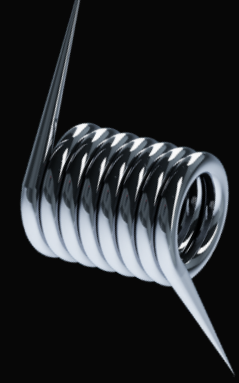


Projet Moteur de Jeux

Marius Jenin Alexandre Lanvin



1 / 1



Sommaire

- Graphe de Scene
- Moteur physique
 - Mouvement
 - Collisions
 - Comportements
- Rendu
 - Blinn-Phong
 - Shadow mapping
- Simulation
 - Scenes

Graphe de scène

3 / 1

NodeSG

- Nœud parent
- Nœuds enfants
- Transformation globale
- Transformation locale



Moteur physique

4 / 1

PhysicsSystem

- RigidBodyVolumes
- Corrections

Moteur physique

4 / 1



PhysicsSystem

- RigidBodyVolumes
- Corrections

RigidBodyVolume

- Liste de Comportements
- Nœud du graphe
 - Position
 - Bounding box

Moteur physique

4 / 1

PhysicsSystem

- RigidbodyVolumes
- Corrections

RigidbodyVolume

- Liste de Comportements
- Nœud du graphe
 - Position
 - Bounding box

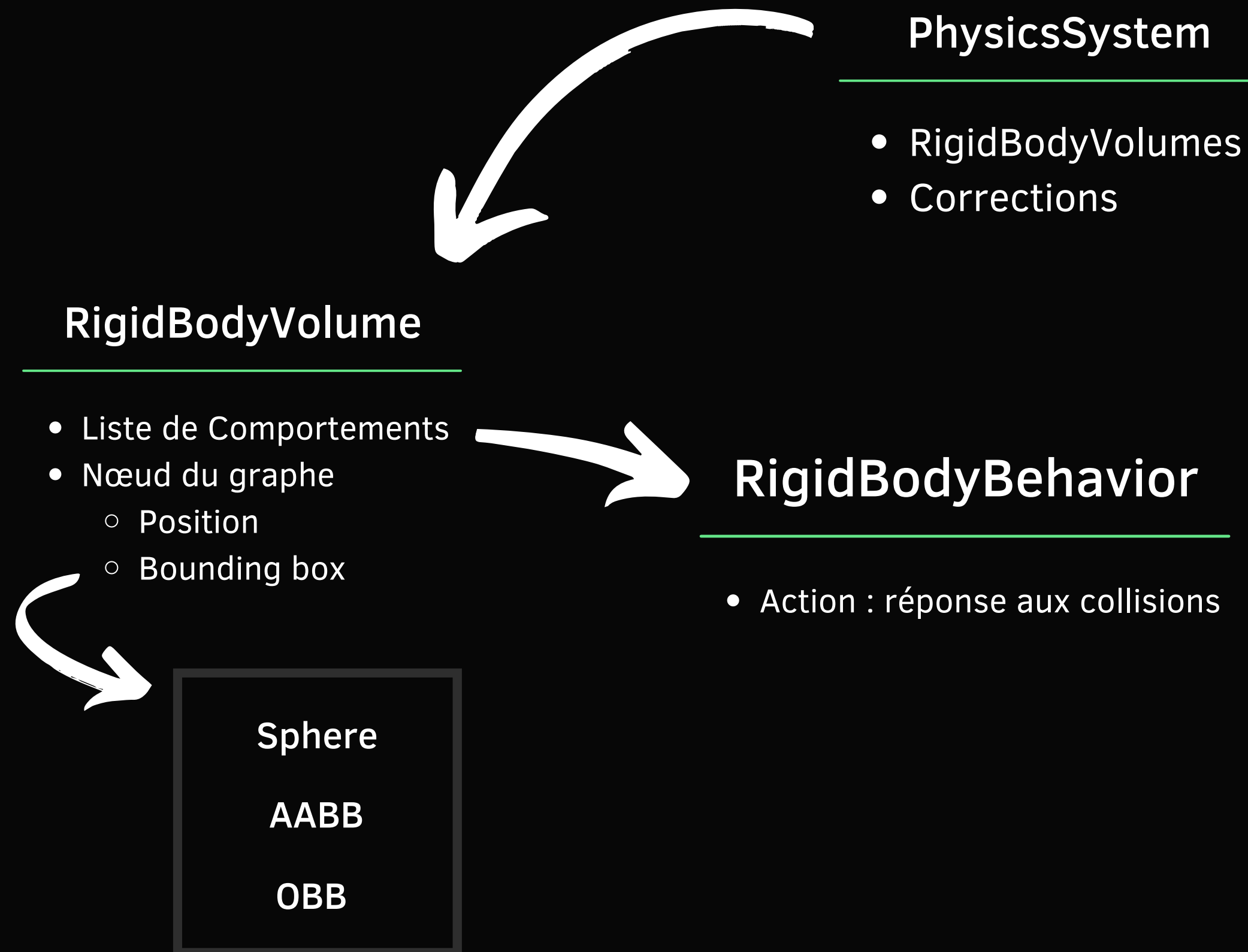
Sphere

AABB

OBB

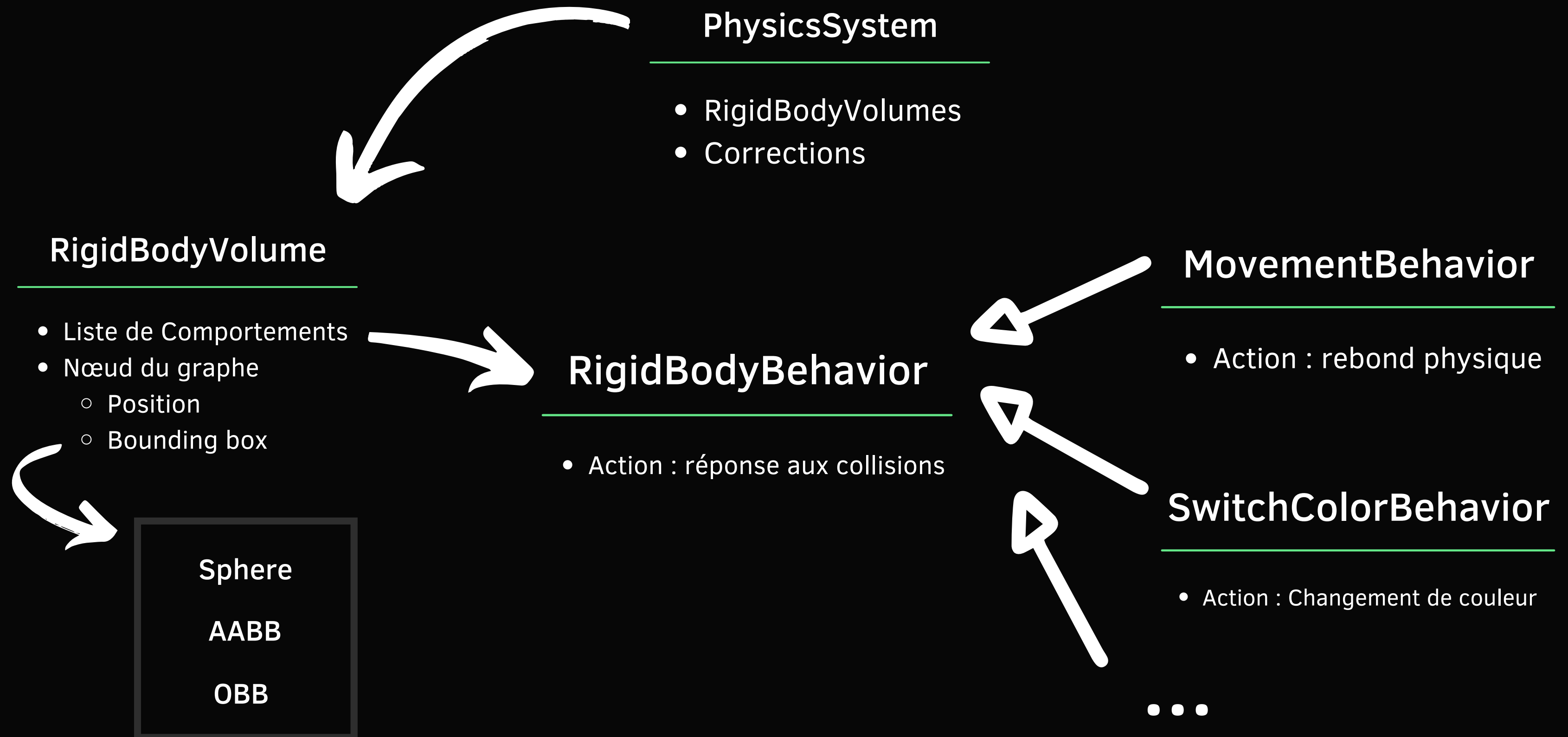
