

Soutenance 1

Black Hole

Mars 2023



Ronan LEBOUCHER, Lorenzo LOMBARDI, Thomas MORIN, Brian PERRET

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Objectifs</b>	<b>4</b>
2.1	Pour cette soutenance...	4
2.2	Pour la prochaine soutenance...	4
<b>3</b>	<b>Création de salles</b>	<b>5</b>
3.1	Méthode de travail	5
3.2	Salle 1	6
3.3	Salle 2	7
3.4	Salle 3	10
<b>4</b>	<b>Modélisation des objets 3D</b>	<b>10</b>
4.1	Introduction	10
4.2	GraviSphère	10
4.3	Bouton et Plaque de Pression	12
<b>5</b>	<b>Gestion du personnage</b>	<b>12</b>
5.1	Introduction	12
5.2	Déplacement du joueur	12
5.3	Inversion de la gravité	13
5.4	Détection du sol	13
5.5	Objet intéragissable	14
5.6	Gestion de la vie du joueur	14
5.7	Animation	15
<b>6</b>	<b>Multijoueur</b>	<b>15</b>
6.1	Introduction	15
6.2	Gestion des clients	15
6.3	Procédure coté client	16
<b>7</b>	<b>Gestion du HDRP</b>	<b>16</b>
7.1	Introduction	16
7.2	La lumière	16
7.3	Pré-calcul des réflexions sur les objets statiques	17
7.4	Gestion de la lumière sur les objets dynamiques	18
7.5	Les volumes	18
<b>8</b>	<b>Interface utilisateur</b>	<b>19</b>
8.1	Introduction	19
8.2	Interface en jeu	19
8.3	Menu	19
<b>9</b>	<b>Site web</b>	<b>19</b>

<b>10 Conclusion</b>	<b>20</b>
10.1 Avance sur le planning . . . . .	20
10.2 Retard sur le planning . . . . .	21

## 1 Introduction

Bonjour, nous sommes Black Hole (anciennement RBLT-corp), l'équipe se compose de Brian, chef de groupe et développeur, Thomas et Ronan qui sont game designers ainsi que notre graphiste : Lorenzo.

Notre équipe est derrière le jeu de reflexion/enigmes nommé UpSide Down se basant sur l'inversion de la gravité grâce à l'objet principal du jeu nommé la "GraviSphère". Le jeu se compose de multiples salles d'énigmes à résoudre en duo. Pour celà le joueur peut interagir avec différents objets tels que des plaques de pression ou encore des cubes que l'on peut déplacer et inverser leur gravité permettant d'ouvrir une porte ou autre mécanisme aidant à la résolution de la salle.

## 2 Objectifs

### 2.1 Pour cette soutenance...

Pour notre première soutenance, nous voulions implémenter les mécaniques de base de notre jeu à savoir : l'inversion de la gravité, le déplacement et l'interaction des objets tels que le fait d'attraper le "cube" ou actionner une plaque de pression pour activer une porte.

Nous voulions aussi implémenter le multijoueur nécessaire au cahier des charges. Et faire en sorte que deux joueurs puissent jouer simultanément dans la même salle.

Conceptualiser les modèles 3D des objets qui intéragissent avec le joueur.

Ainsi que faire en sorte que nos trois premières salles soient fonctionnelles (c'est-à-dire que les mécanique ainsi que les énigmes soient mises en place pour que le joueur puisse résoudre la salle).

De plus, faire en sorte d'avoir un Hub qui fera la liaison entre toutes les salles et constituera le fil rouge de notre jeu et un décor que le joueur retrouvera régulièrement. Cela lui donnera d'avantage de confort dans le jeu, c'est-à-dire que cela cassera le rythme des successions des salles avec un décor familier, ce qui permettra éventuellement de développer de nouvelles idées pour la suite du jeu.

### 2.2 Pour la prochaine soutenance...

Nous aimerais que toutes les salles prévues aujourd'hui soient fonctionnelles et praticables pour les joueurs (sous réserve d'en ajouter ou d'en enlever au fil de notre réflexion).

Ainsi que l'amélioration du multijoueur dans la gestion des changements de salles avec des écrans de chargement.

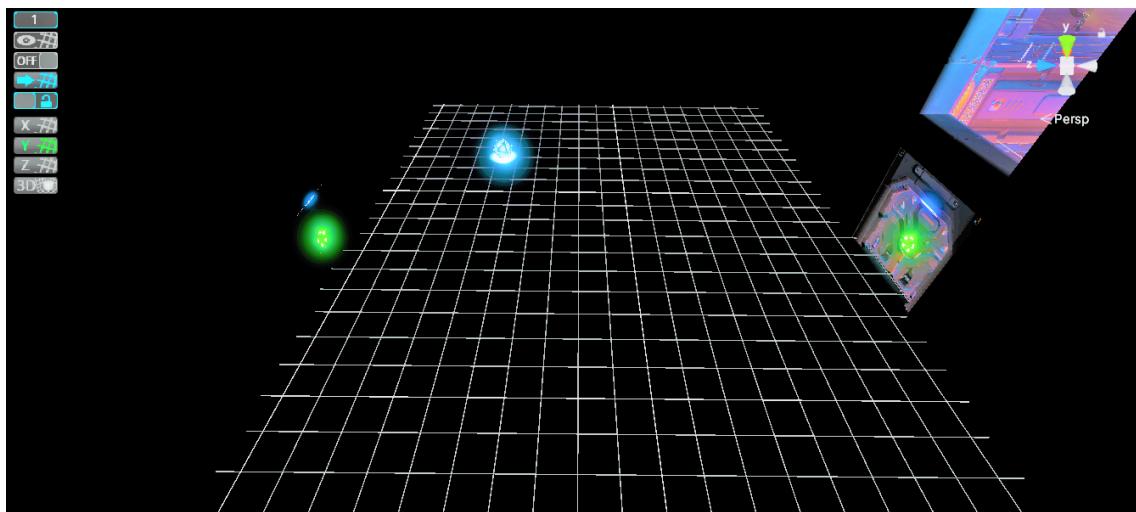
Nous avons enfin pour projet de faire une IA utilisant le "path-finding" pour trouver le chemin le plus court vers le joueur portant l'objet servant à finir le niveau. L'IA n'apparaîtra que dans certains niveaux et sera représenté par un robot. Si le robot rentre en contact avec le joueur, ce dernier lâchera l'objet qui se désintégrera et réapparaîtra à son emplacement d'origine, obligeant les joueurs à tout recommencer. Cela créera une dynamique où un joueur devra porter l'objet, le

rendant vulnérable, pendant que l'autre joueur, en plus d'interagir avec le niveau, devra le protéger du robot en interagissant avec lui pour soit le ralentir, soit le détruire.

