

THE PIT

2GDA

2023 - 2024

Projet semestriel, 2ème année, 2ème semestre

ICAN Paris

Game Design Document

Marcel Gabbardo

Mathieu Brandao

Nicolas Ferretti

Rémi Mostura

INTRODUCTION

Contents

INTRODUCTION

3

L'équipe 6

Pitch 7

Game Pillars 8

Fiche d'identité 9

Overview 3C 10

Overview DA visuelle 11

Overview DA sonore 12

GAME DESIGN 13

TOY DESIGN 14

Intentions de design 15

Références de design 16

3C - Caméra 17

3C - Character 20

3C - Controller 21

Mechanics - Mouvement 22

Mechanic - Jump 25

Emergence constatée 28

Etat critique 29

GAME DESIGN 30

Intentions de design 31

Cible 32

Points forts du toy extraits 33

Conditions de victoire et de défaite 34

Boucles OCR 35

Boucle de gameplay 36

Timeline de gameplay horizontale 37

Game structure 38

Signes et Feedbacks 39

Matrice de Caillois 40

LEVEL DESIGN 41

Philosophies de level design 42

Types de surfaces 43

Système de génération 44

Pattern n°1: tree of life 45

Pattern n°2: pillars of creation 46

Pattern n°3: hanging gardens 47

Pattern n°4: clusters 48

Pattern n°5: citadel 49

Pattern n°6: coliseum 50

Pattern n°7: crown 51

DOCUMENTATION

TECHNIQUE 53

Contraintes importantes 54

Mouvement de l'avatar - Grounded 55

Mouvement de l'avatar - Airborne 56

Mouvement de l'avatar - Jump et Slide 57

Système de chute 58

Système de dégâts et "Life Savers" 59

LD procédural 60

Système de chute de l'étoile 61

Augmentation de la difficulté 62

GAME ART 63

Intentions Globales 64

Contraintes Importantes 66

Références Game Art 68

Références Game Art 69

Intégration Narrative 70

Player 71

Moodboard textures 72

Moodboard formes 73

Concept art objets 74

OBJETS 77

Pattern 1: concept art 78

Pattern 1: rendu final 79

Pattern 2: concept art 80

Pattern 2: rendu final 81

Pattern 3: concept art 82

Pattern 3: rendu final 83

Pattern 4: concept art 84

Pattern 4: rendu final 85

Pattern 5: concept art 86

Pattern 5: rendu final 87

Pattern 6: concept art 88

Pattern 6: rendu final 89

Pattern 7: concept art 90

Pattern 7: rendu final 91

Murs du puits 92

“Life Savers” 93

TECH ART 94

Post processing 95

Visual Effects 96

SOUND DESIGN 97

OVERVIEW 98

Intentions et Contraintes 99

Inspirations et Références 100

Event list 101

EVENTS 102

Event commenté: distance de l'étoile 103

Event commenté: sliding 104

Event commenté: heal and damage 105

event commenté: Landing 106

Mixer Audio 107

COMPTE RENDU 108

RETROSPECTIVES 109

Outils employés 110

Difficultés notables 111

Afterword des développeurs: Marcel 112

Afterword des développeurs: Mathieu 113

Afterword des développeurs: Nicolas 114

Afterword des développeurs: Rémi 115

L'équipe



Marcel Gabbardo

Producer

Game designer

Level designer

Concept artist



Mathieu Brandao

Programmer

VisDev



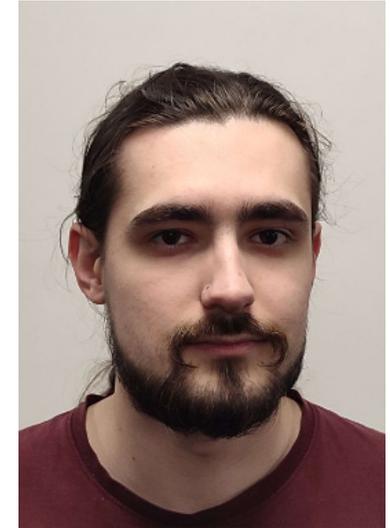
Nicolas Ferretti

3D artist

Concept artist

VisDev

Lighting artist

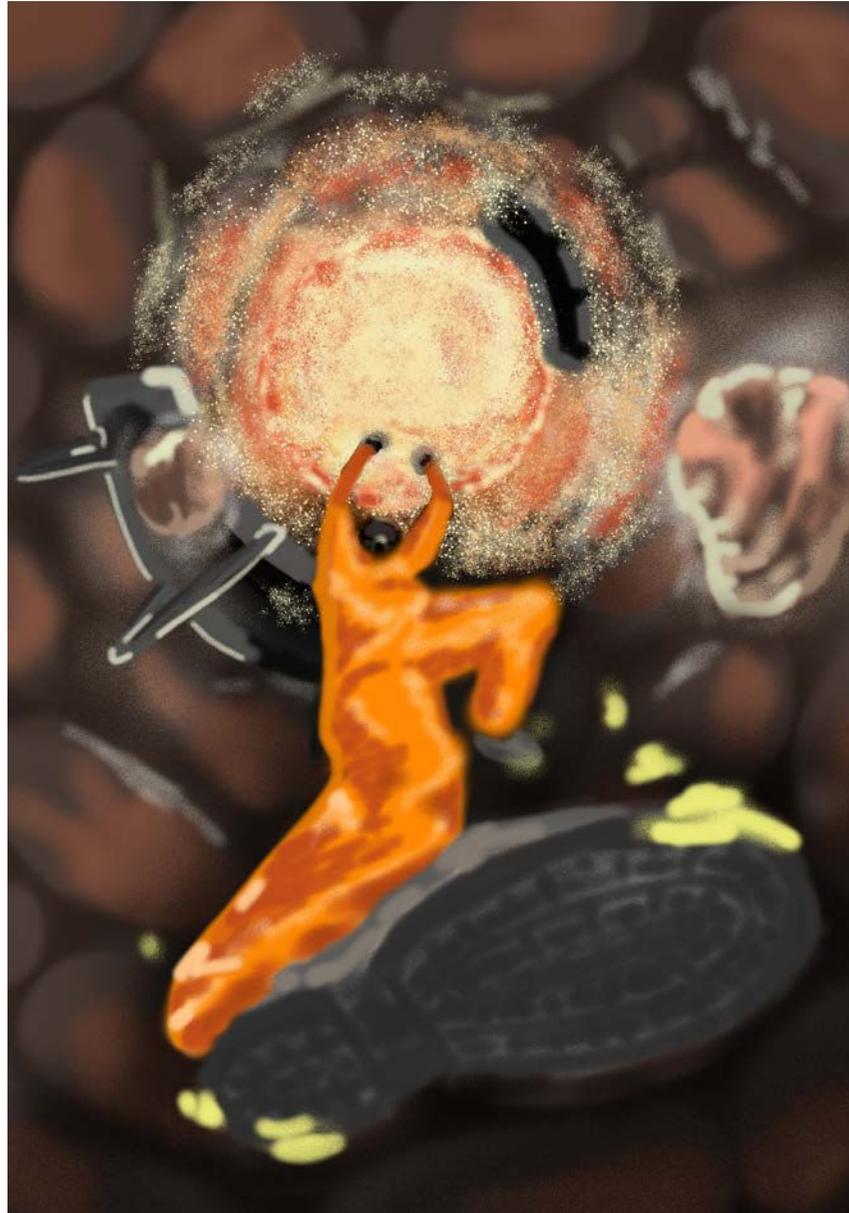


Rémi Mostura

Sound designer

Level designer

Pitch



Nettoyeur dans un puits sans fond, votre pause de 5 minutes est interrompue...

...par une ETOILE ?

QUI TOMBE DROIT DANS LE PUIT ?!

Vous vous retrouvez à descendre de façon effrénée, à sauter par dessus toute barrière, à glisser le long de plaques de métal rouillé.

- à tomber à travers des structures industrielles énormes construites par d'autres avant vous-

sans fin en vue, ni le moindre échappatoire...

Game Pillars



Sensation esthétique de chute

Le joueur va expérimenter une vitesse de chute qu'il connaîtra uniquement de ses rêves, de sports extrêmes, ou de rollercoasters riches en adrénaline. Il sera emporté dans une course folle et intense où le danger l'entourera.

La majorité de cette course consistera en la chute du joueur à travers l'espace à une vitesse croissante.



Pression et Intensité

La tension sera constante, le joueur étant d'une part poursuivi par une boule de feu létale, d'autre part en constant danger de perdre contrôle et de s'écraser à haute vitesse sur les structures qui peuplent le puit.

La clé sera pour lui de gérer sa vitesse et trajectoire tout en faisant un usage pertinent et bien informé des éléments de level design qui l'entourent afin de maintenir un état de jeu aussi confortable que possible.



Flow ininterrompu

Peu importe si le joueur échoue ou réussit, il n'expérimentera jamais une perte complète de momentum et sera souvent plutôt redirigé dans une nouvelle direction, et de nouvelles opportunités de continuer sa descente.

Si le joueur est interrompu ou décide de s'arrêter pour examiner ses possibilités : il sera toujours capable de récupérer la majorité de son momentum précédent dès après avoir sauté de la plateforme.

Fiche d'identité



Univers

Le joueur évolue à travers un environnement industriel sombre et peu accueillant ; vétuste et délabré.

Pitch narratif

Le joueur est un nettoyeur dans un puit sans fond. Sa pause cigarette est interrompue par une étoile qui tombe droit dans le puit.

Il ne lui reste qu'une seule option, descendre plus vite que l'étoile pour survivre ne serait-ce que quelques minutes de plus.

Catégorisation

Genre: Platformer, Falling game

Caractéristiques: First person, Arcade, Endless

Concept

Le joueur devra contrôler sa trajectoire et vitesse en interagissant de façon précise avec l'environnement qui l'entoure, afin de préserver un état de jeu confortable.

Cible

Ce jeu est destiné aux joueurs fans de challenges, et à la recherche de sensations fortes et immersives.

Interfaces

Le jeu est design pour le clavier et la souris, que la majorité des utilisateurs d'ordinateurs possèdent et qui leur confèrent la meilleure précision.

Overview 3C

Camera

Une caméra à la première personne à sensibilité customisable permet à tout joueur d'être immergé dans l'expérience de jeu confortablement.

Le but de cette caméra est de rappeler au joueur les vidéos bodycam de sports extrêmes, sans toute la rigidité de leur mouvement relatif au corps de l'athlète : ici, nous voulons que le joueur ai la complète illusion d'incarner le personnage.

Des effets de mouvement de caméra et de post processing renforcent ce sentiment d'immersion, et communiquent l'ensemble des informations liées à l'état du jeu et du joueur.

Controller

Design pour être joué au clavier et à la souris : le jeu produit des feedbacks immédiats à chaque input du joueur.

Ses inputs sont réduits au minimum (4 inputs de déplacement directionnel, 1 input de saut et les inputs directionnels de contrôle de la caméra), la profondeur du gameplay provenant d'interactions émergentes avec les différents types de structures de level design.

Character

Le joueur incarne un nettoyeur peu décrit, employé par une compagnie pour nettoyer un puit sans fond.

Le personnage est un humanoïde peu caractérisé. Il est une enveloppe jetable que le joueur utilise chaque partie, puis remplacé par une enveloppe identique à la suivante. Cependant, il est important de faire ressentir au joueur toute l'angoisse que son personnage subit en étant poursuivi par une gigantesque boule de feu.

Il peut résister à des impacts forts et se déplacer avec agilité : sa familiarité avec l'environnement apparente.

Le joueur peut le contrôler en le faisant se déplacer octo-directionnellement, et sauter. L'ensemble des autres actions "faites" par le personnage sont contextuelles et naissent par l'interaction entre le personnage et l'environnement.

Le collider du personnage est une capsule, ce qui lui permet de rentrer en collision avec l'environnement de façon cohérente sans s'y coincer.

Overview DA visuelle

Intentions

The Pit est un univers qui doit mêler à la fois angoisse et curiosité aux yeux du joueur.

Son improbabilité doit encourager le joueur à l'explorer, et à réfléchir aux usages des grandes structures qu'il explore.

De grandes structures mécaniques installées par d'autres avant lui composent le puits qui lui même, par surenchère de largeur : est sans fond. Cette profondeur infinie doit donner une impression de vertige considérable au joueur.

Les lumières du jeu doivent lui permettre de facilement lire son environnement sans le sortir de son immersion et de l'univers sombre du puits sans fond.

Références

Les références de The Pit servent en grande partie à la création de structures et d'une ambiance qui immergent le joueur dans un espace fictif vertigineux.

Nous y piochons des idées de textures et d'éclairage, ainsi que de formes mécaniques froides qui seraient cohérente avec l'univers industriel du puits.

Univers et Visuels

The Pit met en scène un puit sans fond dans lequel le joueur s'engouffre sans réel échappatoire. Nous voulons engendrer une peur chez le joueur avec une atmosphère qui ne l'accueille pas, et qui l'antagonise non seulement par sa taille démesurée mais aussi par sa morphologie générale.

Chaque pattern de level design raconte une histoire, et chacune de ces histoires vise à construire un monde froid duquel le joueur (par incarnation de son personnage) est prisonnier.



Overview DA sonore

Intentions

Les intentions de design sonore repose principalement sur deux aspects, la vitesse et la sensation de grandeur.

Quasiment tous les feedbacks liés au joueur évoluent avec sa vitesse et fonctionnent en tandem avec cette dernière.

Cette évolution des sons sert également, en plus de la sensation de vitesse procurée au joueur, à indiquer certaines données telles que la proximité au danger.

Les feedbacks possèdent également un traitement sonore de réverbération qui donne un sentiment d'écho, pour que le joueur comprenne mieux l'immensité du puit qu'il arpente.

Références

Les références sonores principales de ce projet servent principalement à l'ambiance du jeu, ainsi qu'à certains feedbacks particuliers.

Univers Sonore

L'univers sonore de The Pit est situé dans un puit sombre et sans fond et les feedbacks sonores le reflètent. Des sons de métal, de l'écho, l'étoile qui nous poursuit produit un son de boule de feu.



GAME DESIGN

TOY DESIGN

Intentions de design

Références de design

3C - Caméra

3C - Character

3C - Controller

Mechanic - Mouvement

Mechanic - Jump

Emergence constatée

Etat critique

Intentions de design

Rétention de vitesse, flow

Même si le joueur perd de la vitesse ; cette perte doit être justement proportionnée à son erreur de façon à éviter toute frustration chez le joueur. Egalement, il doit être en mesure de récupérer sa vitesse rapidement pour se relancer dans l'action.

Au delà d'un système de gain de vitesse accessible et d'un palier de vitesse minimum dont l'accès est facilité : il faut offrir au joueur des opportunités de gagner rapidement de la vitesse en interagissant pertinemment avec le monde.

Déplacement rapide mais contrôlé

La vitesse est l'aspect principal du jeu. L'ensemble des actions du joueurs vont avoir pour but d'accumuler et de retenir de la vitesse. Il doit être libre d'augmenter et de réduire sa vitesse de façon claire et accessible, et ses interactions avec le monde doivent refléter cette vitesse.

Le joueur doit pouvoir effectuer des techniques avancées de mouvement pour se déplacer de façon précise dans l'espace et parvenir de façon fluide et agréable aux endroits qu'il choisit.

Même lorsqu'il est dans les airs ; il faut lui permettre de se rediriger de façon plaisante. Au delà de toute nécessité de responsivité dans les contrôles ; il faut que ceux-ci soient agréables à utiliser et donnent au joueur une véritable sensation d'opération habile d'un objet à haute vitesse.

Perte de contrôle

Il faut que les déplacements du joueur soient contraints par une multitude de choses.

Premièrement ; on ne peut pas retirer à la sensation de vitesse une certaine inertie et irresponsivité sans lesquelles les mouvements sembleraient trop rigides et mécaniques.

Ensuite, il faut que les chutes du joueur lui donnent toute l'impression de vertige auxquelles elles sont connotées dans le monde réel ; leur aspect majeur étant une perte de contrôle (limitée par l'usage de différents outils).

Aussi, ses déplacements doivent être impactés et définis par l'espace et les objets qui l'entourent ; sans lesquels il n'y aurait ni sensation de vitesse, ni réel intérêt à se déplacer dans l'espace.

Il faut noter, en relation avec les références de game design choisies ; qu'il n'y a véritablement pas de sensation de déplacement sans contact et interaction avec l'environnement.



Références de design



Source engine movement maps

Le joueur peut accumuler de la vitesse à l'infini.

Bien qu'une multitude de chemins différents existent : les joueurs les plus efficaces auront tendance à sauter certaines plateformes à cause de leur vitesse.

Ceci est une bonne chose, une plateforme doit être juste cela : une opportunité offerte au joueur, plutôt qu'un point sur un chemin tracé.

Des outils sont donnés au joueur qui, uniquement lorsque mélangés à certains éléments de level design : lui permettent de gagner en vitesse ou d'effectuer des déplacements autrement impossibles.

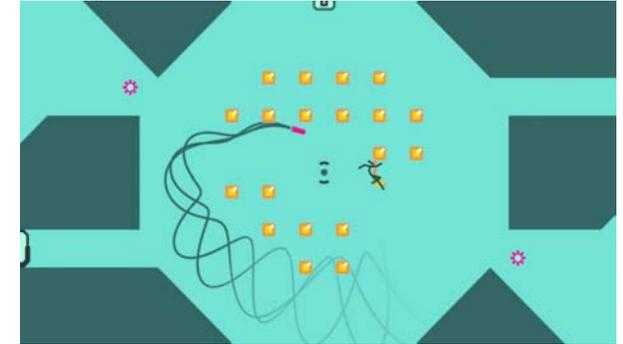


Titanfall 2

L'ensemble des mouvements du joueur sont définis par son interaction avec l'environnement et aucun d'entre eux (lorsqu'effectués efficacement) ne réduit sa vitesse de déplacement.

Des dispositifs existent qui permettent au joueur de rapidement gagner en vitesse, sans limite prédéfinie.

La rétention de la vitesse nécessite des inputs précis en succession rapide et une bonne perception des distances. Au delà des inputs d'ajustement de la trajectoire, les inputs déclencheurs d'actions du personnage liées au mouvement sont fréquents.



N++

Des inputs très simples suffisent à permettre au joueur des déplacements précis et des interactions complexes avec l'environnement.

Le joueur gagne en vitesse lorsqu'il descend une pente ou tombe, et peut la conserver en sautant sur une surface angulée. La direction de son saut est définie par la direction de l'input aussi bien que sa trajectoire actuelle, ou tout contact avec une plateforme angulée. Les dégâts de chute peuvent être annulés en évitant les collisions frontales et perpendiculaires avec des parois.

On arrive à la conclusion qu'il n'y a pas de déplacement véritable sans environnement, et donc sans interaction directe avec ce dernier.

Sensation esthétique recherchée

Pour transmettre l'idée de vitesse du joueur ; un parallèle pratique est le PoV GoPro de sports extrêmes. Avec cet angle de vue à FoV très exacerbé : on ressent une sensation de vitesse décuplée. Cette sensation, bien que factice : demeure la plus accessible et commune.

Un problème de cet angle de vue est sa rigidité ; qui ne fait pas ressentir les secousses de façon suffisamment viscérale (pas d'amortissage et d'inclinaison indépendants, attache à distance constante du corps de l'acteur).

Il est cependant nécessaire d'immerger le joueur dans le regard du personnage, et de lui faire ressentir toutes les pertes de vision, floutages, secousses et inclinaisons : qu'il subit en se déplaçant à haute vitesse et en rentrant en choc avec des parois dures.

Des mouvements de caméra ainsi que du post processing dynamiques nous permettront de retranscrire les mouvements et états du joueur, ainsi que l'état du monde qui l'entoure : et ce de façon élégante et intradiégétique.

Caméra à la première personne

Notre caméra doit donc être à la première personne. Elle tangue et se penche pour marquer les déplacements du joueur. Plusieurs effets de post processing sont appliqués à son rendu ; qui servent à renforcer la sensation de vitesse, aussi bien qu'à rendre certains éléments ou espaces de la vision du joueur plus ou moins intelligibles.

Mais pourquoi ?

Pour parvenir à des problématiques fortes d'immersion, il semble nécessaire de rapprocher la caméra de l'acteur.

La conclusion logique de ce raisonnement est l'agencement de la caméra et l'écriture de ses comportements pour donner une impression d'incarnation diégétique du personnage par le joueur.



Gestion de la charge cognitive

La gestion de la charge cognitive du joueur correspond à plusieurs préceptes de notre design.

Nous devons lui indiquer les informations essentielles sur les dangers qui l'entourent, ainsi que sur le level design qui l'attend sans le forcer à effectuer d'effort trop important.

Mais la composante vertigineuse de notre jeu doit être bien communiquée à travers ces mêmes feedbacks, qui auront donc tendance à augmenter la charge cognitive du joueur ou à troubler sa perception dans certains contextes.

Déplacements contrôlés

La caméra est contrôlée par le joueur avec la souris. Elle effectue une rotation relative aux déplacements de la souris sur les axes verticaux et horizontaux (relatifs au personnage joueur). Il peut regarder dans toutes les directions ; mais la rotation de la caméra est contrainte sur l'axe vertical afin de ne pas rompre l'immersion du joueur avec une torsion impossible du "cou" supposé du personnage.

Post processing

Le post processing esthétique ne sera pas détaillé dans cette section.

Lorsque le joueur est endommagé : un effet de post processing lui occultera partiellement la vision, et fera simultanément office de punition et d'indicateur intradiégétique.

Plusieurs effets de post processing combinés indiqueront au joueur lorsqu'il se rapproche –et excède– le cap de vitesse nécessaire à la prise de dégâts de chute.

L'étoile émet une lumière forte qui indique constamment sa proximité au joueur, même lorsqu'il ne la regarde pas.

Déplacements procéduraux

La caméra, pour communiquer les déplacements du joueur, doit :

Lorsqu'il se déplace latéralement vers la gauche ou la droite : effectuer une rotation sur l'axe forward du joueur dans la direction correspondante pour donner un feedback plus dynamique et marqué.

Lorsqu'il se déplace vers l'avant : se rapprocher du sol (et pas se pencher vers (...), cela réduirait la vision du joueur) pour donner l'impression que le personnage s'allonge vers l'avant pour courir aussi vite que possible.

Lorsque l'étoile qui le poursuit se rapproche trop du joueur : la caméra est secouée pour simuler les secousses du puit en contact avec l'étoile.

Paramètres basiques de la caméra:

Paramètre	Valeur
Field of View	101
Multiplicateur sensibilité souris (défaut)	x1
Multiplicateur sensibilité souris (min)	x0.1
Multiplicateur sensibilité souris (max)	x2



Paramètres mouvements de caméra procéduraux:

Paramètre	Valeur
Abaissement de la caméra lors de déplacement forward	-0.5
Vitesse d'abaissement	-1.4/sec
Inclinaison maximale lors de déplacement latéral	14°
Vitesse de rotation	32°/sec



Paramètres post processing dynamique:

Paramètre	Valeur
Bloom, intensité originelle	2.5
Bloom, intensité max	12
Bloom, intensité, croissance par seconde	7
Bloom, threshold originel	1
Bloom, threshold min	0.5
Bloom, threshold, réduction par seconde	0.5/sec
Vignette, intensité originelle	0.3
Vignette, intensité max	0.6
Vignette, intensité, croissance par seconde	0.2/sec



3C - Character

Lore du personnage

Le personnage joueur est un humanoïde, agent de nettoyage qui travaille dans un puit sans fond ; où il nettoie les déchets qui sont collés aux parois et s'assure de l'intégrité de la structure. Il est habillé conséquemment, de façon à ne pas se salir et à pouvoir respirer au sein des vapeurs nauséabondes qui s'accumulent (le trou n'est pas aéré).

Son histoire qui précède le jeu n'est pas racontée, mais toute ambition est écrasée par la volonté de survivre une fois le jeu lancé.

Il se déplace rapidement et avec agilité et est naturellement résistant aux impacts et peut tomber de haut sans se blesser. Le personnage est habitué à son travail, et parcourt efficacement les échafaudages qui l'entourent.

Besoins, volontés de design

Il est nécessaire que le personnage se détache bien de l'environnement qui l'entoure, tout en rentrant dans la thématique industrielle. Cela doit aussi être le cas dans son viewmodel, qui communique son état. Il doit pouvoir effectuer des mouvements complexes et agiles.

Aussi, il ne doit pas être trop caractérisé : c'est un personnage qui meurt et meurt encore dans un jeu d'arcade, sans dire mot ou interagir avec le monde autrement qu'en se déplaçant.

Morphologie

Le personnage est humanoïde, de taille moyenne (175cm). Il porte des vêtements amples anti-contamination qui dissimulent entièrement son corps. Ses vêtements lui permettent d'effectuer des mouvements complexes sans restriction.

Compétences

Le personnage peut:

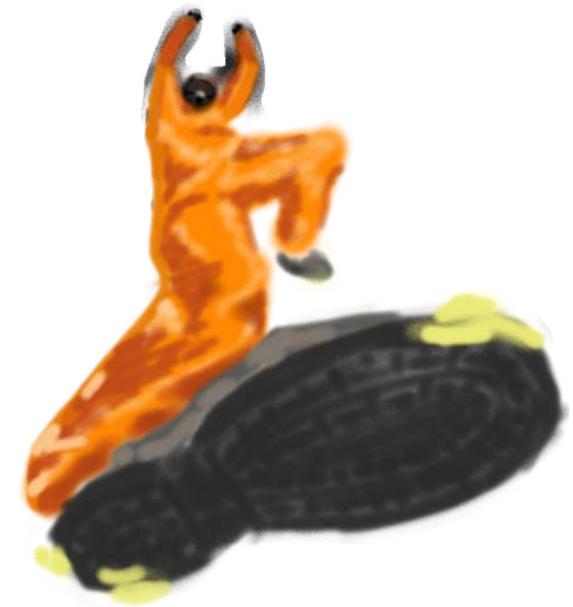
Se déplacer en courant dans toutes les directions

Sauter haut

Regarder autour de lui

Collider

Le collider du personnage joueur est un collider capsule qui lui permet des collisions consistentes même lorsqu'il se retourne, ainsi que d'interagir de façon intuitive avec d'autres colliders de formes variées.



3C - Controller

Interfaces supportées

Le jeu supporte principalement les inputs mouse & keyboard. Cette interface permet au joueur d'effectuer des déplacements précis en changeant rapidement d'input, et en utilisant la souris pour s'orienter

Ils permettent à la démographie majoritaire respective de Steam et itch, qui sont les plateformes principales de publication indépendante de jeu vidéo, de jouer au jeu s'il y est un jour publié.

Cependant, le jeu pourrait gagner à employer une manette de console ; dont le panning smooth de caméra donne une sensation de survol et de planeur très agréable. Aussi, les déplacements analogues donnent au joueur un contrôle directionnel plus précis et moins mécanique qui pourrait bénéficier au gamefeel du jeu.

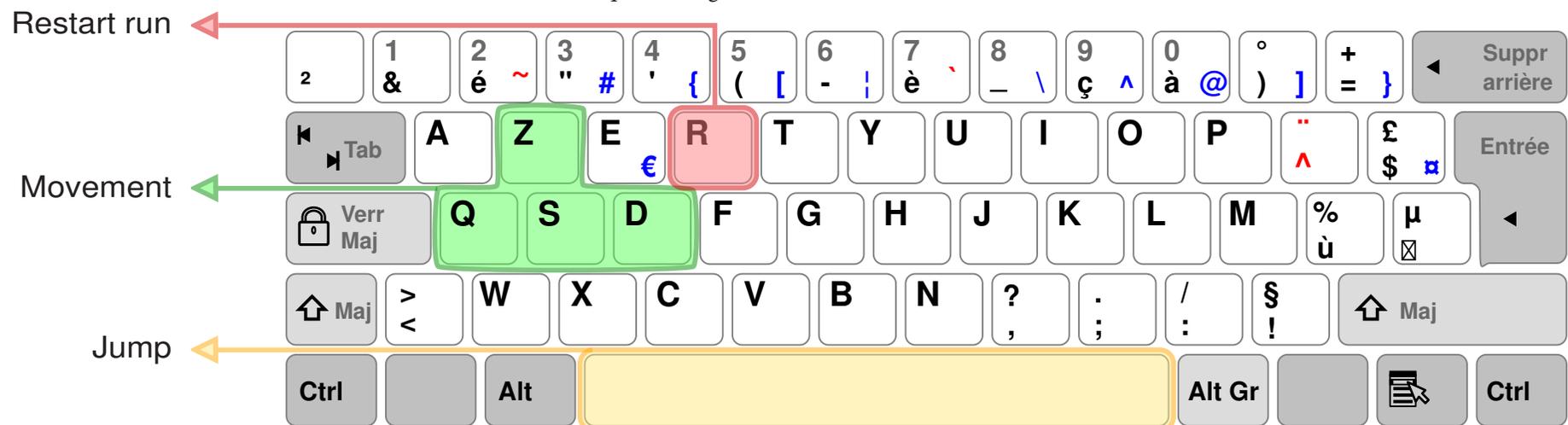
Sensation esthétique de contrôle

Les contrôles doivent répondre immédiatement aux inputs du joueur et être accompagnés de feedbacks clairs et distincts.

Le joueur doit pouvoir être précis dans tous ses déplacements qui doivent être entièrement exempts de quelconque frustration, mais doit tout de même ressentir la résistance viscérale de l'environnement, de la physique du jeu et principalement des limitations humaines de son avatar. Les extrêmes, que ce soit l'immobilité ou la haute vitesse, doivent être rapidement atteignables.

Les transitions directionnelles doivent se faire de façon fluide pour faciliter l'accès à un état de flow et le joueur ne doit jamais être désorienté à cause de saccades trop puissantes ou d'une déconnection entre sa trajectoire et ses inputs ainsi que la direction dans laquelle il regarde.

Chaque seconde passée à effectuer même les déplacements les plus simples doit être agréable et prenante : car là est le sujet principal du jeu.



Mechanics - Mouvement

Mouvement dans 8 directions

Le joueur peut se déplacer octo directionnellement, dans les directions cardinales et intermédiaires relatives à sa propre rotation.

La transition d'une direction à l'autre ne se fait pas instantanément, mais avec une inertie légère (cela évite d'invalider la sensation de momentum accumulé, ou de donner un feedback trop saccadé qui pourrait être désagréable).

Lorsqu'il se déplace au sol ; la vitesse de déplacement du joueur suit un cap prédéfini qui dépend de la direction dans laquelle il se dirige.

Vers l'avant

Le cap de vitesse est le plus élevé lorsque le joueur se déplace vers l'avant (s'il se déplace diagonalement, le cap de l'input directionnel forward/backward est utilisé).

Même si le cap de vitesse est plus élevé ; le taux de gain de vitesse n'est pas augmenté et le joueur prends plus longtemps à atteindre le cap.

En arrière

C'est lorsque le joueur se déplace vers l'arrière qu'il est le plus lent. Ce type de mouvement existe principalement pour d'éventuels ajustements précis ; le joueur doit être encouragé au possible à aller vers l'avant et à sauter, puis réfléchir ensuite.

Latéral

Avec un cap de vitesse intermédiaire, les déplacements latéraux du personnage ont pour but principal d'ajuster sa trajectoire et devraient être principalement utilisés en combinaison avec le déplacement vers l'avant.

Lorsque le joueur se déplace en diagonale, le cap de vitesse "avant" ou "arrière" est utilisé, plutôt que le cap "latéral".

Lorsqu'airborne

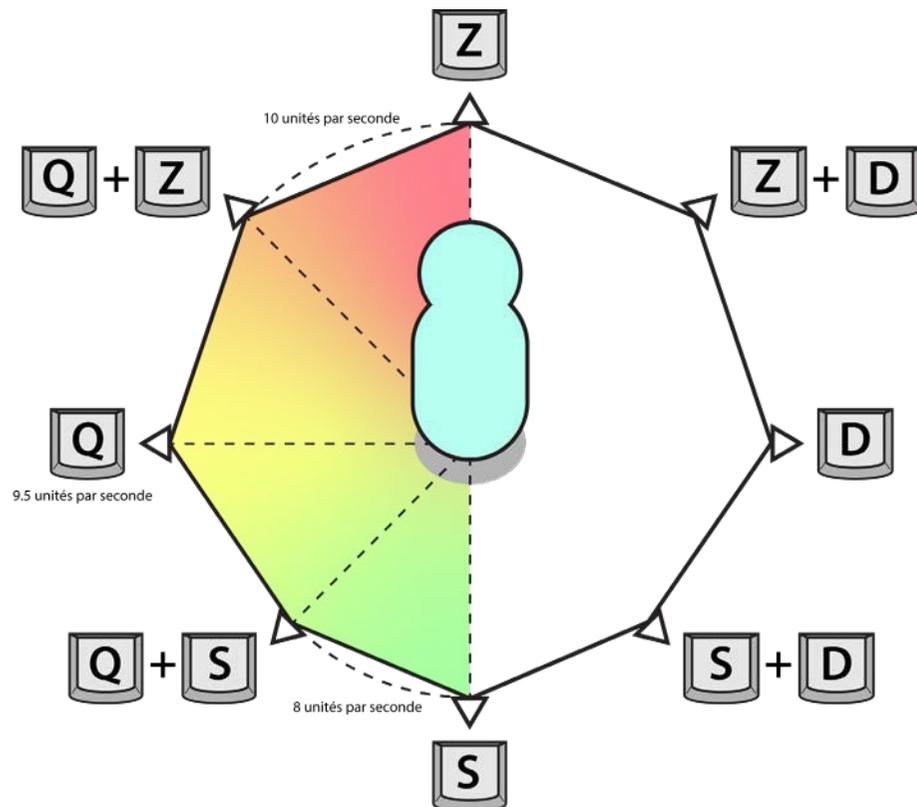
La vitesse de redirection horizontale du joueur est réduite tant qu'il est airborne : lui donnant une impression de résistance aérienne et réduisant l'aspect irréel et mou d'un caractère contrôlé traditionnel.

Cependant, le cap de vitesse est plus élevé pour permettre au joueur de faire plein usage de la vitesse qu'il a accumulé et de se rediriger sur des chutes plus longues : où il prend plus le risque de prendre des dégâts.

Il peut transférer sa vitesse verticale (de chute) en vitesse horizontale en sautant sur des pentes.

