

**University of Ferhat Abbas Setif -1**  
**FACULTY OF SCIENCES**



# **Compte rendu de projet (Partie Simulation 3D)**

**Groupe Responsable de  
la Simulation**

**IDTW G3**

# I-Explication de TP(Chimie)

## 1- Objectifs :

Vérifier le degré chlorométrique d'une eau de Javel par un titrage indirect.

## 2- Manipulation :

### Principe de titrage :

Excès d'ions iodure est ajouté à un volume connu de solution d'eau de Javel en milieu acide. Le diiode ( $I_2$ ), obtenu par réaction entre les ions hypochlorite  $ClO^-$  et iodure  $I^-$ , est ensuite titré par des ions thiosulfate  $S_2O_3^{2-}$  de concentration connue. La quantité d'ions hypochlorite s'en déduit. Ce titrage est dit «indirect» car ce n'est pas la solution d'eau de Javel que l'on titre, mais le diiode formé par l'addition d'un excès d'ions iodure sur l'eau de Javel.

### Matériels et produits :

Matériel	Produits
- Béchers - Erlenmeyers - Pipette graduée de 10 mL - Fiole jaugée de 100 mL - Eprouvette graduée - Burette graduée	- Eau de Javel de commerce - Solution d'iodure de potassium (KI) 0,1 M - Solution d'acide chlorhydrique (HCl) 1 M - Solution de thiosulfate de sodium ( $Na_2S_2O_3$ ) 0,1 M - Empois d'amidon - Eau distillée

## 3- Titrage :

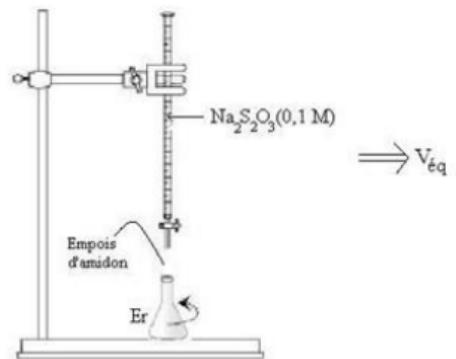
- Remplir la burette avec la solution de thiosulfate de sodium  $Na_2S_2O_3$  (0,1 M) et ajuster au zéro.

- Verser la solution de thiosulfate de sodium jusqu'à ce que la couleur de la solution contenue dans l'rlenmeyer devienne jaune pâle. Ajouter alors quelques gouttes d'empois d'amidon. Continuer à verser la solution titrante mais goutte à goutte jusqu'à la décoloration complète de la solution.

Noter le volume obtenu ( $V_{éq}$ ). Réduction du diiode:



Oxydation du thiosulfate :



Soit la réaction de titrage notée réaction 3 :



**4- Exploitation des résultats de mesure :**

- À l'équivalence, les réactifs sont introduits dans les proportions stœchiométriques données par l'équation de titrage. D'après la réaction de titrage (réaction 3), la quantité d'ions thiosulfate apportés par la solution titrante est le double de celle des molécules de diiode initialement présentes dans l'rlenmeyer, soit :  $nI_2 = n(S_2O_3^{2-})/2$ .

- D'après la réaction 2 :  **$nClO^-$  initialement présents dans le prélèvement de volume V de la solution diluée  $S'$  =  $nI_2$  formé.**

- Or  $[ClO^-](aq)S' = nClO^- \cdot V$  avec  $V$  le volume de la prise d'essai de la solution diluée  $S'$  donc :  $V = 10,0 \text{ mL} = 10^{-2} \text{ L}$ .

- La solution commerciale est  $F$  fois plus concentrée, on a alors  $[ClO^-](aq)S = F [ClO^-](aq)S'$ .

- Au cours de la fabrication de l'eau de Javel, il se produit la réaction 1 d'équation :  $Cl_2(g) + 2 (Na^+(aq) + OH^-(aq)) \rightarrow (Na^+(aq) + ClO^-(aq)) + (Na^+(aq) + Cl^-(aq)) + H_2O(l)$  (réaction 1) Ainsi, on constate qu'une mole de dichlore peut conduire à la formation d'une mole d'ions hypochlorite, soit :  $nCl_2 = nClO^-$ . Pour 1 litre de solution :  $nCl_2 = [ClO^-](aq)S$

## II-Creation et export des objets 3D



Avec l'aide de Blender comme modeleur 3d nous avons pu créer les objets nécessaires au test et faire une scène primitive avec rendu

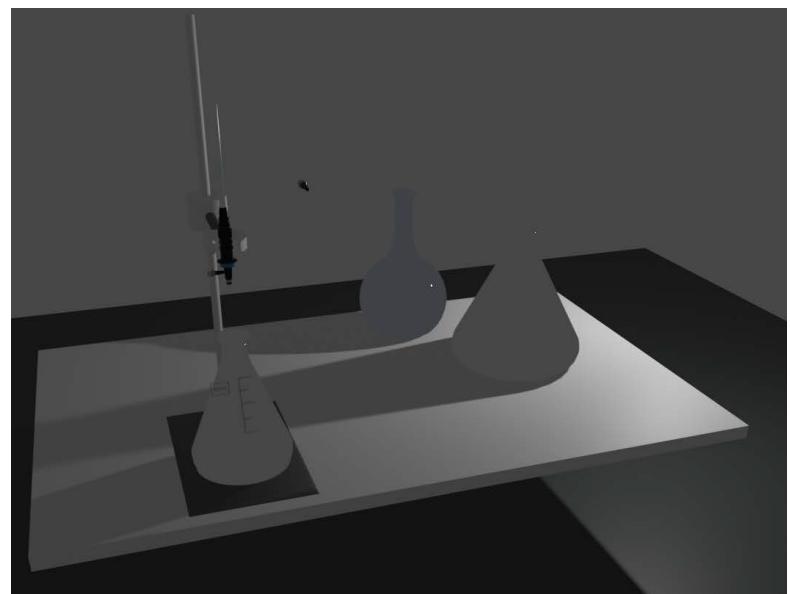


Perspective de Camera (Blender)

Après le rendu en jpeg (rendu Blender), nous avons exporté les objets créés avec l'aide de Blender sous forme de fichier avec l'extension spécifique (.gltf) dont nous aurons besoin pour notre importation au web basée sur WebGL plus tard.