### 3 Création de salles

#### 3.1 Méthode de travail

Pour la création des différentes salles et des différents décors que compte le jeu, nous (Ronan et Thomas, les deux game designers) devions trouver une méthode ainsi qu'un rythme de travail permettant de travailler efficacement et de se partager le travail. La première étape de la création d'une salle ou d'un décor et de se réunir et de réfléchir aux mécaniques ainsi qu'au déroulement de la salle. Au fil de notre réflexion, nous esquissons un schéma général de la salle, réunissant nos idées et le prémissé du déroulement de la salle. A partir de ce schéma, nous décidons qui se chargera de quelle partie de la salle, et sur quel créneau horaire chacun travaillera. Le choix d'un créneau horaire différent est nécessaire car si nous travaillons simultanément sur la même salle, alors un push sur le projet provoquera un conflit entre nos deux versions de celui-ci. Mais malgré cette séparation du travail, notre coopération au sein du projet n'est pas impactée. En effet, chaque idée de design et chaque idée de configuration de salle ou encore d'éénigme est partagée et discutée entre nous, et permet de garder un design homogène ainsi que de chacun compléter les idées de l'autre. Concernant le côté concret de la création de salles, un outil nous a été très utile : Snap. Cet outil permet d'afficher une grille sur les axes X, Y ou Z sur la scène Unity, et nous permet de placer les objets de la salle ou encore les murs ou les sols de manière plus précise et simplifiée, sans avoir à les placer au pixel près et à l'angle près (illustration ci-dessous).



### 3.2 Salle 1

Cette première salle sert de tutoriel, c'est pourquoi la phase de schématisation et de conceptualisation de la salle était surtout centrée sur le dosage en terme de difficulté, et la manière d'agencer la salle pour faire appréhender au joueur les mécaniques de base du jeu.

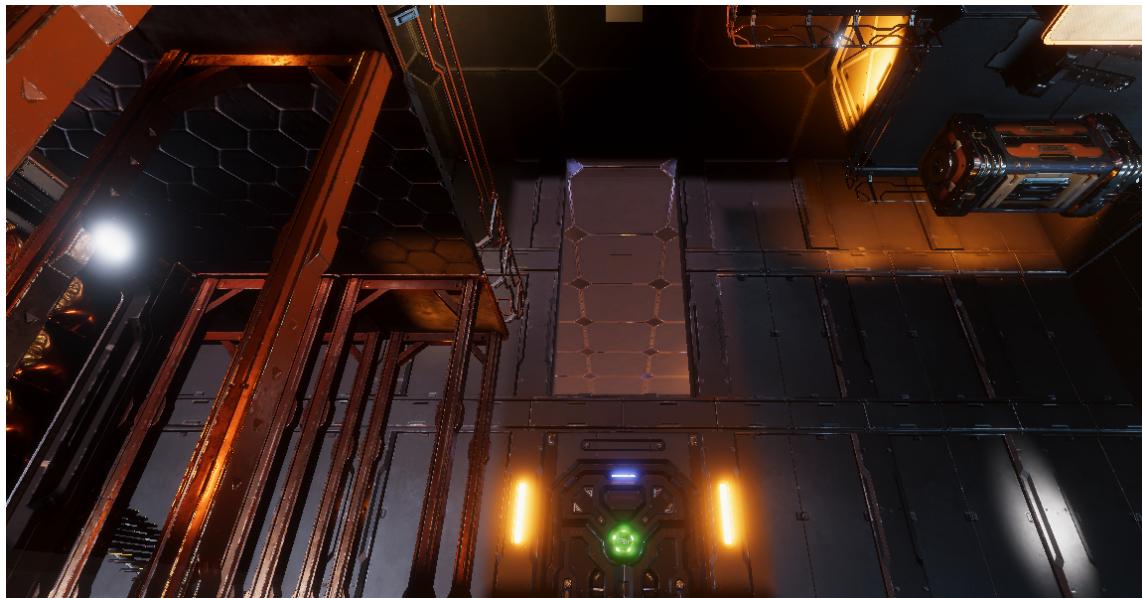
Pour cette salle chaque joueur est dans sa propre salle identique à l'autre pour leur faire découvrir les boutons et l'inversion de gravité. Pour résoudre cette salle, chaque joueur devra explorer la salle pour trouver deux boutons et les actionner, chaque bouton étant situé dans des couloirs annexes et opposés géographiquement, qui permettra d'ouvrir la porte de sortie qui se situe à l'étage et dont l'accès se fait uniquement avec l'inversion de gravité.



### 3.3 Salle 2

Cette seconde salle permet de mettre le joueur en condition du multijoueur, et de faire découvrir les premières phases de réflexion en coopération, avec les deux joueurs dans la même salle. Elle permet également de faire expérimenter au joueur l'utilisation du cube, et faire preuve de coopération et de réflexion pour sortir de la salle. Elle est constituée de 3 parties principales, et de 2 parties annexes à ces dernières. Sa conception a été intéressante car elle nécessitait vraiment de la réflexion par rapport aux énigmes en coopération.

