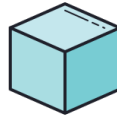


Manuel utilisateur

Moteur 3D - 7Physics



Équipe 3 : Noa AMMIRATI, Fanny DELNONDEDIEU, Quentin GENDARME, Pierre LOTTE, Théo PIROUELLE, Éléa TURC



ENSEEIHT
Département Sciences du Numérique
1APP SN 2020-2021

1	Introduction	2
2	Installation/Utilisation de l'application	3
3	Présentation générale de l'application	4
4	Manipuler des objets 3D	5
4.1	Ajouter un objet 3D	5
4.2	Supprimer un objet 3D	5
5	Gestion de la caméra	6
5.1	Gestion du zoom	6
5.1.1	Zoomer	6
5.1.2	Dezoomer	6
5.2	Se déplacer sur la scène	7
5.2.1	Déplacements basiques	7
5.2.2	Déplacement vertical	7
5.3	Tourner la caméra	7
6	Animer la scène 3D	8
6.1	Ajouter des forces	8
6.2	Supprimer des forces	9
6.3	Lancer la simulation	9
6.4	Arrêter la simulation	9
7	Importer et exporter un projet 3D	10
7.1	Récupérer un projet 3D	10
7.2	Sauvegarder un projet 3D	11
8	Utilisation de l'API	13
8.1	Exemple d'utilisation	14
8.1.1	Résultat	15

1. Introduction

L'objectif de ce manuel est de donner à l'utilisateur les bases nécessaires à la bonne utilisation de l'application. L'application représente un moteur 3D permettant de réaliser des simulations de notions de physique élémentaires. Un moteur 3D est un ensemble de fonctions permettant de représenter des objets dans un monde 3D, permettant de les manipuler et de les afficher. Un autre objectif d'un moteur 3D est la gestion des interactions avec l'environnement (forces appliquées).

Les principales fonctionnalités du moteur 3D sont :

- L'affichage de formes 3D prédéfinies et personnalisables (cube, sphere, pyramide, cylindre, etc.)
- L'utilisation de la caméra (permettant de changer d'angles de vue)
- L'animation des différentes formes contenues dans la scène 3D.

2. Installation/Utilisation de l'application

Pour pouvoir utiliser l'application, vous devez avoir la version Java8 installée. Pour installer cette version, référez-vous aux instructions suivantes : [Installation Java8](#)

Par la suite, pour lancer l'application, il faut exécuter le fichier 7Physics.jar avec la commande suivante :

- Pour Linux : `java -jar 7Physics.jar`
- Pour Windows : double-click sur le fichier 7Physics.jar
- Pour MacOS : `java -jar 7Physics.jar`



Si une version ultérieure de Java est installée, le programme pourrait ne pas se lancer. Si tel est le cas, il sera nécessaire de supprimer les versions plus récentes présentes sur le système ou bien de forcer l'utilisation de Java8.

3. Présentation générale de l'application

Voici l'interface de l'application :

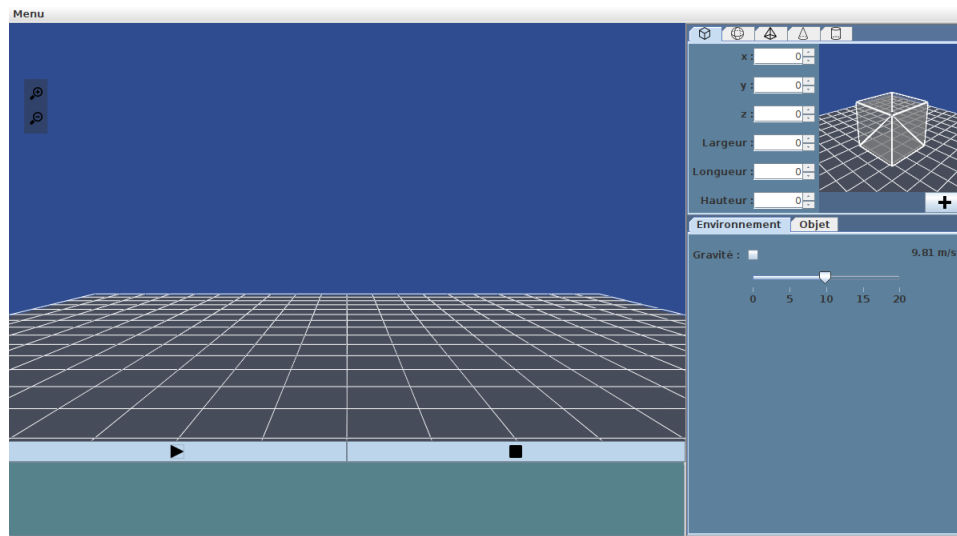


Figure 3.1 – Interface de l'application

L'application est structurée en 3 parties :

