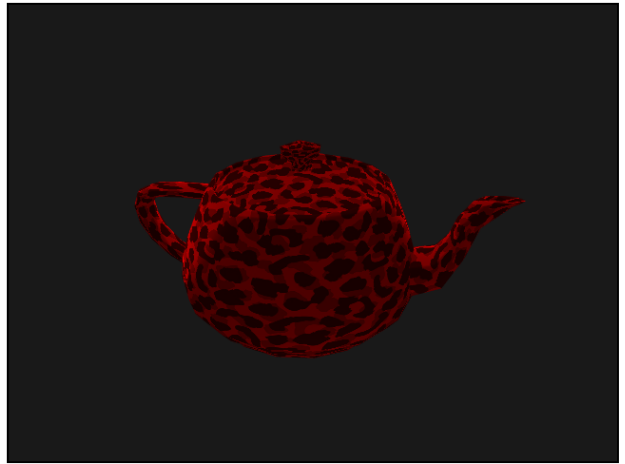
PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 10
metalness 0.8
ao 0

localhost:8000/tp-cg-2020/

Applications Before (dernier MA... Entertainment La Vie anglais francais Shopping

59.8



Parameters

viewPosition[0] 10 [1] 20 [2] 40
lightPosition[0] 0 [1] 40 [2] 20
lightColor[0] 1 [1] 1 [2] 1
objectColor[0] 1 [1] 0 [2] 0

PBR parameters (TO DO)

albedo[0] 1 [1] 0 [2] 0
roughness 10
metalness 0.5
ao 0

Après avoir ajouté la texture, on obtient le résultat comme l'image ci-dessus (à droite). On voit que la texture est bien appliqué en prenant compte les calculs de lumière.