Moteur physique

4 / 1



Moteur physique

4/1



Moteur physique

Mouvement

5 / 1

PhysicsSystem

Mis à jour indépendamment du
taux de rafraîchissement.

Moteur physique

Mouvement

PhysicsSystem

Mis à jour indépendamment du
taux de rafraîchissement.

Différentes intégrations

A toutes les frames on met à
jour les positions des Rigidbody
en fonction de l'intégration
choisie paramétrable

Moteur physique

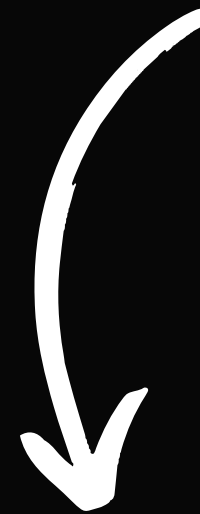
Mouvement

PhysicsSystem

Mis à jour indépendamment du
taux de rafraîchissement.

Différentes intégrations

A toutes les frames on met à
jour les positions des Rigidbody
en fonction de l'intégration
choisie paramétrable



Euler

Application de la vitesse
à chaque frame

Moteur physique

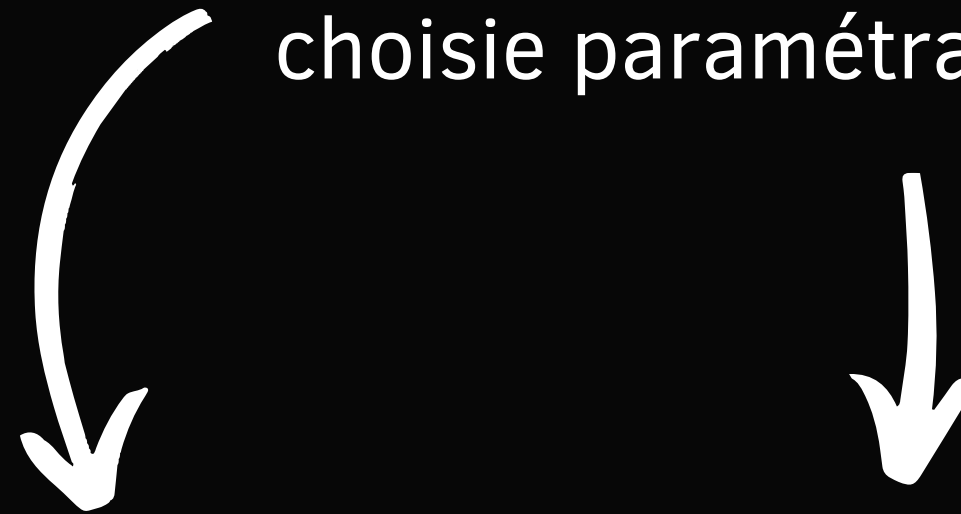
Mouvement

PhysicsSystem

Mis à jour indépendamment du taux de rafraîchissement.