- La partie centrale représente la scène 3D : c'est ici que l'utilisateur va visualiser les objets 3D ainsi que les interactions.
- La partie située au-dessous de la scène 3D permet d'interagir avec les objets en lançant une simulation (bouton play). Elle permet aussi d'afficher la liste des objets présents sur scène.
- La partie de droite permet de paramétrer la scène : ajout d'un cube, ajout d'une sphère, activation de la gravité, ajout de force sur les différents objets, suppression d'un objet.


Dans la suite de ce manuel, vous trouverez l'explication des différentes fonctionnalités fournies par l'application vous permettant alors la prise en main de celle-ci.

4. Manipuler des objets 3D

4.1 Ajouter un objet 3D

Pour pouvoir ajouter un objet 3D sur la scène 3D, il faut sur la partie supérieur droite de l'interface, sélectionner la forme souhaitée.



Pour afficher un **cube** sur la scène 3D, il faut sélectionner le bouton :  et remplir les informations demandées.

Pour afficher d'autres formes, il suffit de sélectionner les boutons adjacents.

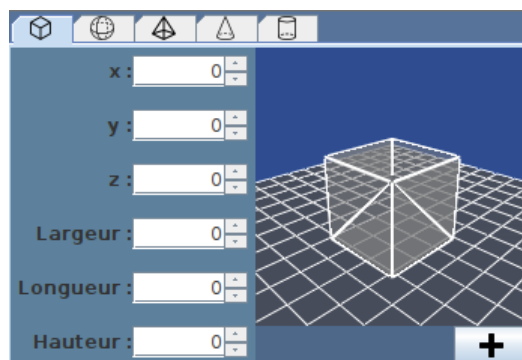


Figure 4.1 – Ajout d'une forme

4.2 Supprimer un objet 3D


Afin de supprimer un objet 3D, il faut tout d'abord le sélectionner dans la partie inférieure de l'interface en cliquant sur un des boutons présent. Une fois cela fait, rendez-vous dans l'onglet « Objet » en bas à droite de l'interface. Vous n'avez plus qu'à cliquer sur le bouton supprimer : .



Figure 4.2 – Suppression d'une forme

5. Gestion de la caméra



Afin de manipuler la caméra au clavier il est important de cliquer sur la scène en premier lieu. Sans cela, la caméra ne bougera pas.

Afin d’observer la scène sous plusieurs angles, il est possible de manipuler la caméra. Les opérations possibles sont :

- Zoomer/Dézoomer
- Déplacements basiques (Gauche, droite, avant, arrière)
- Se déplacer verticalement
- Faire tourner la caméra.

5.1 Gestion du zoom

Lors de la visualisation de la scène que vous avez créé, il vous est possible de régler le niveau de zoom utilisé afin de voir un objet plus en détail ou bien encore de voir une scène de taille importante en intégralité. Afin de réaliser ces actions, référez-vous aux explications suivantes.



Attention, la gestion du zoom ne peut s’effectuer qu’à la souris.

5.1.1 Zoomer

Il existe 2 possibilités pour zoomer sur la scène.

- La première est d’utiliser le bouton prévu à cet effet présent sur la gauche de votre écran représenté par ce symbole :
- La seconde est d’utiliser la molette de votre souris en la faisant glisser vers le haut.

5.1.2 Dezoomer

Il existe 2 possibilités pour dezoomer la scène.

- La première est d’utiliser le bouton prévu à cet effet présent sur la gauche de votre écran représenté par ce symbole :
- La seconde est d’utiliser la molette de votre souris en la faisant glisser vers le bas.



Afin de mieux maîtriser votre environnement et de bénéficier d’une précision accrue, veuillez privilégier l’utilisation de la molette. En effet, les boutons latéraux sont limités à une manipulation prédéfinie du zoom et ne sont pas réglables.