Au début de la salle, un joueur actionnera une plaque de pression au plafond pour ouvrir la porte en bas, et permettre au deuxième joueur d'accéder à la deuxième partie de la salle et d'actionner à son tour une plaque de pression qui permettra au premier joueur d'accéder à la deuxième partie de la salle à son tour (la porte étant verrouillée si un joueur ne se situe pas sur une des plaques de pression).



Ensuite, une deuxième porte verrouillée ne s'ouvre que si les deux joueurs appuient en même temps sur deux boutons distincts (avec un délai permettant un léger décalage sur la pression simultanée) qui se situent chacun à l'opposé, dans des pièces dont l'entrée se situe en bas pour l'un et en haut pour l'autre (avec une vitre dans chaque pièce donnant sur l'intérieur de la pièce principale pour que les deux joueurs se voient et se coordonnent). Une fois les deux boutons pressés, la porte reste ouverte définitivement.

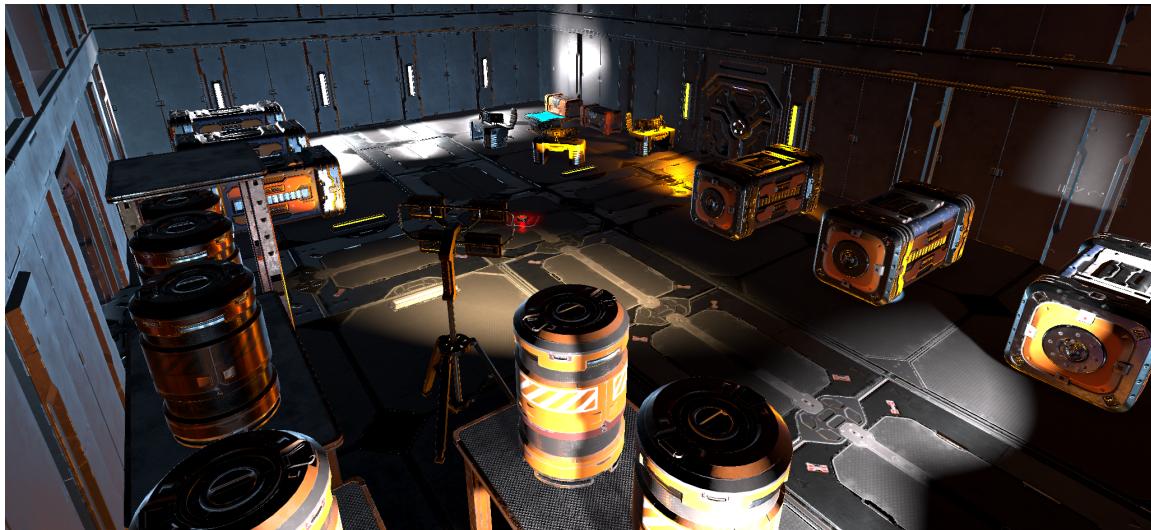


L'image ci-dessus est couloir annexe en bas.



L'image ci-dessus est le couloir annexe en haut.

Dans la troisième (et dernière) partie de la salle, les joueurs récupéreront un cube qui sera utile une fois que les deux joueurs seront revenus à la première partie de la salle (aux plaques de pression).



En réitérant le processus réalisé au début de la salle au niveau des plaques, où une porte cachée dans le décor abritera une dernière plaque de pression sur laquelle les joueurs devront poser le cube pour rendre permanente l'ouverture de la porte de sortie de la salle.



L'image ci-dessus est la décoration où la porte est cachée.

Cette succession d'énigmes dans cette salle nécessitait une plus grande réflexion car nous souhaitions créer une salle où chaque joueur jouerait un rôle dans son déroulement, sans que celle-ci puisse être finie seul, tout en faisant découvrir les nouvelles mécaniques du cube et des plaques de pression, et en gardant l'inversion de gravité comme principale mécanique. La conception de la salle en gardant en tête toutes ces contraintes représentait un challenge conséquent, ce qui nous a permis à la fois de trouver le déroulement de cette salle et des idées pour de prochaines salles.

### 3.4 Salle 3

Cette troisième salle est plutôt centrée sur la recherche d'indices dans l'environnement, la résolution d'énigmes, et la recherche de portes dérobées. La salle est divisée en deux couloirs où les joueurs sont séparés et résolvent chacun des énigmes (5 chacun). A chaque énigme, le joueur devra choisir quelle porte emprunter en fonction d'un indice trouvé par l'autre joueur, de l'éénigme à résoudre ou encore de la présence d'une porte dérobée.

Quand les deux joueurs arrivent dans la salle, ils doivent choisir chacun un couloir à emprunter. Prenons le joueur 1 qui prend le couloir de gauche et le joueur 2 le couloir de droite. Dans la salle 1 du joueur 1, le joueur devra trouver, en explorant la salle, un numéro qui indique la porte numérotée que le joueur 2 devra prendre, sachant que sa 1ere salle ne contient pas d'indice, pour que celui-ci atteigne sa deuxième salle. Dans cette dernière, il trouvera à son tour un indice qui permettra au joueur 1 de prendre la bonne porte et atteindre sa salle 2. Dans cette salle l'indice trouvé est un chiffre à l'envers, le joueur 2 comprendra donc qu'il doit inverser la gravité et choisir entre les 3 portes au plafond et non celles au sol. Une fois inversé, l'ordre des portes n'est pas le même que quand il était en bas et de ce fait, il doit s'adapter. Partant de cette même logique les joueurs trouveront par exemple un code césar et ou encore des portes dérobées (dont une au plafond).

Cette salle nous a demandé de la réflexion dans la mise place des énigmes et dans la façon dont on les adaptera à leur salle, de punir les mauvais choix de portes pour chacun des joueurs et le fait que l'on ait privilégié la réflexion vis-à-vis des mécaniques du jeu (l'inversion de gravité par exemple).