Différentes intégrations

A toutes les frames on met à jour les positions des Rigidbody en fonction de l'intégration choisie paramétrable



Euler

Application de la vitesse à chaque frame

Verlet

Application de la vitesse moyenne entre 2 frames

Moteur physique

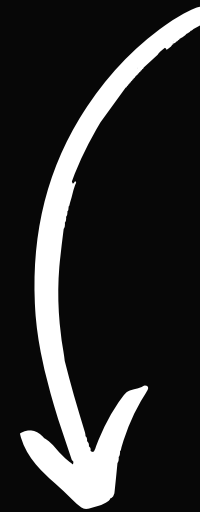
Mouvement

PhysicsSystem

Mis à jour indépendamment du taux de rafraîchissement.

Différentes intégrations

A toutes les frames on met à jour les positions des Rigidbody en fonction de l'intégration choisie paramétrable



Euler

Application de la vitesse à chaque frame



Verlet

Application de la vitesse moyenne entre 2 frames



Runge-Kutta Classique

Application de la vitesse en appliquant l'approximation de Runge-Kutta en 4 itérations

Moteur physique

Collisions

Collisions

- Collisions entre les boites englobantes
- Impulsions pour résoudre ces collisions
- Correction avec une projection linéaire

```
struct Collision {  
    bool colliding;  
    glm::vec3 normal;  
    float depth;  
    std::vector<glm::vec3> contacts;  
    RigidBodyVolume* rigid_body_1{};  
    RigidBodyVolume* rigid_body_2{};
```

Moteur physique

Comportements

7 / 1

RigidBodyBehavior

- Action
- update_physics
- update_render

Moteur physique

Comportements

RigidBodyBehavior

- Action
- update_physics
- update_render



Action exécutée lors d'une collision avec un autre RigidBodyVolume

Moteur physique

Comportements

RigidBodyBehavior

- Action
- update_physics
- update_render



Action exécutée lors d'une collision avec un autre RigidBodyVolume

Appelé à chaque mise à jour du système physique.

Moteur physique

Comportements

RigidBodyBehavior

- Action
- update_physics
- update_render



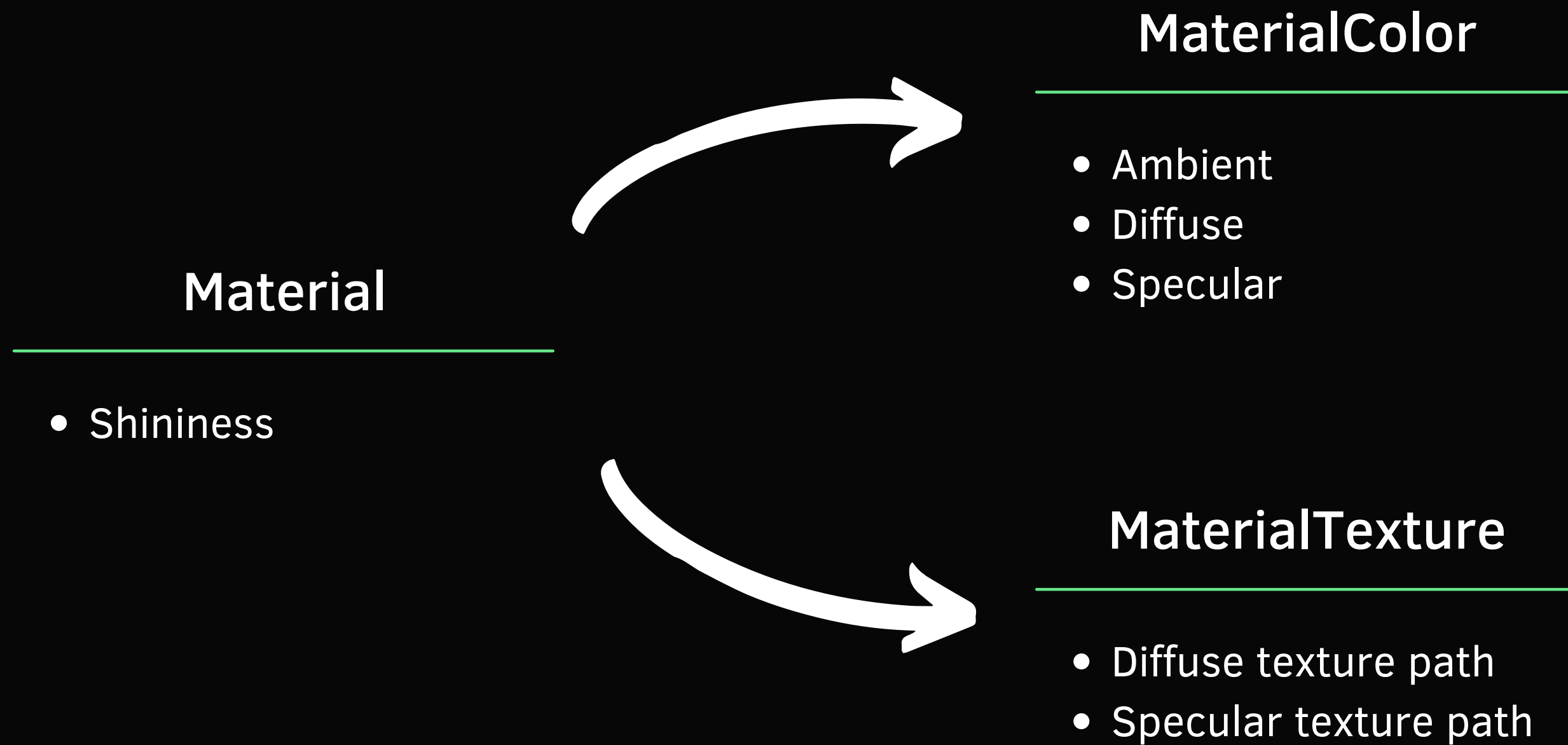
Action exécutée lors d'une collision avec un autre RigidBodyVolume

Appelé à chaque mise à jour du système physique.