5.2 Se déplacer sur la scène

Afin de vous balader au sein de la scène pour visualiser celle-ci sous différents points de vue, il vous est possible de déplacer la caméra. Vous pourrez ainsi déplacer la caméra sur 3 axes différents que nous allons voir.



Attention, les déplacements sur la scène ne peuvent être effectués que via le clavier et les raccourcis présentés.

5.2.1 Déplacements basiques

Les premiers déplacements que nous allons voir sont les déplacements basiques gauche, droite, avant et arrière. Ces déplacements se feront à l'aide des touches ZQSD du clavier.

- **Z** : Avancer.
- **Q** : Se déplacer vers la gauche.
- **S** : Reculer.
- **D** : Se déplacer vers la droite.

5.2.2 Déplacement vertical

Ensuite, il peut être utile de bouger la caméra verticalement. Pour cela, nous utiliserons les touches Espace et Shift.

- **ESPACE** : Monter la caméra.
- **SHIFT** : Descendre la caméra.

5.3 Tourner la caméra

Enfin, pour finir, le dernier mouvement possible est la rotation de la caméra. Pour cela, il suffit simplement de cliquer sur la scène 3D puis de glisser la souris vers l'endroit où vous souhaitez tourner la caméra en maintenant le clic de la souris.



Attention, la rotation de la caméra ne peut se faire que via la souris.

6. Animer la scène 3D

Une fois les objets ajoutés à la scène, il est possible de leur appliquer des forces personnalisées puis de lancer la simulation qui prendra donc en compte ces forces pour déplacer les objets de la scène.

6.1 Ajouter des forces

La première étape est donc d'ajouter des forces. Pour cela, il est nécessaire de préselectionner la forme souhaitée dans la partie inférieure de l'interface en cliquant sur un des boutons semblable au bouton vert de la figure suivante :

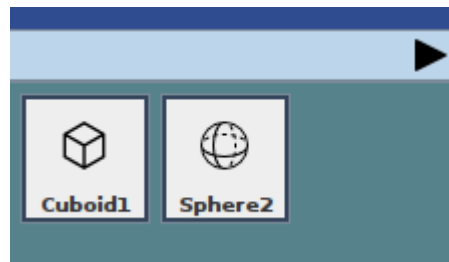


Figure 6.1 – Sélection d'une forme

Une fois la forme sélectionnée, la partie inférieure droite de l'interface, dans l'onglet « Objet », sera mise à jour afin d'afficher les informations liées à l'objet ajouté. Il est alors possible d'ajouter des forces en cliquant sur le bouton + disponible en haut de la table intitulée « Forces ».

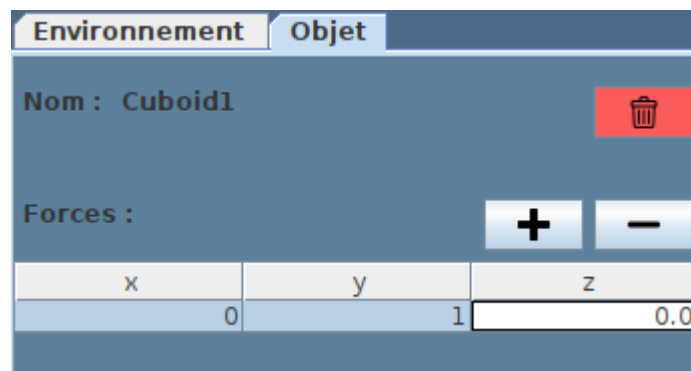


Figure 6.2 – Détails d'une forme

Une fois cela fait, et dès que vous changerez les valeurs de la ligne qui s'est ajoutée à la table, la force sera exercée sur l'objet.

6.2 Supprimer des forces

Afin de supprimer des forces, rendez-vous dans les détails de l'objet et cliquez sur la ligne contenant la force à supprimer. Une fois cela fait, cliquez sur le bouton - de la table intitulée « Forces ».

6.3 Lancer la simulation

Afin de lancer la simulation, vous pouvez simplement appuyer sur le bouton Play disponible dans la partie inférieure de l'interface.