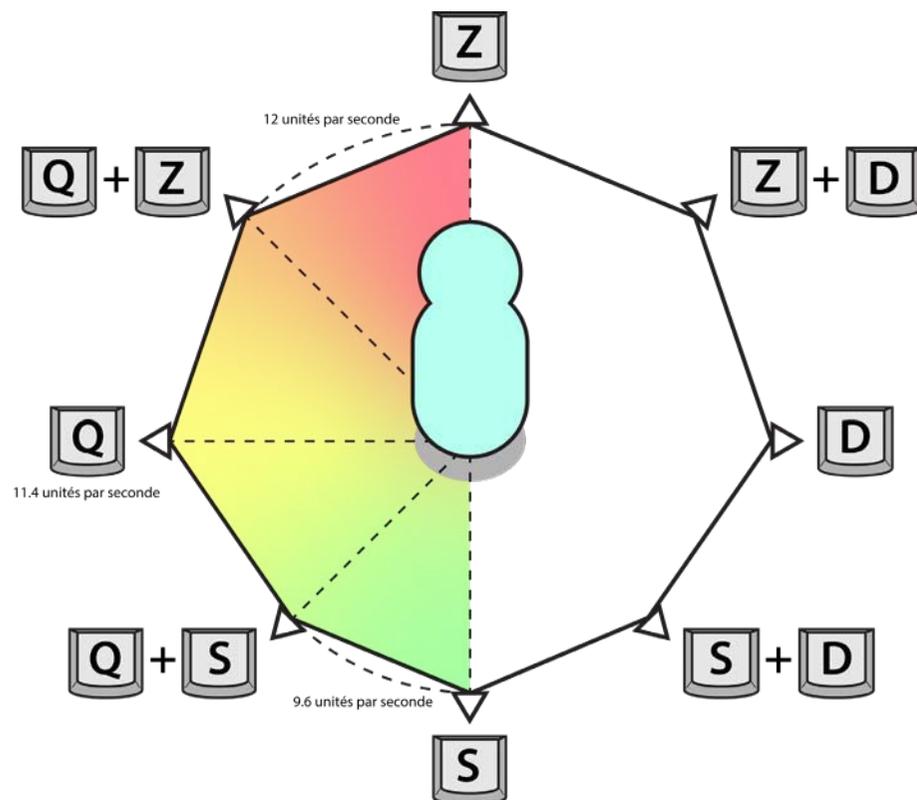
Paramètres mouvement grounded:

Paramètre	Valeur
Cap de vitesse grounded	10
Accélération	9/sec
Multiplicateur de mouvement latéral	x0.95
Multiplicateur de mouvement arrière	x0.8



Paramètres mouvement airborne:

Paramètre	Valeur
Cap de vitesse horizontale	12
Accélération	9/sec
Multiplicateur de redirection horizontale	x0.4



Mechanic - Jump

Intentions de design

Les déplacements grounded du joueur avec les 4 touches directionnelles correspondant à une pure nécessité dans le character controller : nous voulons que le saut fasse office d'un outil versatile et évolutif que le joueur pourra (par le simple clic d'un bouton) utiliser pour effectuer une variété de mouvements complexes et émergents.

En appliquant le même travail sur le game feel que nous avons apportés au mouvement grounded, nous entreprenons dans cette mechanic d'incorporer des fonctionnements et interactions profondes sans complexifier son usage basique.

Pour ce faire, nous prenons en compte l'interaction principale et la plus variée du jeu : celle du collider du joueur avec des pentes (dont l'inclinaison peut être extrêmement variée). Également, nous nous intéressons aux mécanismes émergents qui font du saut un outil de gain de vitesse agréable.

Force verticale

Le saut consiste en une impulsion importante vers le haut. Ce dernier emmène le joueur jusqu'à une certaine hauteur très rapidement, mais le met également dans un état de gamefeel faible où il flotte.

Pour palier à ça, nous augmentons fortement (de environ 50%) la vitesse de chute du joueur dès qu'il a atteint l'apex de son saut, lui permettant de rapidement atteindre un premier palier de vitesse de chute (correspondant à 30% de sa vitesse de chute maximale).

Distance horizontale parcourue

Lorsqu'il saute, le joueur subit également une impulsion modérée dans la direction de son forward qui fait du saut un outil plus versatile, qu'il peut par exemple utiliser pour aller plus loin ou tomber plus vite, ou même se rediriger rapidement.

Cette impulsion est augmentée lorsque le joueur est en état de sliding pour renforcer le flow et lui permettre d'accumuler plus de vitesse (en partie par potentielle impulsion contraire à celle qui le fait aller vers le haut).

Evolution contextuelle

Pour contribuer au flow de la glissade du joueur le long des pentes, et pour augmenter la versatilité de l'action du saut : la direction considérée "up" sur laquelle l'impulsion verticale est ajoutée est relative à l'angle de la plateforme sur laquelle se tient le joueur.

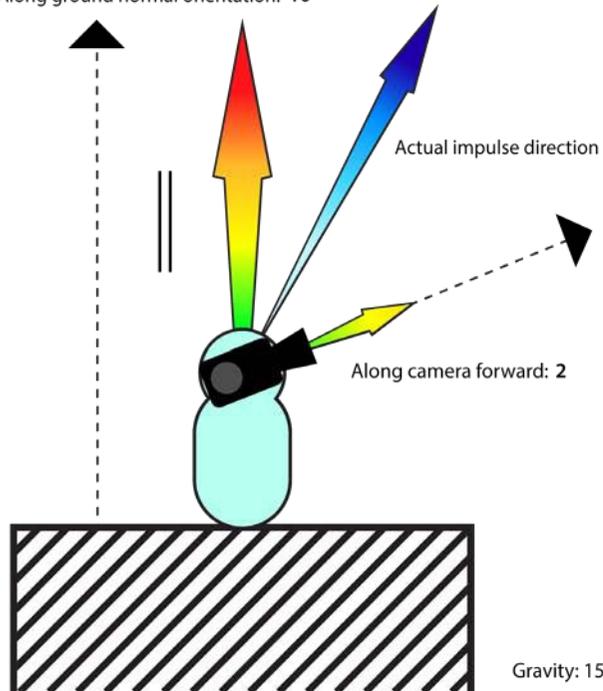
Couplée à l'impulsion sur le forward de la caméra du joueur : cette inclinaison du saut renforce drastiquement le flow de façon intradiégétique (la prise de momentum par saut n'annule pas le momentum déjà obtenu), et force le joueur à réfléchir différemment au saut aussi bien qu'à l'inclinaison de chaque bloc de level design.

La gravité est décuplée lorsque le joueur glisse, pour lui permettre d'y gagner plus de vitesse de chute et pour assurer qu'il reste bien collé à la pente peu importe son inclinaison.

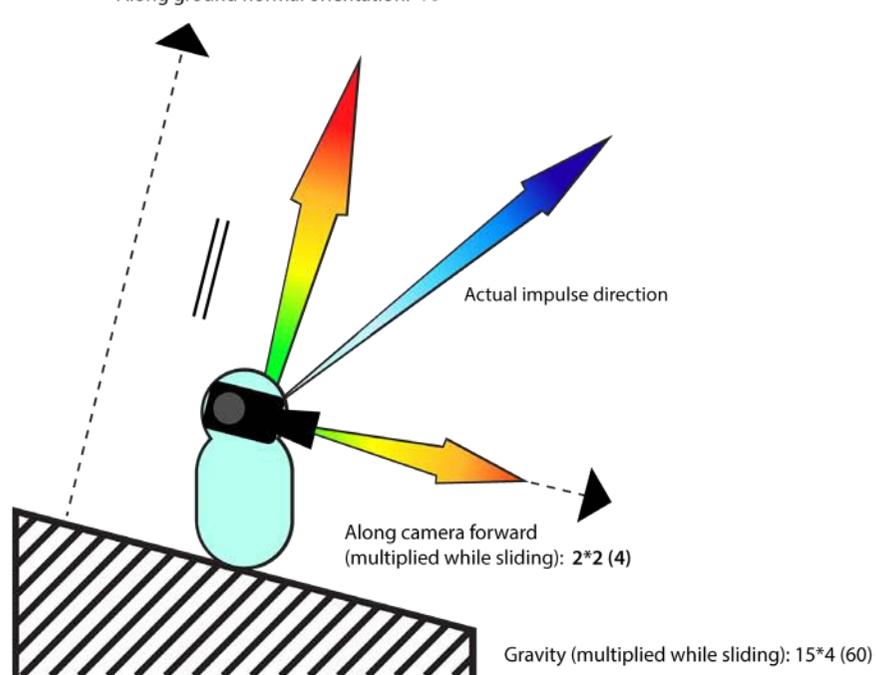
Paramètres saut et gravité:

Paramètre	Valeur
Force upward du saut	10
Force forward du saut	2
Multiplicateur de force forward (while sliding)	x2
Gravité	15
Multiplicateur de gravité (while sliding)	60

Along ground normal orientation: 10

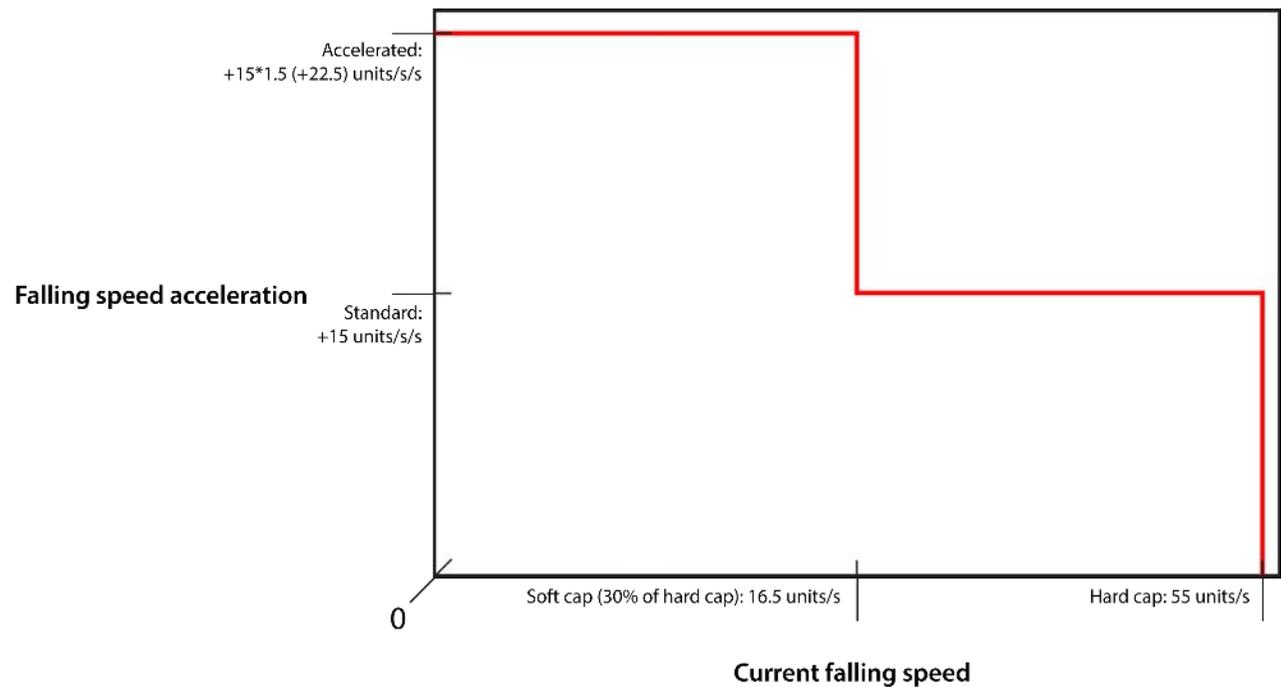


Along ground normal orientation: 10



Paramètres vitesse de chute:

Paramètre	Valeur
Soft cap de vitesse de chute (30% du hard cap)	16.5
Hard cap de vitesse de chute	55
Gravité standard (lorsqu'au dessus du soft cap)	15
Gravité augmentée (x1.5 lorsqu'en dessous du soft cap)	22.5



Emergence constatée

Wallrun/ Walljump

Le collider trigger qui vérifiait l'état airborne du joueur était positionné de telle façon que le joueur était considéré comme "grounded" au contact avec n'importe quelle surface ; murs et plafonds compris.

Cela signifiait que ce dernier pouvait maintenir la touche de saut pour se transformer en véritable balle rebondissante.

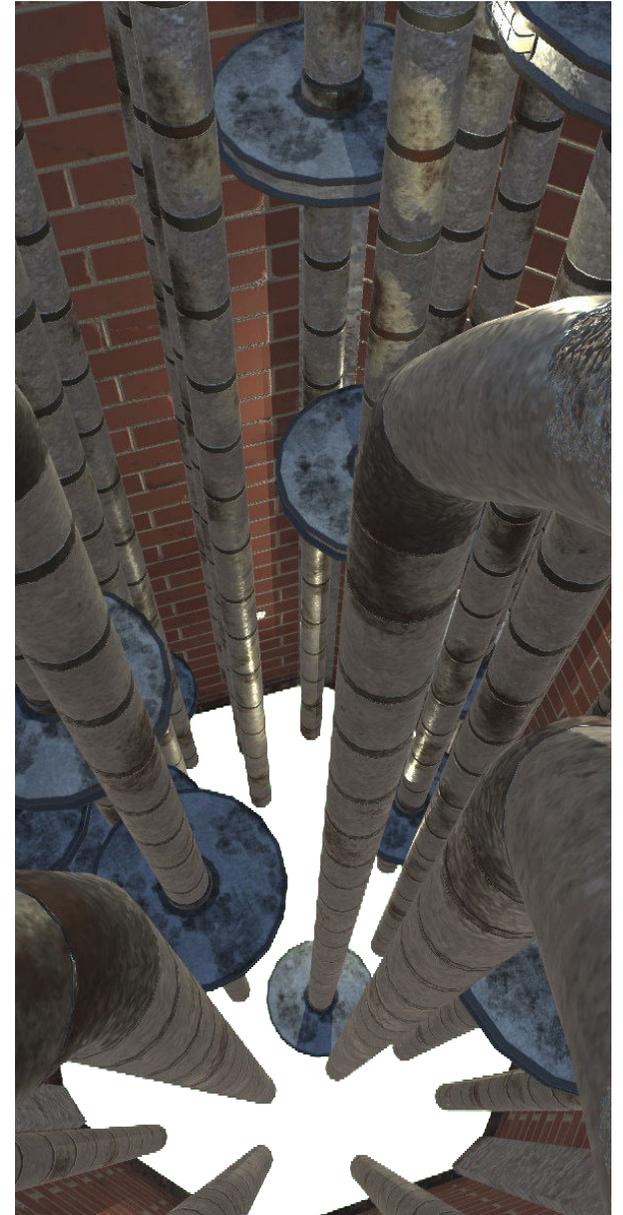
Le gamefeel qui en découlait correspondait parfaitement à l'interaction proche entre le joueur et l'environnement que l'on recherchait, mais allait à l'encontre de nos buts pour le caractériser. Nous avons donc décidé de conserver cette dynamique transformatrice du saut selon la surface touchée, tout en la limitant aux pentes sur lesquelles le joueur pourra véritablement se poser.

Slide clipping

Lorsque le joueur tombait exactement sur le bord d'une pente, il était projeté très rapidement dans sa chute et atteignait instantanément une vitesse parfois trop élevée pour que les colliders ne l'arrêtent.

Nous avons décidé de modérer ce comportement dans la mesure du possible pour réduire les bugs de collision, mais prenons en due considération la récompense méritée par un joueur qui exécute un contact parfait avec l'environnement.

Pour conserver cette dynamique, nous intégrons le système de saut en cours de glissade de façon à ce que les sauts du joueur effectués en fin de glissade lui permettent de gagner plus de vitesse.



Etat critique



Obstruction des déplacements

L'état critique atteint à l'instar de la volonté du joueur est celui où ses mouvements sont bloqués par le level design.

Ce dernier se sent alors presque comme enlisé dans de la boue ; où il doit se débattre pour récupérer sa liberté.

Malgré l'intérêt d'apporter une résistance aux mouvements du joueur, c'est un moment de gamefeel faible où l'attrait de la vitesse de chute importante n'est pas présent.

Perte de contrôle

Lorsque le joueur atteint une certaine vitesse de chute (différente selon le contexte dans lequel il se trouve : des briques de level design qui le redirigent et altèrent sa trajectoire forçant une perte de contrôle prématurée) : il commence à perdre le contrôle de son personnage par vertu du temps réduit entre plusieurs éléments de level design.

Alors, il doit être très attentif aux opportunités qui se présentent à lui pour se rediriger par interaction avec le level design.

GAME DESIGN

Intentions de design

Cible

Points forts du toy extraits

Conditions de victoire et de défaite

Boucles OCR

Boucle de gameplay

Timeline de gameplay horizontale

Game structure

Signes et feedbacks

Matrice de Caillois

Intentions de design

Pousser le joueur vers la descente

La mécanique centrale ajoutée au jouet pour en faire un jeu est un équivalent de timer qui transforme sa descente en course poursuite.

Alors, toutes les problématiques de bon usage de l'environnement sont sublimées et deviennent essentielles au jeu, et plus seulement des options à la disposition du joueur.

Bien qu'être un ajout à part entière d'une nouvelle mécanique, le timer existe en accordance parfaite avec la tendance principale du jeu, et l'option la plus facilement atteignable par le joueur.

Ce timer est lancé principalement pour exercer une pression sur le joueur, et le pousser à continuer sa descente. Les redirections qu'il subit en tombant de pente en pente sont suffisantes pour l'obliger à allonger sa descente jusqu'à la prochaine plateforme stable.



Récompenser la bonne lecture de l'espace

En parallèle avec ces nouvelles problématiques : la bonne compréhension de l'environnement qui entoure le joueur devient un challenge non seulement d'observation, mais aussi de vitesse d'observation et parfois de réaction.

Les indications doivent être claires, ou servir une difficulté en bonne cohérence avec leur utilité ou rôle. Cependant, dans un jeu à forte prétention vertigineuse, les informations doivent être communiquées de façon à rendre le joueur confus et le forcer à voir outre un voile qui lui floute (figurativement la vision).

Par exemple, le danger imminent représenté par l'étoile pourra éblouir le joueur ou noyer les formes du level design dans sa lumière : le rendant plus difficile à lire et augmentant la tension ressentie par le joueur.



Préservation et récupération

L'interaction avec l'environnement est salvatrice lorsque bien effectuée ; le joueur pouvant interrompre sa chute et ralentir en entrant en contact avec ce dernier (que ce soit pour prendre moins de dégâts, ou pas de dégâts du tout), ou se sauver de la prise de dégâts en tombant précisément sur certains blocs de level design spécifiques.

Pour préserver le flow du joueur à travers ces interactions principalement ralentissantes, nous récompensons leur bonne exécution (et l'accès à des espaces difficiles à atteindre) par des blocs de level design qui permettent au joueur de rapidement reprendre la vitesse perdue. Ces blocs de level design doivent aussi être utilisés aux endroits où les joueurs ont de grandes chances d'échouer, pour modérer leur frustration.



Cible

Difficulté

La difficulté de base du doit être suffisamment basse pour permettre à tout joueur qui respecte ses mechanics centrales d'y jouer agréablement.

Cependant, la prise de risque doit être récompensée, et cette récompense doit être un biais d'allègement de la charge cognitive du joueur.

Le jeu doit donc être une expérience facile de prise en main qui offre aux joueurs des opportunités d'alléger certains aspects de la difficulté en confrontant des challenges augmentés sur d'autres aspects.

Durée de jeu

Une partie doit durer au moins 6 minutes.

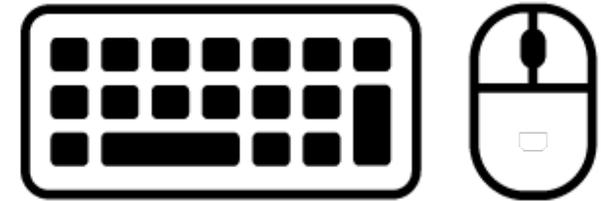
Ces 6 minutes doivent être libres de toute interruption frustrante et fluides, et offrir au joueur toute l'étendue de l'expérience du jeu. Au delà de ces 6 minutes, la charge cognitive accumulée ainsi que la fatigue doivent commencer à rendre le jeu plus difficile.

6 minutes est une durée de jeu qui bénéficie à ce genre de jeu arcade, et qui pourrait –lorsqu'il contiendrait plus de contenu, et par renouvellement cyclique des contenus à des étapes prédéfinies de progression– permettre au joueur de sample l'ensemble de ses selling points avant de l'emmener plus en profondeur dans les systèmes de jeu.

Interfaces utilisateur

Le clavier et la souris ont été choisis pour donner accès au jeu à quiconque possède un ordinateur.

La souris permet également de se rediriger rapidement, et confère au joueur un contrôle précis sur sa trajectoire.



Points forts du toy extraits

Accumulation de vitesse

L'accumulation de vitesse du joueur n'est pas la tendance du jeu, mais demeure la poursuite principale du joueur.

Cette tension lors de la prise de vitesse n'est pas sans intérêt, mais signifie que les moments d'échec et de réussite sont tous deux à basse tension.

Nous décidons donc de nous orienter vers la facilitation du gain de vitesse de chute (pour que la majorité du gameplay se produise aux plus hautes vitesses possibles), en ajoutant plutôt des contraintes externes qui en font un état à forte tension innée.

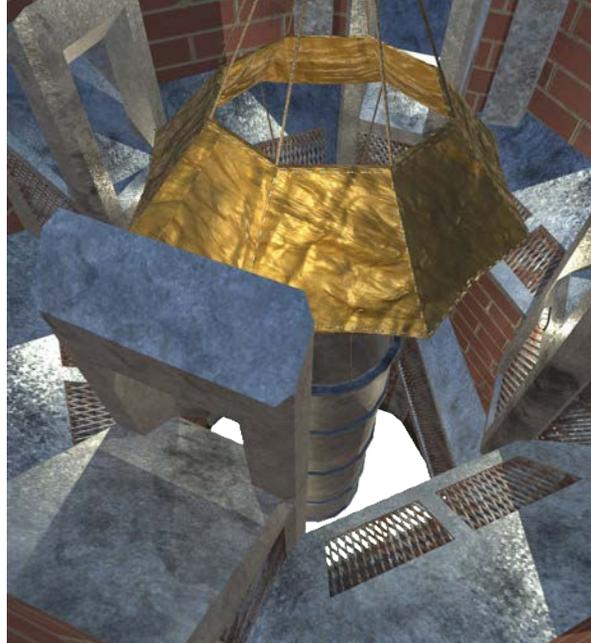


Intensité et vertige

En due proportion à l'étendue du vertige ressenti par le joueur lors de ses chutes les plus rapides : nous devons ajouter une mechanic qui l'y pousse à force égale.

Même si ce dernier craint le vertige et la perte de contrôle : ces derniers doivent être son unique option.

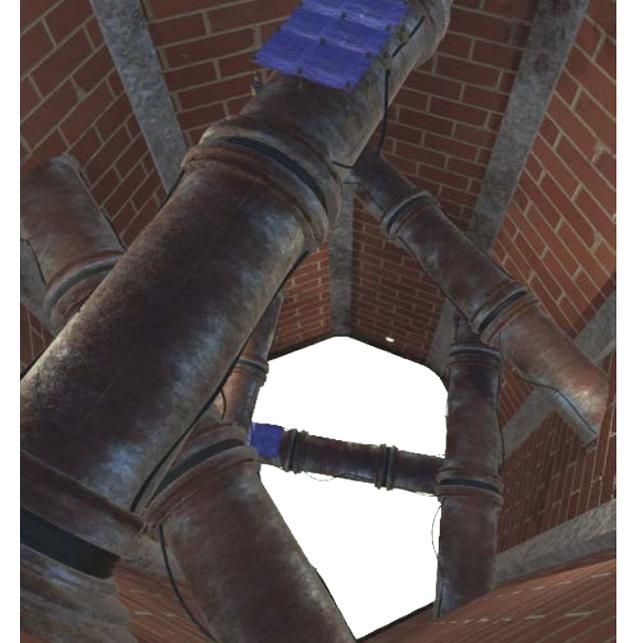
Lorsque le joueur perd contrôle : le retour à un état neutre (après lequel il entreprend de récupérer sa vitesse de chute perdue) est ressenti presque comme une renaissance, un retour à la vie.



Interactions émergentes

Ne voulant pas ajouter de nouvelles mechanics liées aux contrôles du personnage : nous avons décidé d'augmenter et renforcer les options du joueur purement en travaillant davantage sur les interactions avec l'environnement.

Ces dernières ne comportent pas l'inclusion de nouvelles mechanics, et tout travail qui leur est porté correspond à une amélioration directe et tangible des piliers porteurs de notre character controller.



Conditions de victoire et de défaite

Condition de défaite 1: Immolation

Une gigantesque boule de feu (étoile, soleil) poursuit le joueur à haute vitesse : et l'oblige à tomber toujours plus vite pour lui échapper.

Sa proximité est indiquée au joueur par une source de lumière puissante qui semble presque dévorer l'environnement, ainsi que des effets de caméra shake.

La vitesse de chute de l'étoile est dynamique, lui permettant de ne jamais être abandonnée complètement par le joueur tout en ralentissant lorsqu'elle s'en approche afin de le garder dans un état de tension sans apporter trop de difficulté.

Celle ci tue le joueur lorsqu'il rentre en contact avec elle, et sert de motivateur unique (et suffisamment dramatique pour y suffire) à la continuation de la descente dans le puit.

Paramètres étoile:

Paramètre	Valeur
Vitesse minimum	12.5
Vitesse minimum jusqu'à distance	25
Vitesse de base	30
Vitesse de base atteinte à distance	60

Condition de défaite 2: Chute létale

Nous devons maintenant nous tourner vers comment modérer la descente du joueur tout en maintenant un degré élevé de tension.

Notre approche est d'ajouter des dégâts de chute au jeu, qui sont régis par deux facteurs. Le joueur meurt lorsqu'il a accumulé deux dégâts de chute, mais ces derniers se résorbent après une courte durée.

Premièrement, le joueur doit avoir atteint un certain cap de vitesse pour prendre des dégâts et un autre cap de vitesse (qui coïncide avec sa vitesse maximale) pour mourir instantanément lors du contact avec le level design.

Deuxièmement, le joueur doit avoir déjà chuté pendant une durée ininterrompue de 1.8 secondes, afin de lui permettre de tomber aussi vite qu'il le souhaite : tant qu'il tombe d'élément de level design en élément de level design.

Paramètres dégâts de chute:

Paramètre	Valeur
Vitesse de prise de dégâts du joueur	35
Vitesse de mort du joueur	55
Temps de sécurité	1.8
Temps de buffer de dégât	0.5
Temps de résorption des dégâts	5

Victoire et récompense

Malgré l'absence de victoire possible : la réussite des actions du joueur équivaut à une multitude de résultats.

Premièrement, la stabilisation (ou maintien de la stabilité) de l'état de jeu le plus confortable. Celui ci est principalement celui où le joueur est à une distance suffisante de l'étoile, sans pour autant être en risque de prise de dégâts de chute.

Cependant : jouer de façon méthodique et se préserver en sécurité par rapport aux dégâts de chute oblige le joueur à rester en proximité élevée de l'étoile (de laquelle il ne va pas assez vite pour s'éloigner). Cela engendre des feedbacks éblouissants et une nuisance cognitive qui viennent rompre la tranquillité du joueur.

Alors, celui ci peut décider d'alléger sa charge cognitive première en accélérant : ce qui équivaut à une autre augmentation de la charge cognitive du joueur qui doit désormais faire attention à ne pas prendre de dégâts de chute, et réagir suffisamment rapidement aux obstacles.

Boucles OCR

Court terme

Objectif	Challenge	Reward
glisser	atteindre une pente, challenge d'input ainsi que d'approche du level design	Gain de vitesse de chute, sécurité et gestion pleine de la vitesse du joueur
ralentir	interagir avec l'environnement de façon pertinente se poser sur une plateforme, ou glisser	sécurité par rapport aux dégâts de chute, repos du joueur, allègement de charge cognitive
se rediriger	s'aligner diligemment avec une surface pour y rebondir, et faire bon usage des contrôles	atteinte d'une partie du level design autrement inaccessible, reward appropriée
atteindre un bloc de LD proche	se déplacer correctement, éviter les obstacles sur son chemin, bien comprendre les metrics	usage contextuel du bloc de LD pour un but particulier
time son saut pour une tech (sauter sur un élément incliné vers le bas pour prendre en vitesse)	se positionner correctement & appuyer au bon moment (positionnement + timing)	gain de vitesse fulgurant

Moyen terme

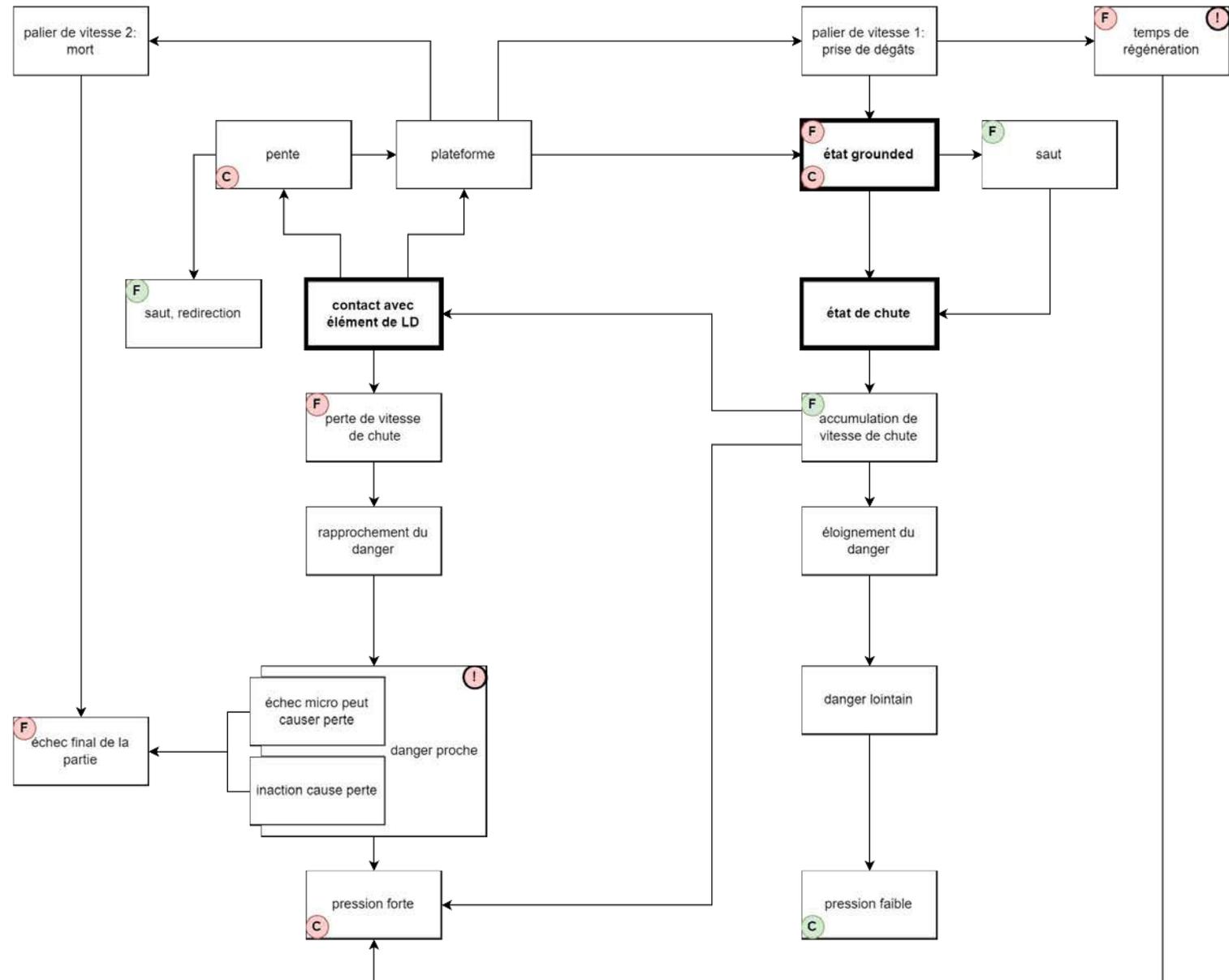
Objectif	Challenge	Reward
gagner en vitesse de chute	interagir avec les patterns de LD, tout en s'assurant de ne pas prendre trop de dégâts	Eloignement de l'étoile, gain rapide de distance parcourue
atterrir sur un lifesaver	s'aligner avec le lifesaver en avance, connaître sa position ou réagir rapidement à son apparition	Secour du joueur, toute vitesse de chute annulée, mais tout risque de prise de dégâts annulée
s'éloigner de l'étoile	réussir à gagner suffisamment de vitesse de chute, et à la maintenir pour ne pas s'en rapprocher	Décroissance de la charge cognitive, game feel et lisibilité renforcés
traverser un pattern de LD	planifier ses déplacements en bonne connaissance de la forme globale du pattern	parvenir au pattern suivant, renouvellement du gameplay et nouvelles opportunités
maîtriser un mouvement particulier	atteindre l'endroit où ce mouvement est employé suffisamment constamment pour le cibler	progression tangible, première étape de la première étape vers la maîtrise
établir un plan de sa route à travers le LD visible	reconnaître les blocs de LD visibles et comprendre les metrics	ne pas perdre en vitesse / ne pas prendre de dégâts

Long terme

Objectif	Challenge	Reward
battre un high score	survivre suffisamment longtemps, et aller suffisamment vite pour ne pas utiliser trop de temps	augmentation du nombre displayed sur death screen, out rewards
maîtriser une route sur un pattern	parvenir au point d'entrée de la route, et l'entreprendre avec des angles et vitesses variés	progression de la maîtrise, facilitation de l'atteinte de meilleurs scores
maîtriser un pattern (ensemble des routes possibles)	explorer le pattern sous tous ses angles, et prendre en due considération ses connexions aux autres patterns	progression drastique de la maîtrise, qui facilite potentiellement drastiquement l'accès aux lifesavers
maîtriser les mouvements (sauts/slides/techs/etc.)	essayer de se détacher d'habitudes établies, et explorer des opportunités qui se situent au delà de ses routes habituelles	possibilité de réagir à toute situation dangereuse sous réserve de compréhension des paramètres qui définissent le danger

Boucle de gameplay

- (F) impact flow
- (C) impact charge cognitive
- (●) impact positif
- (●) impact négatif
- (!) état critique



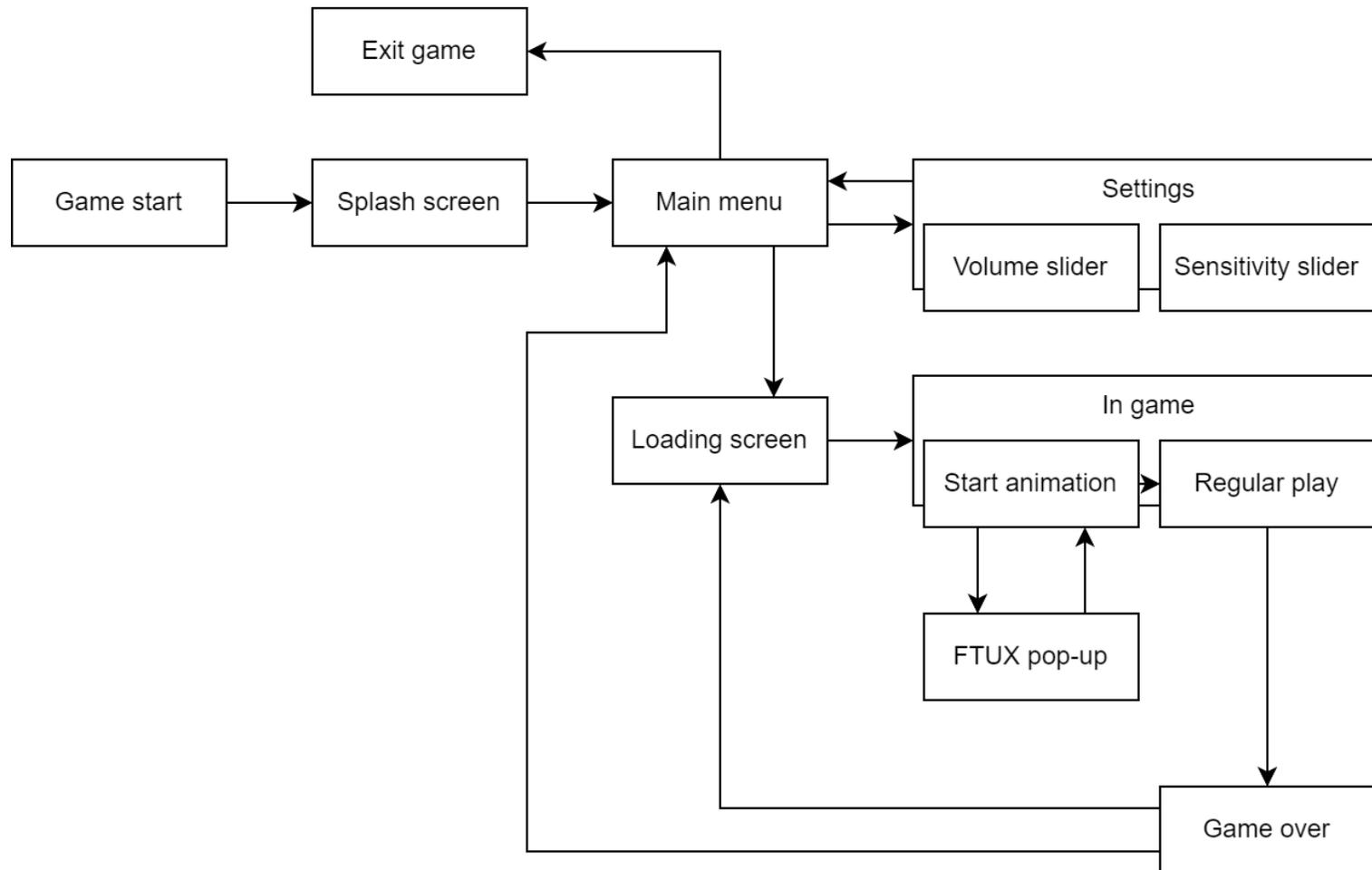
Timeline de gameplay horizontale

Explication

La timeline de gameplay horizontale est un outil de Game Design que nous avons mis en place au cours de ce projet, et qui a pour but de cataloguer les éléments de jeu et leur impact (sur la gamestate et sur la charge cognitive du joueur) au sein de chaque étape de gameplay. Celle ci nous a servi pour rationaliser l'impact cohérent de chaque feedback, ainsi qu'à identifier les états critiques du jeu.