Appelé à chaque frame

Rendu

8 / 1

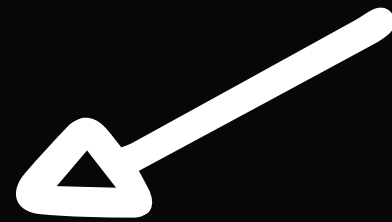


Rendu

9 / 1

Light

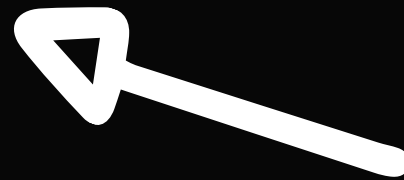
- Liste de comportements



PositionLight



DirectionLight

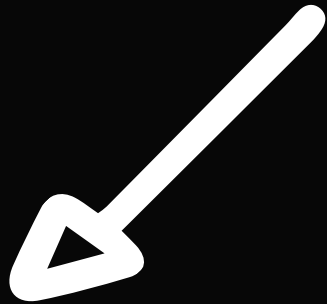


SpotLight

Rendu

10
1

LightBehavior



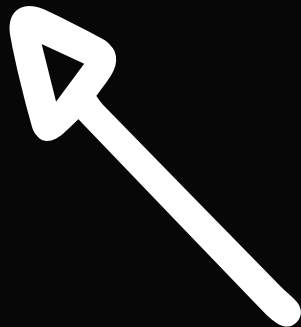
PositionLightBehavior

- Position
- Attenuation



DirectionLightBehavior

- Direction



SpotLightBehavior

- Angles du spot

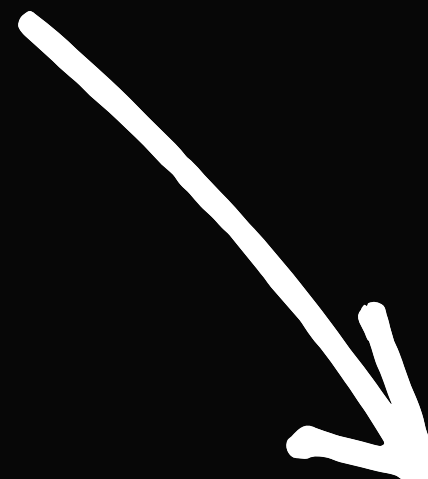
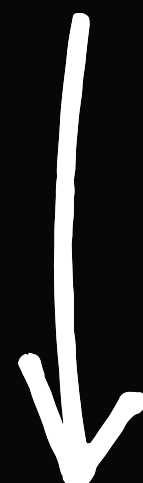
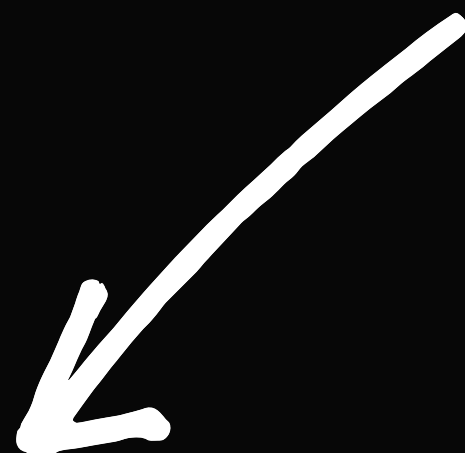
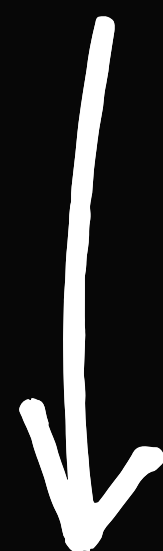
Rendu

11
1

PositionLight

SpotLight

DirectionLight



PositionLightBehavior

- Position
- Attenuation

SpotLightBehavior

- Angles du spot

DirectionLightBehavior

- Direction

Rendu

Blinn-Phong

Blinn-Phong

Nous utilisons ce modèle d'éclairage en ajoutant de l'atténuation en fonction de la distance à la lumière de l'objet