## 4 Modélisation des objets 3D

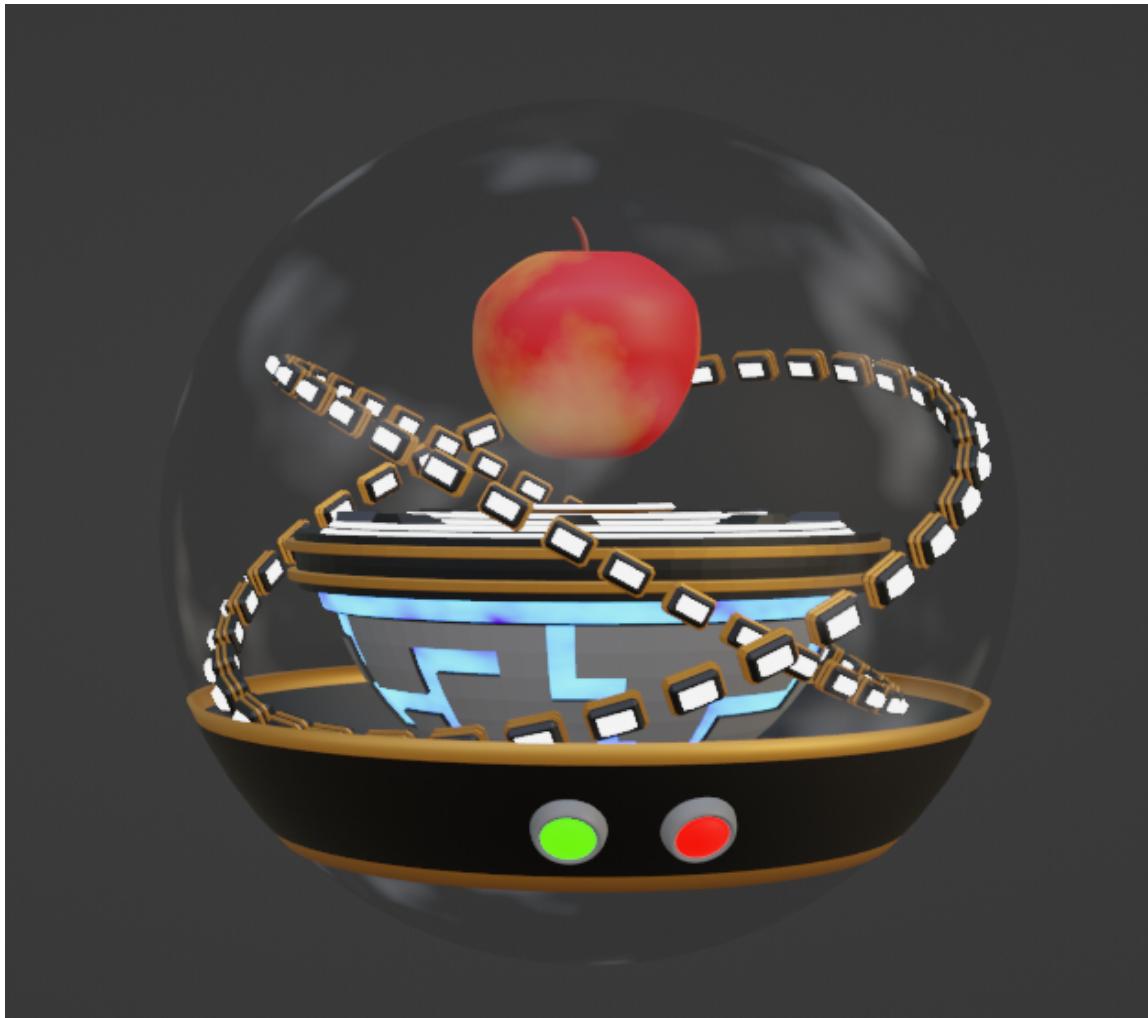
### 4.1 Introduction

Pour la création des salles, un pack d'asset graphique a été utilisé. Cependant aucun objet de ce pack ne pouvait être utilisé en tant qu'objet interagissable pour les joueurs. De ce fait, Lorenzo le graphiste du groupe s'est occupé de modéliser tous les objets manquants au bon fonctionnement du jeu.

### 4.2 GraviSphère

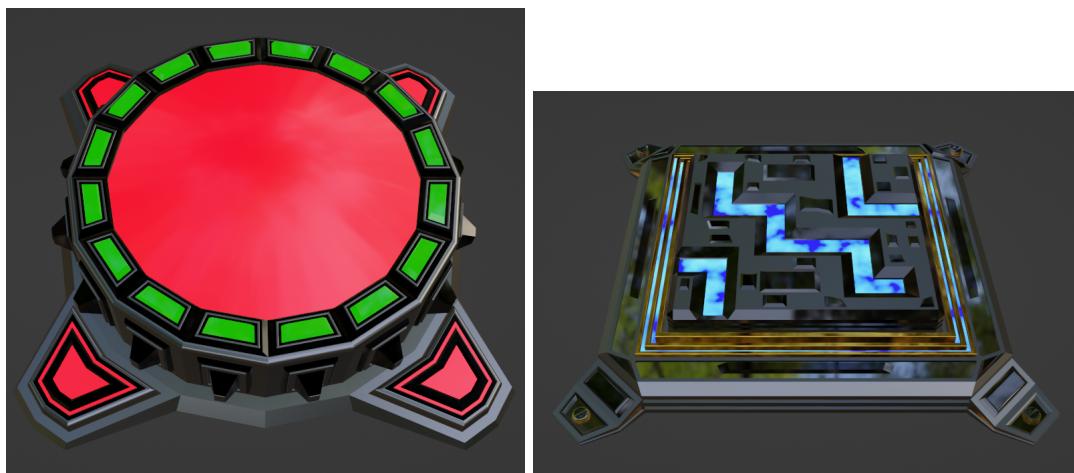
La GraviSphère est l'objet phare du jeu : celui qui permet au joueur de changer sa gravité. Étant en permanence dans la main du joueur, cet objet sera toujours dans le champs de vision, c'est pour cela qu'il fallait un objet qui se différencie des autres, un objet qui fait l'identité du jeu, comme le Portal-Gun dans Portal. Comme cet objet doit changer la gravité de son possesseur, il fallait que dans son design, on retrouve un élément qui fait penser à la gravité, et généralement, la première chose qui vient en tête est la pomme de Newton, grâce à laquelle l'histoire veut que Newton ait eu la révélation sur l'existence de la gravité. Ainsi la GraviSphère est constituée d'une pomme flottante au dessus d'une plateforme futuriste avec deux anneaux lévitant autour, le tout dans une sphère de verre. De cette façon, on a un objet, soit trop futuriste pour être un objet des temps modernes soit trop complexe pour être un objet terrien empêchant ainsi les joueurs de savoir précisément où ils évoluent et laissant une place de mystère dans cet environnement.