Gameplay Phase	Grounded state		Transition to airborne state		Airborne state		Transition back to grounded state	
	Game element	Impact	Game element	Impact	Game element	Impact	Game element	Impact
Inputs	Directional inputs Mouse look	Light to Medium Medium	Jump input	Light to Medium	Directional inputs Mouse look	Medium Heavy		
Visual Information	Movement feedback Environment reading Star feedback Motion blur	Light Medium to Heavy Void to Heavy Void to Heavy	Movement feedback	Light	Falling speed indicator Movement feedback Motion blur	Medium to Heavy Light Void to Heavy	Impact feedback Damage taken	Light to Heavy Heavy
Audio Information	Footsteps Star proximity Sliding	Light Void to Heavy Light	Jump sound	Light	Falling speed indicator Platform grazing Fainting	Medium to Heavy Heavy Heavy	Impact feedback Impact dampener Damage taken Death to fall damage Life saved by dampener Landing	Light to Heavy Medium Heavy Heavy Heavy Medium
Deep Interactions	Momentum gain Collisions Sliding	Medium Medium to Heavy Heavy	Jump angles	Void to Heavy			Collisions Sliding Fall damage	Heavy Heavy Void to Heavy
Knowledge	Metrics Interactions	Medium to Heavy Medium to Heavy			Metrics	Heavy	Speed preservation	Light to Heavy
Reflection	Trajectory planning	Heavy			Trajectory planning	Heavy		
Pressure	Meteorite proximity Damage taken	Medium to Heavy Void to Heavy	Meteorite proximity Damage taken Clarity of descent path	Medium to Heavy Void to Heavy Light to Heavy	Meteorite proximity Damage taken Fall damage risk	Light Void to Heavy Light to Heavy	Meteorite proximity Damage taken Understanding of landing spot	Light to Heavy Void to Heavy Void to Heavy
Illustration								

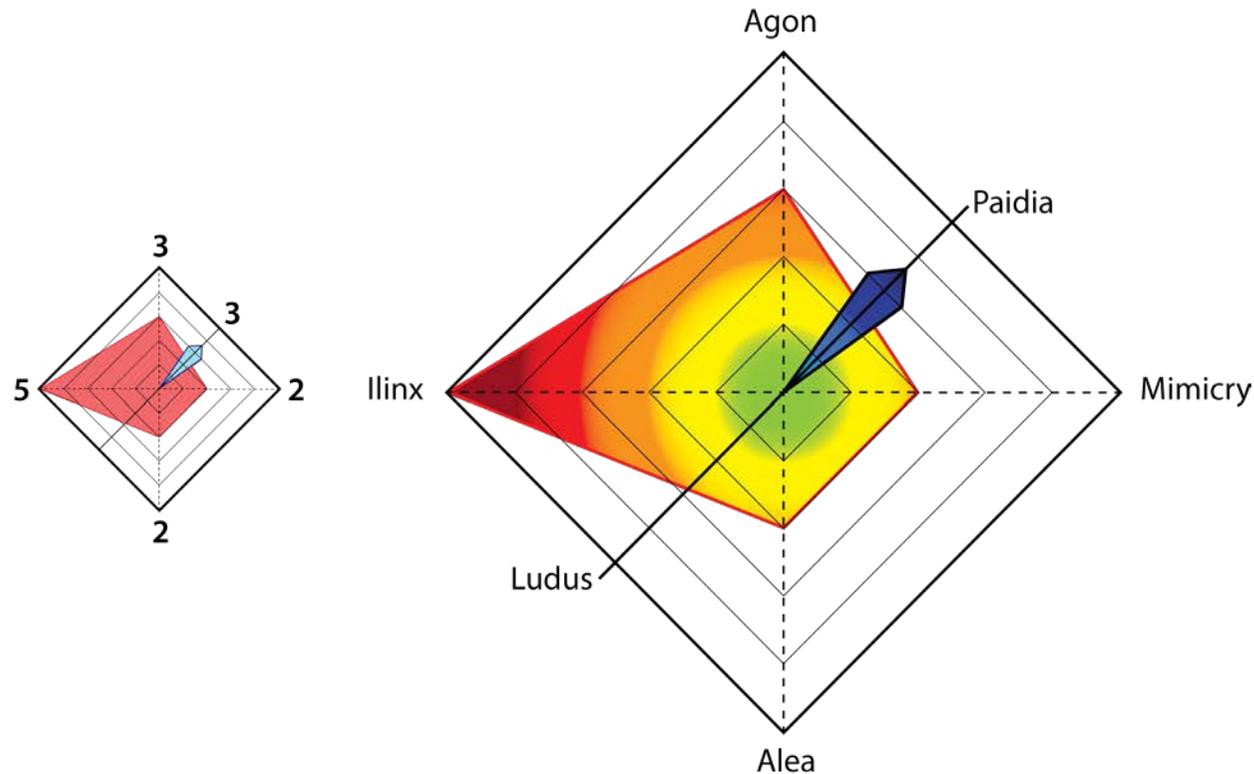
Game structure



Signes et Feedbacks

Type	Catégorie	Signs	Event, Objets concernés	Feedbacks
Visual	Player Camera (Movement)		Forward movement, grounded Player (Camera)	Camera lowering
Visual	Player Camera (Movement)		Lateral movement, grounded Player (Camera)	Rotation in input direction, along player forward axis
Audio	Player Movement (Grounded)		Footsteps Player, Ground	Repeated sounds, change in volume, type and speed according to movement speed
Audio	Player Movement (Sliding)		Slide Player, Slope	Continuous sound, changing * player speed
Audio	Player Movement (Grounded)		Jump Player, Ground	Single sound
Visual	Player Camera	Camera shake (position) perpendicular to player's forward axis	Star proximity Player (Camera)	
Both	Meteorite	Post-processing and sound * distance to star(after a certain threshold)	Star proximity Star, Player	
Visual	Meteorite	A light source is attached to the star, very strong, engulfs whatever is within range	Burning (light source) Star	
Both	Player Camera (Post-Process)	Fainting, scaling with time (after reaching threshold) - resets if speed goes under threshold	Falling speed threshold exceeded Player (Camera, post-process)	
Audio	Player Movement (Airborne)	Wind resistance continuous noise * falling speed	Falling ambiance (+ mov. feedback)	Wooshing wind noises to indicate player will take damage
Audio	Player Movement (Airborne)		Collision with environment (impact)	Clanking sound * falling speed
Audio	Health and Death		Fall damage Player, Ground	Damage SFX (bone crushing, clothes ruffling, flesh ripping) + tinnitus
Both	Health and Death	Video and audio filters last while player is damaged	Damaged State Player	
Both	Health and Death	"Damaged State" filters fade out, and a sound effect plays	Health Regen Player	
Both	Health and Death		Death to fall damage Player	Sound effect plays, other sound effects stop, Game Over screen
Audio	Health and Death		Death to Meteorite Meteorite, Player	Sound effect (visuals are already applied due to proximity)
Audio	Player Movement (Airborne)		Fall on impact dampener Player, Impact dampener	Sound effect when player falls into an impact dampener
Visual	Scene visuals	Fog effect	Distance filter Camera	Fog strength increases * distance travelled this run
Visual	Player Camera		Game Start Camera	Camera locked looks up to star, then looks down to player forward with slight headbob
Visual	Environment		Lighting Lamps	Pointlight on lamp

Matrice de Caillois



Agon: Le joueur n'est pas en compétition avec autrui, mais il est victime d'une tension constante et croissante. Egalement, la présence d'un système de scoring offre au jeu une composante compétitive par la comparaison des scores.

Alea: La génération du niveau est procédurale, et la perte de contrôle du joueur peut l'envoyer sur des trajectoires inattendues. Cependant, ces aspects ne définissent pas la boucle de gameplay : elles forcent seulement le joueur à s'adapter.

Ilinx: Les feedbacks et metrics du jeu sont construits autour de l'altération dramatique de la perception du joueur. La vitesse étant le coeur du jeu, elle est traduite par des effets visuels et sonores drastiques.

Mimicry: L'entièreté des feedbacks sont communiqués de façon intradiégétique, et il n'y a pas d'UI visible en jeu. Cependant, le joueur ne peut pas voir son personnage dans le jeu et ce dernier n'est pas caractérisé.

Paidia: La player agency est au coeur du jeu. Même si ce dernier possède des modalités d'action limitées : l'interaction entre ces dernières et l'environnement donnent naissance à un corpus très vaste d'options à tout moment donné.

Ludus: Mais le joueur doit tout de même respecter des règles strictes, sous réserve de sa propre mort. Cela vient pencher la balance vers le Ludus, sans tout de même la renverser.

LEVEL DESIGN

Philosophies de level design

Types de surfaces

Système de génération

Pattern n°1: Tree of Life

Pattern n°2: Pillars of Creation

Pattern n°3: Hanging Gardens

Pattern n°4: Clusters

Pattern n°5: Citadel

Pattern n°6: Coliseum

Pattern n°7: Crown

Philosophies de level design

Liberté du joueur

Il est nécessaire que le joueur ai plusieurs option à sa disposition à tout moment donné, pour éviter que le jeu ne devienne un rail runner glorifié.

Il faut éviter les sauts obligatoires ou inutiles, afin que le joueur s'oblige à lire sans cesse son environnement et à ne jamais le négliger.

Flow note : Pas de mauvais choix

Aucune des propositions de trajectoire ne doit activement interrompre le joueur. Plutôt, l'emploi des trajectoires optimales doit générer un gain de vitesse visible qui récompense les joueurs.

Il ne faut pas que la prise du bon choix génère de latence ou d'interruption du flow.

Opportunités visibles

Les chemins alternatifs proches doivent être immédiatement visibles pour que le joueur puisse rapidement décider de les employer ou non.

Ceux qui sont plus lointains doivent également être visibles, afin de tease d'autres éventualités au joueur.

Flow note : Pas de regrets

Les signes de ces opportunités doivent être facilement reconnaissables pour que les joueurs ne passent pas au delà à leur prochaine rencontre.

Alors, ils peuvent anticiper leur arrivée et s'aligner préalablement avec eux.

Opportunités atteignables

Il faut que le joueur puisse se déplacer horizontalement à une vitesse suffisante pour atteindre des chemins alternatifs sans trop le détourner de sa descente.

Flow note : Pas de détours

Les déplacements horizontaux ne doivent pas avoir d'impact négatif trop important sur la prise de vitesse de chute. Le joueur ne doit pas être puni pour son exploration des opportunités qui lui sont proposées.

Usage #1: Obstacle

L'environnement peut avoir un impact négatif sur l'état du joueur de deux façons:

Premièrement, il peut le ralentir et le coincer momentanément ainsi que contraindre ses déplacements. Ceci ne doit arriver que sur des courtes durées, mais quand même résulter en une situation critique pour le joueur.

Secondairement, en lui infligeant des dégâts de chute. Une mauvaise approche du niveau correspond à l'ignorer –que ce soit en n'évitant pas les obstacles, ou en les évitant trop souvent et atteignant une vitesse critique–.

Usage #2: Lifesaver

Si l'atteinte d'un cap de vitesse critique peut résulter en l'échec du joueur, l'environnement qui a tendance à le ralentir peut lui permettre d'éviter d'atteindre cet état critique.

Usage #3: Redirection

Lorsque le joueur est au sol, il possède d'avantage de contrôle sur sa direction de déplacement. L'interaction avec l'environnement peut également rapidement rediriger le joueur.

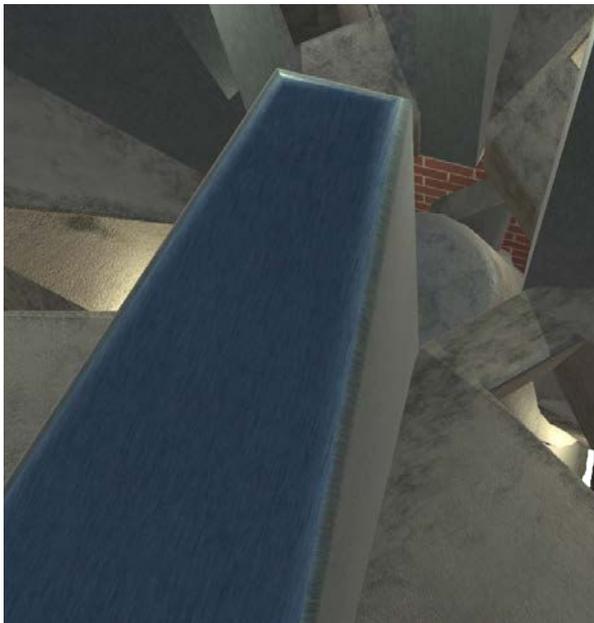
Types de surfaces

Ground

Le ground est la plateforme la plus basique du jeu.

Elle permet au joueur de s'arrêter et de se rediriger librement, tout en le ralentissant progressivement jusqu'à atteindre l'immobilité s'il n'effectue pas d'input de déplacement.

Egalement, l'état grounded lui applique un cap de vitesse normatif auquel il sera abaissé même dans le cas d'inputs de déplacements.



Slide

Le slide est une plateforme à basse friction, qui n'a pas tendance à ralentir les déplacements du joueur. Ce dernier peut cependant être immobilisé si le bas d'une pente coïncide avec un autre collider.

La slide comporte une behavior qui permet aux joueurs qui se déplacent dessus de rapidement gagner en vitesse de chute, et d'avoir plus de contrôle sur la direction dans laquelle leurs sauts les propulse.

Les slides peuvent apparaître sous plusieurs formes, mais seront presque tout le temps inclinées afin d'éviter que le joueur soit "stranded" sur un slide plat ; comme un poisson hors de l'eau.



Lifesaver

Le lifesaver est une surface avec une propriété qui lui est exclusive, et très importante à la gameplay loop : celle de réduire à néant la vitesse de chute du joueur, sans lui infliger de dégâts de chute.

Ce dernier permet au joueur de (peu importe sa vitesse de chute) se sauver de la prise de dégâts, avant de rapidement reprendre sa route. Il est utilisé principalement soit comme modération de la difficulté dans un pattern où les joueurs ont tendance à prendre des dégâts trop souvent, ou à des emplacements difficiles à atteindre lorsqu'on va trop vite, pour récompenser les joueurs qui préservent leur sang froid dans un état critique.



Systeme de generation

Patterns

Nous employons une logique de pattern qui nous permet de concevoir des structures predefiniees et design intentionnellement, avant de les inclure dans un systeme de generation procedurale qui les agence les unes par rapport aux autres principalement aleatoirement.

Ces patterns sont de hauteur normee et nous nous efforcons de maintenir la duree de chacun d'entre eux dans une meme tranche de temps.

Chaque pattern apporte au joueur une experience differente, et celui ci doit s'adapter rapidement a leur apparition et les apprendre au fur et a mesure qu'il joue au jeu afin de les parcourir plus efficacement.

Aléatoire

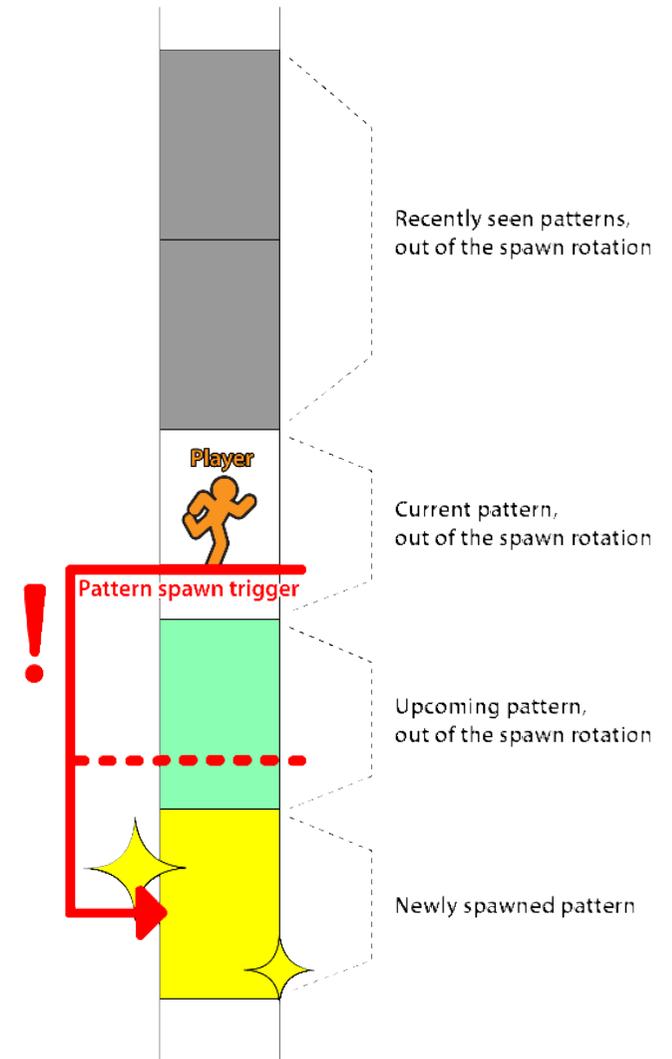
Notre systeme de spawn de pattern n'est pas entièrement aleatoire : il choisit en effet parmi les patterns qui ne font pas partie des 3 derniers que le joueur a pu explorer, afin d'assurer le bon renouvellement du gameplay et d'eviter qu'un meme pattern apparaisse en chaîne.

La rotation des pattern est modifiee de façon incrementale et aleatoire a chacune de leur apparition pour diversifier davantage les transitions entre un pattern et le suivant.

Mesures de sécurité

Afin d'assurer des transitions fluides d'un pattern a l'autre, et d'eviter au joueur toute situation "dechech et mat", nous construisons les structures internes a chaque pattern de façon a ce qu'elles puissent supporter le joueur de l'entree a la sortie du pattern, sans trop definir sa trajectoire.

Ce dernier est alors capable de se diriger dans la bonne direction pour atterrir sur le nouveau pattern sans prendre de degats, sous reserve de sa connaissance dudit pattern.



Pattern n° 1: tree of life

Intentions de level design et niche

Une structure unique qui supporte les déplacements du joueur sur une hauteur importante, et à laquelle il pourra rester proche le temps de ce pattern : ce dernier a pour but de servir de lifeline au joueur.

Dans ce pattern, nous explorons la design space d'un environnement presque bi-dimensionnel, où le pylône central déplié est presque comparable à un chemin linéaire (qui connaît cependant quelques embranchements). Le joueur peut cependant décider de s'en détacher pour sauter certaines étapes du pattern, et chuter jusqu'aux prochaines.

Flow principal

Le joueur va principalement glisser le long de plateformes, ce qui va alléger sa charge cognitive (il n'a besoin que de se rediriger légèrement pour rester dans l'axe des pentes).

Autrement, le joueur peut décider d'employer des chemins optionnels qui lui demanderont des inputs précis afin d'être pratiqués.



Modulation possible

L'angle des pentes et leur longueur peuvent être altérés, ainsi que leur largeur.

L'agencement vertical de ces dernières: la hauteur à laquelle elles se situent relative aux pentes précédentes et suivantes (ce paramètre a un impact considérable sur le game feel, car des chutes plus courtes forcent le joueur à se rediriger plus rapidement), ainsi que leur alignement avec ces dernières (impact fortement contextuel à évaluer au cas par cas) peuvent également être modifiés.

Visuels

Ce pattern doit être un silo, ou un objet qui semble important et justifie premièrement les éléments de level design qui lui sont attachés, puis ensuite l'importance que l'on veut que le joueur lui voue.

Narration

Ce pattern est le plus technologiquement chargé de tous, et doit donc nécessairement être plus solide. Cependant, les éléments de pattern extérieurs qui lui sont attachés peuvent être vétustes et détruits.

Pattern n°2: pillars of creation

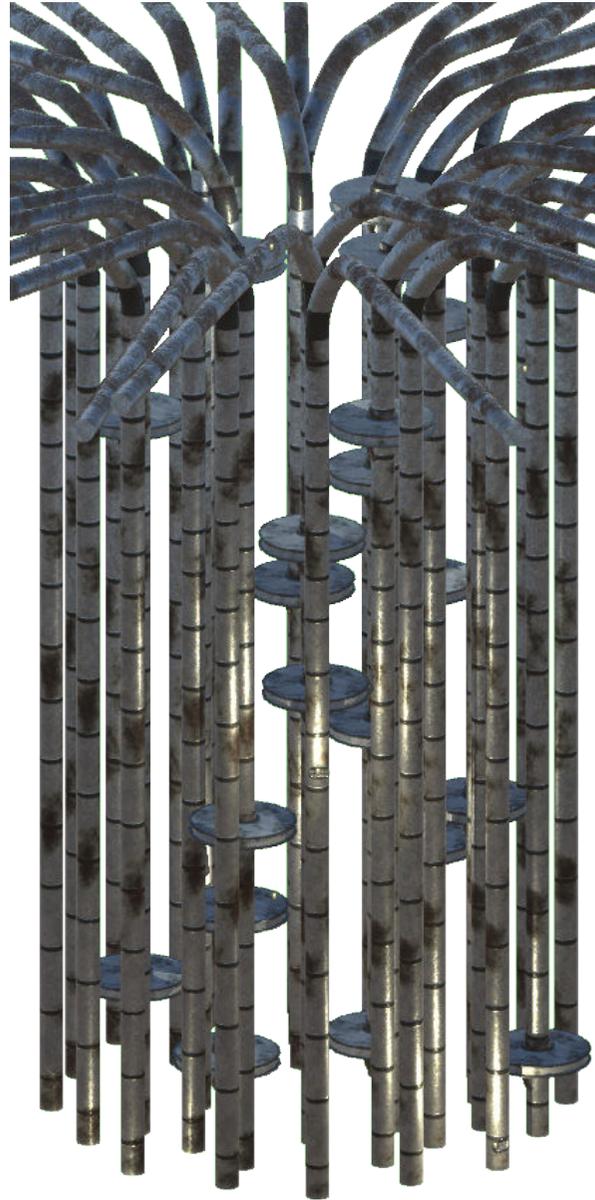
Intentions de level design et niche

Le but de ce pattern est de faire obstruction aux déplacements du joueur, verticaux aussi bien que horizontaux. Simultanément, il doit offrir au joueur une multitude de plateformes qu'il peut employer ou sauter comme bon lui sent.

La niche remplie par ce pattern est de le forcer à zigzaguer et à rapidement se rediriger horizontalement pour aller de plateforme en plateforme.

Flow principal

Le joueur est encouragé à se diriger vers le coeur du pattern, où un grand nombre de plateformes sont à sa disposition. Si ce dernier ne saute pas de plateformes, il prendra excessivement longtemps à traverser ce pattern. Autrement, il devra faire usage de sa compréhension des metrics du jeu pour déterminer s'il peut sauter des plateformes sans prendre de dégâts.



Modulation possible

L'agencement des tuyaux ainsi que l'orientation de leur partie horizontale peut moduler la difficulté que le joueur aura à s'y faufiler.

Les plateformes sont ici agencées pour offrir au joueur des chemins logiques et variés, faciles à suivre. Cependant, leur disposition peut être modifiée pour convenir à d'autres buts (par exemple, une spirale de plateforme pourrait obliger le joueur à déplacer sa souris de façon particulière, et engendrer un gamefeel tout nouveau).

Visuels

Visuellement, les éléments de ce pattern sont individuellement simples : mais leur multiplicité et proximité cause un effet de profondeur où chaque tuyau fait office de ligne de fuite.

Narration

La césure spontanée des tuyaux est une piste insuffisamment explorée que nous souhaitons développer davantage pour donner une dimension "détruite" ou "sectionnée" au pattern.

Pattern n°3: hanging gardens

Intentions de level design et niche

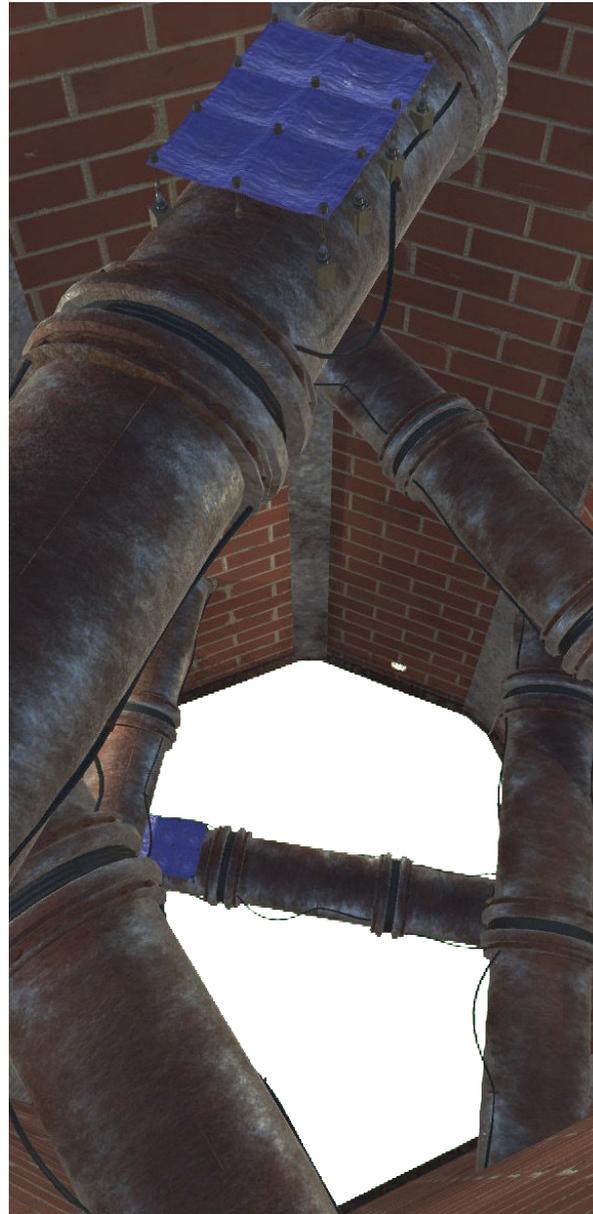
Ce pattern représente l'exploration d'un design space autrement non exploité dans le projet: celui de la plateforme courbée polyvalente. Les tuyaux font en effet office de plateforme lorsque le joueur tombe droit dessus, et de pente lorsqu'ils tombent sur leur extrémité.

Ce pattern représente une opportunité rare pour le joueur de décider de se déplacer horizontalement pour s'aligner avec précision avec la prochaine plateforme, ou d'en sauter pour accumuler plus de vitesse.

Flow principal

Le joueur glisse de tuyau en tuyau, et est récompensé avec un gain de vitesse proportionnel à sa précision.

S'il ne parvient pas à interagir avec les tuyaux de façon pertinente: il peut aussi tomber droit jusqu'aux lifesavers disposés en contrebas.



Modulation possible

L'agencement des tuyaux est libre. Leur position et alignement les uns par rapport aux autres peut avoir plusieurs impacts sur la facilité et fluidité des déplacements du joueur (qui ne sont pas à confondre : forcer le joueur à aller jusqu'aux limites de ses metrics de déplacement pour y atteindre à coup sur une plateforme nous permet de créer des chemins difficiles mais "absolus", fiables).

L'inclinaison des tuyaux est à éviter, car elle résulte en des collisions désagréables lors de l'arrivée en bas des tuyaux: là où ils rejoignent la paroi du puit.

Visuels

Ces tuyaux doivent faire figurer des détails qui permettent d'éviter de leur donner une apparence trop simple, surtout en considération de leur grande taille.

Narration

Ces tuyaux sont anciens et partie intégrante du puit : mais ils ont survécu au passage du temps et semblent toujours remplir leur rôle.

Pattern n°4: clusters

Intentions de level design et niche

Ce pattern est un chemin courbé, que le joueur devra prendre avec plus ou moins de caution selon l'angle dans lequel il y entre. Au delà de ce chemin sont situés des chemins de rattrapages alternatifs, qui vont permettre au joueur de glisser pour accumuler de la vitesse et se garder à l'abris des dégâts de chute avec une difficulté proportionnelle à sa distance de la coupe à laquelle les poutres concernées sont rattachées.

Ce pattern récompense fortement le joueur qui arrive à se positionner intelligemment et à rapidement réagir pour rejoindre la partie avantageuse de la structure.

Flow principal

Le joueur glisse à l'intérieur d'une coupe, vers la suivante. S'il dérape et ne parvient pas à y pénétrer : il peut se rattraper sur la coupe suivante, ou les poutres qui les supportent.



Modulation possible

Détruire l'alignement des coupes détruit simultanément le flow du pattern, qui réside dans la capacité du joueur à interagir avec les parties moins pratiques de la structure pour rejoindre les plaques de métal, ou même les coupes en aval.

La rotation des supports des coupes peut avoir un impact positif sur le flow du pattern en réduisant les espaces vides, mais qui réduira la récompense offerte au joueur qui parvient à se raccorder à la structure.

Visuels

La différence de couleur et le patterne imprimé sur le métal des coupes permet au joueur de rapidement les reconnaître et discerner du reste du pattern.

Narration

L'utilité de cette structure n'étant pas apparent, nous nous efforçons de communiquer l'absence d'un élément complémentaire qui s'en servirait comme support.

Pattern n°5: citadel

Intentions de level design et niche

Le but de ce pattern est de proposer au joueur une multitude de points d'entrée, situés plus ou moins haut sur le pattern. Ces derniers lui donnent accès à des parties du pattern qui changent selon la hauteur à laquelle il se situe.