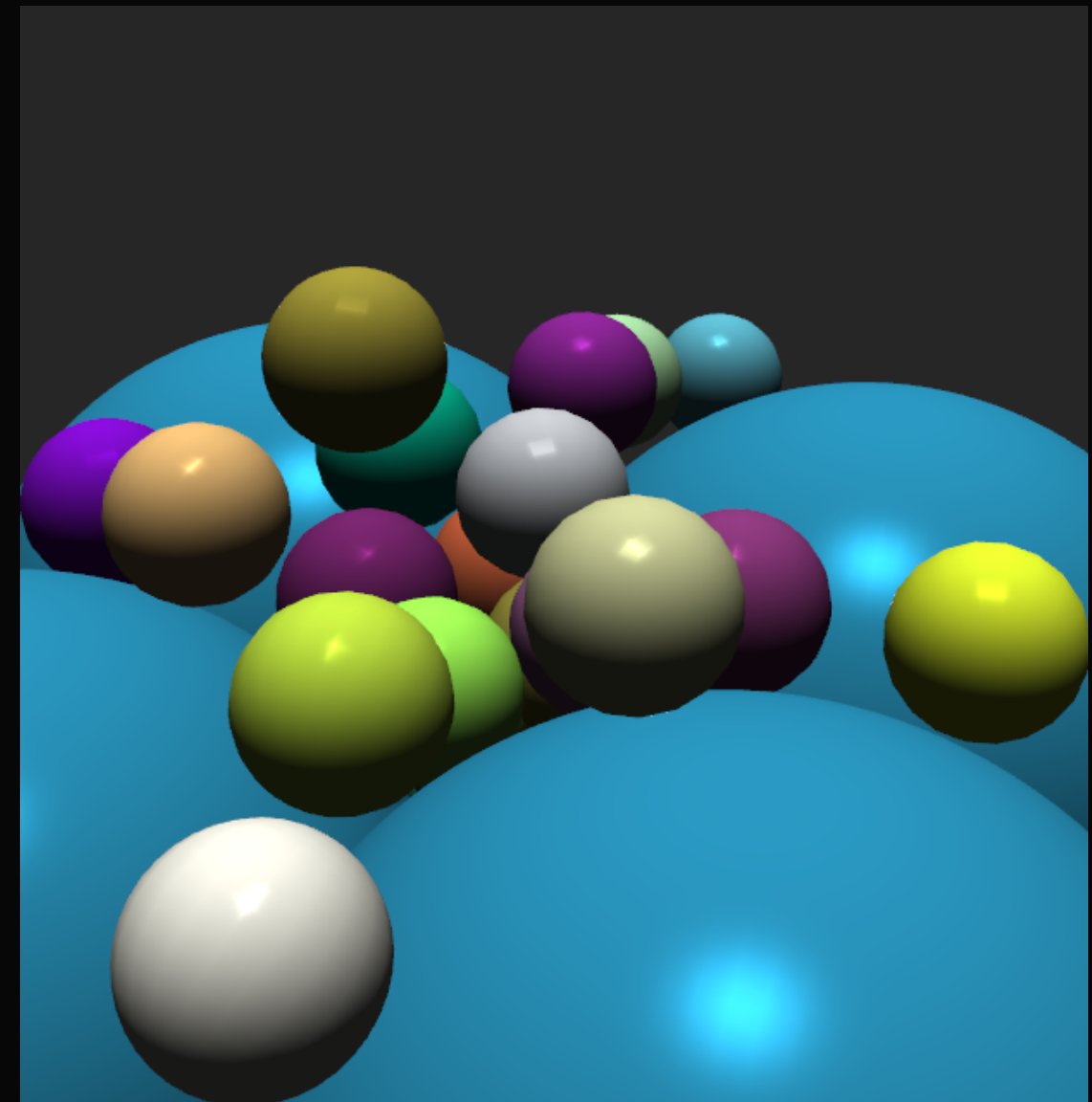
Rendu

Blinn-Phong

Blinn-Phong

Nous utilisons ce modèle d'éclairage en ajoutant de l'atténuation en fonction de la distance à la lumière de l'objet

La couleur des objets est la somme de l'éclairage calculé par ce modèle pour toutes les lumières de la scène