Figure 6.3 – Lancement de la simulation

6.4 Arrêter la simulation

Afin d'arrêter la simulation, vous pouvez simplement appuyer sur le bouton Stop disponible dans la partie inférieure de l'interface.



Figure 6.4 – Arrêt de la simulation

7. Importer et exporter un projet 3D

Il est possible d'enregistrer un projet 3D en cours sous forme d'un fichier pour le continuer plus tard à l'aide de la fonctionnalité « importer un projet 3D ». Cet import peut aussi être fait à partir de fichiers créés par d'autre moteur 3D.

7.1 Récupérer un projet 3D

Pour importer un projet, il suffit d'ouvrir le menu déroulant en haut à gauche de l'écran et de sélectionner « Importer un projet ». Une fenêtre de navigation de fichier s'ouvrira et vous pourrez ainsi choisir le fichier à importer. Il est possible qu'une erreur apparaisse lors de l'importation pour cause d'une extension de fichier non valide (extension différente de « .obj »).

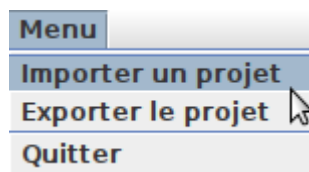


Figure 7.1 – Sélection de l'importation sur le menu déroulant

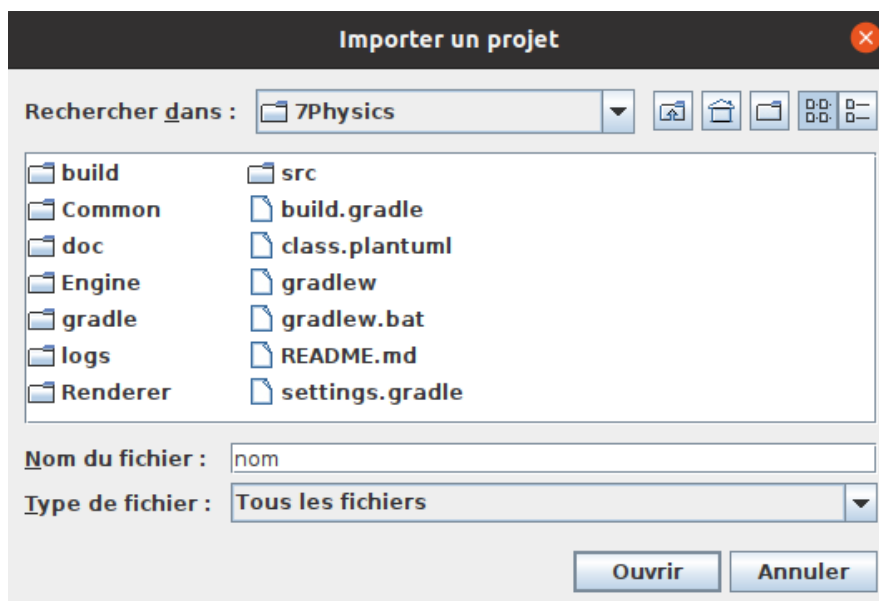


Figure 7.2 – Navigateur de fichier permettant de sélectionner un fichier

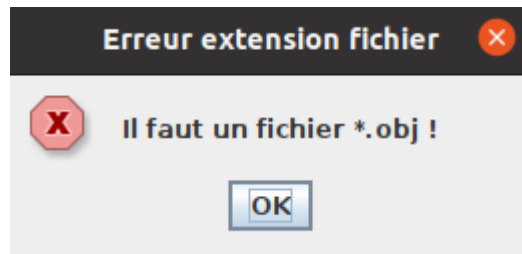


Figure 7.3 – Erreur lors de l’importation d’un fichier

7.2 Sauvegarder un projet 3D

De la même manière, pour exporter le projet en cours et créer un fichier **.obj* de celui-ci, on utilise le menu déroulant. En sélectionnant « Exporter un projet », une fenêtre s’ouvrira pour choisir l’emplacement et le nom du fichier à enregistrer. Il est possible qu’une erreur apparaisse lors de l’exportation pour cause de scène 3D vide (pas de formes 3D présentes sur la scène).