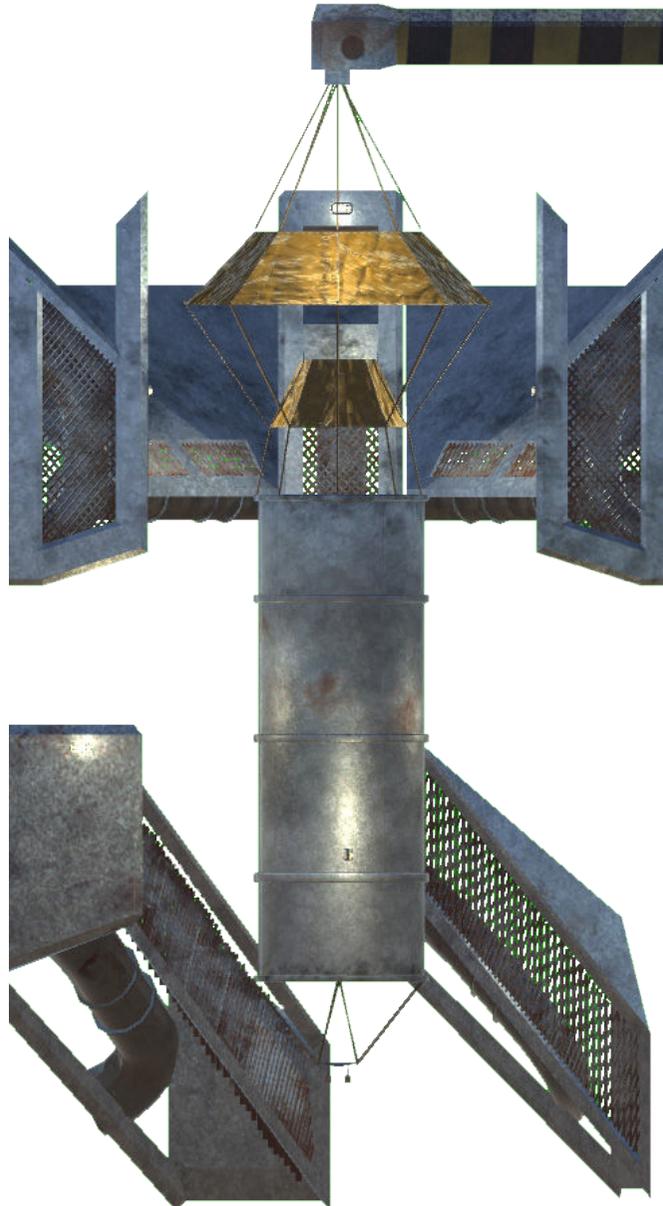
Le but de ce pattern est de donner au joueur des choix véritables non par embranchement, mais en modifiant les résultats de toute décision prise : selon une variable changeante (ici, la vitesse et la hauteur du joueur ont un impact fort sur le résultat de ses interactions avec l'environnement).

Flow principal

Le joueur glisse le long d'une pente avant de tomber librement, puis se rattraper sur une pente en contrebas.

S'il arrive à pénétrer dans le tuyau central du pattern : il aura l'opportunité de s'aligner avec un lifesaver qui se trouve à sa sortie pour nullifier sa vitesse et traverser le pattern sans danger.

Cela signifie également qu'un joueur en perte de contrôle qui arrive sur ce pattern aura une opportunité salvatrice.



Modulation possible

L'angle et alignement verticale des pentes peuvent être ajustés pour faciliter la transition d'une à l'autre.

Egalement, il est possible de permettre au joueur de plus rapidement rentrer dans le tuyau central si on déplace ou agrandis certaines de ces pentes, ce qui peut avoir un impact important sur les routes considérées précieuses pour le joueur.

Visuels

Structurellement, ce pattern est l'un des plus complexes du jeu. Il est nécessaire que cette complexité ne nuise pas à la bonne lisibilité de la structure, nous optons donc pour des textures mates et des formes dessinées pour renforcer les contours et délimitations de chaque plateforme.

Narration

Les meilleures façons de justifier la complexité d'une telle structure dans le contexte de notre univers était de créer une facilité technologique de complexité équivalente, ou de la justifier par une zone de stockage technique mal planifiée ou surpeuplée. Ici, il s'agit d'un hub de tuyauterie.

Pattern n°6: coliseum

Intentions de level design et niche

Ce pattern a pour but de faire glisser le joueur dans une cuve, et de le forcer à ajuster son angle d'arrivée selon sa vitesse afin de ne pas être interrompu par la partie inférieure du pattern.

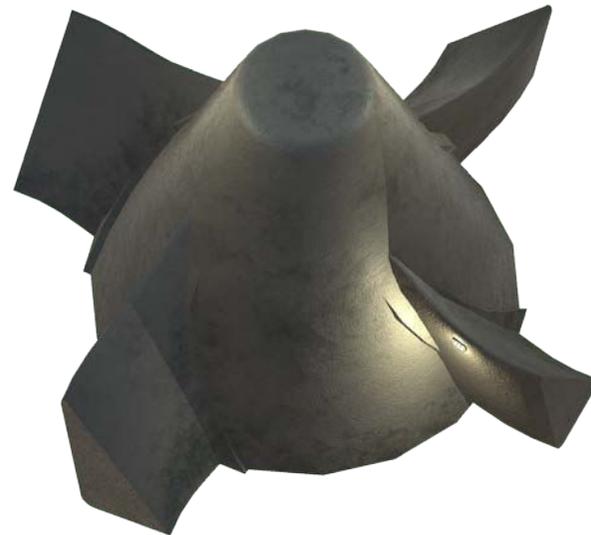
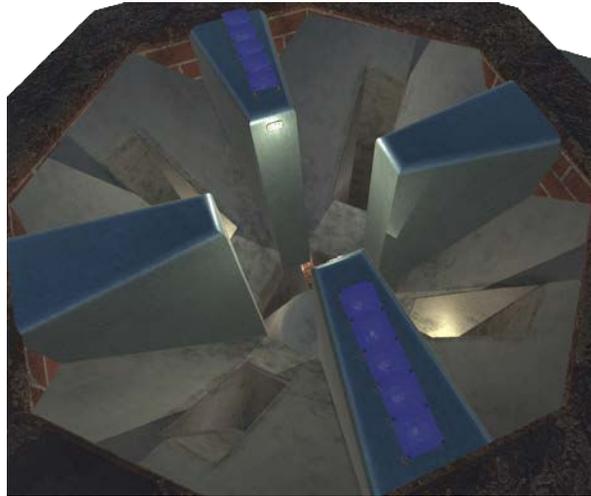
Chacune des deux parties (supérieure et inférieure) peuvent avoir des impacts qui varient drastiquement, selon si le joueur les approche de façon cohérente ou pas.

La niche de ce pattern réside principalement dans sa section en deux blocs distinct, et dont l'utilité du subséquent dépend entièrement de l'approche du précédent.

Flow principal

Le joueur glisse le long d'une pente de la première moitié du pattern, puis descend sur la deuxième moitié (premièrement sur le corps principal du béton, puis sur les extrusions qui le relie aux parois du puit).

Il peut également passer dans un des trous dans les pentes pour traverser le pattern plus rapidement, ou s'arrêter sur une des plateformes plates en haut du pattern.



Modulation possible

La taille et orientation de la majorité des éléments de ce pattern peut être considérée, mais chaque élément devra alors être modifié simultanément pour préserver leur bon alignement et ne pas endommager le flow du pattern.

Principalement, la largeur du haut de la seconde partie du pattern peut être augmentée ou réduite selon la précision que l'on souhaite requérir de la part du joueur.

Visuels

Afin d'éviter une homogénéité déplaisante inhérente à des structures épaisses telles que celle-ci : il est nécessaire de diviser le pattern en au moins deux matériaux différents.

Narration

Ce pattern s'inscrit dans le contexte du jeu comme solidificateur structurel du puit, qui agit comme une voûte et simultanément comme un stoppeur pour tout objet qui tomberait dans le puit.

Pattern n°7: crown

Intentions de level design et niche

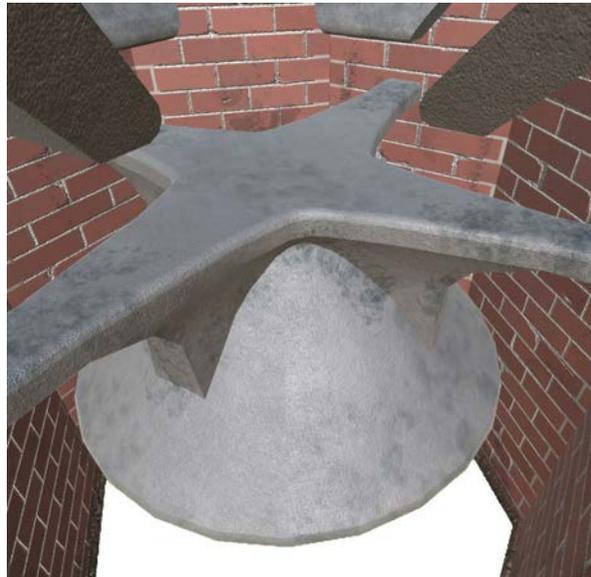
Similairement au pattern #6, ce pattern en forme de sablier force le joueur dans un entonnoir et l'oblige à s'aligner pour continuer sa descente dans la deuxième moitié du pattern.

Contrairement au pattern #6, ce pattern ne punit pas l'échec du joueur avec un ralentissement, mais avec une prise de dégâts causée par le dénivelé important entre les différentes parties de la structure.

La niche de ce pattern est d'obliger le joueur à se rediriger rapidement pour glisser dans une direction, puis dans l'autre.

Flow principal

Le joueur glisse le long d'une pente avant de se retourner pour en employer une autre. Ce processus est répété jusqu'à 4 fois au sein du pattern : et demande principalement un ajustement à la souris.



Modulation possible

La hauteur des extrémités de chaque pente peut être modifiée pour simultanément impacter leur angle, et les chances que le joueur a de prendre des dégâts lorsqu'il entre dans le pattern.

La hauteur des éléments relatifs les uns aux autres définit grandement la difficulté que le joueur pourra avoir à se faufiler entre eux.

Visuels

Tout comme le pattern #6, la forme encombrante de ce pattern nous dirige vers des structures bétonneuses, de stabilisation du puit. Ici nous explorons davantage les pistes de béton brutaliste, ou qui rappellent des formes explorées par Le Corbusier ou Oscar Niemeyer.

DOCUMENTATION TECHNIQUE

Contraintes importantes

Performances

Durant toute la période de développement, une attention particulière a été apportée à la gestion des performances, en effet dû à la vitesse que le joueur peut atteindre, il a été important de s'assurer que la physique, les collisions et l'apparition du LD suivent et interagissent correctement avec l'avatar.

Pour s'assurer du bon fonctionnement de cela et pour maîtriser le plus possible la vitesse du joueur en chute, nous avons décidé de simuler nous même la gravité. Grâce à cela, nous sommes capable de la moduler pour palier à nos besoins, par exemple pour accélérer la vitesse de l'avatar en début de chute.

Character Controller basé sur la physique du moteur

Tous les mouvements de notre avatar sont basées sur le moteur physique de Unity, par conséquent certains problèmes connus de ce moteur comme le clipping à haute vitesse sont présents dans notre jeu.

Nous avons donc dû, au maximum, essayer de contourner ces soucis.

Le principal système que nous avons mis en place pour palier à ce problème de clipping est un cap de vitesse sur la chute de l'avatar.

Pour permettre de glisser sur des surfaces sans modifier la vitesse du joueur directement nous avons opté pour l'utilisation de Physics Materials pour enlever la friction de certains objets.

Boucle infinie de LD

Notre intention étant de faire un platformer infini, il est important de mettre en place une logique de génération de LD qui va permettre au joueur de ne pas avoir l'impression de voir le même pattern en boucle.

Il était aussi important d'empêcher le joueur de voir les patterns apparaître car il n'était pas possible pour nous d'en générer trop simultanément sans risquer de causer des problèmes de performances. Pour cela nous avons utilisé le système de "fog" de Unity qui nous permet de limiter de manière intradiégétique la distance de vue du joueur, cachant ainsi l'apparition du LD. Ce "fog" nous sert aussi à augmenter la difficulté, en montant sa densité au fur et à mesure que le joueur descend.

Mouvement de l'avatar - Grounded

Fonctionnement "Grounded"

Les collisions et les mouvements de notre avatar sont presque entièrement physics-based, c'est-à-dire que l'on laisse majoritairement au moteur physique de Unity la tâche de s'en charger.

Notre jeu se déroulant principalement de manière verticale, nous simulons nous-même la gravité afin de pouvoir la modifier pour répondre à nos besoins. Cette gravité a une valeur de base de 15 unités par seconde.

En fonction de la direction dans laquelle le joueur veut que l'avatar se déplace, un multiplicateur est appliqué au vecteur de mouvement afin de le ralentir lorsqu'il ne va pas en avant.

De base, ce multiplicateur est de 80% en arrière et de 95% sur les côtés.

L'avatar se déplace par accélération, lors de l'input celle-ci ne part pas de 0% mais de 50% de sa capacité maximale afin de donner au joueur la sensation de se déplacer dès l'input et pas après 1 ou 2 secondes, quelque soit la direction.

Au sol, un cap est appliqué à la vitesse horizontale de l'avatar afin de l'empêcher de perdre le contrôle dans un état où il doit être posé.

Détails technique

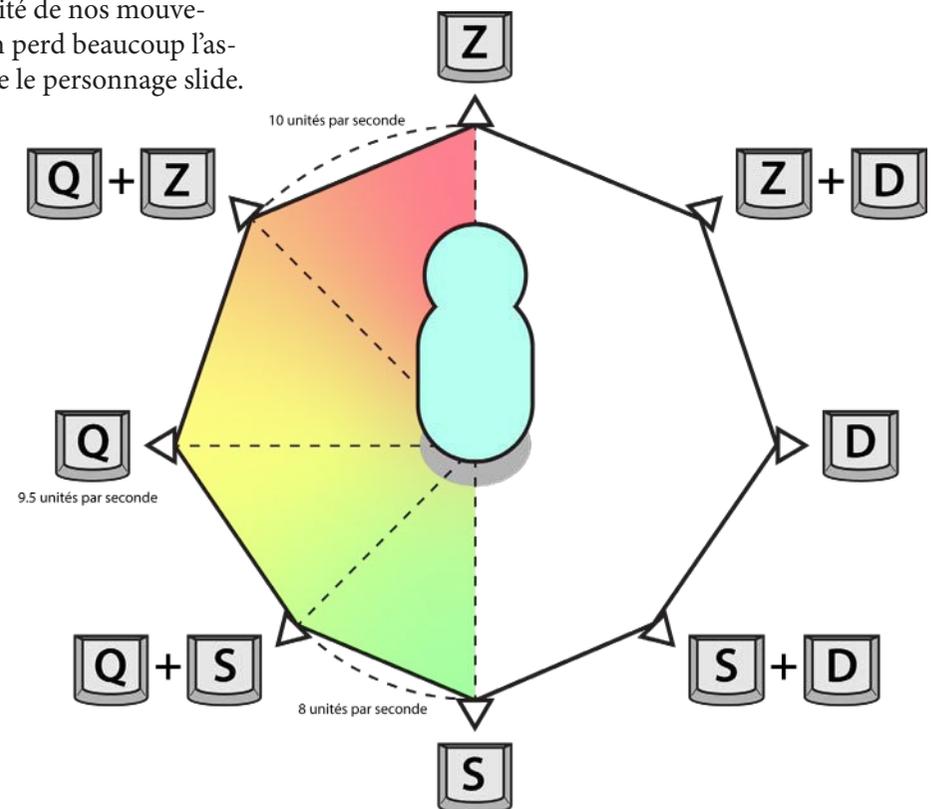
Bien que l'utilisation de "FixedUpdate" est fortement recommandé lors de l'utilisation du moteur physique de Unity, tout notre système (hors gravité) est appliqué via "Update".

La raison de cela est que celui-ci permet de transmettre un feeling et une sensation en jeu plus proche de nos intentions de design que lorsque l'on utilise "FixedUpdate".

Sous "FixedUpdate", la majorité de nos mouvements sont assez rigides et on perd beaucoup l'aspect flottant que l'on a lorsque le personnage slide.

Paramètres mouvement grounded:

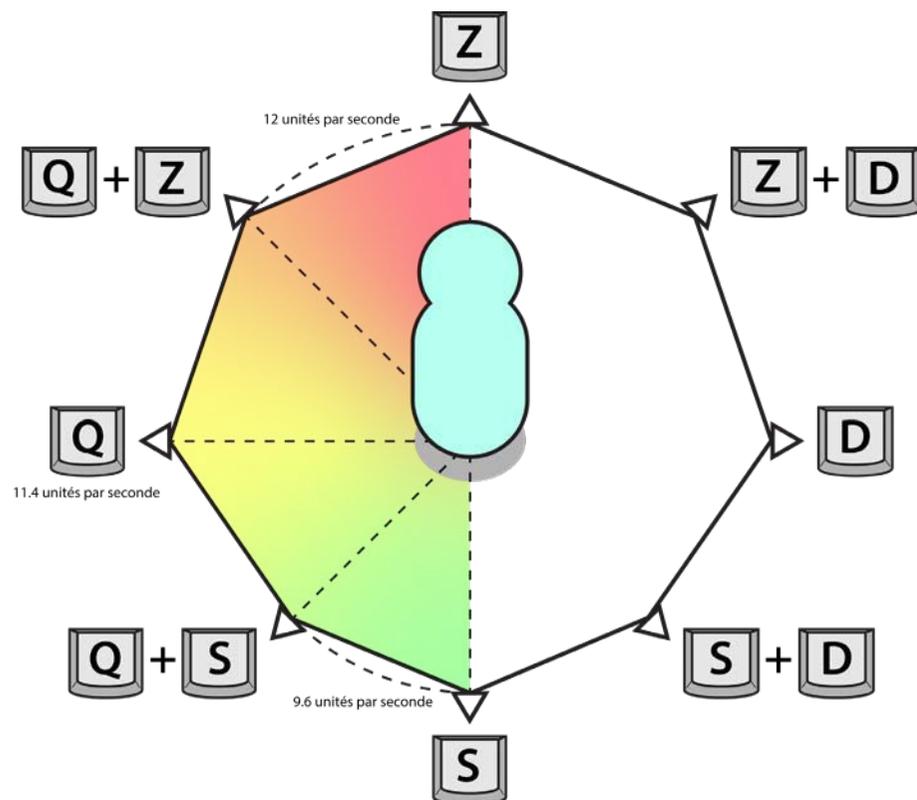
Paramètre	Valeur
Cap de vitesse grounded	10
Accélération	9/sec
Multiplicateur de mouvement latéral	x0.95
Multiplicateur de mouvement arrière	x0.8



Mouvement de l'avatar - Airborne

Paramètres mouvement airborne:

Paramètre	Valeur
Cap de vitesse horizontale	12
Accélération	9/sec
Multiplicateur de redirection horizontale	x0.4



Fonctionnement "Airborne"

Quand le joueur saute, son support et la direction de la caméra va avoir un impact sur la direction et la force du saut.

Lorsqu'il le déclenche, une impulsion de 10 unités par seconde est appliquée sur un axe perpendiculaire à la plateforme sur laquelle il est, à ça est ajoutée une impulsion de 2 unités par seconde dans la direction où regarde la caméra.

Les collisions et les mouvements de notre avatar étant majoritairement physics-based, en vol, le personnage conservera une certaine inertie qui évite une sensation de rigidité dans son mouvement.

Les paramètres de déplacement sont les mêmes en l'air qu'au sol, à la différence que l'accélération de base est bridée à 40% en plus des réductions en cas de mouvement vers les côtés ou l'arrière.

Un cap est appliqué à la vitesse horizontale de l'avatar afin d'éviter une perte de contrôle totale lors des déplacements aériens.

Nous avons aussi appliqué un cap de vitesse verticale pour éviter d'atteindre des vitesses si élevées que le personnage traverserait le LD, celui-ci étant fait pour ralentir le joueur s'il cherche à partir en chute libre.

Mouvement de l'avatar - Jump et Slide

Fonctionnement du saut

Quand le joueur saute, son support et la direction de la caméra va avoir un impact sur la direction et la force du saut.

Lorsqu'il le déclenche, une impulsion de 10 unités par seconde est appliquée sur un axe perpendiculaire à la plateforme sur laquelle il est, à ça est ajoutée une impulsion de 2 unités par seconde dans la direction où regarde la caméra.

Fonctionnement "Sliding"

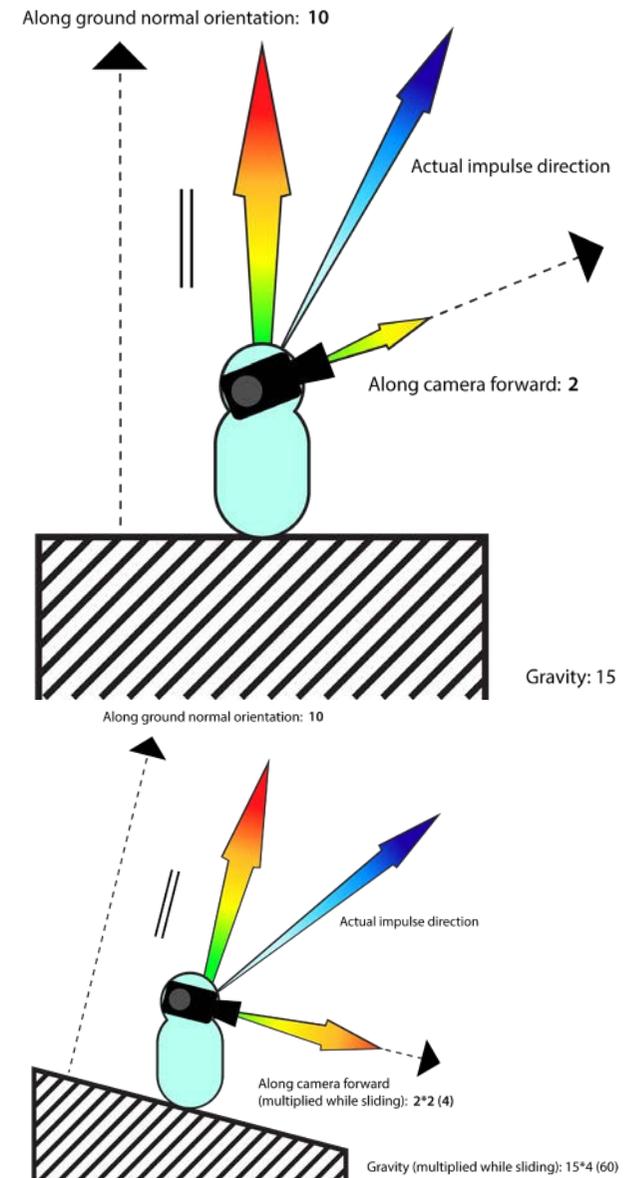
Le comportement de l'avatar en slide est basiquement le même qu'à l'état grounded.

Il y a 3 différences :

- L'avatar n'a pas de friction avec les plateformes qui le rendent dans cet état
- Sa gravité est multipliée par 4 pour permettre une meilleure sensation de glissement (60 unités par seconde)
- Sa force de saut par rapport à la caméra est multipliée par 2 (4 unités par seconde)

Paramètres saut et gravité:

Paramètre	Valeur
Force upward du saut	10
Force forward du saut	2
Multiplicateur de force forward (while sliding)	x2
Gravité	15
Multiplicateur de gravité (while sliding)	60



Système de chute

Paramètres vitesse de chute:

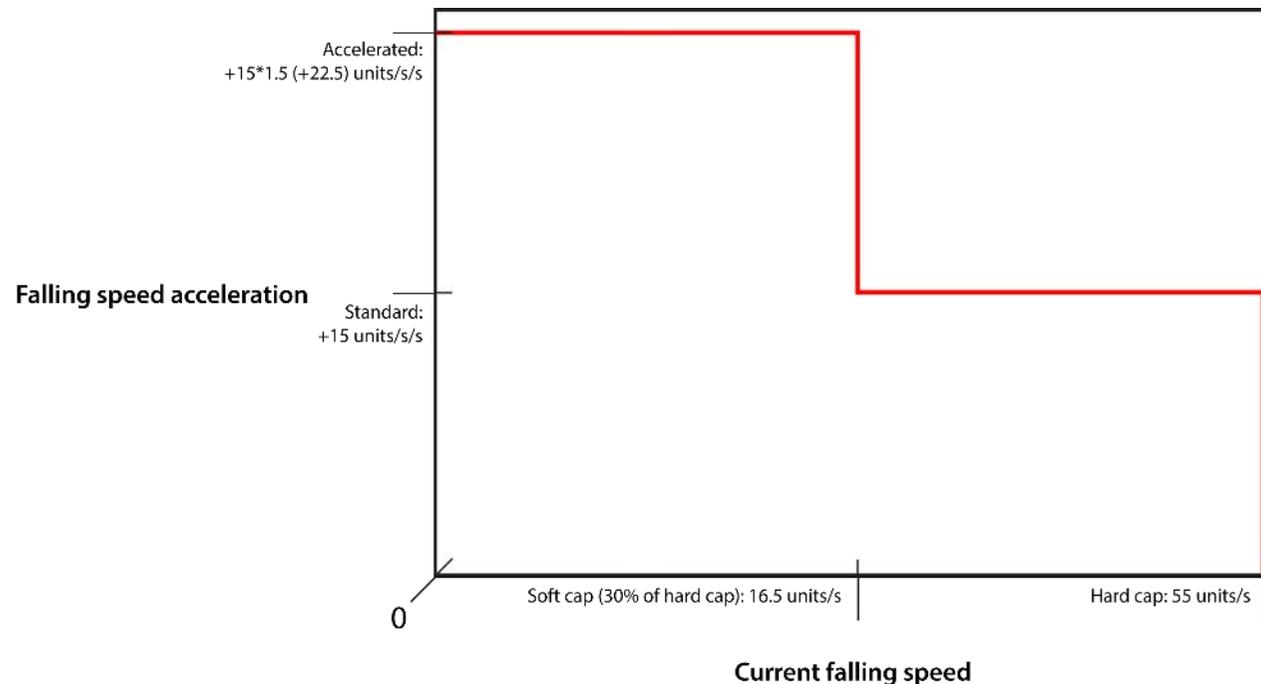
Paramètre	Valeur
Soft cap de vitesse de chute (30% du hard cap)	16.5
Hard cap de vitesse de chute	55
Gravité standard (lorsqu'au dessus du soft cap)	15
Gravité augmentée (x1.5 lorsqu'en dessous du soft cap)	22.5

Fonctionnement

Lorsque le joueur est airborne, il est affecté par notre simulation de la gravité, appliquant une force continue vers le bas ainsi qu'une accélération pour le faire tomber de plus en plus rapidement.

En début de chute, jusqu'à atteindre le palier de vitesse "minimale" (16.5 unités par seconde), cette gravité est augmentée de 50%, passant d'une force de 15 unités à une force de 22.5 unités.

Il repasse ensuite à une force de gravité "normale" avant de ne plus être affectée lorsqu'il atteint le cap de vitesse de 55 unités par seconde.



Système de dégâts et “Life Savers”

Fonctionnement des dégâts

Notre système de dégâts est basé sur 2 éléments :

- Le temps de chute
- La vitesse de chute

Si le temps de chute de l'avatar est inférieur à 1.8s, le joueur ne prendra pas de dégât.

Cependant, si ce temps est supérieur à 1.8s, le joueur sera “vulnérable” si sa vitesse dépasse certains paliers :

- 25 unités par seconde, il prend un dégât s'il n'est pas déjà blessé, sinon il est en risque de mourir
- 55 unités par seconde, il est en risque de mourir

Dans le cas où le joueur remplit les conditions pour mourir sur sa chute, un timer de 0.5s est lancé à partir du moment où il remplit la condition de défaite pour donner un peu plus de temps au joueur pour rattraper sa chute.

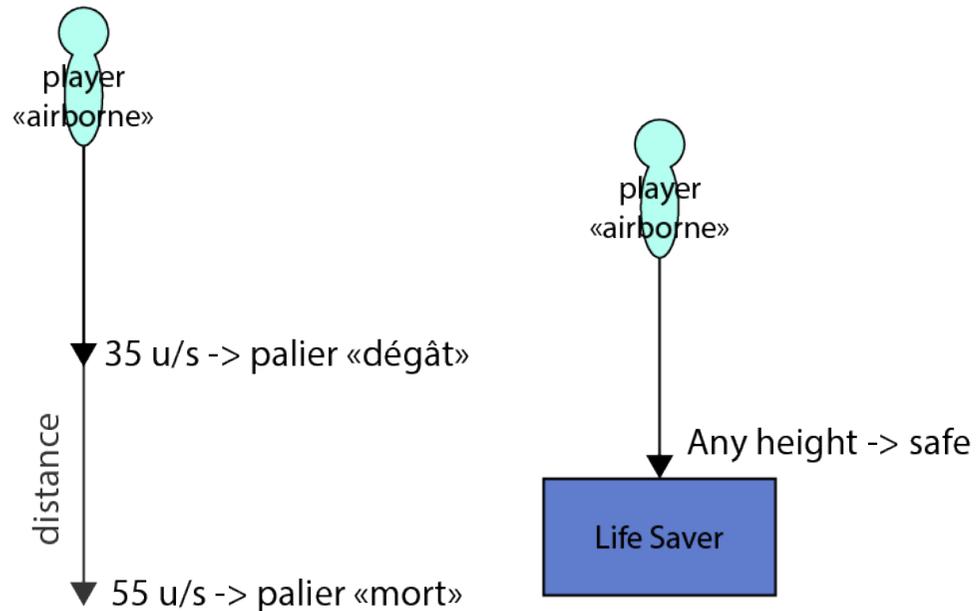
Fonctionnement des “Life Savers”

Le “Life Saver” est un objet qui permet au joueur de rattraper sa chute, quelle que soit la hauteur.

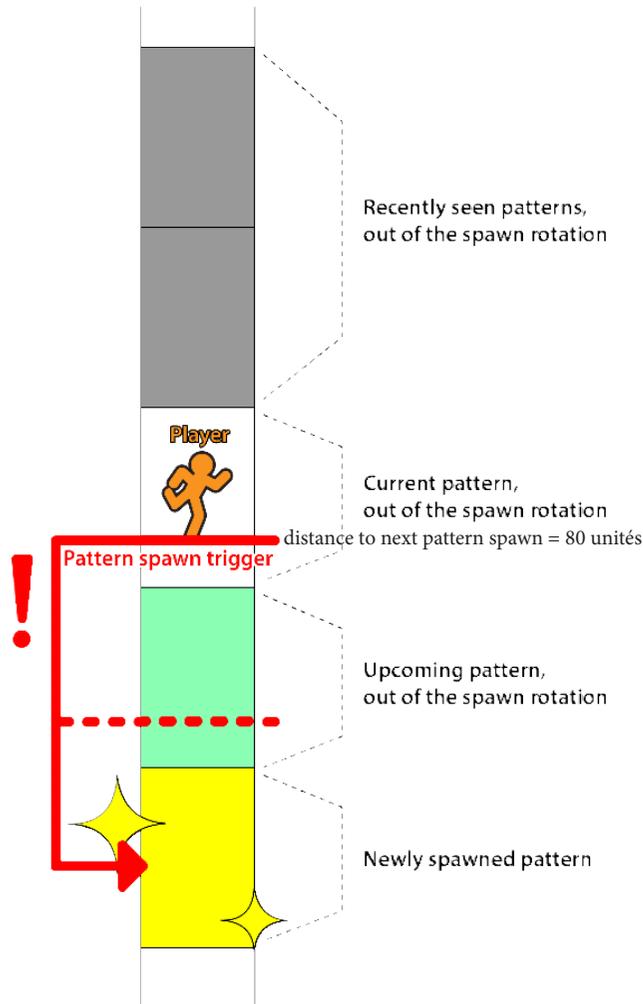
Son fonctionnement est simple :

Si le joueur tombe dessus, sa vitesse verticale est réduite à 0 au contact, l'empêchant de prendre des dégâts.

Le “Life Saver” devient ensuite intangible tant que l'avatar est en contact avec, afin de lui permettre de continuer de se déplacer sans difficulté.



LD procédural



Anatomie d'un pattern

Le pattern est la section de level design prédéfinie et stockée dans un prefab que nous appelons à intervalles de distance régulière pour permettre au joueur de continuer sa descente. Ils sont d'une hauteur normée de 100 unités.

Processus de sélection

Avant son apparition, nous choisissons procéduralement un pattern, qui n'est pas encore activé dans la scène, au sein d'un tableau où ils sont tous regroupés.

En effet, nos patterns existent tous dans la scène par défaut : ce qui nous évite tout lag éventuel causé par leur instanciation ou destruction.

Les patterns sont désactivés lorsqu'ils sont trop éloignés en haut du joueur.

Leur activation tardive garantit une impression de permanence du puit (on ne les voit pas disparaître en regardant vers le haut), et évite une répétition d'un même pattern plusieurs fois consécutivement, ou dans un interval trop réduit.

Trigger de l'apparition

Lorsque le joueur se rapproche de la fin d'un pattern : le pattern après le suivant est appelé, et apparaît directement en dessous.

Dès qu'il apparaît, l'origine de ce nouveau pattern est enregistré comme nouveau point de référence pour l'appel du pattern d'après.

Transformation

Afin de minimiser la sensation de répétition des patterns, on applique une rotation aléatoire par multiples de 45° aux patterns lorsqu'ils apparaissent.