Rendu

Shadow maps

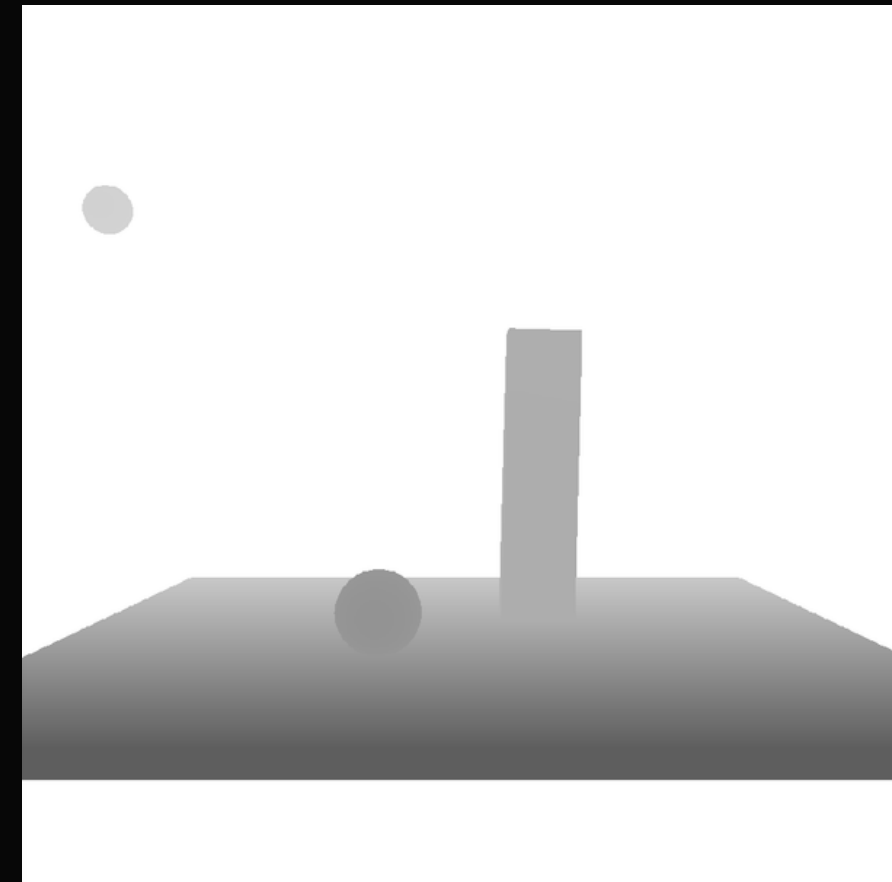
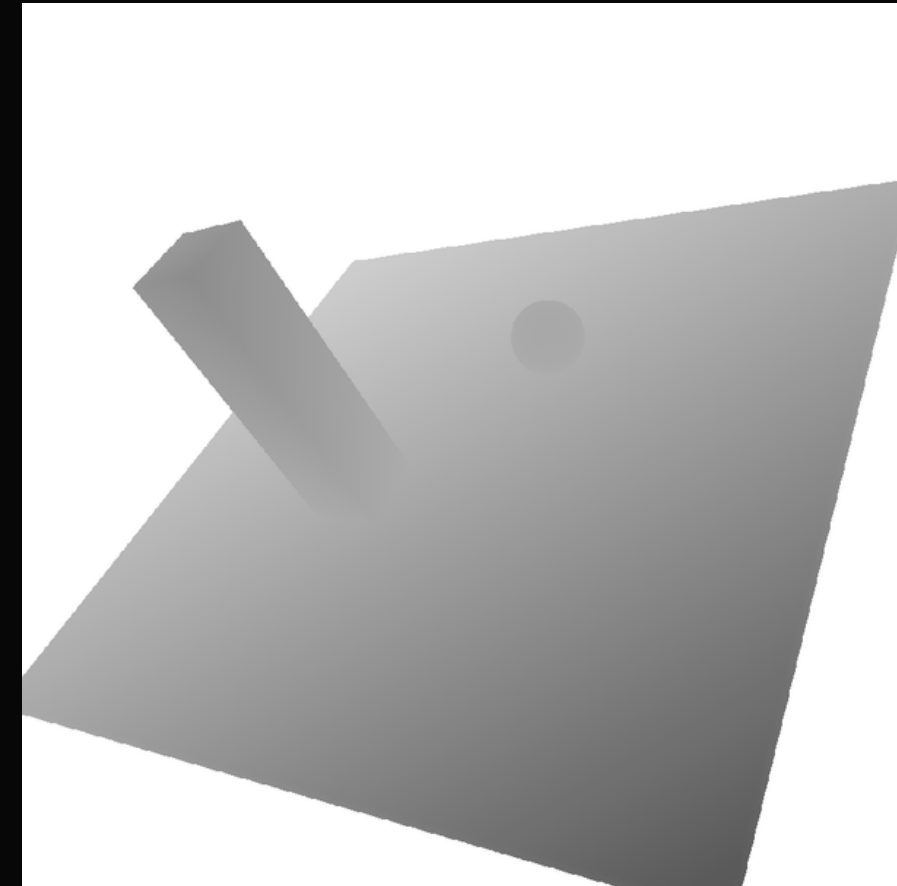
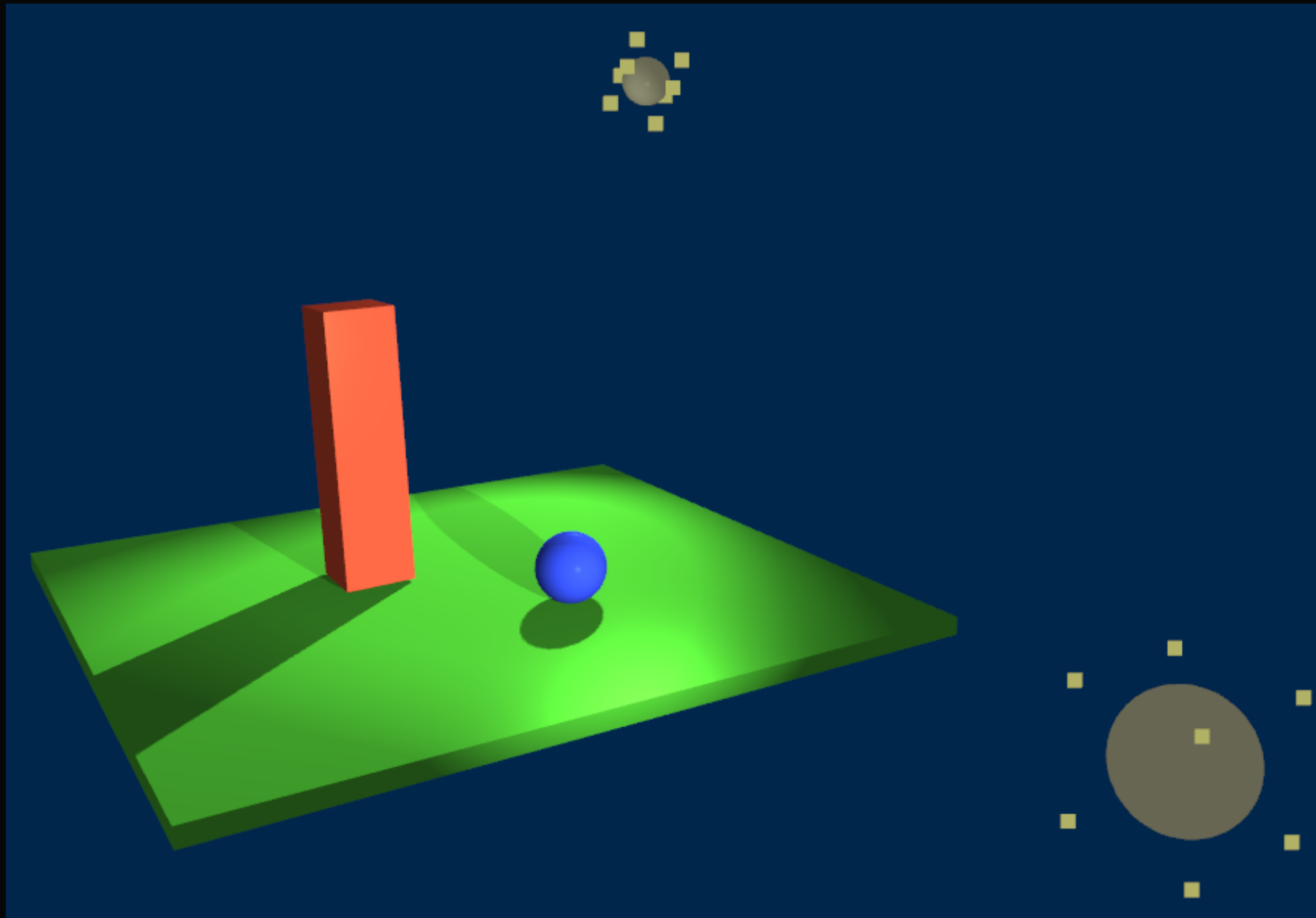
Shadow maps