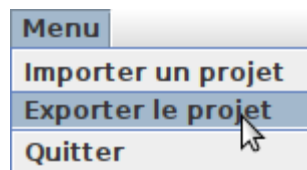


Figure 7.4 – Sélection de l’exportation sur le menu déroulant

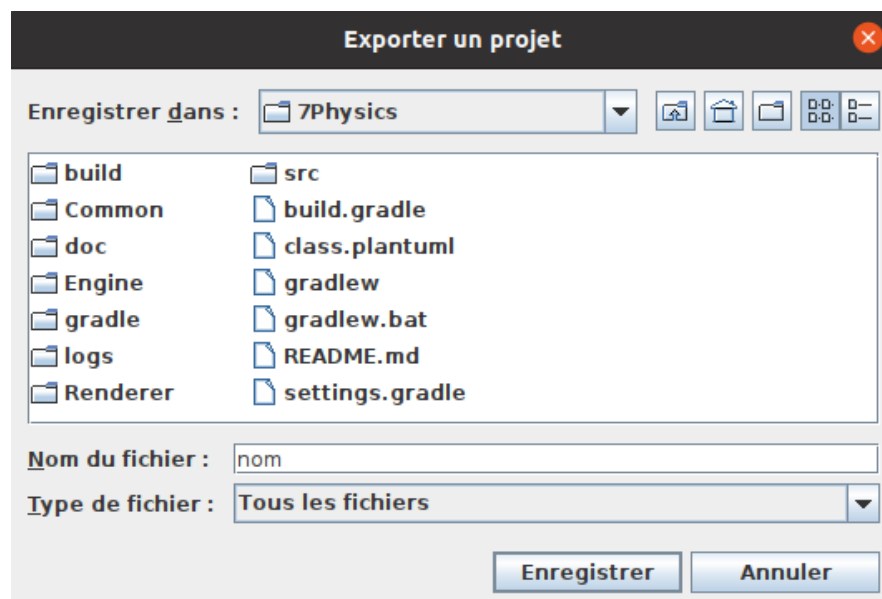


Figure 7.5 – Navigateur de fichier permettant de sauvegarder un fichier

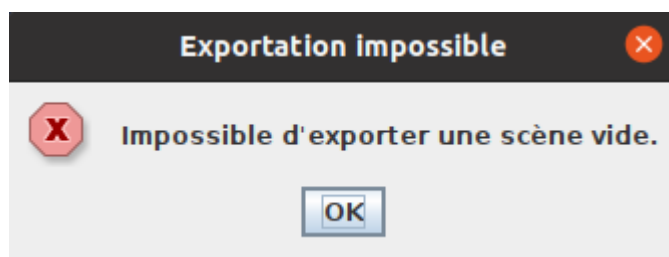


Figure 7.6 – Erreur lors de l’exportation d’un fichier

8. Utilisation de l'API

Pour utiliser les fonctionnalités de 7Physics il est nécessaire de tirer ses dépendances. Pour ce faire, il est nécessaire d'installer le plugin [JitPack](#) à votre gestionnaire de dépendances.

Exemple avec *Gradle* :

```
repositories {  
    mavenCentral()  
    maven {  
        url 'https://jitpack.io'  
        metadataSources {  
            artifact() //Look directly for artifact  
        }  
    }  
}
```

Vous pourrez ensuite ajouter les dépendances de 7Physics à votre projet :

```
dependencies {  
    implementation 'com.github.7Physics:Common:master-SNAPSHOT'  
    // Si vous souhaitez utiliser le moteur de rendu :  
    implementation group: 'org.jogamp.jogl', name: 'jogl-all-main', version: '2.3.2'  
    implementation group: 'org.jogamp.gluegen', name: 'gluegen-rt-main', version: '2.3.2'  
    implementation 'com.github.7Physics:Renderer:master-SNAPSHOT'  
    // Si vous souhaitez utiliser le moteur physique :  
    implementation 'com.github.7Physics:Engine:master-SNAPSHOT'  
}
```



De part la façon dont a été structuré 7Physics il est possible d'utiliser le moteur de rendu avec un moteur physique différent, et inversement.