Après le prototype validé par tous les membres du groupe, Lorenzo s'est occupé de modéliser la GraviSphère au propre. Cependant, durant la modélisation, il y eut deux légers problèmes. Le premier était le choix des anneaux. De nombreux modèles ont été proposés, mais à chaque fois, il manquait un petit quelque chose, mais à force un modèle est sorti du lot mettant d'accord tous les membres du groupe. Le deuxième problème était l'export de certains matériaux de Blender vers Unity. En effet, les matériaux d'émission (des textures devant produire de la lumière) disparaissaient d'un logiciel à l'autre, rendant ainsi l'objet qui se démarquait par ses nombreuses parties lumineuse en un objet fade et sombre. Après de longues recherches et de nombreuses exportations, le modélisateur du groupe a pu enfin trouver la manière de faire en sorte que la GraviSphère puisse être exporté avec toutes ses textures et ses lumières, conservant ainsi le charme qu'elle avait sur le logiciel de modélisation Blender.



### 4.3 Bouton et Plaque de Pression

Dans le pack d'asset utilisé pour faire les salles, il manquait les objets nécessaires pour que les joueurs puissent interagir avec l'environnement. Pour cela, il fallait un bouton pouvant rendre l'ouverture des portes temporaire ou définitive et une plaque de pression pour ouvrir la porte soit avec le poids d'un joueur soit avec le poids d'un "cube" (qui est une icosphère). Le plus gros défi était de modéliser des objets 3D qui s'intègrent bien avec le pack d'asset des salles. Étant en hight poly (possédant énormément de détails), cela a donné un défi à notre modélisateur qui maîtrisait seulement le low poly (peu de détails) à devoir se surpasser. Pour cela, il a dû faire des modèles 3D en low-poly avec suffisamment de détail pour simuler un objet hight poly et ne pas faire tâche dans des salles très détaillées.



Dans l'ordre : le bouton et la plaque de pression.

## 5 Gestion du personnage

### 5.1 Introduction

Le personnage est l'objet le plus important dans un jeu. En effet, c'est ce que le joueur va directement contrôler, il est donc impératif que cela lui procure une expérience agréable et intuitive au travers des différentes actions qu'il aura à sa disposition tel que le contrôle du personnage et ses collisions ou bien l'interaction avec les autres objets.

### 5.2 Déplacement du joueur

Le joueur peut déplacer son personnage comme la plupart des fps modernes (avant, arrière, gauche, droite). Le déplacement du personnage se fait via le biais d'un "rigidbody" permettant d'appliquer une force via un vecteur (x,y,z) au "rigidbody" possédant une vitesse. Cette force est calculée en fonction des différentes touches que le joueur presse ainsi que l'orientation de la caméra que le joueur contrôle avec sa souris. A noter que le joueur a la possibilité d'effectuer un saut à la condition que le personnage soit en contact direct avec un sol.

### 5.3 Inversion de la gravité

L'inversion de la gravité se fait simplement en inversant la gravité du joueur, si celui-ci a une gravité de -9,81 et appuie sur la touche associée, alors sa gravité passera à 9,81.

L'inversion de la gravité amène quelques problématiques. En effet lors du changement il est nécessaire de faire une rotation de la partie graphique du personnage ainsi que la caméra pour que le plafond devienne le sol.



Un joueur regarde un autre joueur marchant au plafond.

### 5.4 Détection du sol

Une problématique récurrente dans les jeux vidéos est la détection du sol qui peut parfois s'avérer compliquée dans certaines circonstances telles que les escaliers (le joueur ne doit pas glisser). De plus, dans notre cas, le joueur peut inverser la gravité ce qui a pour effet d'inverser également son modèle 3D, ce qui complexifie le processus. Pour cela, nous utilisons un "raycast" permettant de détecter entre deux points donnés si il y a un objet avec un "collider" (le sol, un escalier, un cube, ...) La distance entre ces deux points doit être ajustée pour que le joueur ne puisse pas faire de double saut mais aussi que la distance soit suffisante si le joueur marche sur un "collider" arrondi. Cette distance pose quelques problèmes dans certaines circonstances. En effet, le joueur est soumis en permanence à une gravité mais sur des surfaces inclinées telles que des escaliers, ce qui provoque un glissement sur le joueur. Pour résoudre ce problème nous désactivons simplement la gravité si le joueur se trouve sur un escalier (d'autres méthodes équivalentes existent mais sont plus complexes avec le changement de gravité).