La scène pouvant avoir plusieurs lumières, nous devons avoir une carte de profondeur différente par lumière.

Rendu

Shadow maps

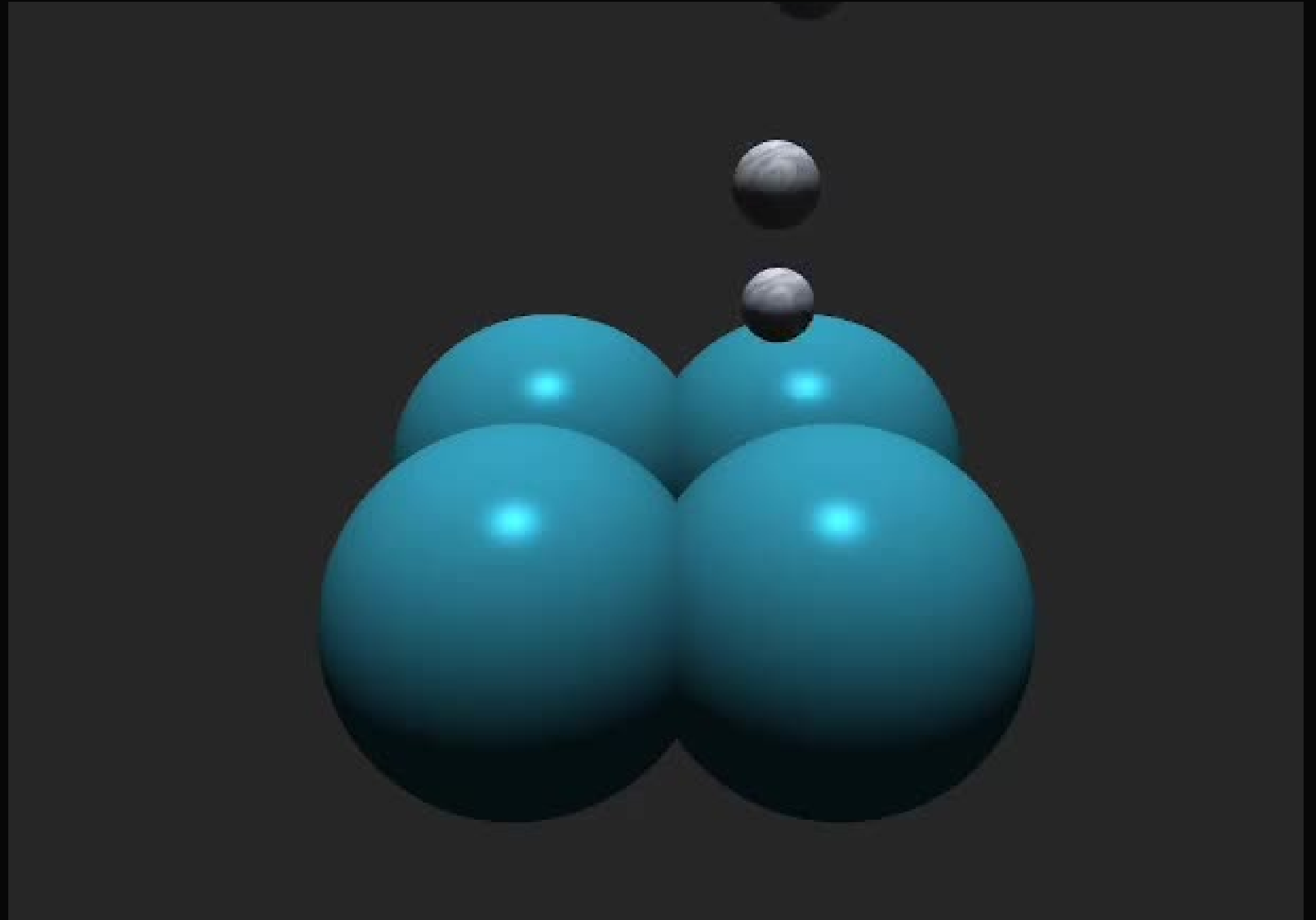
14
1



Simulations

Scene simulation

15 / 1



Simulations

Scene Laboratoire

Character

L'utilisateur peut interagir avec cette scène par le biais d'une classe personnage.

Simulations

Scene Laboratoire

Character

L'utilisateur peut interagir avec cette scène par le biais d'une classe personnage.



Body

- Nœud du personnage
- Peut seulement se tradater

Simulations

Scene Laboratoire

Character

L'utilisateur peut interagir avec cette scène par le biais d'une classe personnage.