Systeme de chute de l'étoile

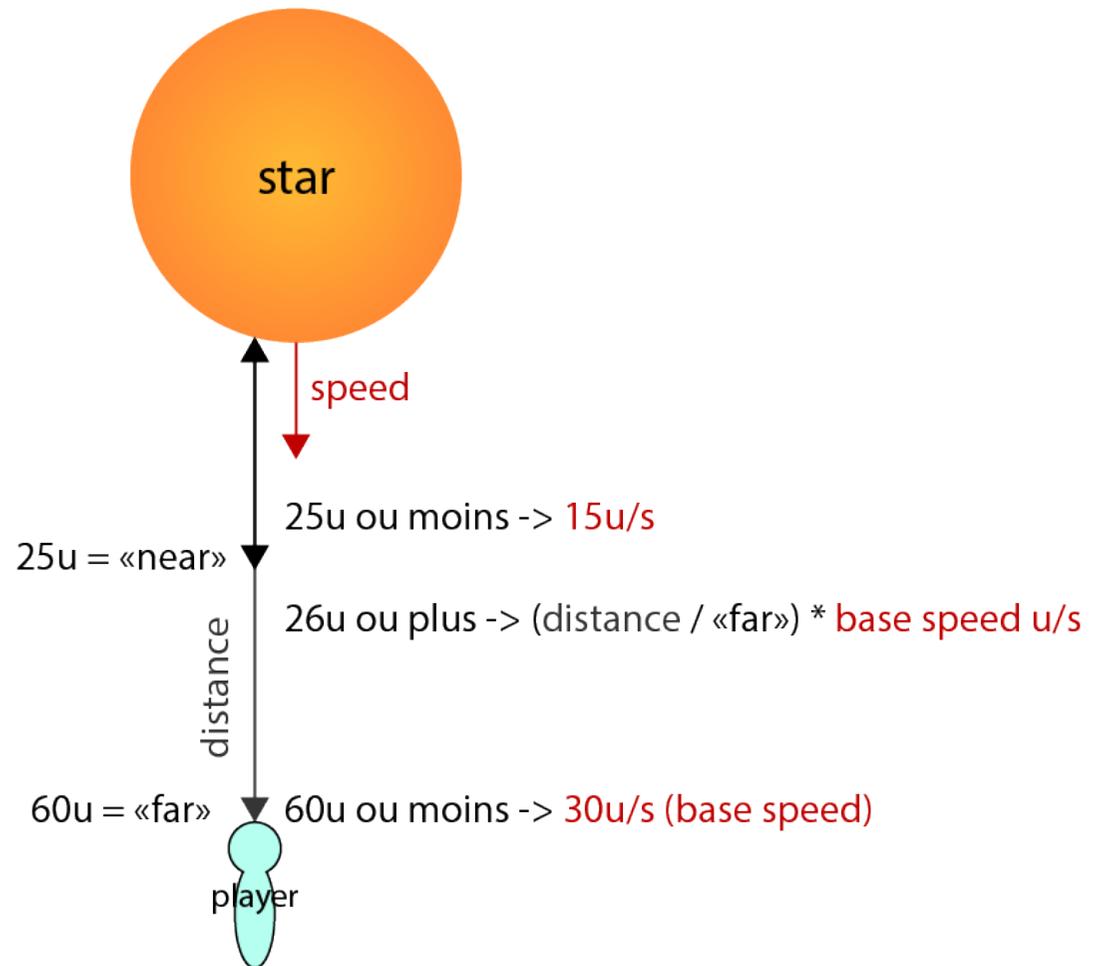
Fonctionnement

La vitesse de l'étoile qui suit le joueur dans sa chute est dynamique, elle dépend de sa distance au joueur.

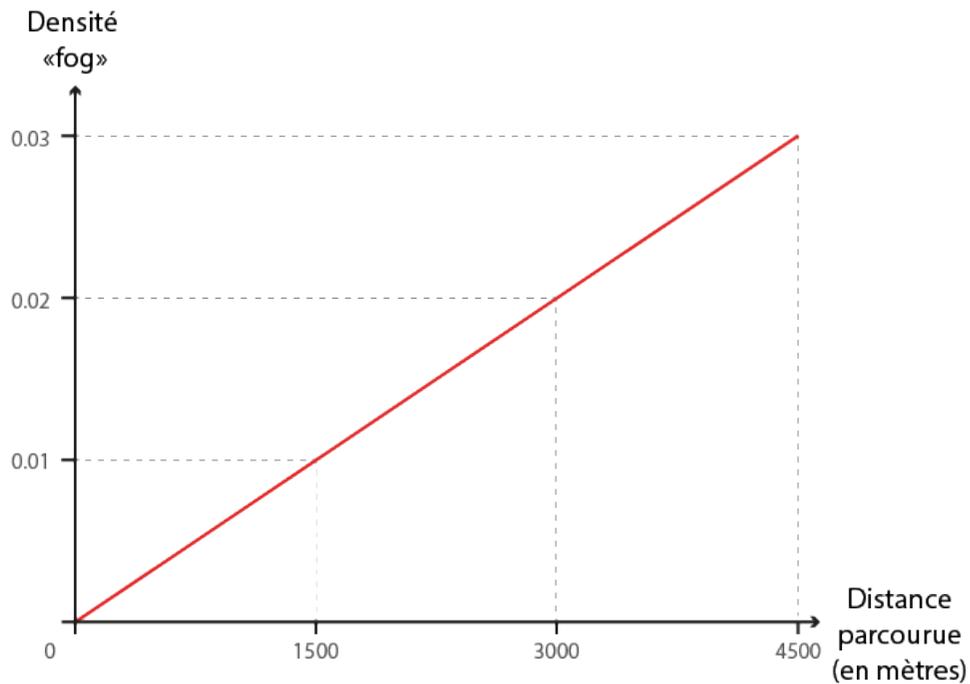
Nous avons fait ce choix car il était difficile de trouver une vitesse à laquelle la difficulté serait présente sans être complètement excessive ou à laquelle elle serait inexistante.

Pour cela, on utilise 2 paliers de distance entre l'étoile et le joueur :

- "near" : si la distance entre le personnage et l'étoile est inférieure à ce palier la vitesse de base de l'étoile est divisée par 2.
- "far" : si la distance entre le personnage et l'étoile est supérieure au palier "near" alors la vitesse de l'étoile sera multipliée par la distance entre le personnage et l'étoile, le tout divisé par le palier "far" (x0.75 à 45u, x1 à 60u, x1.5 à 90u, etc.).



Augmentation de la difficulté



Fonctionnement

On utilise un fog pour limiter la distance de vue du joueur afin d'éviter qu'il voie le LD apparaître, cependant ce fog nous sert aussi à gérer la difficulté du jeu.

Plus le joueur descend, plus la densité du fog augmente, réduisant la distance de son champ de vision, au maximum, le joueur peut voir à peine un demi-pattern, le forçant à user de sa mémoire pour ne pas se mettre dans une situation difficile en s'engageant de la mauvaise manière dans un pattern.

La distance à laquelle le joueur peut voir, passé 4500 mètres, reste assez éloigné pour que le joueur puisse quand même se rediriger par réflexe au besoin.

GAME ART

Intentions Globales

Univers et storytelling

La direction artistique de The Pit penche vers un univers réaliste à l'aide des éléments qui le composent visuellement et à la fois complètement improbable dû aux événements qui s'y produisent (étoile tombant dans un mystérieux puits sans fond)

En effet afin de plonger le joueur en immersion dans un univers fictif peu tangible il nous faut utiliser tous les outils à notre disposition (éclairage, éléments 3D, sons et textures de qualités). Ainsi le joueur investi dans l'univers de The Pit est tout aussi bien happé par son univers curieux que son gameplay.

Les éléments présents dans le jeu ne justifient pas les événements qui s'y produisent mais les concrétisent. Il est important de garder une part de mystère qui laisse place à l'imagination du joueur : "À quoi servait ce puis ?", "Pourquoi cette étoile nous tombe dessus ?", "À quoi sert cet élément mécanique ?"

La curiosité que suscite un univers si étrange permet au joueur de construire un imaginaire puissant autour du puits et des événements extraordinaires qui s'y produisent.

Visuels et effets

Les éléments visuels de The Pit sont semi-réalistes, on devine leur utilité mais ils ne possèdent pas tous d'équivalence directe à des objets du monde réel.

Le player est en immersion à l'aide d'effets visuels à la première personne qui mettent en avant le sentiment de chute. En effet la plupart des effets en jeu sont intradiégétiques de manière à ne pas perturber le joueur tout en traduisant ce qu'il se passe autour de lui.

Les éléments 3D sont au service du Level Design et du ressenti de vitesse afin de mettre en avant la façon dont le joueur se déplace dans l'environnement avant que celui-ci soit mis en avant.



Visuels et effets

Les éléments 3D étants au service du gameplay se doivent de traduire leur utilité à l'instant où ils sont vus par le joueur.

Sans sortir le player de l'univers, chaque forme doit être comprise très rapidement par le joueur afin d'alléger sa charge cognitive.

En fonction de la situation, les éléments qui sont utiles au joueur sont plus ou moins mis en avant pour le diriger implicitement dans une direction qui lui sera avantageuse.

Enfin, chaque section de jeu a son identité propre permettant de faire la différence entre les patterns rapidement. Ainsi le joueur n'a qu'à identifier une forme ou une couleur pour savoir ce qui l'attend plus loin dans le puit.



Contraintes Importantes

Level Design et feeling de jeu

Lors de la création des modèles 3D de The Pit, il était important de respecter les éléments de Level Design que nous avons créés.

Chaque modèle se devait de traduire l'intention principale du module de Level Design qu'il représentait.

Pour cela, il a fallu simplifier certains modèles ainsi que certains détails dans le but de rendre le modèle 3D plus lisible en jeu et que le joueur ne passe pas à travers certains éléments 3D de manière incohérente.

Lorsque le joueur arrive en jeu il doit faire face à la fois à un univers cohérent qui se respecte mais il doit aussi pouvoir en tirer les éléments qu'il peut utiliser pour se créer un imaginaire autour du puits.

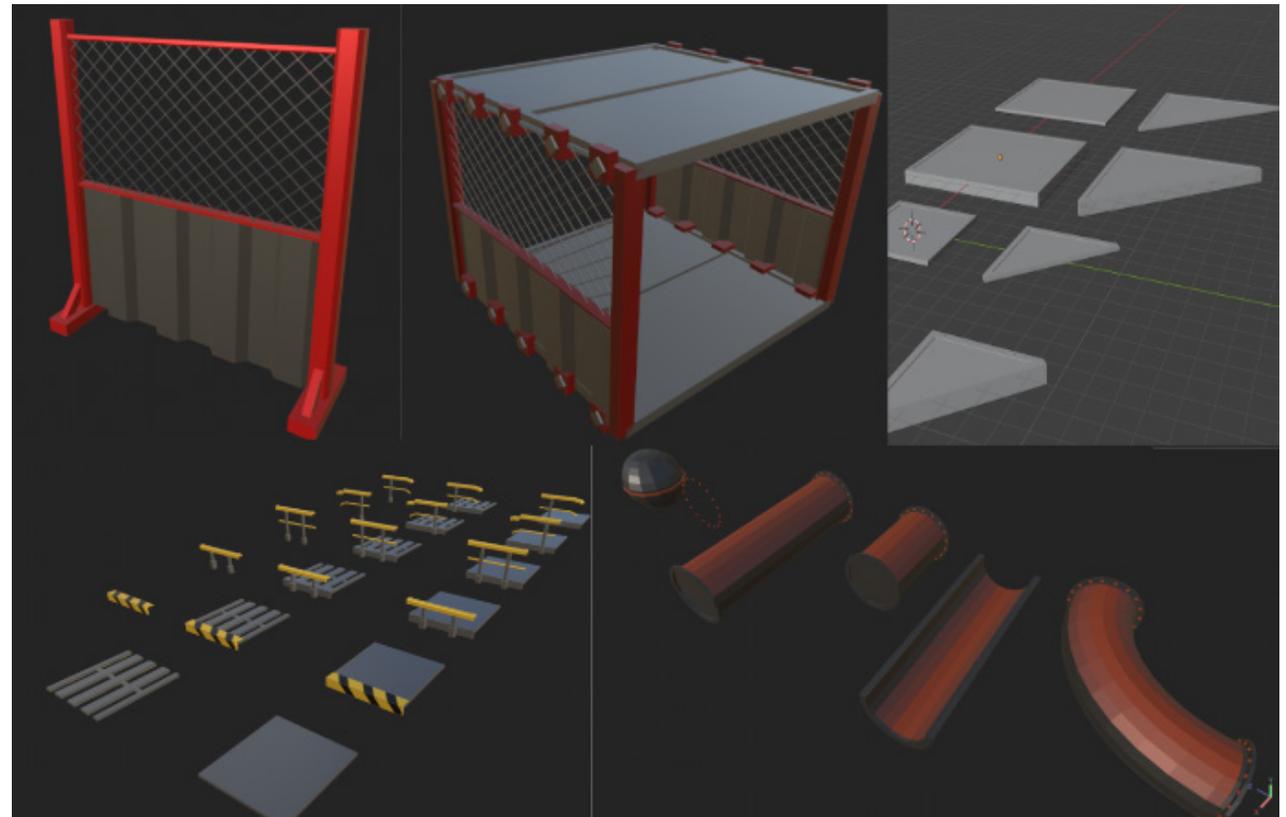
Il était important lors de la création de chaque modèle de se mettre dans la peau du joueur et de regarder les textures et les géométries de près afin de les paufiner et qu'elles ne paraissent pas grossières aux yeux du joueur.

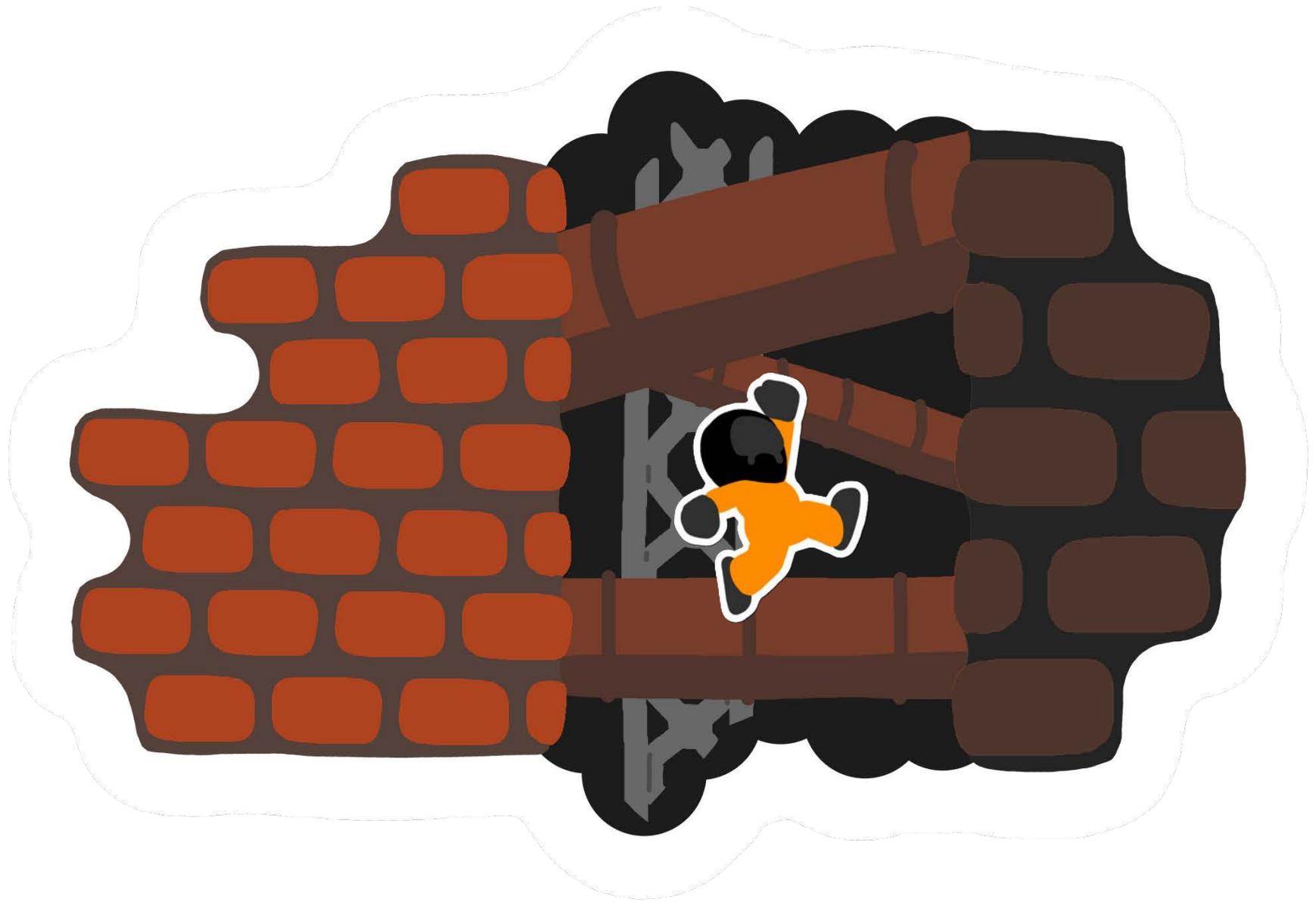
Enfin, il est extrêmement important que le joueur puisse identifier chaque élément rapidement afin de fluidifier son gameplay. La plupart des éléments ne seront vus que d'un seul angle alors celui-ci doit être mis en valeur.

Blocs de Level Design

Lors de la création de The Pit nous avons d'abord pensé les patterns comme étant un assemblage de plusieurs blocs prédéfinis comme les slides des tunnels ou encore des barrières.

Nous avons donc créé plusieurs visuels pour ces blocs que nous n'avons finalement pas utilisés au profit de plus grands blocs de LD bien plus précis et efficaces.





Références Game Art

Quake est une grande inspiration qu'il s'agisse de la forme ou des textures des murs ainsi que du nombre de polygones utilisés dans les modèles 3D. D'autres décors des Quake plus récents sont intéressants à étudier. En effet, ils constituent des éléments qui peuvent être intégrés dans notre jeu: tuyauterie et autres plaques de métal ainsi que plateformes rouillées. Certains décors et uniquement certains sont exploitables dans notre projet.



Iron Lung utilise des couleurs ainsi que des textures extrêmement intéressantes pour notre projet. L'espace de jeu bien que très restreint est extrêmement sombre, on ne voit qu'à 2 m autour de soi. Il serait bien malgré tout que nous puissions voir plus loin dans notre projet afin de visualiser la plateforme sur lesquelles le joueur doit atterrir.



The Pony Factory insiste sur une ambiance sombre avec certains points lumineux qui attirent le regard. Les textures racontent une histoire.

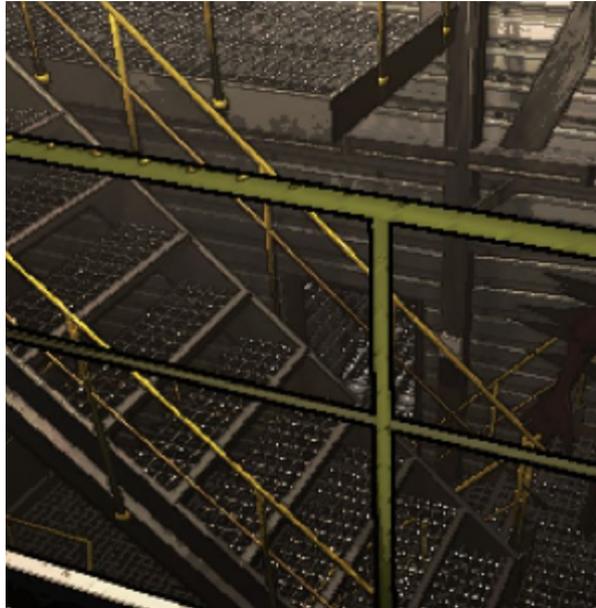


Saw the game met en place un décor mécanique rouillé, abandonné avec une architecture brutaliste intéressante. Nous pourrions nous inspirer de l'ambiance de ce jeu pour constituer notre projet.

Projet Blary nous inspire dans le nombre de polygones utilisés ainsi que dans son ambiance lumineuse. Il ne laisse voir que les 20 m qui se situe en face du joueur, le reste est pris dans la pénombre, laissant une part de mystère dans l'environnement du joueur.



Références Game Art



Dans Lethal company, les plateformes, barrières, ainsi que les murs sont tous aussi intéressants que tout le reste de nos références, ceux-ci respectent une esthétique graphique qui correspond à ce que nous voulons donner comme aspect final dans notre jeu. De plus, le personnage principal du jeu correspond au character design que nous cherchons à donner au personnage, ne laissant transparaître qu'une tenue sans humanité.

Viscera simulator, met en place une ambiance d'entreprise potentiellement malhonnête nettoyant des cadavres à l'aide d'employés probablement sous-payés. Ceux-ci possèdent des tenues de nettoyeur technique qui sont exploitables dans notre univers.



Intégration Narrative

Cohérence et univers

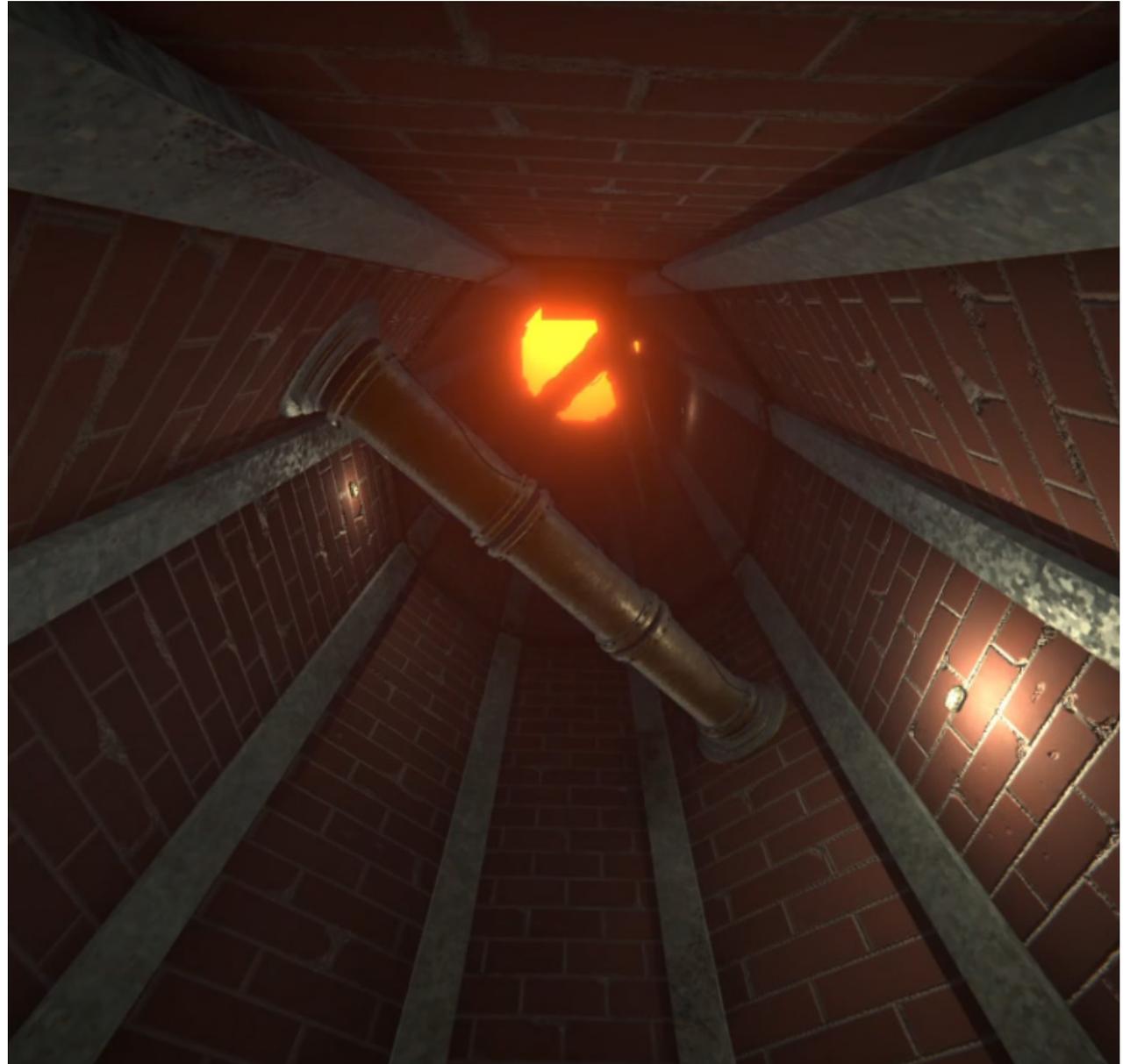
L'univers de The Pit étant complètement fictif, il est important d'y plonger totalement le joueur afin qu'il s'y sente investi.

Pour cela, chaque pattern créé a son utilité dans le puits et est tangible aux yeux du joueur.

La disposition de la rouille ou bien de l'usure permet à la fois de raconter l'histoire des éléments visuels et de faciliter la lecture de jeu.

Lors de sa première entrée dans le puits, nous ne laissons pas le choix au joueur que d'accepter son destin le forçant à fuir une étoile sécrétant sur lui. De cette manière le joueur n'a pas le choix que de s'accaparer cet univers s'il veut progresser dans le jeu.

Il était donc important que les visuels correspondent à ce que le joueur s'attendait à retrouver dans un puits sans fond.

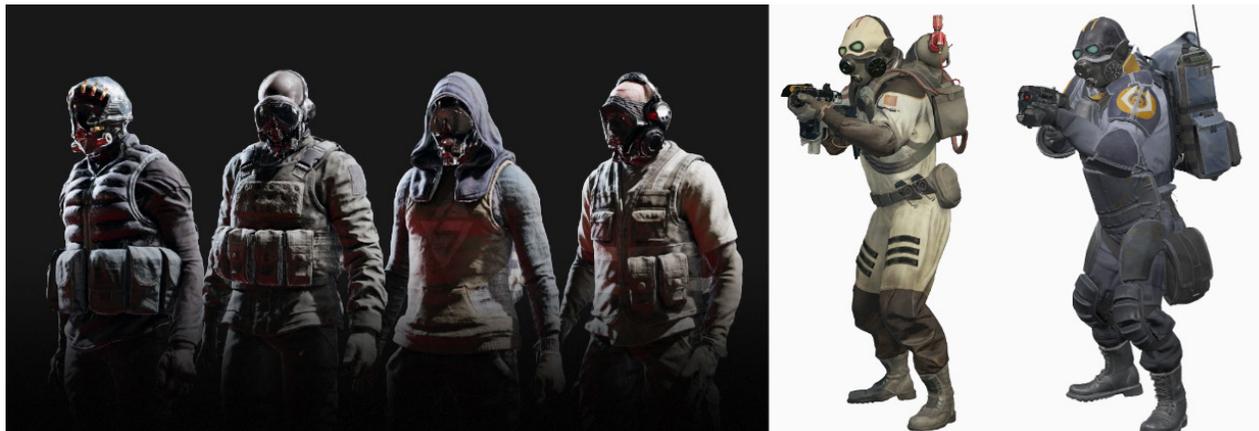


Player



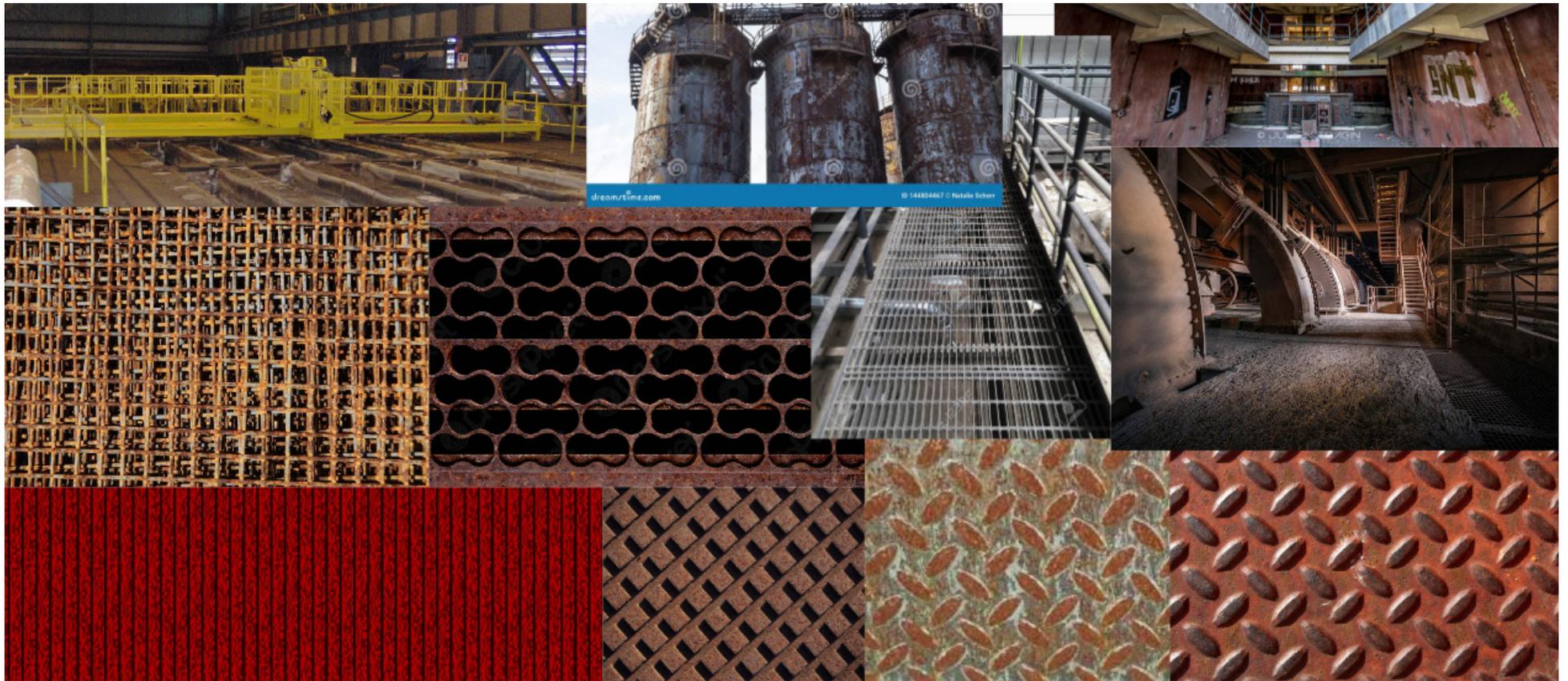
Pour les références du modèle du personnage, nous souhaitons avoir une tenue recouvrant complètement le personnage principal. Cette même tenue doit être technique et correspondre à l'univers mécanique du jeu, mais doit aussi traduire les capacités du joueur.

Ces éléments de tenues de travailleur dans une compagnie respectant des codes strictes nous ont permis de créer un univers dans lequel une mystérieuse entreprise emploie des nettoyeurs dans un puits sans fond.

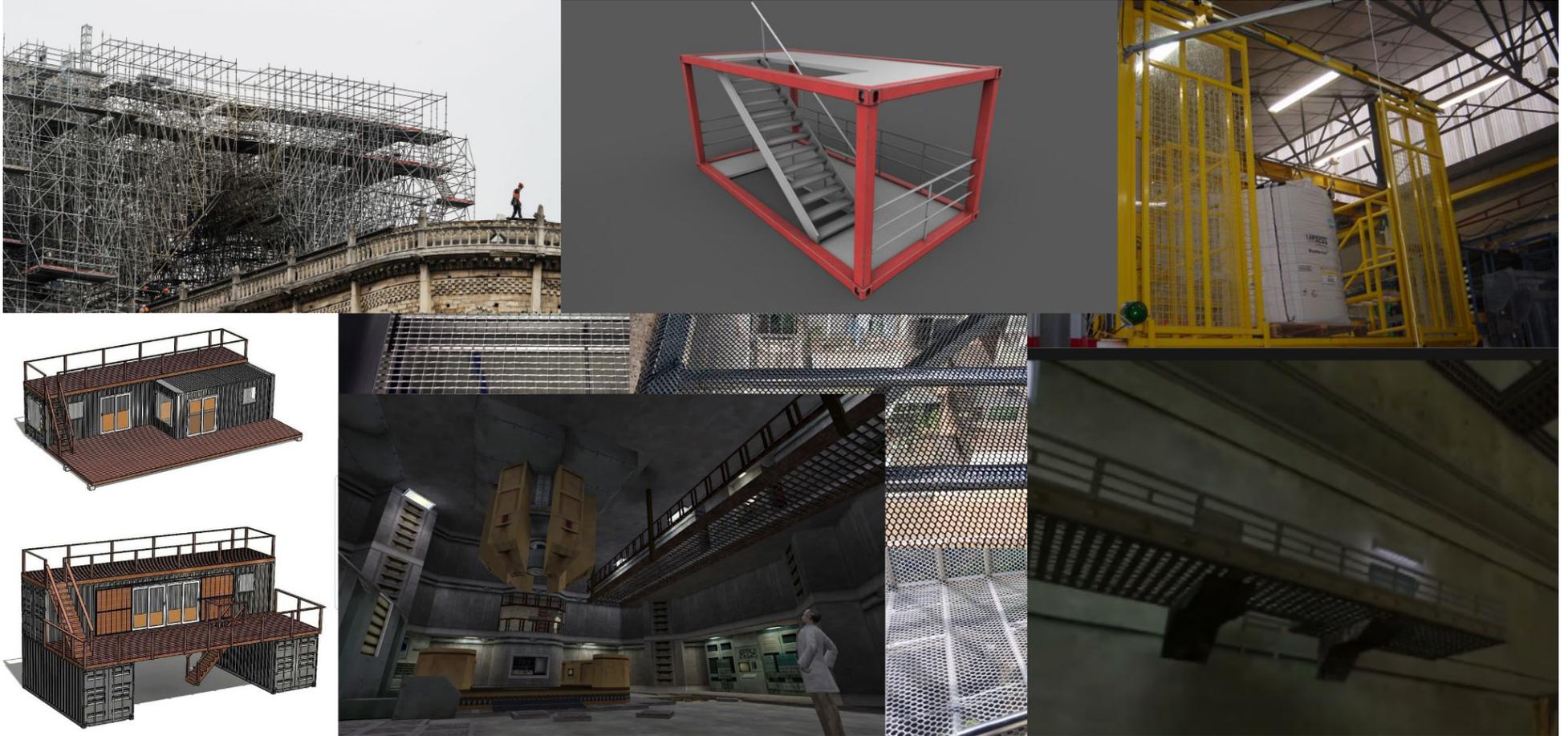


Ces tenues ne distinguent pas entre les employés en question et font partie d'une déshumanisation du personnage principal volontaire. La vie de ce personnage n'a que peu d'importance aux yeux de la compagnie qui l'emploie.