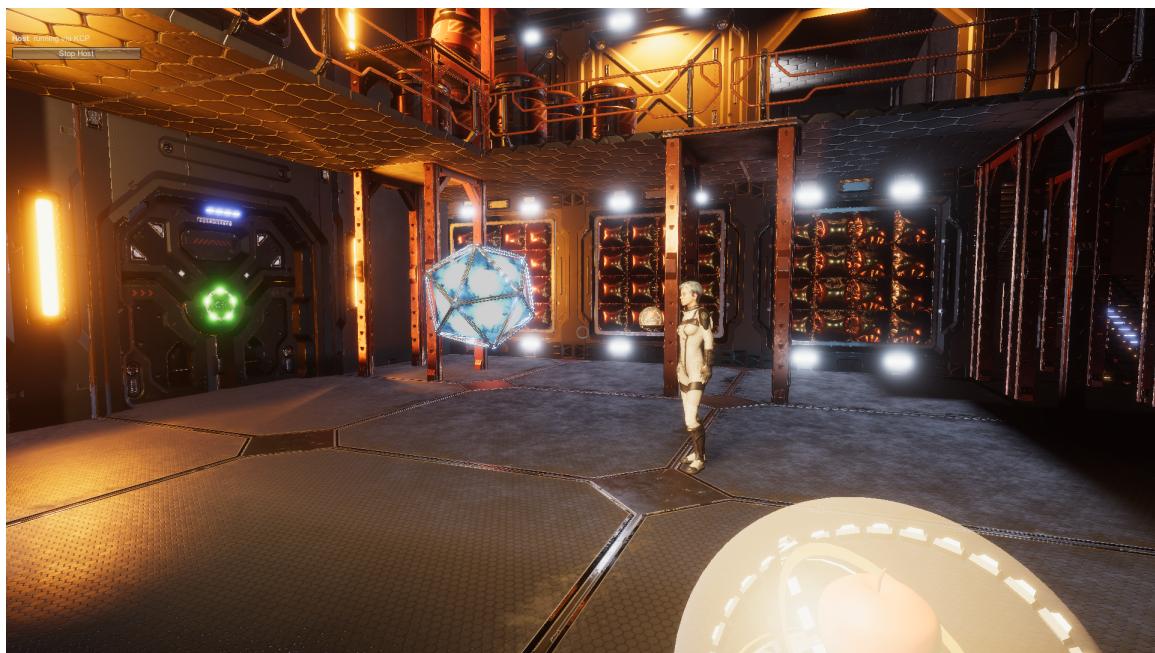
A noter que l'utilisation d'un "character controller" aurait été bien plus pratique en proposant directement ces fonctionnalités, mais ces mêmes fonctionnalités ne gèrent absolument pas le changement de gravité.

## 5.5 Objet intéragissable

Le joueur a la possibilité d'intéragir avec certains objets dans la salle (cube, bouton, plaque de pression, ...) lui permettant de réaliser diverses actions (ouvrir des portes, déplacer des objets, ... )

Pour faciliter la compréhension du joueur par rapport à la manière qu'il a de savoir avec quel objet il a la possibilité d'intéragir, nous avons mis en place un détourage en blanc de l'objet en question ainsi qu'une aide pour savoir quelle est la touche associée à l'action pour cet objet (hormis le cube qui est l'objet le plus manipulé, pour lequel nous avons décidé qu'il n'était pas nécessaire d'afficher ses aides). Parmis ces objets, certains peuvent être déplacés, tel que le cube, pour actionner une plaque de pression à la place d'un joueur par exemple. Pour cela, le joueur peut appuyer sur la touche associée pour attraper l'objet qui se met à flotter dans les airs vers l'endroit où le joueur regarde.

De plus, l'objet en question synchronise sa gravité avec celle du joueur tant que celui-ci le garde en main, il peut donc le laisser au plafond en activant une plaque de pression reliée à une porte pour que le joueur passe par cette dernière. Enfin, si le joueur bloque l'objet qu'il tient dans la main et qu'il s'en éloigne trop, alors celui-ci va automatiquement le lâcher.



Joueur portant le "cube".

## 5.6 Gestion de la vie du joueur

Chaque joueur possède un nombre de points de vie, qu'il peut perdre à cause de certains pièges ou bien gagner avec la régénération naturelle. La régénération intervient seulement quelques secondes après les derniers dégâts subis par le joueur.

En cas de mort, le joueur réapparaît au dernier point de réapparition par lequel il sera passé (se trouvant généralement au début des salles).

A noter que le joueur ne prend pas de dégats de chute.

### 5.7 Animation

Le personnage possède de nombreuses animations permettant principalement une meilleure expérience pour les joueurs. Un grand nombre de ces animations est lié au déplacement du personnage et se joue en fonction de l'état du personnage (marche, court, saute, inverse la gravité, ...). Mais certaines animaions se jouent en continu lorsqu'aucune autre n'est jouée, c'est le cas de l'animation "idle" du personnage permettant d'imiter sa respiration en ajoutant ainsi du mouvement dans l'objet pour une meilleure expérience.



Animation du joueur en train de marcher.

## 6 Multijoueur

### 6.1 Introduction

Upside Down est un jeu uniquement en duo, le multijoueur est donc nécessaire pour respecter le cahier des charges. Pour cela nous utilisons Mirror qui offre une bonne documentation et l'accès à une grande communauté tout en répondant aux contraintes de notre jeu, à savoir un besoin d'avoir une très faible latence entre les joueurs pour ne pas avoir de possibles décalages lors d'actions rapide ainsi que de pouvoir lancer facilement des parties en local avec un personnage hébergeant le serveur (tout en étant client) et un autre joueur qui est uniquement client.

### 6.2 Gestion des clients

Chaque nouveau client qui se connecte prend le contrôle d'un personnage qui lui est attribué grâce a un ID. De plus, il faut faire en sorte que les différents scripts ne se "mélangent" pas entre les différents personnages. Pour cela nous utilisons une fonction de Mirror permettant d'identifier si le joueur est le joueur local et donc ne pas interpréter les mouvement des autres personnages.

Le mode multijoueur ajoute aussi tout un tas de complications, principalement dues à la synchronisation des personnages, à celle des animations ou bien de certaines variables comme l'identification du joueur qui tient le cube pouvant entraîner de nombreux bugs.

### 6.3 Procédure côté client

L'implémentation du multijoueur permet de synchroniser un grand nombre de données entre le serveur et les clients mais cela est coûteux en performance et donc il ne faut pas en abuser. Pour cela, Mirror permet de faire des calculs uniquement côté client ce qui a pour effet de diminuer le coût de performances dû à la synchronisation.