Body

- Nœud du personnage
- Ne tourne jamais



Camera

- Vue du personnage
- Peut tourner

Simulations

Scene Laboratoire

Character

L'utilisateur peut interagir avec cette scène par le biais d'une classe personnage.



Body

- Nœud du personnage
- Ne tourne jamais



Camera

- Vue du personnage
- Peut tourner



Item

- Objet dans la main

Simulations

Scene Laboratoire

Déplacement

Les directions de translation du personnage sont calculées en fonction des transformations de la caméra.

```
//Compute translation relative to camera direction
glm::vec3 forward(0, 0, -1);
glm::vec3 forward_vec = character_cam_trsf->apply_to_vector(forward);

forward_vec[1] = 0.; //disable flight
forward_vec = glm::normalize(forward_vec);

glm::vec3 up(0, 1, 0);
glm::vec3 right_vec = glm::cross(forward_vec, up);
```


Simulations

Scene Laboratoire

Interactions

Le personnage peut attraper et lancer un objet de la scène.




Attraper un objet revient à le retirer du système physique et l'ajouter comme nœud fils du personnage.

Simulations

Scene Laboratoire

Interactions

Le personnage peut attraper et lancer un objet de la scène.



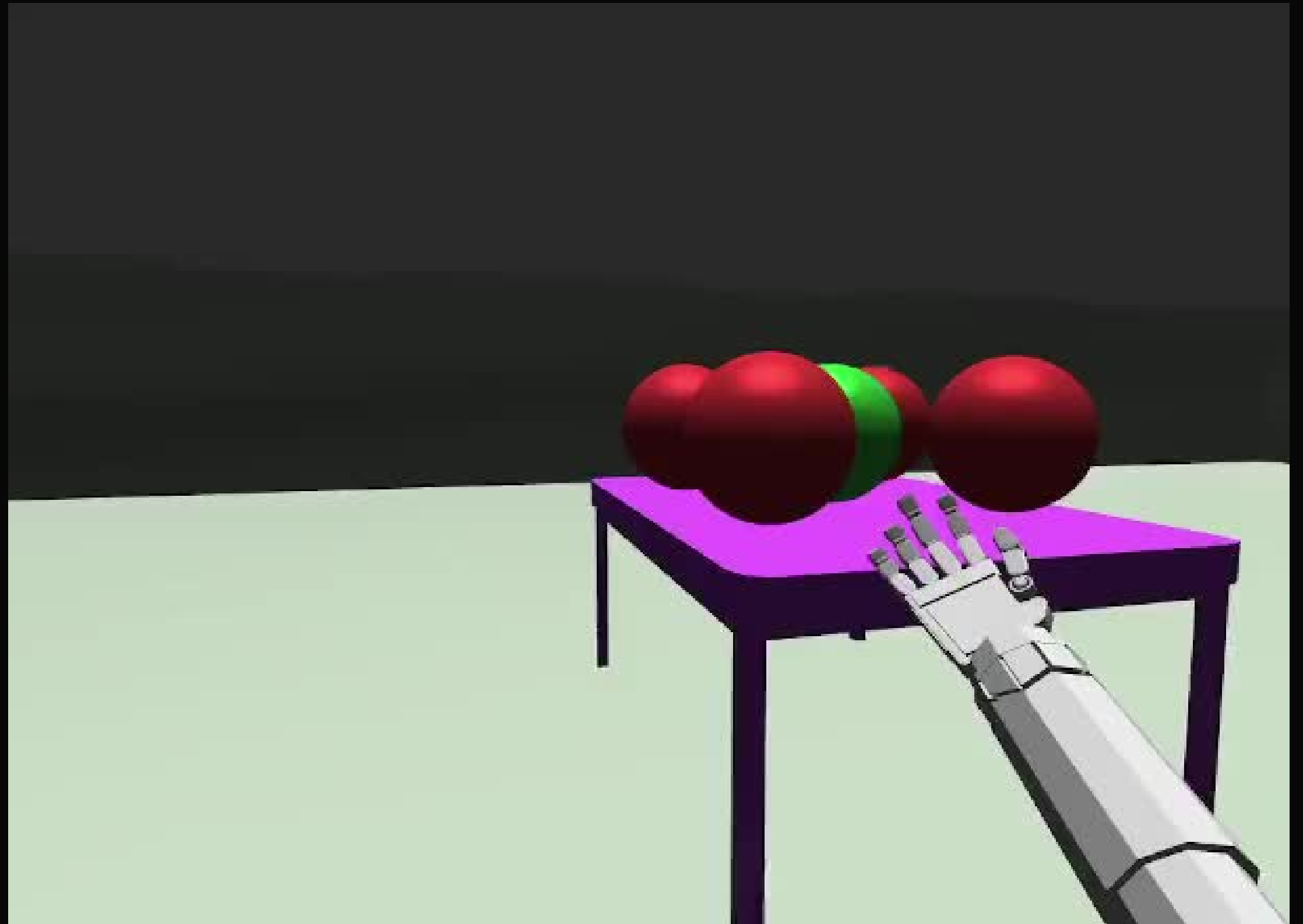
Attraper un objet revient à le retirer du système physique et l'ajouter comme nœud fils du personnage.

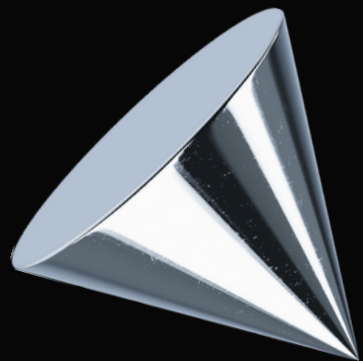
Pour lancer l'objet on applique une impulsion linéaire.

Simulations

Scene Laboratoire

19 / 1





**Merci de
votre
attention**