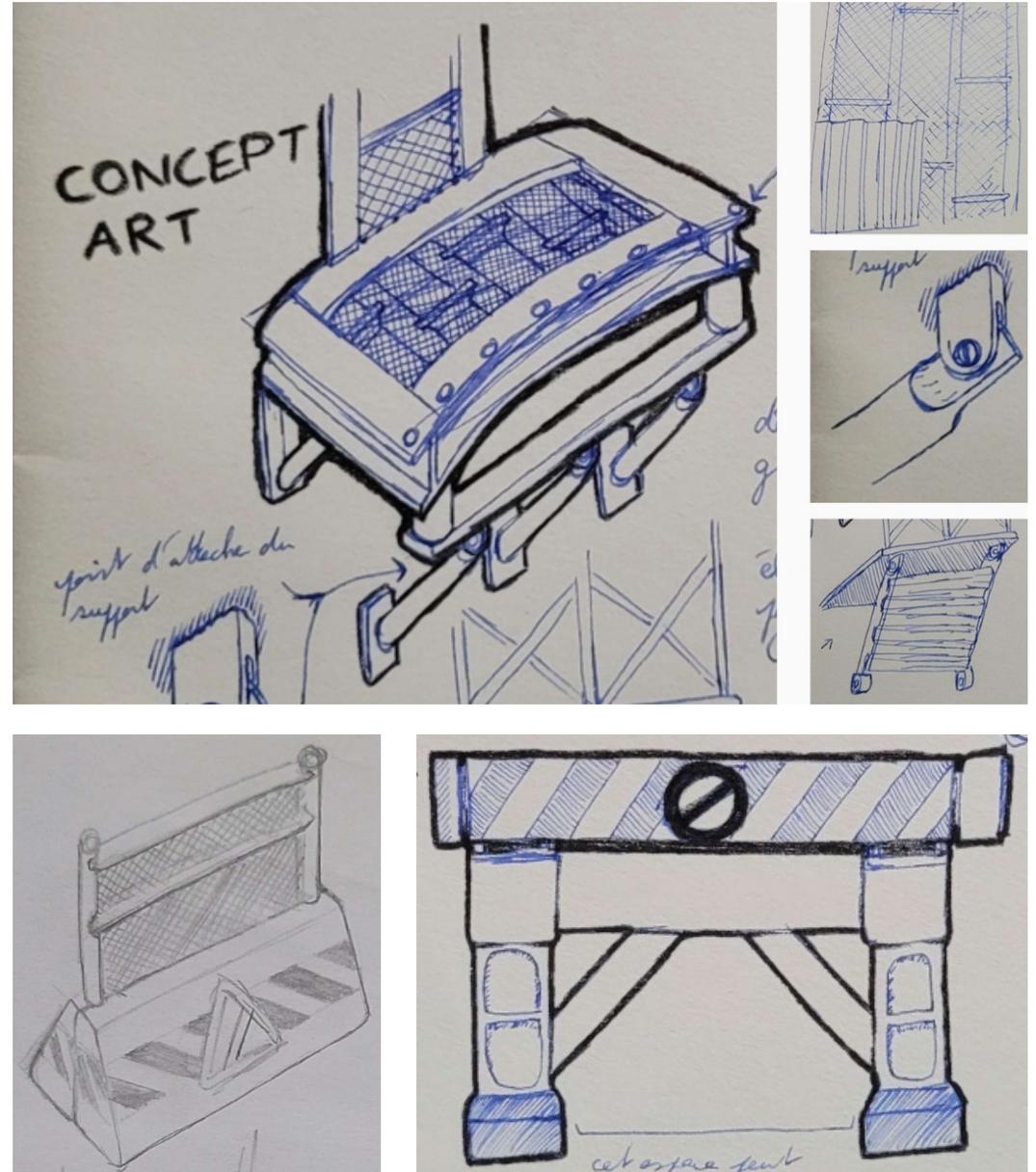
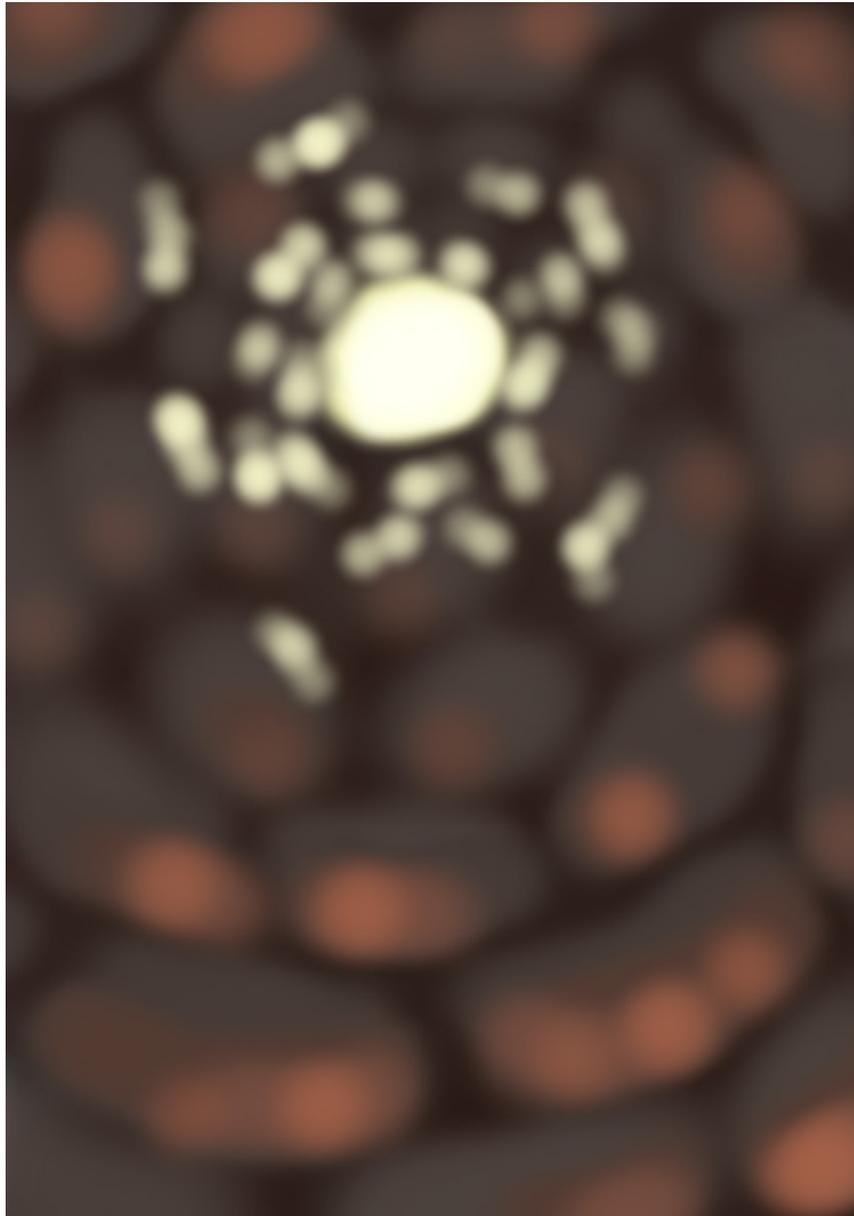
Moodboard textures

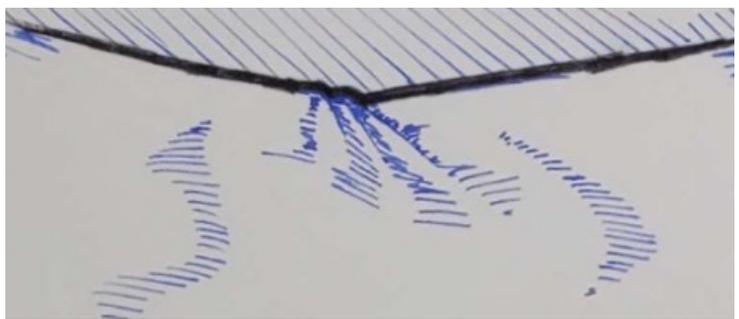
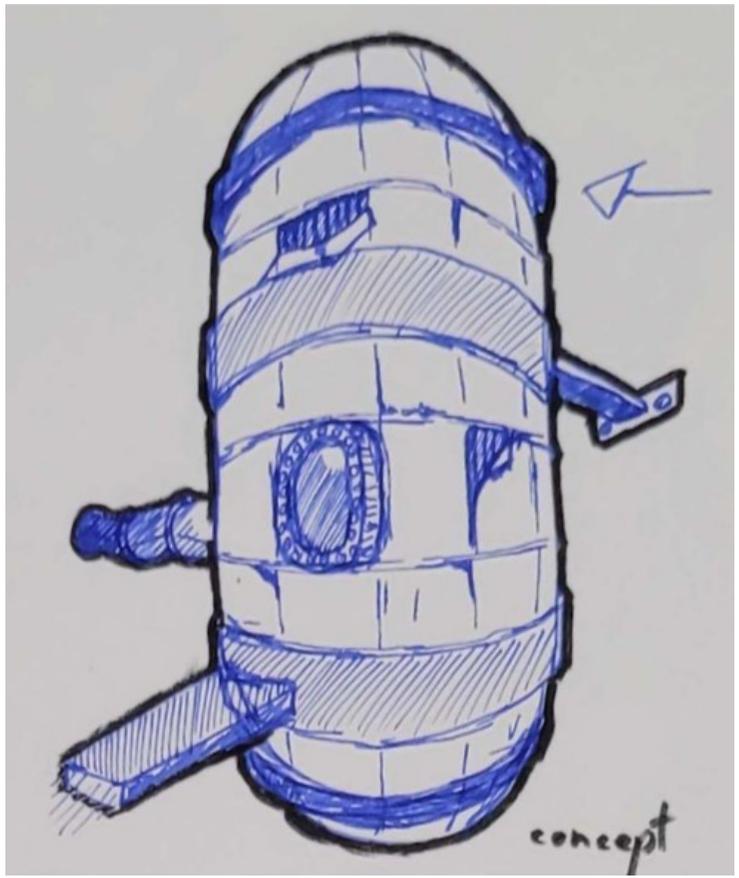


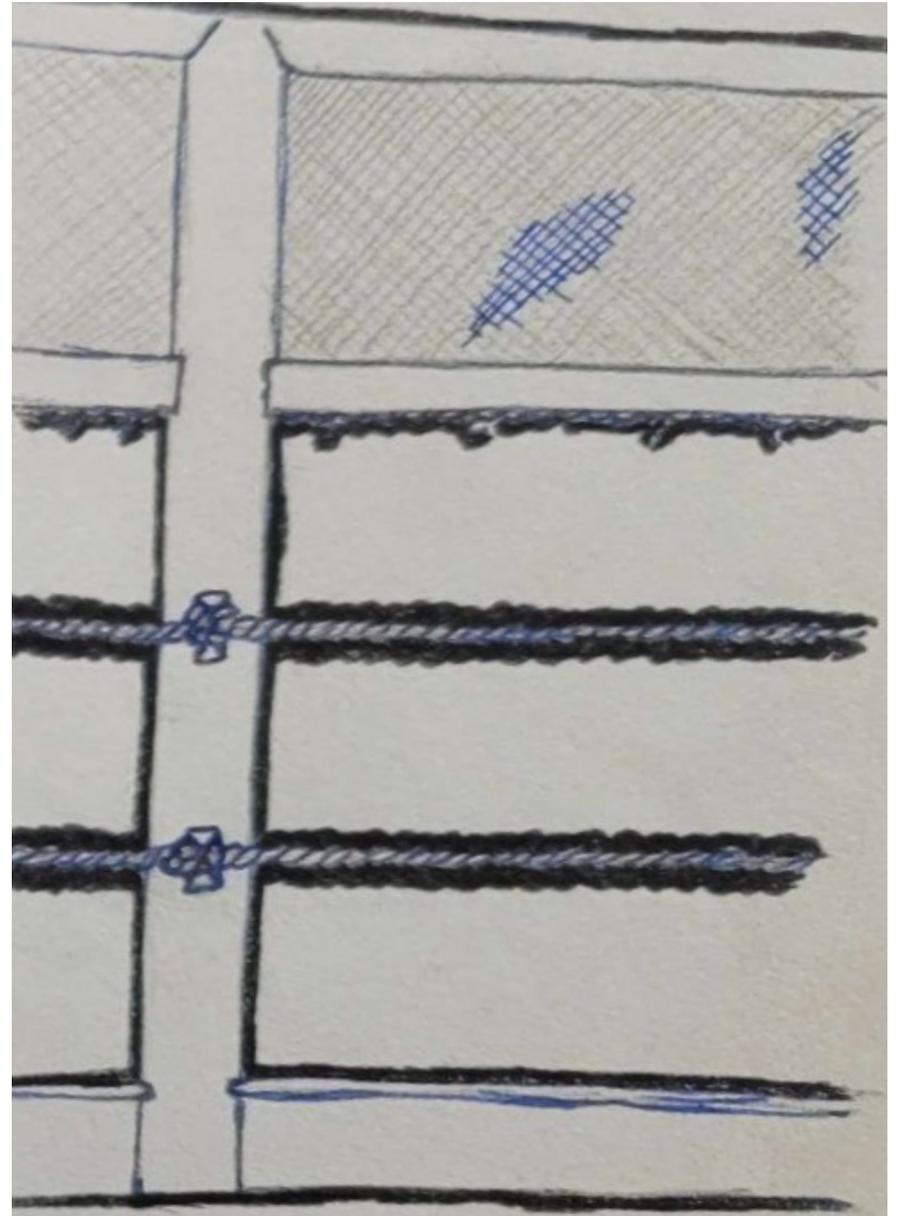
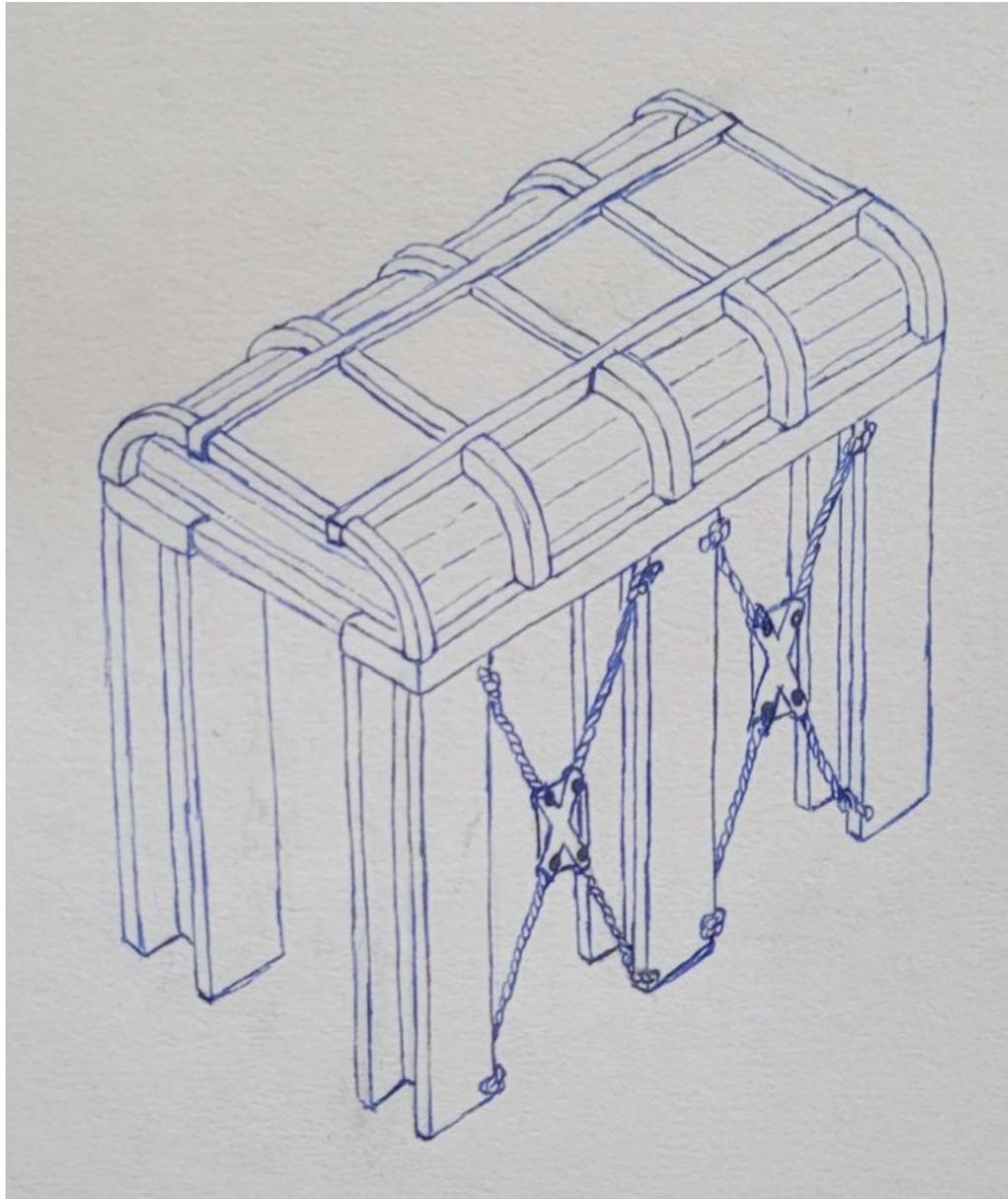
Moodboard formes



Concept art objets







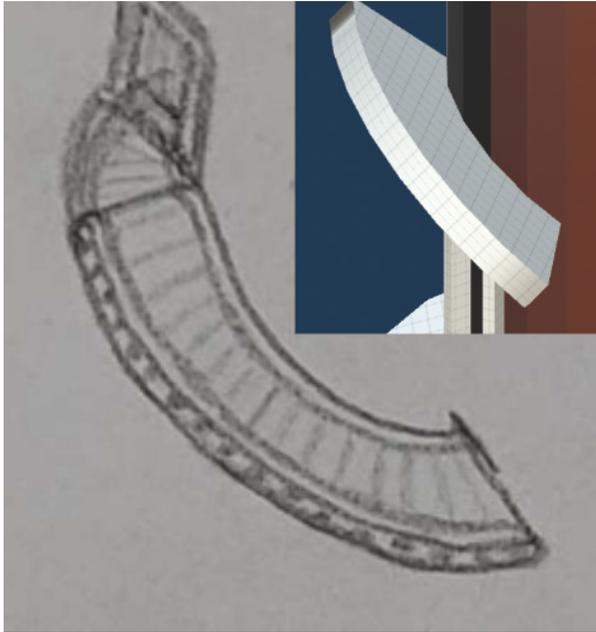
OBJETS

Fiches documentaires individuelles des pattern de level design

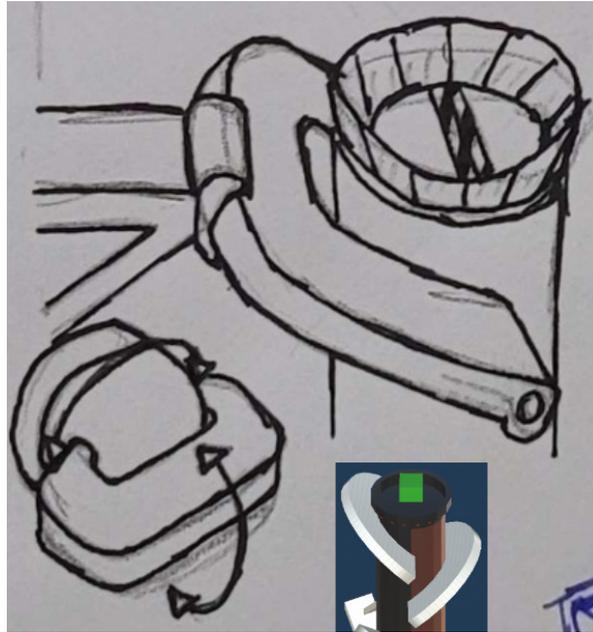
Murs du puits

Lifesaver

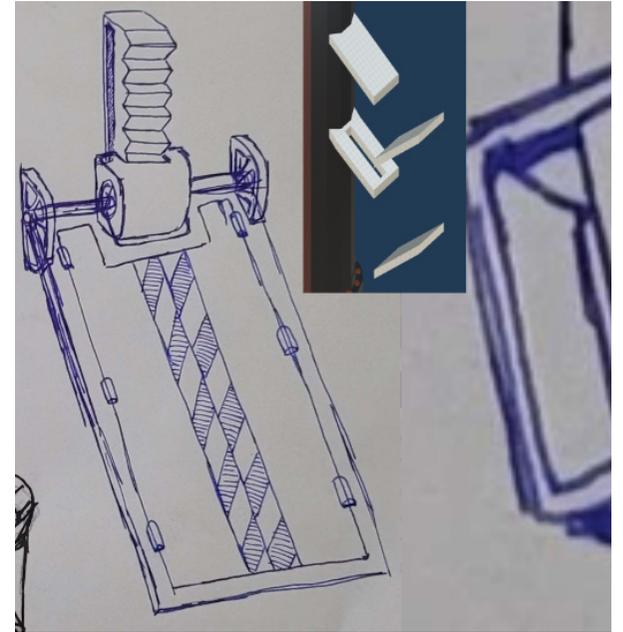
Pattern 1: concept art



Conveyor belt

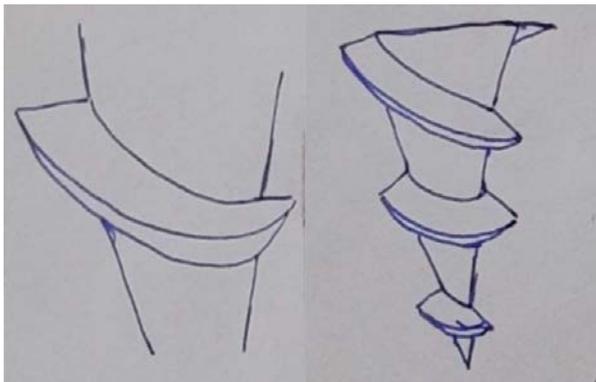


Metal holding devices, holding drill in place



Moveable platform (gates)

Cosmetic drillbit pointing down, not implemented

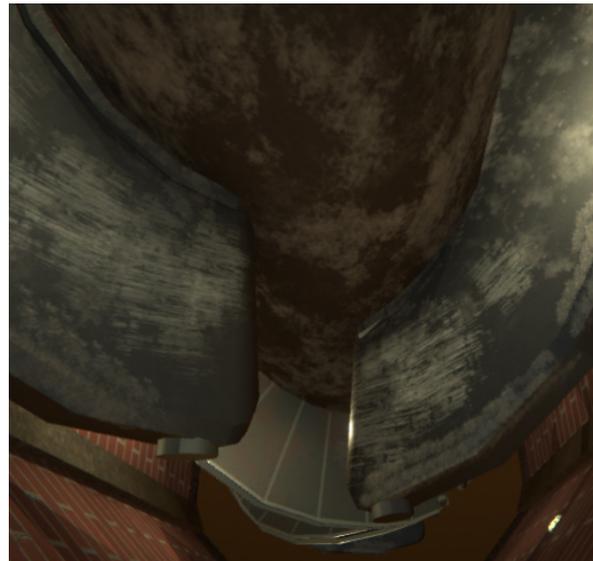


Pattern 1: rendu final



Ce pattern est inspiré d'un silo de stockage autour duquel tourne le player. Les éléments sur lequel le joueur slide ont tous plus ou moins la même forme (tapis roulants et pinces en métal), de façon à ce que le joueur ne questionne par leur utilité en pleine chute.

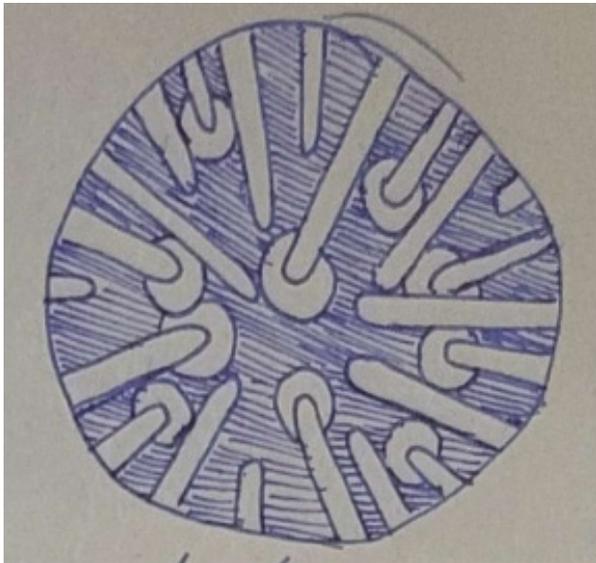
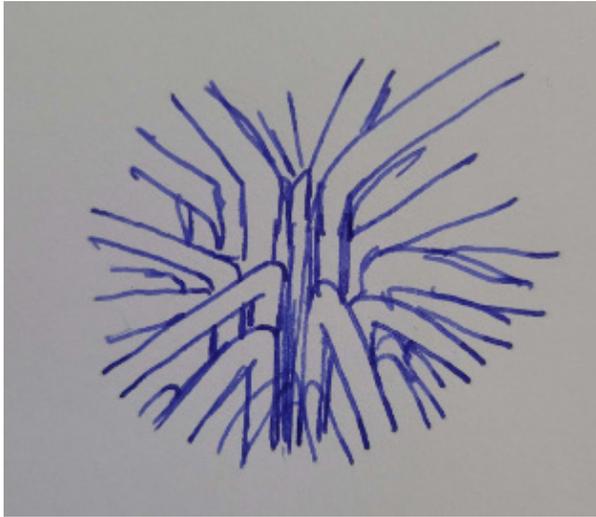
Les plateformes avec les marquages jaunes sont, elles, des plateformes ascenseurs qui servent à remonter des objets du puits. Les marquages de sécurité sur les portes de ces plateformes aident le joueur à comprendre qu'il faut se retourner lorsqu'il fait face à l'une d'entre elles tout en lui indiquant la direction à suivre.



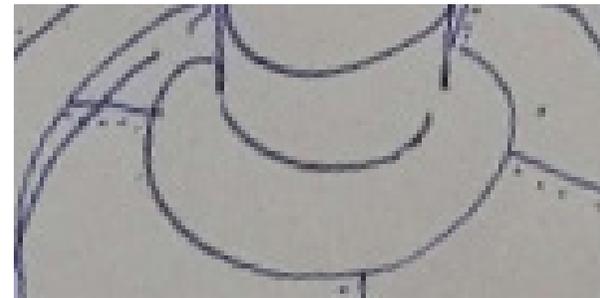
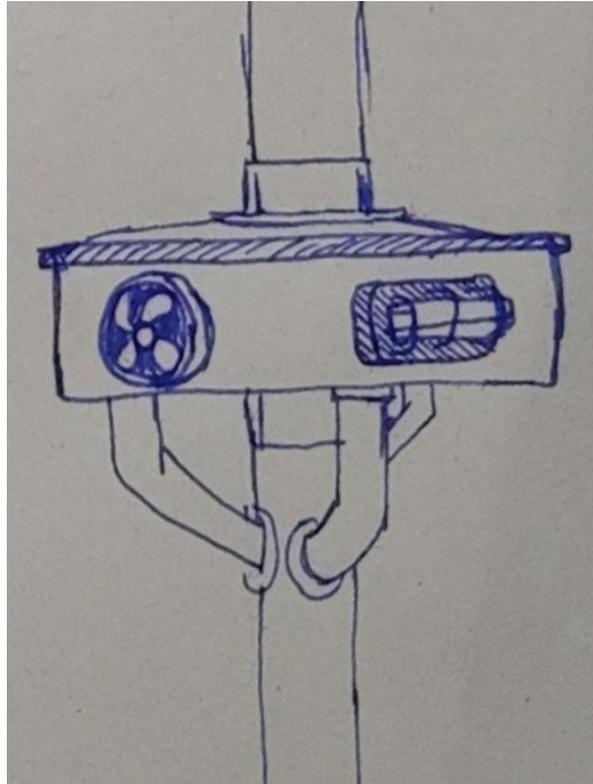
Les tapis roulant s'inscrivent dans la même logique de transport de matériaux que les plateformes montantes même si certains se sont abîmés ou effondrés avec le temps.

Les lumières sont placées de manière à rendre la circulation fluide peu importe le passage emprunté et à mettre en avant les opportunités proposées par le pattern.

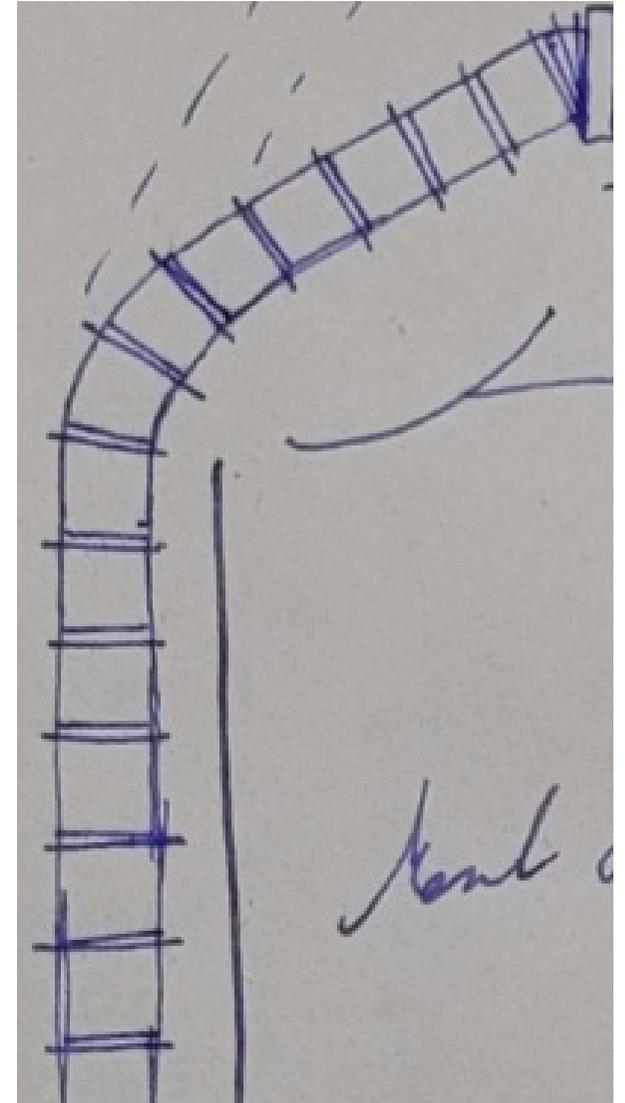
Pattern 2: concept art



Vue de dessus

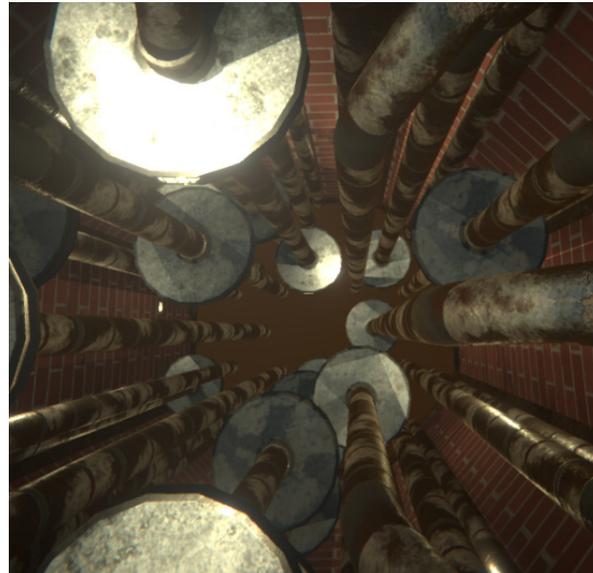
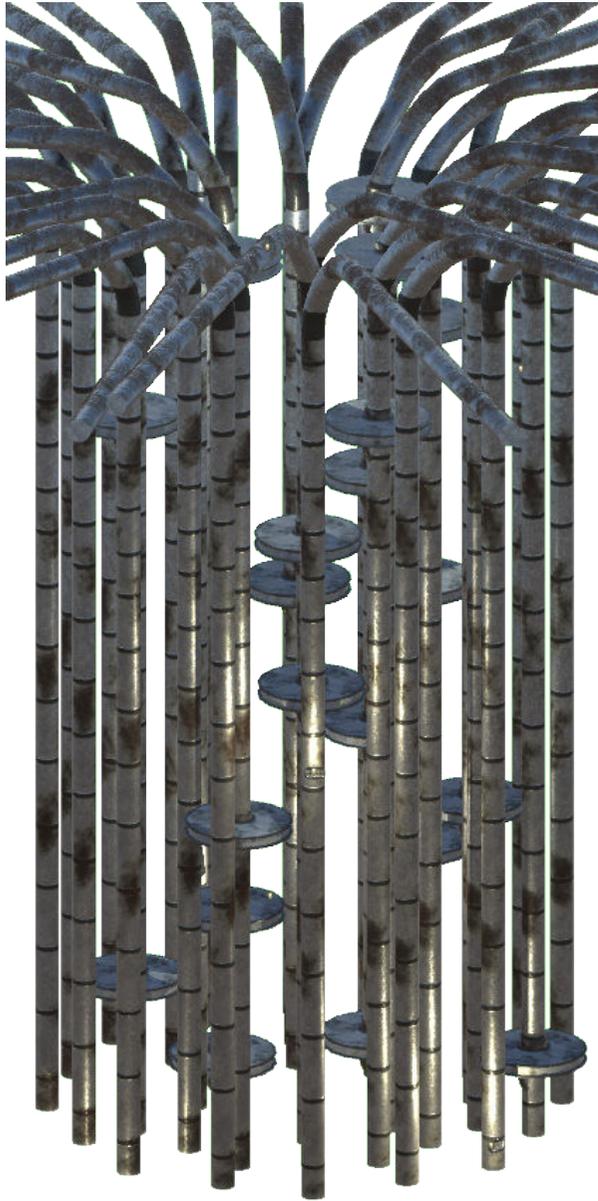


Module plateforme



Torsion du tuyau

Pattern 2: rendu final



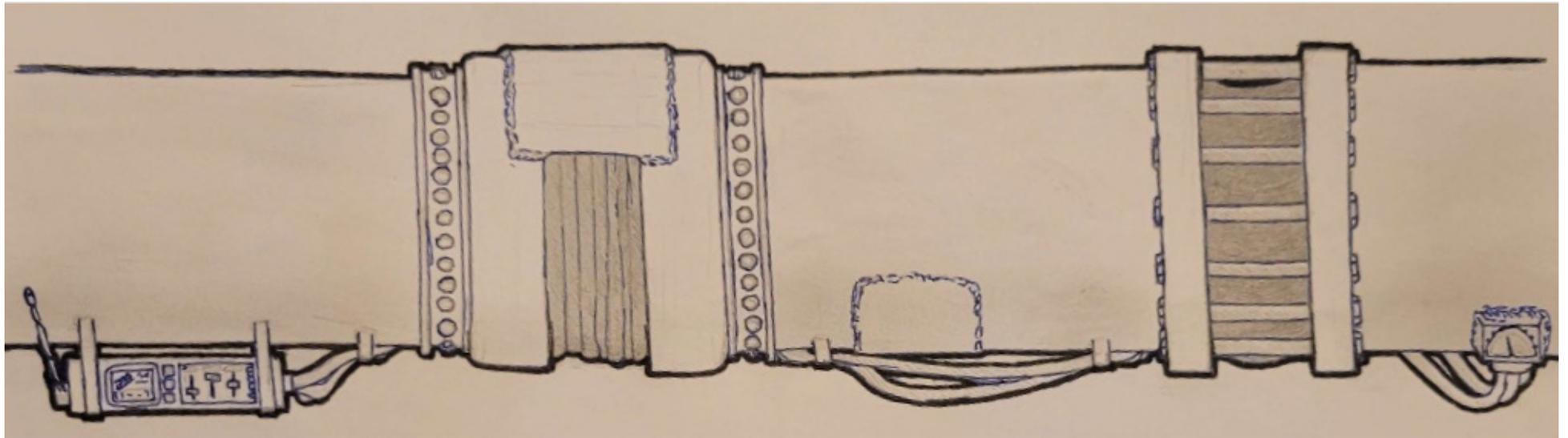
Ce pattern présente un amas organique de tuyaux fuyants en une direction avec des marquages et des plateformes appuyant la sensation de vitesse de chute lorsque le joueur le traverse.