Objet (.gltf) exporté



Rendu Primitive

### III-Imports des objets 3D



en utilisant html, la bibliothèque threejs (GLTFLoader) et sur la base des méthodes webGL, nous avons pu importer l'objet nécessaire dans la partie 3d de la page.

A screenshot of a terminal window and a web browser. The terminal window shows a portion of a ThreeJS script with code for setting up a renderer, loading GLTF files, and adding materials to objects. The browser window shows a 3D scene with a tall, thin gray cone on a small base, a smaller gray cylinder on a rectangular base, and a gray rectangular object with a circular top. The browser's address bar shows the URL 127.0.0.1:5500/index%20mvm.html.

```
// Set up the renderer
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

// Load the GLTF object
var loader = new THREE.GLTFLoader();
loader.load( 'object6.gltf', function ( gltf ) {
    // Add a material to the object.
    object = gltf.scene;
    gltf.scene.traverse(function (child) {
        if (child.isMesh) {
            child.material = new THREE.MeshStandardMaterial({color: 0xffffffff});
        }
    });
    object.position.setX(-4);
    object.position.setY(-0.01);
    object.position.setZ(4);
    object.rotation.z = 0;
    object.rotation.y = -2;
    // o.material.color.set(red);
    //-----GRID-HELPER-----
    var size = 10;
    var divisions = 10;
    var gridHelper = new THREE.GridHelper( size, divisions );
    scene.add( gridHelper );
    //-----GRID-HELPER-----
    object.scale.set(0.1, 0.1, 0.1);
    object.traverse(function(child) {
        if (child.isMesh) child.material.envMap = envMap;
    });
    scene.add( gltf.scene );
    animate();
});
```

ThreeJS Script pour l'importation

Nous avons importé l'objet en utilisant GLTFLoader() à partir de trois js et nous suivons les instructions nécessaires pour implémenter un objet 3d avec webGL ,

puis nous avons fait l'animation.

A screenshot of a code editor showing a portion of the ThreeJS script. It includes the 'animate' function which calls requestAnimationFrame and renderer.render. The code is highlighted in blue and yellow.

```
// Animate the scene
function animate() {
    requestAnimationFrame( animate );
    renderer.render( scene, camera );
}
animate();
```

Animation

A screenshot of a code editor showing the full ThreeJS script. It includes the setup for the renderer, loading of the GLTF file, traversing the scene to add materials, and defining the 'animate' function. The code is highlighted in blue and yellow.

```
// Set up the renderer
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

// Load the GLTF object
var loader = new THREE.GLTFLoader();
loader.load( 'object6.gltf', function ( gltf ) {
    // Add a material to the object.
    object = gltf.scene;
    gltf.scene.traverse(function (child) {
        if (child.isMesh) {
            child.material = new THREE.MeshStandardMaterial({color: 0xffffffff});
        }
    });
    object.position.setX(-4);
    object.position.setY(-0.01);
    object.position.setZ(4);
    object.rotation.z = 0;
    object.rotation.y = -2;
```

Objet

## IV-Interactions avec les objets 3D

En utilisant ThreeJS :



```
// Animate the scene
function animate() {
  requestAnimationFrame( animate );
  renderer.render( scene, camera );
}
animate();

// Add mouse control
var isDragging = false;
var previous.mousePosition = {
  x: 0,
  y: 0
};

document.addEventListener('mousedown', function(event) {
  isDragging = true;
});
document.addEventListener('mousemove', function(event) {
  // Make sure the object is selected
  if (!isDragging) {
    return;
  }
  var deltaMove = {
    x: event.clientX-previous.mousePosition.x,
    y: event.clientY-previous.mousePosition.y
  };

  object.position.x += deltaMove.x*0.01;
  object.position.y -= deltaMove.y*0.01;

  previous.mousePosition = {
    x: event.clientX,
    y: event.clientY
  };
});
document.addEventListener('mouseup', function(event) {
  isDragging = false;
});

</script>
</body>
</html>
```

## L'interaction avec le mouvement de la souris

En ajoutant cet ensemble de lignes à l'animation, nous pourrons interagir/déplacer l'objet sélectionné avec la souris

```
111
112     // Add mouse control
113     var isDragging = false;
114     var previousMousePosition = {
115         x: 0,
116         y: 0
117     };
118
119     document.addEventListener('mousedown', function(event) {
120         isDragging = true;
121     });
122     document.addEventListener('mousemove', function(event) {
123         // Make sure the object is selected
124         if (!isDragging) {
125             return;
126         }
127         var deltaMove = {
128             x: event.clientX-previousMousePosition.x,
129             y: event.clientY-previousMousePosition.y
130         };
131     });
132 }
```

```
130     };
131
132     object.position.x += deltaMove.x*0.01;
133     object.position.y -= deltaMove.y*0.01;
134
135     previous.mousePosition = {
136         x: event.clientX,
137         y: event.clientY
138     };
139 });
140 document.addEventListener('mouseup', function(event) {
141     isDragging = false;
142 });
```