## 7 Gestion du HDRP

### 7.1 Introduction

Le "High Definition Render Pipeline" ou HDRP est un pipeline de rendu physiquement réaliste permettant une gestion de la lumière très précise et réaliste pour créer tout type d'atmosphère (un matin de brume ou bien une exploration sous marine). Cependant le calcul de la lumière est très coûteux en ressource, il faut donc savoir doser le nombre d'effets et leurs qualités pour que le jeu fonctionne de manière fluide sur la plupart des ordinateurs. Un moyen simple de faire cela est de proposer plusieurs qualités graphiques en modifiant la quantité d'effets visuels ou leurs qualités.

### 7.2 La lumière

Pour créer une source lumineuse telle qu'une lampe il faut procéder à deux étapes :

La première est de placer une source lumineuse qui permet de paramétrier comment la lampe éclaire les autres matériaux et donc de créer les reflets. Les sources lumineuses possèdent un grand nombre de paramètres permettant de créer tous les types de lampes (puissance, angle et portée de diffusion, température, ...) permettant de créer toute une atmosphère avec différentes lampes qui donnent un rendu très agréable, ce qui participe à l'immersion. De plus, les sources lumineuses offrent 3 modes de rendu ("Realtime", "Mixed", "Baked"). Le mode "Realtime" permet faire le calcul de cette lumière en permanence, ce qui est très gourmand et nécessite donc d'être utilisé sur des objets dynamiques. Le mode "Bake" quant à lui permet de faire un pré-calcul de la lumière et de stocker ses valeurs dans une texture qu'il ajoutera lors de l'exécution du jeu, permettant un faible coût de performance mais cela ne marche que pour les objets statiques. Enfin le mode "Mixed" fait un mélange des deux. Il faut donc jouer avec ces 3 modes en fonction du rôle de la lampe pour éviter un maximum de calculs de lumière inutiles.

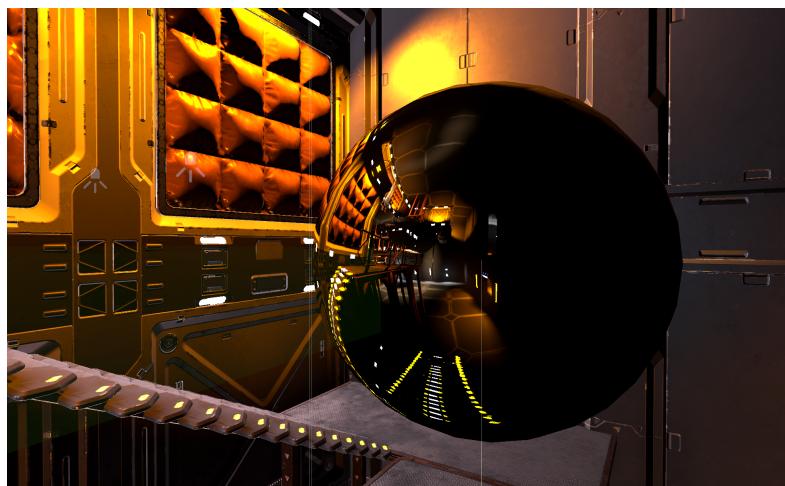
La deuxième étape est de faire en sorte que la texture de la lampe émette la même couleur que la source lumineuse pour donner l'impression que c'est cette partie de la lampe qui éclaire. Pour cela, il suffit de jouer sur les paramètres du matériau.



Les réflexions sur un tuyau.

### 7.3 Pré-calcul des réflexions sur les objets statiques

Le pré-calcul des réflexions permet d'avoir une très bonne lumière sans avoir une grande quantité de calcul, mais avec l'inconvénient de ne fonctionner que pour les objets statiques. Le pré-calcul de la lumière s'opère avec différents composants. Cela commence par les lampes comme vu plus haut en modifiant le mode de rendu, mais il existe aussi les "Reflexion probes" dont le fonctionnement peut être comparé à une caméra 360 qui stocke dans un périmètre donné toutes les réflexions (pré-calcul) et les stocke dans des images 360. Lors de l'exécution du jeu, il suffit de superposer les images 360 avec les textures d'un objet pour recréer les réflexions et ainsi avoir des réflexions réalistes avec bien moins de calculs à effectuer.



Vue d'une "Light Probe", on peut y voir l'image 360.

## 7.4 Gestion de la lumière sur les objets dynamiques

Cependant, le pré-calcul des lumières pose un réel problème pour les objets dynamiques car leur réflexion change en fonction de leur distance avec les différentes lampes les éclairant. Nous sommes donc obligés de calculer en temps réel la lumière pour ces objets ce qui consomme bien plus de puissance de calcul et ce qui nous oblige donc à limiter au maximum le nombre d'objets dynamiques avec un calcul des lumières en temps réel.

De plus, il existe justement un procédé permettant de réduire le nombres de caluls pour chacun de ces objets. Pour ce procédé il suffit d'utiliser un "Light Probe Group" qui permet de placer dans une carte un grand nombre de points (plusieurs milliers) qui fonctionne un peu de la même manière que les "light probes" vu plus haut, mais au lieu de faire une image celle-ci stocke l'orientation des différentes réflexions et sources lumineuses ("lightmap") qui passent par chacun de ces points. Cela permet de faire en sorte que lorsque le joueur se déplace, au lieu de calculer en permanence les 30 lumières qui l'éclairent, de simplement prendre 4 points dans les "Light Probe Group" et de recréer toute la lumière et les réflexions. On passe donc d'un calcul du nombre de source de lumière à seulement 4 en permanence, et donc permet d'avoir un éclairage fidèle même sur les objets dynamique sans une grande quantité de calculs de lumière.



Vue des nombreux composants gérant la lumière.

## 7.5 Les volumes

Les volumes permettent de changer les paramètres de rendu de manière locale en délimitant des zones. Cela permet entre autres de créer différentes atmosphères en fonction de la pièce dans laquelle se trouve le joueur. Ces paramètres de rendu sont, par exemple, l'exposition, les ombres, le flou de mouvement, ou encore la fumée.