Ces tuyaux ne sont plus utilisés, leur état est déplorable et ils ont rouillé à cause de l'eau et des débris qui tombent dessus depuis toutes ces années.

La forme si particulière de ce pattern vu du dessus laisse imaginer des lianes tombantes dans la jungle, accentuant le sentiment de vertige.

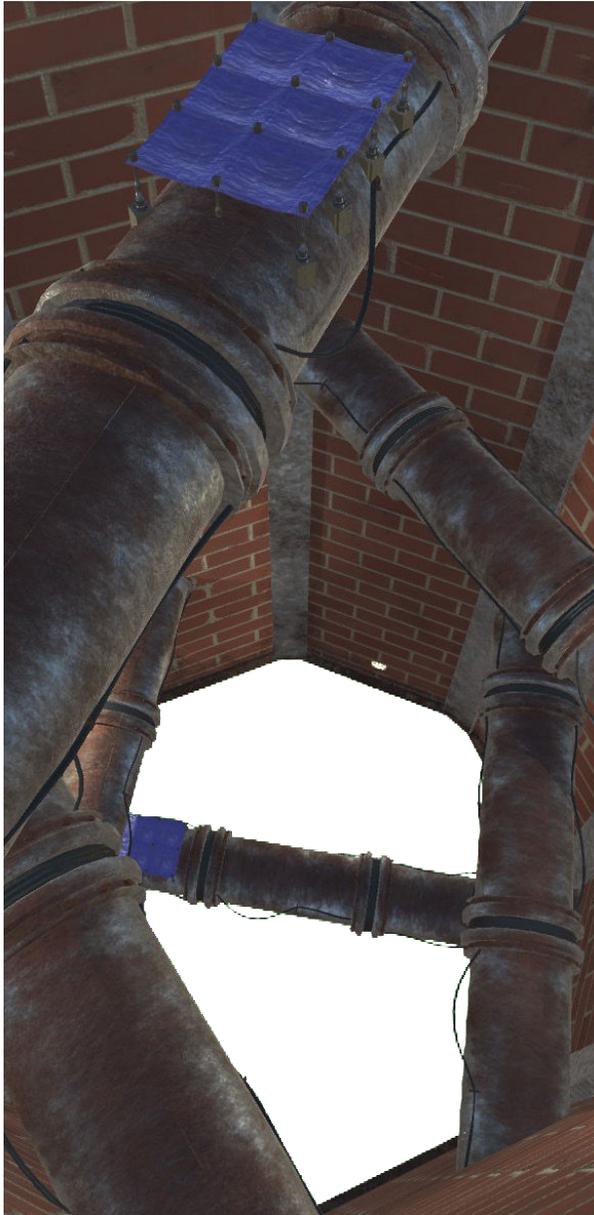
Les plateformes, quant à elles peuvent être interprétées comme des filtres qui sont désormais hors service dû à l'état actuel des tuyaux. En effet la coupure brutale des tuyaux laisse imaginer un problème de conception des créateurs de ce puits sans fond.

Pattern 3: concept art



Tuyau avec plusieurs modules techniques

Pattern 3: rendu final



Les tuyaux de ce pattern sont évidemment bien plus solides que ceux du pattern numéro 2 dû aux éléments potentiellement dangereux qu'ils transportent.

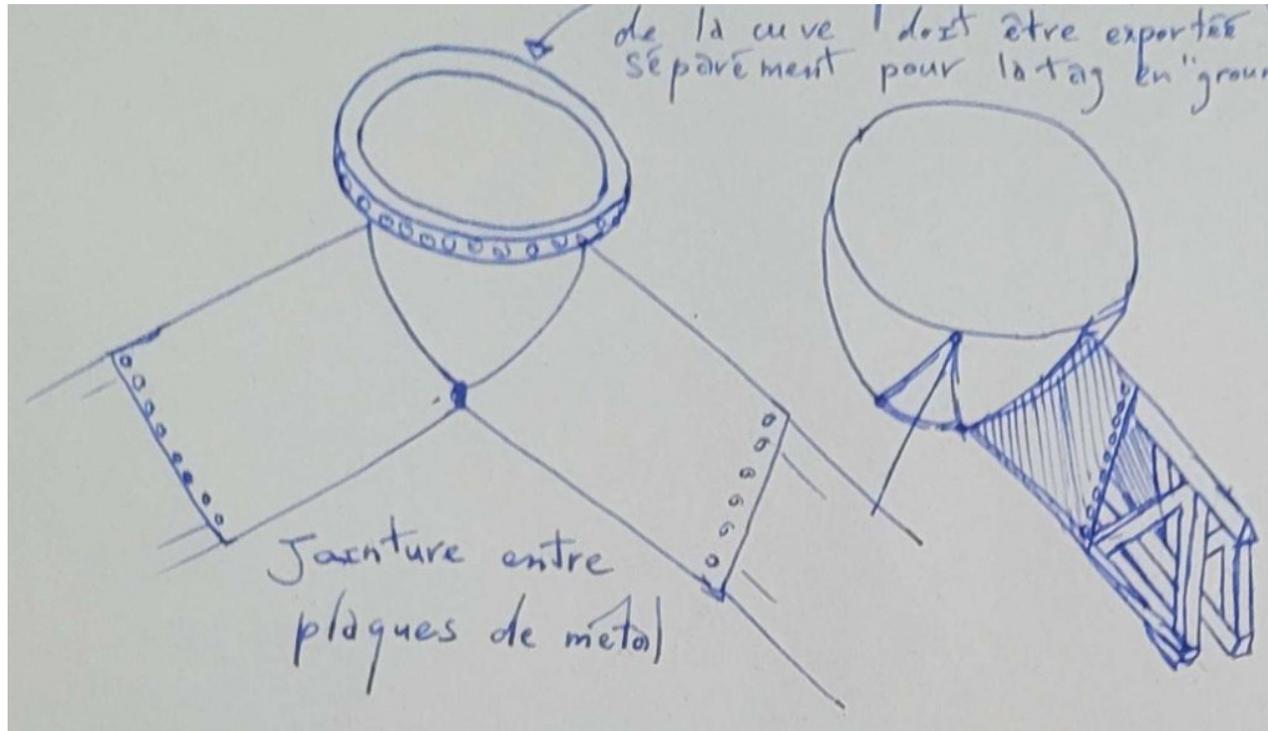
Malgré tout, leur usure est tout de même avancée dû aux différents éléments qui tombent dans le puits.

Plusieurs câbles électriques longent les tuyaux, ceux-ci peuvent servir à réguler la pression ou à contrôler l'état du contenu du tuyau.

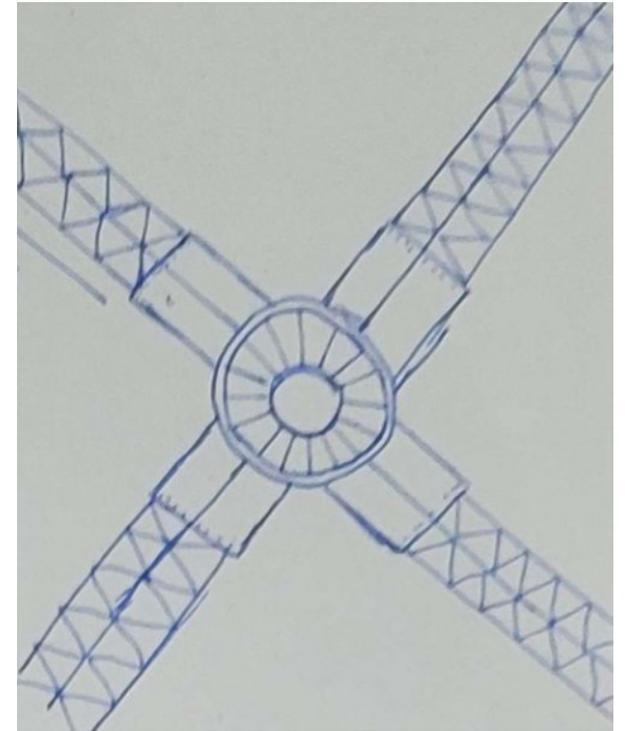
Malgré tout, nous pouvons constater la mauvaise organisation de la structure de ce puit dû à la façon dont les tuyaux y sont positionnés. En effet, leur agencement ne fait que peu de sens et rappellent la façon dont les tuyaux du pattern 2 étaient positionnés (disposition organique et peu régulée).



Pattern 4: concept art



Coupe et attache des supports



Proportions des clusters

Pattern 4: rendu final



Les éléments en forme d'entonnoir de ce pattern invitent logiquement le joueur à s'y plonger pour les enchaîner les uns à la suite des autres.

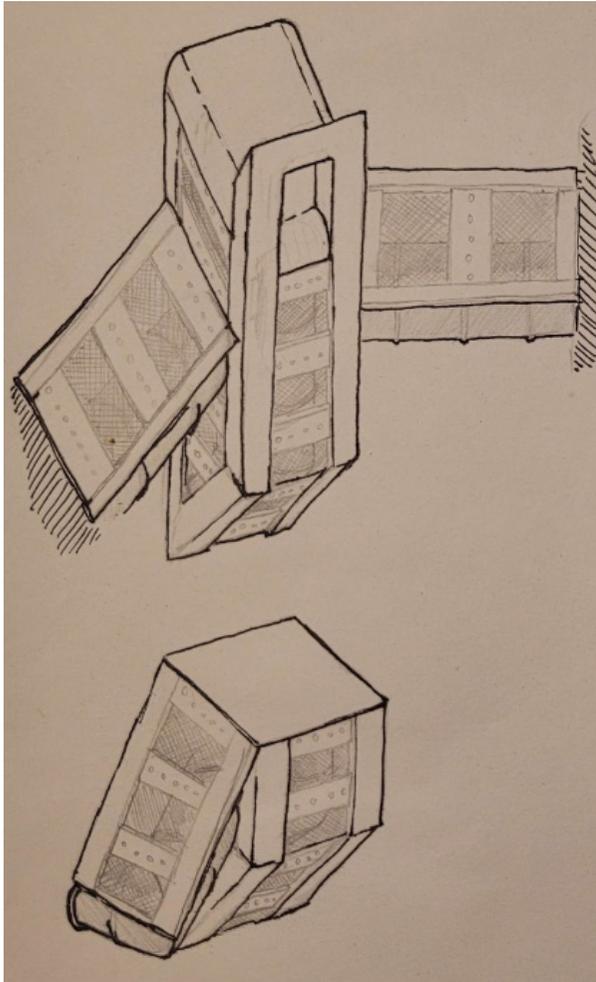
Les barreaux qui tiennent les entonnoirs sont volontairement d'une autre couleur et troués afin de pousser le joueur à se déplacer vers le centre des entonnoirs.

Les marquages sur le métal convergent vers le centre de chaque entonnoir donnant un visuel hypnotisant lorsqu'on traverse plusieurs d'entre-eux d'affilé. Cet effet récompense le joueur et ne dépend que de sa précision.

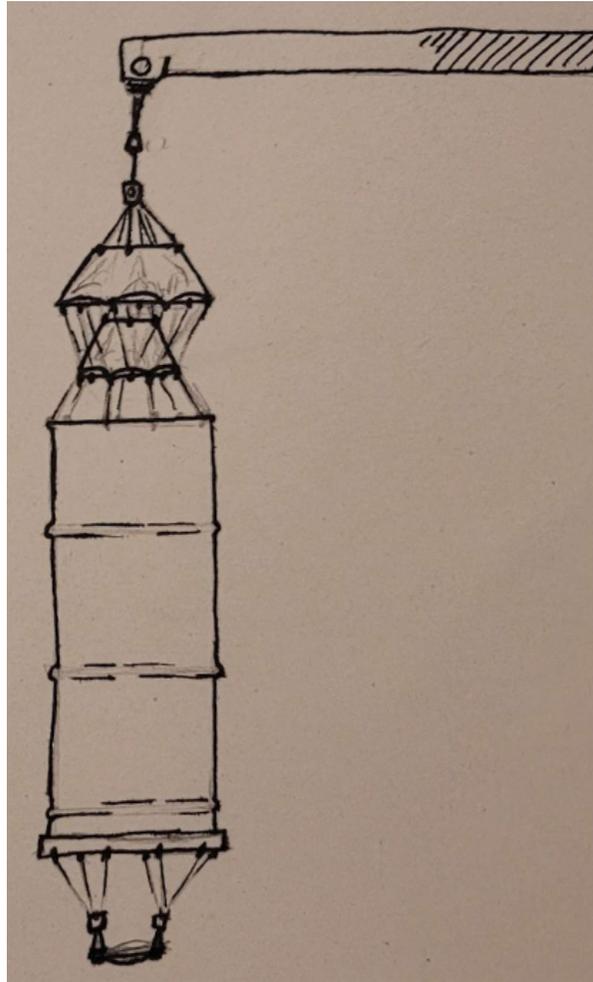
Les marques de griffures au centre des entonnoirs laissent penser que des éléments passaient dedans avant que le joueur ne les traverse mais il n'est jamais clairement expliqué à quoi servaient ces structures.

Enfin chaque entonnoir est éclairé de façon à aider le joueur à s'orienter à la sortie d'un entonnoir et ainsi de pouvoir les enchaîner sans demander une trop grande charge cognitive.

Pattern 5: concept art

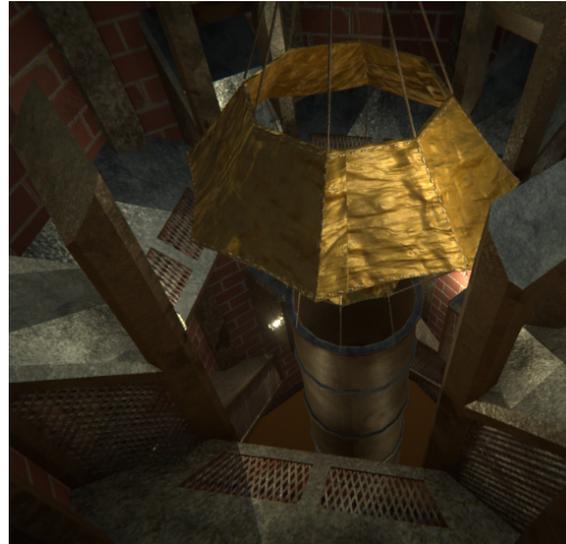
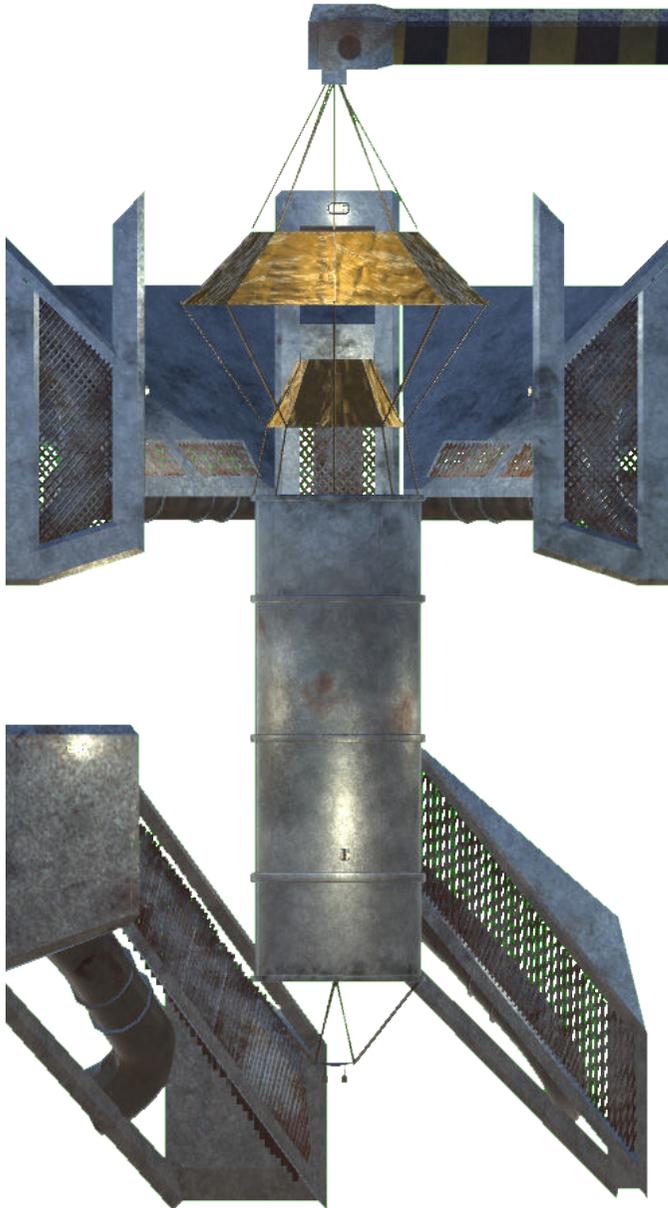


Poutres et grillages, justifications narratives



Ilot central, justification structurelle

Pattern 5: rendu final



Ce pattern est assez complexe visuellement comparé aux autres mais c'est à l'image des possibilités de chemin qu'il représente. En effet, le joueur peut utiliser pleins de passage différents tout en restant dans une section cohérente.

En effet, toutes les plateformes de ce pattern sont destinés à la protection des tuyaux qui passent derrière les grillages.

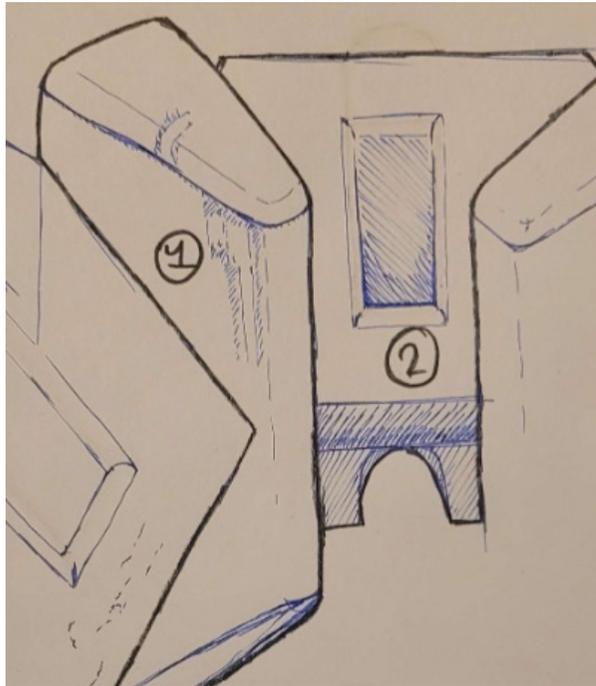
Même les bâches protègent le nouveau tuyau en cours d'installation qui se trouve au milieu.

Il était important que malgré une texture partagée avec celle des "Life Savers", les bâches jaunes ne fassent pas penser au joueur qu'elle lui sauveront la vie. C'est pour cela que la couleur ainsi que la forme différencient énormément de celle des "Life Savers".

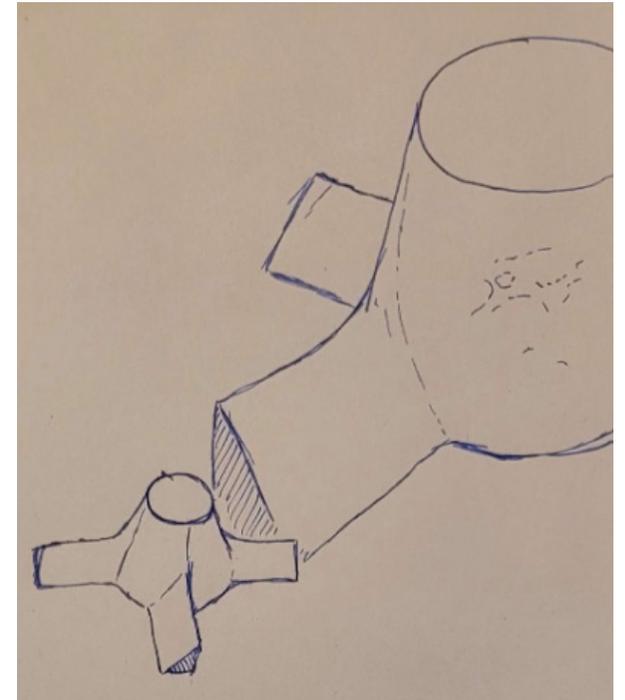
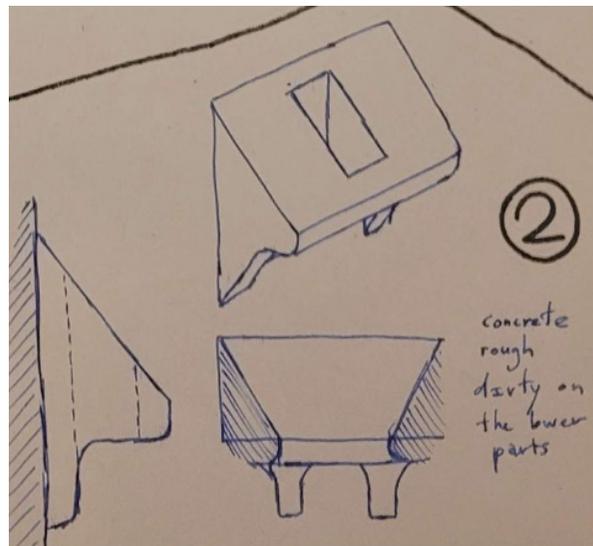
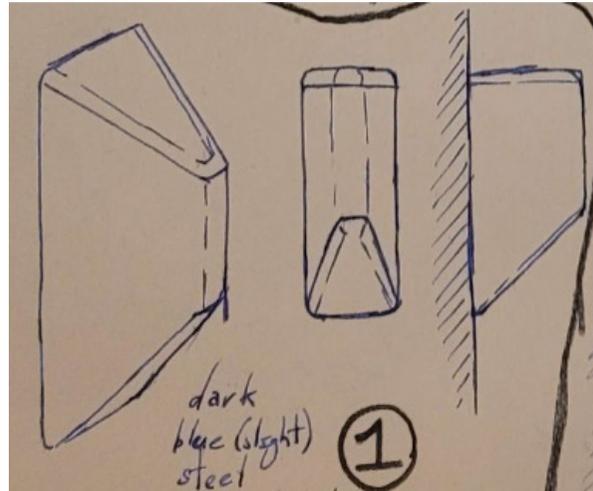
Les éléments comme les cordes qui soutiennent le tuyau central peuvent être traversés par le joueur car ils ne sont pas mis en évidence et ne servent qu'à expliquer le fonctionnement de ce pattern.

Enfin, il était important que malgré sa complexité, le pattern ne demande pas au joueur de se concentrer sur ces visuels pour le comprendre. C'est pour cela que les surfaces utilisables sont mises en avant avec des lumières.

Pattern 6: concept art

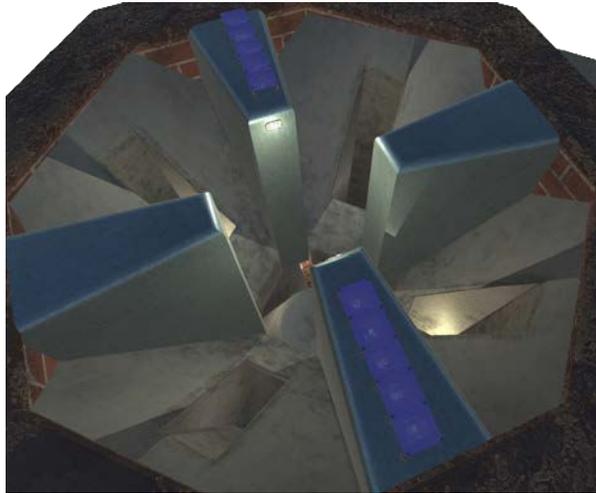


Partie supérieure du pattern, et ses composants



Partie inférieure du pattern

Pattern 6: rendu final



Ce pattern est une immense structure en béton prenant la forme globale d'un entonnoir pour sévasser sur sa partie inférieure.

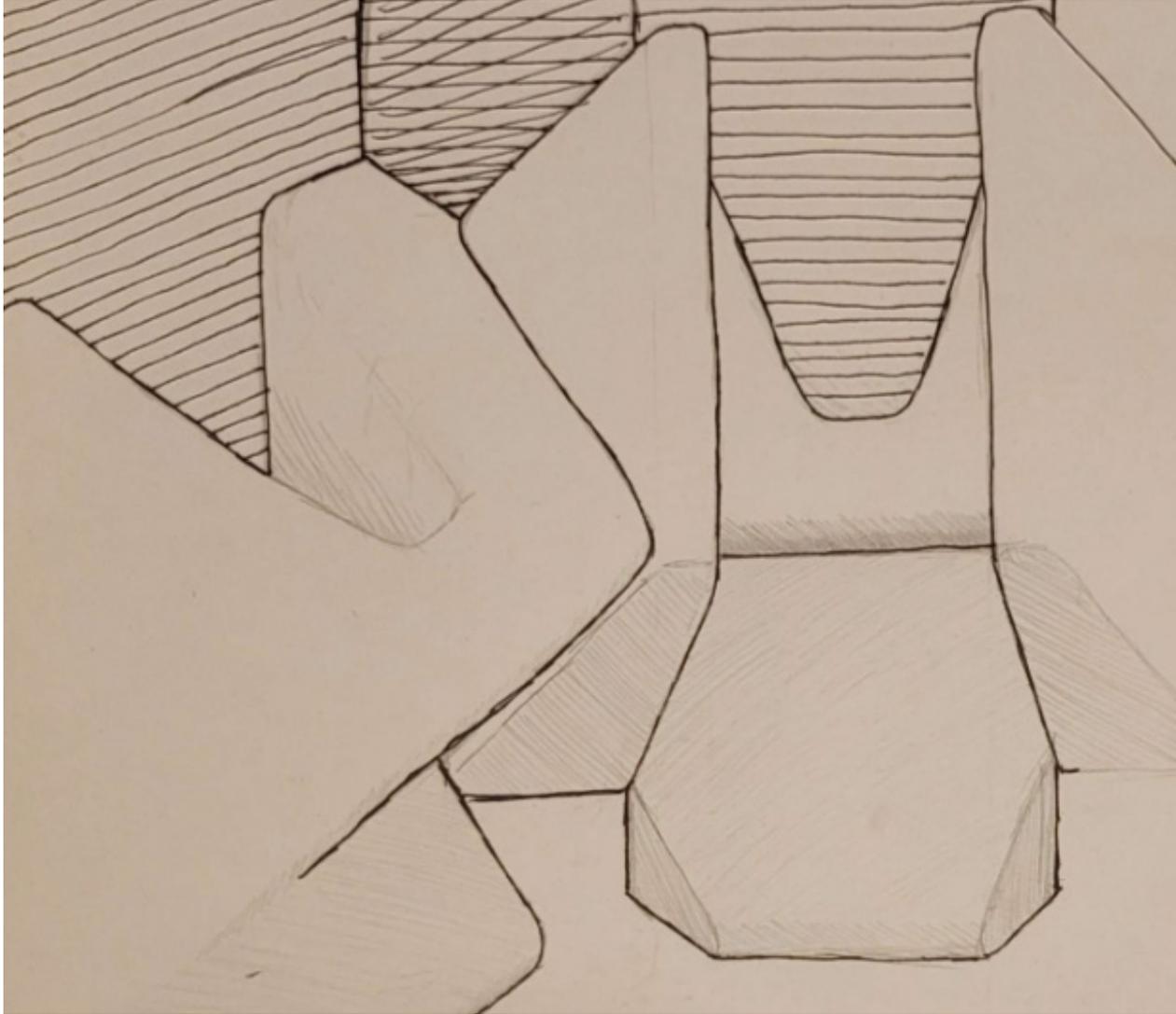
Cette structure était probablement une structure de tri des éléments qui tombent dans le puit mais de par son aspect bétonné, on en déduit qu'elle n'est pas finie ou ne doit plus être utilisée.

Cette structure sert aussi d'élément de fortification du puits, solidifiant sa structure et ses fondations.

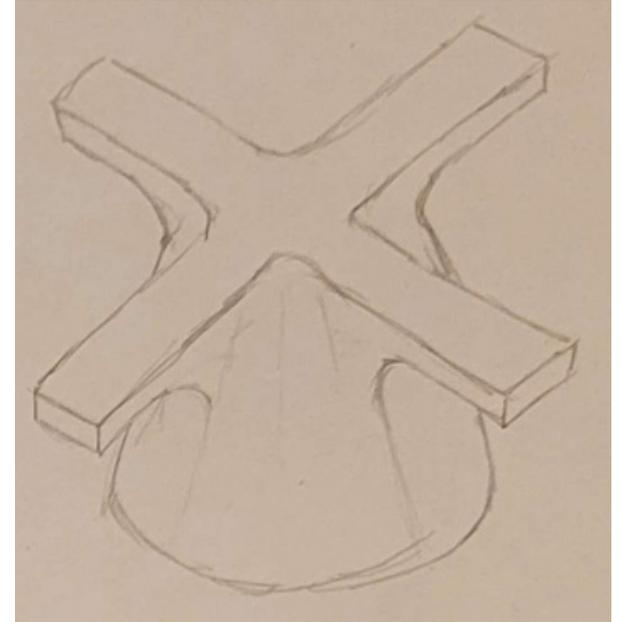
On devine tout de même le passage d'anciens débris au centre de l'entonnoir à cause de la saleté et de la pourriture qui s'est incrustée dans le béton.



Pattern 7: concept art



Partie supérieure du pattern



Partie inférieure du pattern

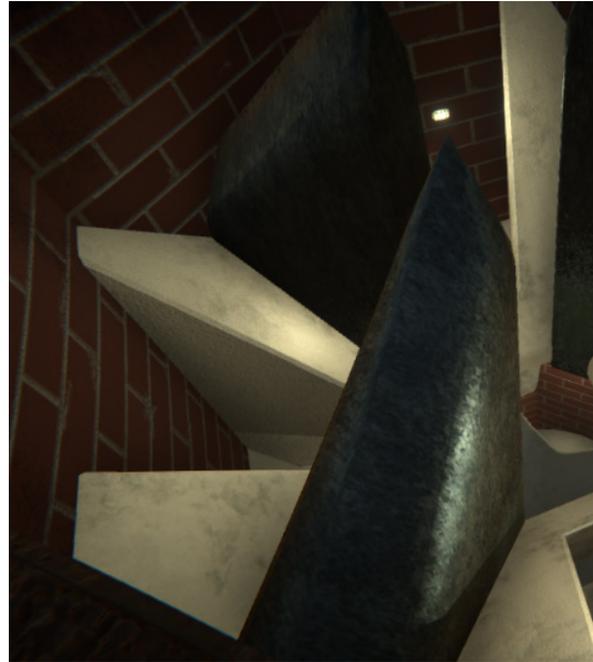
Pattern 7: rendu final

Ce pattern présente une structure très similaire à celle du pattern 6. En effet, on en déduit que plusieurs structures servant à maintenir le puits sous différentes formes existent en son sein. Malgré tout, celle-ci paraît plus récente et donc plus performante que sa forme précédente.

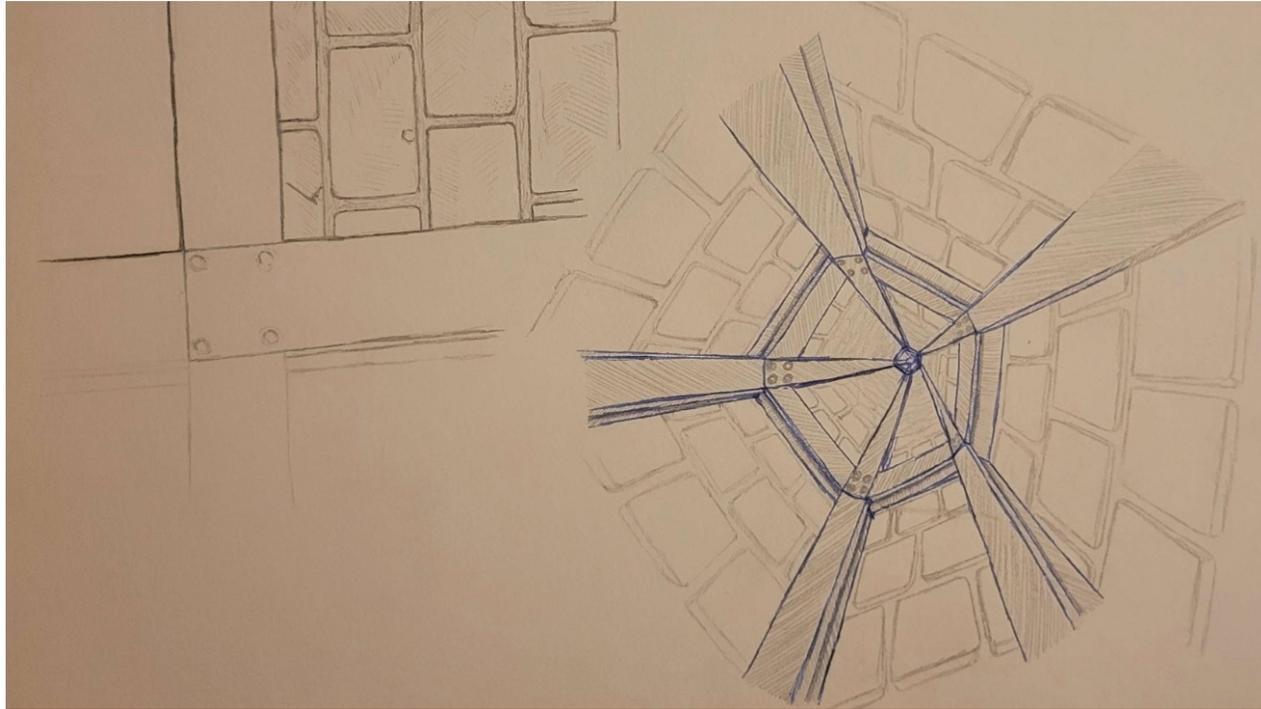
En effet, cette structure est plus équilibrée et plus complexe que sa forme précédente. On y trouve moins de saletés et elle paraît plus complète.