8.1 Exemple d'utilisation

```

public static void main(String... args) {
    JFrame frame = new JFrame("Exemple 7Physics");

    Camera camera = new Camera(new Position(-3, 1, 0));
    camera.rotateVertical(-Math.PI/8);

    // On crée la scène 3D
    Scene3D scene = new Scene3D(camera);
    // C'est un panel comme les autres, on peut lui modifier ses dimensions et
    // l'ajouter à un panel
    scene.setPreferredSize(new Dimension(1000, 600));
    frame.add(scene);

    frame.pack();

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);

    // On crée le world qui contiendra les objets physiques
    World world = new World();
    world.setGravity(new Vec3(0, -1, 0));

    Position wallPosition = new Position(0, .15, -1);

    scene.addObject(new Cuboid(3, .1, 2))
        .setPosition(wallPosition);
    world.addPhysicObject(new Cuboid(3, 2, .1))
        .setPosition(wallPosition)
        .setDynamic(false);

    Random r = new Random();
    r.nextFloat();

    for(int i = 0; i < 100; i++) {
        // Position partagée entre l'objet graphique et l'objet physique :
        // fait le lien entre eux.
        Position spherePosition = new Position(r.nextFloat()*6-3,
            (i+2)/3f,
            r.nextFloat()*6-3);

        // On ajoute la sphère graphique
        scene.addObject(new Sphere(.1, 3))
            .setPosition(spherePosition)
            .setColor(Color.ORANGE, Color.RED);
        // On ajoute la sphère physique
        world.addPhysicObject(new Sphere(.1, 3))

```

```
.setPosition(spherePosition)
.addForce(new Vec3(r.nextFloat()*4-2, 0, r.nextFloat()*4-2));
}

// On lance la simulation à 60 mise à jour par seconde
world.startStepLoop(60);
}
```

8.1.1 Résultat

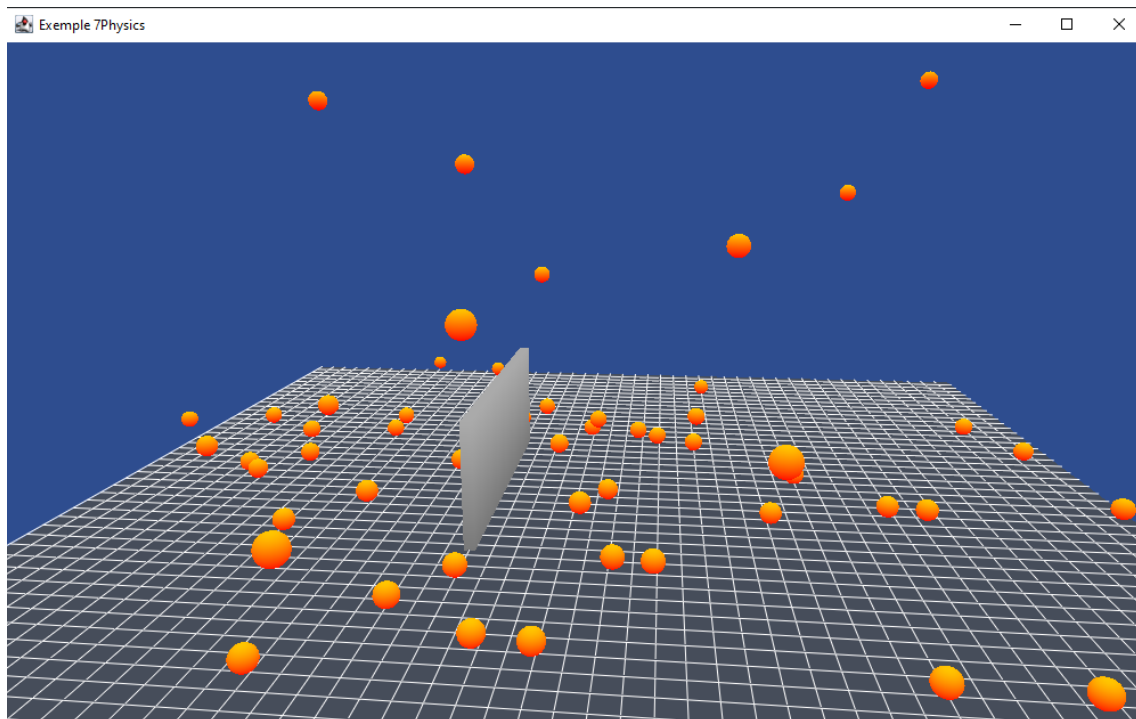


Figure 8.1 – Rendu obtenu à partir du code précédent