## 8 Interface utilisateur

### 8.1 Introduction

L'interface utilisateur va être une surcouche sur l'écran de chacun des joueurs. Cette surcouche peut être uniquement cosmétique ou bien donner des réelles informations comme les menus ou son état de santé.

### 8.2 Interface en jeu

Actuellement notre jeu n'a qu'un viseur pour dire au joueur où est-ce qu'il est en train de regarder et donc d'interagir plus facilement avec les objets qui l'entourent.

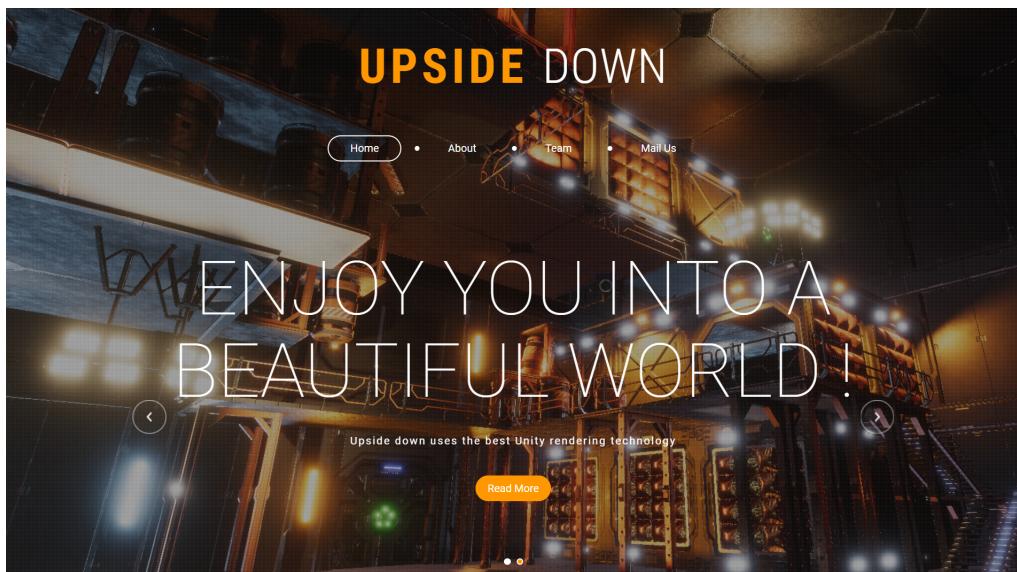
Nous avons fait le choix de ne pas ajouter d'autres choses en permanence tel que la barre de vie pour bénéficier au maximum de l'immersion en jeu.

### 8.3 Menu

Notre jeu possède un certain nombre de menu tels que le menu pause ou le menu des paramètres. Le menu pause est un menu très classique permettant de quitter le jeu, reprendre la partie ou accéder au menu des paramètres. Le menu des paramètres est plus intéressant car celui-ci permet de modifier un grand nombre de paramètres tels que la résolution ou bien la qualité graphique.

Nous avons aussi créé un menu pour personnaliser la majorité des touches dans le jeu. De plus, ces touches sont sauvegardées dans un fichier Json et rechargées automatiquement à chaque démarrage du jeu.

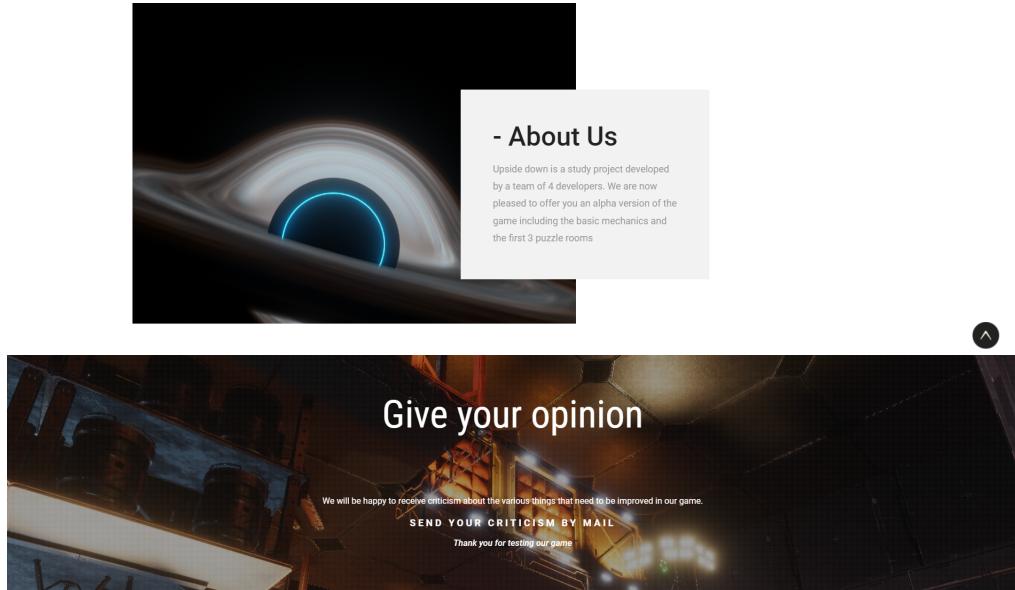
## 9 Site web



## Welcome

Upside Down is made up of a number of puzzle rooms using gravity reversal that you have to solve with the help of your teammate.

[Download now !](#)



## Our Team



## 10 Conclusion

### 10.1 Avance sur le planning

Nous ne sommes pas vraiment en avance à tous les niveaux. Notre avance se situe au niveau des premières salles (1 et 2) où nous avons la décoration et la lumière de faites, ce qui n'était pas présent dans les objectifs que l'on s'était fixés pour cette soutenance.

## 10.2 Retard sur le planning

Nos seuls retards sur le planning se situent au niveau du hub, que l'on devait avoir commencé, ainsi que sur la salle 3 qu'on devait avoir finie et rendue fonctionnelle, ce qui sera très bientôt le cas. On a donc un très léger retard sur certains aspects mais également un petit peu d'avance sur d'autres aspects.