Malgré cela, on reconnaît les murs de métal bleu ainsi que l'entonnoir en béton permettant de maintenir la structure du puit dans un état plus ou moins correct.

L'aspect béton sur les pentes est le même que sur le pattern 6, ce qui fait tout de suite comprendre au joueur que les mouvements qu'il va devoir effectuer sur ce pattern sont similaires à ceux du pattern 6 mais il y trouvera une manière différente de s'y faufiler s'il exploite tous les volumes du pattern.



Murs du puits



Les murs du puits sont constitués de briques rouges de grande taille qui rappellent les usines de la révolution industrielle.

Celles-ci sont en plutôt bon état au vu du peu de personnes qui passent pour les abîmer mais sont tout de même un peu sales ou dégradés par endroit à cause de leur ancienneté.

Selon les patterns, les briques sont soutenues par des poutres massives en métal qui permettent de rendre la structure plus solide et moins plate visuellement.

Le but des briques était de profiter des lignes horizontales qui composent le mur pour donner un effet stroboscopique à la chute du joueur. En effet, ces lignes servent de repère dans l'espace et permettent au joueur d'évaluer sa vitesse d'un pattern à un autre.

Les murs sont sectionnés entre chaque pattern, permettant au joueur de visualiser la fin de l'épreuve qu'il est entrain de parcourir et d'avoir un nouveau repère dans l'espace qui coupe le motif répétitif des briques rouges.



“Life Savers”

Le “Life Saver” est un élément incontournable pour se rattraper d’une chute trop longue, c’est pour cela qu’il s’agit d’un élément visible de loin et reconnaissable afin que le joueur puisse aisément les repérer.

Il s’agit d’une simple bâche bleue de chantier que le joueur identifie comme bénéfique dès les premières secondes de jeu.

Lorsque que le joueur tombe de trop haut et que sa vision se trouble, alors les “Life Saver” deviennent “lumineux” afin d’aider le joueur à survivre à sa chute mortelle.



TECH ART

Post processing

Visual effects

Post processing

Les filtres de post processing servent l'ambiance et les différents éléments du jeu. Etant en première personne, nous avons utilisé un effet de fisheye permettant de donner un aspect plus "rond" au champ de vision, plongeant le joueur dans l'immersion.

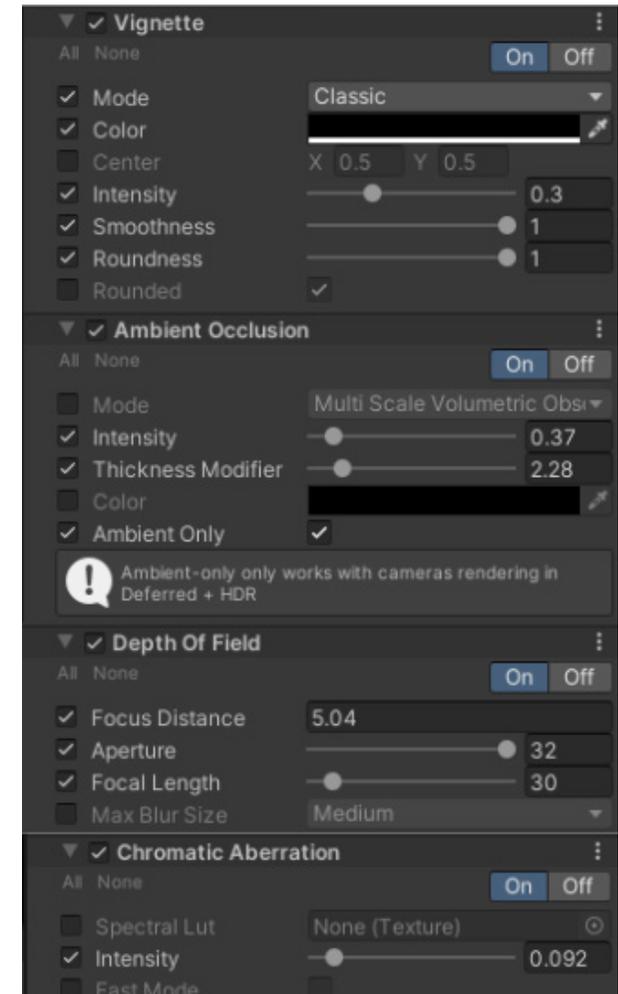
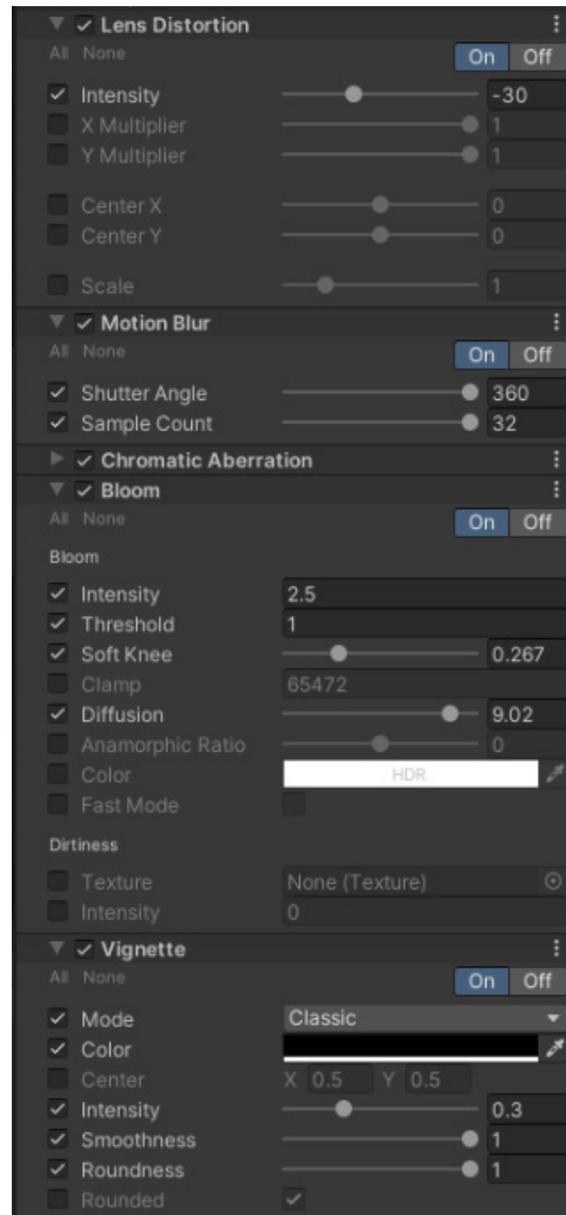
L'occlusion ambiante a permis d'ancrer les différents éléments dans le décor ainsi que de les rendre moins flottants en accentuant les ombres dans les recoins.

Le flou de mouvement permet au joueur de ressentir une certaine fluidité et immersion lors des mouvements de caméra rapides qui peuvent être effectués en jeu en plus de la vitesse importante en chute.

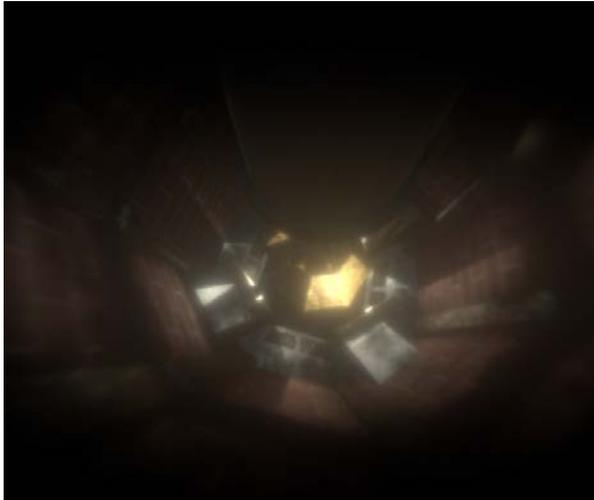
Le bloom sert, quant à lui, à rendre les lumières un peu plus aveuglantes, notamment celle de l'étoile qui nous tombe dessus qui doit traduire un vrai sentiment de danger lorsqu'elle est proche.

Celle-ci doit être éblouissante et intimidante.

La vignette sert à noircir les contours dans le but de compléter la profondeur de champ, donnant au joueur un aspect plus réaliste au champ de vision sans prendre le pas sur la lisibilité de jeu.



Visual Effects



Les éléments de type VFX viennent se rajouter par dessus le champ de vision du joueur afin de lui communiquer des informations de statut de manière intradiégétique.

Lors de la prise de dégât du joueur un filtre rouge se dépose sur le champ de vision afin de lui faire comprendre qu'il est dans un état critique. Le filtre est moins puissant sur le centre de l'image et concerne principalement la vision périphérique du joueur.

De cette façon le joueur est informé de son état actuel de manière claire sans trop impacter son gameplay.



A l'inverse, lors d'une chute trop longue, un voile sombre et flou vient se déposer sur le champ de vision du joueur. Ce voile représente la perte de conscience du personnage car sa chute dure trop longtemps.

Cela sert à indiquer au joueur qu'il est sur le point de prendre des dégâts de chute.

Ce voile met en évidence les "Life Savers", rendant la lisibilité du LD plus compliquée mais aidant le joueur à se réceptionner sur un "Life Saver".

Tous ces éléments apparaissent de façon intradiégétique de manière à garder le joueur en immersion dans l'ambiance sombre et dangereuse du puits.

En plus de ces filtres un effet de shaking sur la caméra apparaît lorsque l'étoile se rapproche trop du joueur, mettant une pression en plus de la lumière intensive provoquée par l'astre.

SOUND DESIGN

OVERVIEW

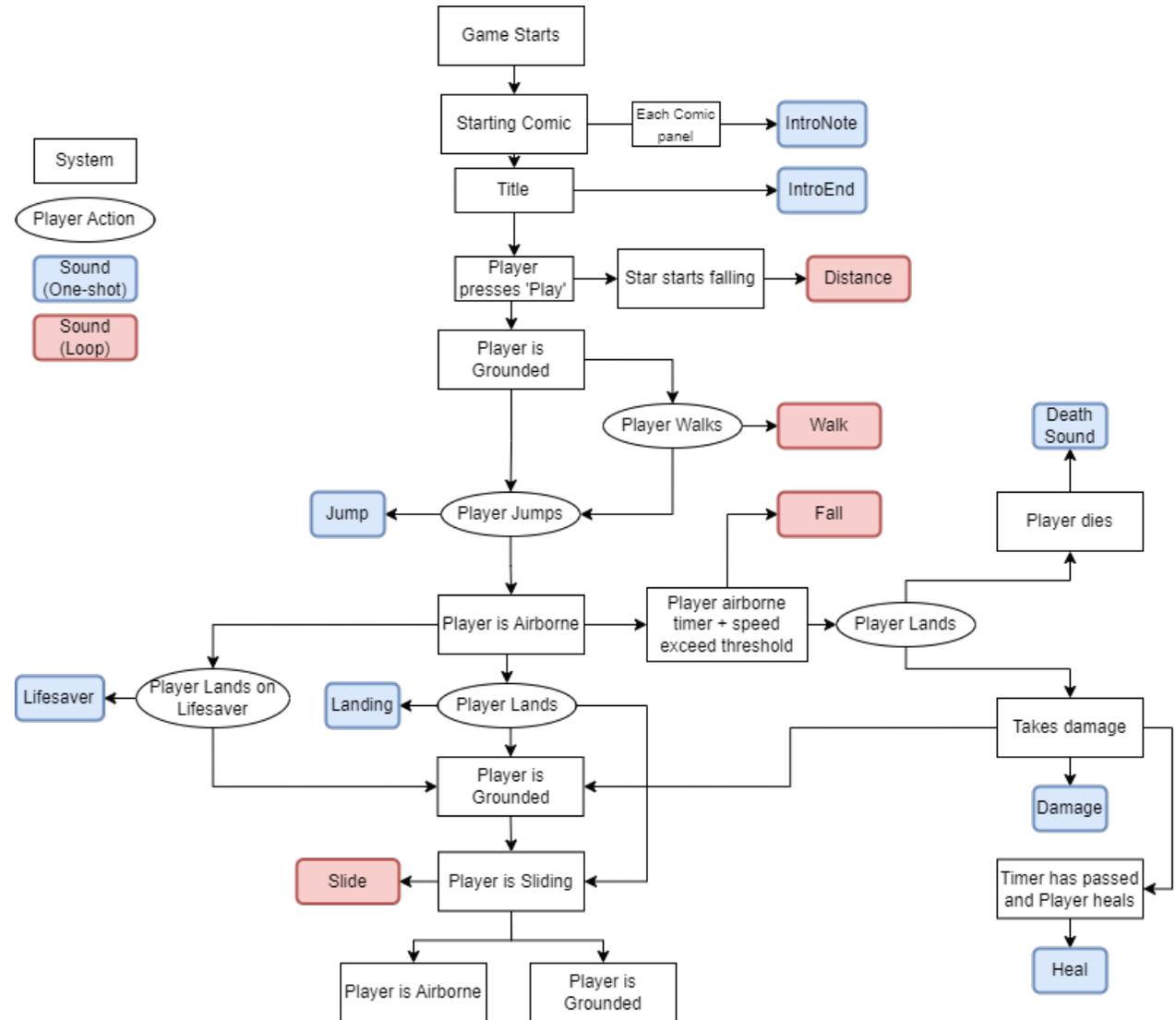
Intentions et contraintes
Inspirations et références
Event list

Intentions et Contraintes

Contexte

The Pit est un jeu de chute où le joueur essaie d'échapper à une étoile qui le poursuit constamment en descendant dans un puit sans fond, détruisant tout sur son passage. De ce fait le joueur est dans un état constant de mouvement, essayant de réguler sa chute avec les structures du puit pour à la fois ne pas se faire carboniser par l'étoile et ne pas mourir d'une chute trop conséquente. L'univers est industriel, froid et sombre, opposé à la radiance et la chaleur de l'étoile qui nous poursuit. A cet effet, l'univers sonore s'accorde avec l'univers visuel par plusieurs facteurs :

- Absence de musique, pour créer une atmosphère oppressante et rajouter de la clarté aux feedbacks
- Un modificateur de reverb sur les sons pour renforcer l'ambiance du puits sans fond et donner une sensation de grandeur
- Un gameplay basé sur la vitesse qui influence les paramètres de plusieurs sons, comme le volume ou le reverb



Boucle de prédiction sonore

Inspirations et Références

Inspirations et Références

Les références principales pour la DA sonore de ce projet sont Lost In Vivo et Lethal Company.

Lost In Vivo est un jeu d'exploration horreur avec une ambiance sonore qui possède de longs moments de silence pesants et des sections beaucoup plus bruyantes et oppressantes. L'univers sonore est très intéressant car il possède des traitements sonores qui reflètent très bien l'ambiance du jeu, notamment le filtre donnant de l'écho à tous les sons pour évoquer l'environnement du jeu.

Lethal Company est un autre jeu avec une ambiance similaire, il se déroule dans des environnements industriels sombres similaires aux nôtres et des traitements sonores similaires à ceux de Lost In Vivo. Les sons de Lethal Company sont soumis à un filtre similaire à celui de Lost In Vivo et possèdent un timbre similaire à celui que nous recherchons.

Ces deux jeux sont également dépourvus de musique, ce qui joue en la faveur d'une ambiance plus oppressante ainsi que d'un sentiment d'isolation.

Subnautica est un jeu qui possède un feedback très pertinent, il s'agit d'un jeu de plongée où le joueur possède une barre d'oxygène, si cette dernière arrive à zéro, il meurt. Lorsque le joueur va mourir à cause de cette dernière les sons du jeu subissent une perte de volume progressive ainsi qu'un muffling, cependant il peut tout de même survivre en remontant à la surface à la dernière seconde. Cet état est similaire à la chute de notre projet, le joueur peut perdre le contrôle de sa chute et rentrer dans un état où il est essentiellement 'mort' car il ne peut plus toucher le sol sans mourir, notre jeu possède cependant des éléments de Level Design sur lesquelles le joueur peut atterrir sans mourir.



Event list

Priorité	Type d'Objet (Folder)	Event	FMOD File Name	Loop	Type	Finition	Trigger	Script Unity	Unity Trigger	Description	Reference
2	Player	Walking 4-5 sons (variations d'un même son)	Walking	No	2D Event	Intégré	Player walks	S_SoundController		Walking on metal	https://www.youtube.com/watch?v=olm2&t=2032 lethal company metal footsteps
3	Player	Fall 3 paliers	Fall	Yes	2D Event	Intégré	Player speed exceeds threshold	S_CtrlPosiProcess		Wind resistance sounds, big fall, wind passing by	
4	Player	Landing 3 states Deafens Walking while played	Landing	No	2D Event	Intégré	Player lands after a fall	S_SoundController		Bonk metallic	
1	Environment	Platform Creaking	Platform_Creak	No	2D Event	Abandoned	Timer sur plateforme, atteint 0 le creak joue. Réduit naturellement, lorsque le joueur est grounded, réduction du timer de manière aléatoire lorsque le joueur prend un pas.			Metal squeak, metal grind, creaking	

Priorité

La colonne priorité est un outil d'organisation, elle donne un degré d'importance aux feedbacks sonores pour donner un point de repère pendant la production, quels feedbacks à designer/intégrer en premiers.

Type d'Objet (Folder)

Cette catégorie définit à quel type de feedback appartient l'évènement et dans quel dossier FMOD il se situe, les codes couleurs de l'évènement list sont identiques à ceux des fichiers FMOD.

Event

Event précise de quel feedback il s'agit et ajoute des notes importantes supplémentaires.

FMOD File Name

Cette colonne contient le nom exact de l'évènement sur le FMOD du projet, elle permet de trouver facilement l'évènement sur le logiciel.

Loop

La colonne loop sert à indiquer si l'évènement est un son qui effectue une boucle parfaite.

Type

Cette catégorie indique s'il s'agit d'un son spatialisé ou d'un évènement 2D.

Finition

La colonne finition permet de suivre l'état de finition des feedbacks : s'ils sont designés, implémentés, abandonnés, etc.

Trigger

Le trigger est l'expression verbale de la condition d'activation désirée pour un évènement. Elle aide pour l'intégration dans le moteur de jeu.

Script Unity

Cette colonne renseigne le script unity utilisé pour déclencher l'évènement.

Unity Trigger

Cette catégorie renseigne le code qui permet de déclencher l'évènement.

Description

La description du son que l'on veut que l'évènement produise, avec des notes additionnelles sur les effets ou l'implémentation souhaitée.

Reference

Cette colonne note les références utilisées pour les sons si il y'en a.



EVENTS

Fiches documentaires individuelles

Event commenté: distance de l'étoile

Layering

Cet événement est constitué de plusieurs ambiances sonores différentes qui évoluent en fonction de la proximité du joueur avec l'étoile qui le poursuit.

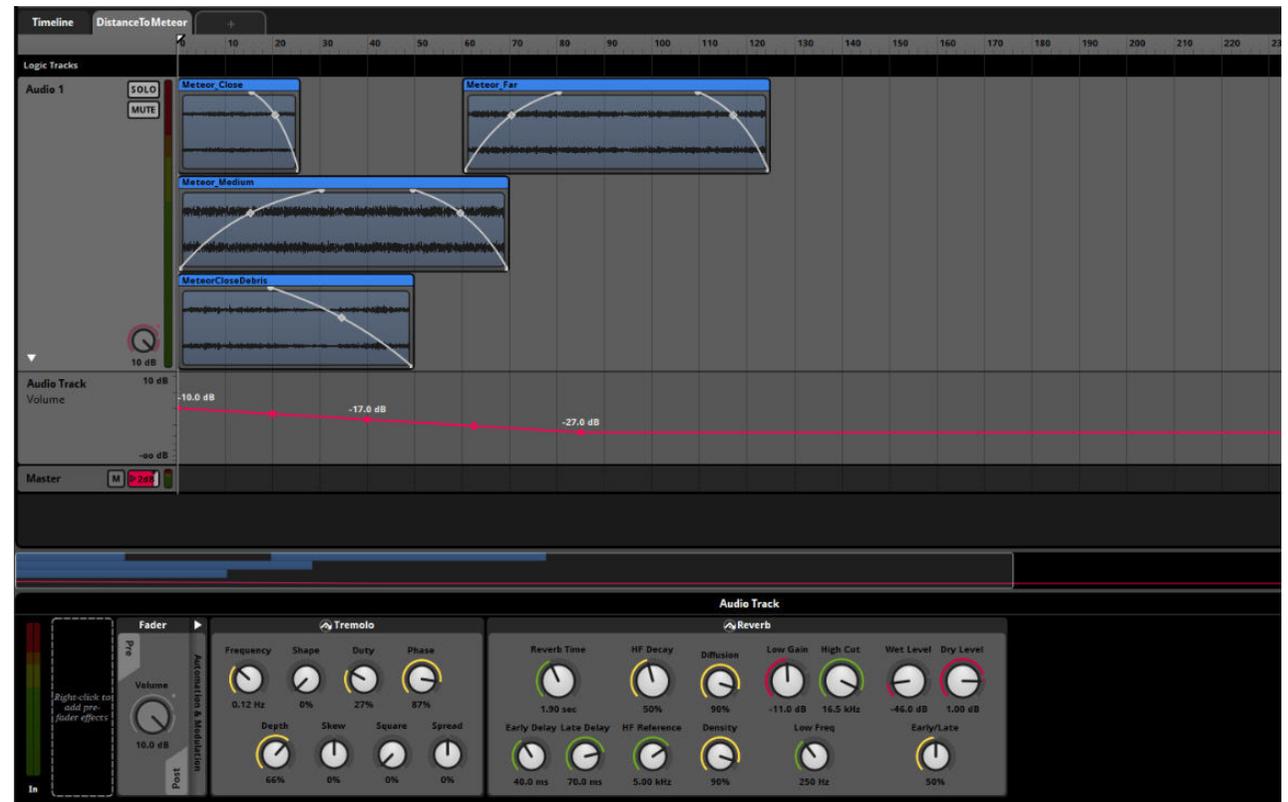
Le son le plus proche est un son de flamme qui rappelle une boule de feu incandescente, c'est un feedback supplémentaire qui sert à alerter le joueur qu'il est très proche de l'étoile. Il se joue en même temps que le camera shake de proximité à l'étoile.

Le feedback le plus long est le son de rumbling, il s'agit d'un bruit de grondement de séisme accompagné de ruptures de métaux. C'est un son qui sert à établir une tension et à marquer un "midway" entre la distance où l'étoile se rapproche plus vite du joueur et celle où elle le tue.

Une track intermédiaire à ces deux dernières composée exclusivement de bruits de métaux compressés et cassants sert à faire la transition entre ces deux feedbacks.

Enfin, lorsque le joueur est assez éloigné il entend un son de vent plus distant et avec de l'écho, à la fois pour lui indiquer que l'étoile est loin de lui et aussi donner plus d'envergure au puit sans fond dans lequel le joueur évolue.

Sur Unity, on récupère par script la valeur de la distance du joueur à l'étoile depuis un autre script, que l'on assigne à une variable qui vient actualiser la parameter sheet FMOD. Cette fonction est placée dans un void Update().



```
_distancetometeor = _distanceScript.distToPlayer;  
_instanceDistance.setParameterByName("DistanceToMeteor", _distancetometeor);
```

Event commenté: sliding

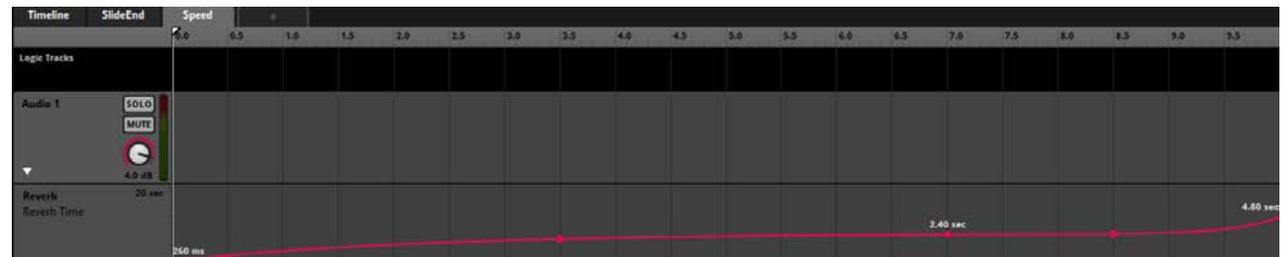
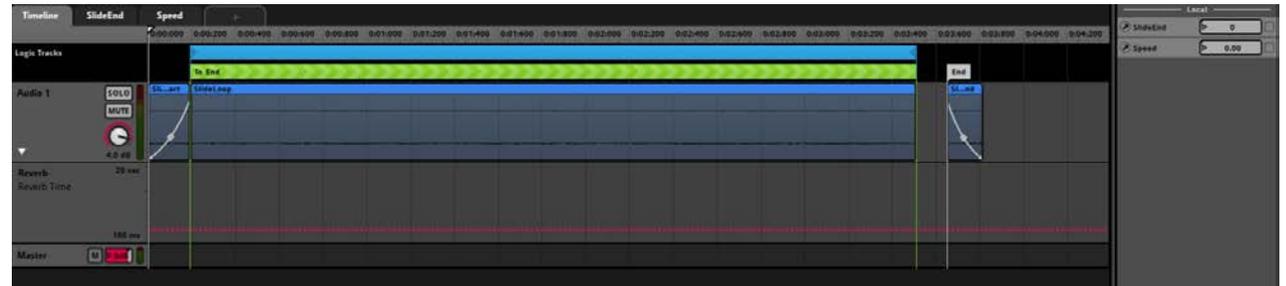
Event Logic

L'évènement de slide est un event assez important car le joueur est amené à effectuer beaucoup de glissades au cours d'une partie.

Le problème de cet event est qu'il n'est pas rare de slide pour une durée très courte pouvant même être de moins d'une seconde. Pour obtenir un son qui marche sur des slides longs ou extrêmement courts on utilise un son séparé en trois. La première partie est celle du début du son, elle s'enchaîne sur une boucle parfaite dans une loop region. Cette loop region est superposée à une transition region qui transitionne sur la fin du son en fonction de la parameter sheet SlideEnd.

Sur Unity, nous utilisons une fonction pour lancer une instance de l'évènement lorsque le joueur rentre dans l'état sliding. Une autre fonction sert à vérifier si le joueur quitte cet état, si son état est différent de l'état sliding le jeu change le paramètre de SlideEnd pour que le son transitionne sur la fin.

Le son est également modifié selon la vitesse du joueur, plus le joueur est rapide plus le feedback sonore aura un reverb important grâce à une automatisation.



```
Message Unity | 0 références
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Slide"))
    {
        _instanceSlide.setParameterByName("SlideEnd", 0);
        _instanceSlide.start();
    }
}
```

```
#region Slide
if (_movementScript._playerState != State.Sliding)
{
    _instanceSlide.setParameterByName("SlideEnd", 1);
}

_speedSlide = (_movementScript.GetComponent<Rigidbody>().velocity.magnitude / _movementScript._maxGroundedSpeed) * 10;
_instanceSlide.setParameterByName("Speed", _speedSlide);
#endregion
```

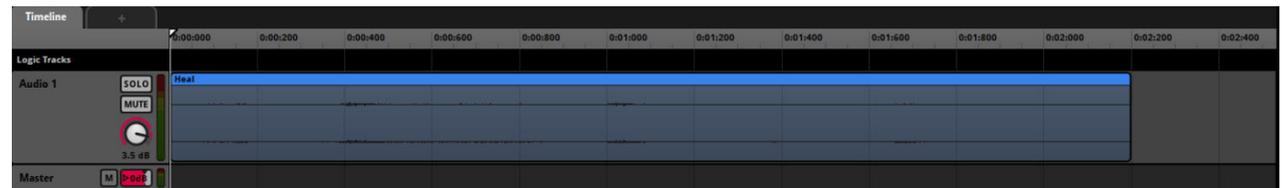
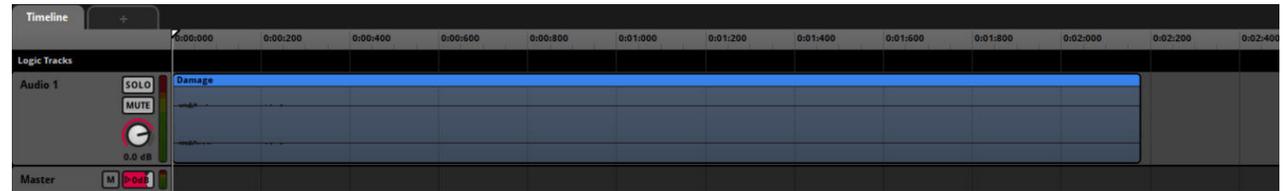
Event commenté: heal and damage

Event Logic

L'évent de dégâts est un bruit distinct de craquement d'os accompagné d'acouphènes, les acouphènes sont un son très reconnaissable pour un feedback important qu'est la prise de dégât.

Le soin est un son de respiration, comme quelqu'un qui reprend son souffle avec un 'woosh' traité avec du reverb.

Les sons sont intégrés en PlayOneShot dans des méthodes appelées directement dans les triggers des changements d'état.

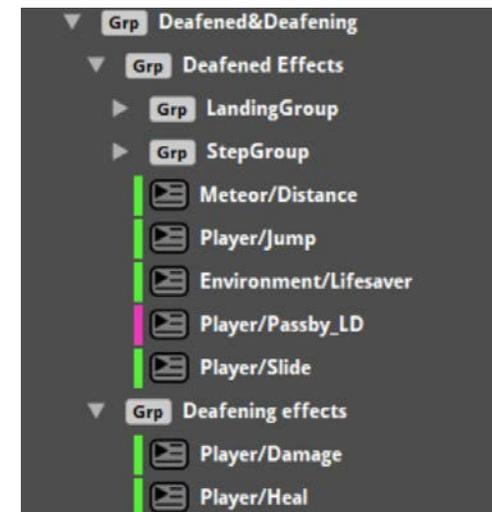


```
if (_hpRefreshTimer >= _timeToRefreshHP)
{
    _redDmgFilter.SetActive(false);
    _hpRefreshTimer = 0f;
    _soundController._playHealSound();
    _dmgStatus = DmgStatus.Healthy;
}
```

```
1 référence
public void _playDamageSound()
{
    FMODUnity.RuntimeManager.PlayOneShot("event:/Player/Damage");
}

1 référence
public void _playHealSound()
{
    FMODUnity.RuntimeManager.PlayOneShot("event:/Player/Heal");
}
```

```
case DmgStatus.Healthy:
    break;
case DmgStatus.Damaged:
    if (_airborneTime > _timeToDie || !_noTimeToDie)
    {
        if (_dmgStatus == DmgStatus.Damaged)
        {
            _whyDie = CauseOfDeath.Fall;
            _dmgStatus = DmgStatus.Dead;
            Debug.Log("airborneTime on death : " + _airborneTime);
        }
        else
        {
            _dmgStatus = DmgStatus.Damaged;
            _soundController._playDamageSound();
            Debug.Log("airborneTime on damage : " + _airborneTime);
        }
    }
    break;
```

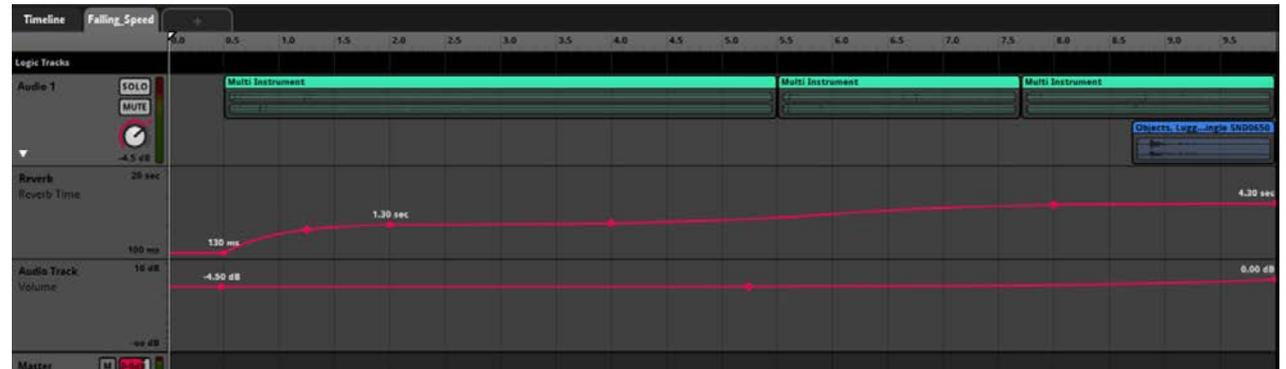


event commenté: Landing

Event Logic

Cet event est constitué de plusieurs sons, ce sont des bruits de pas qui gagnent en nombre et en intensité pour simuler un atterrissage maladroit et hâtif. Au point le plus haut du son nous ajoutons un bruit d'impact de métal pour un feedback supplémentaire.

Pour l'intégration nous jouons le son au moment de l'atterrissage en récupérant la vitesse maximale atteinte lors du saut.



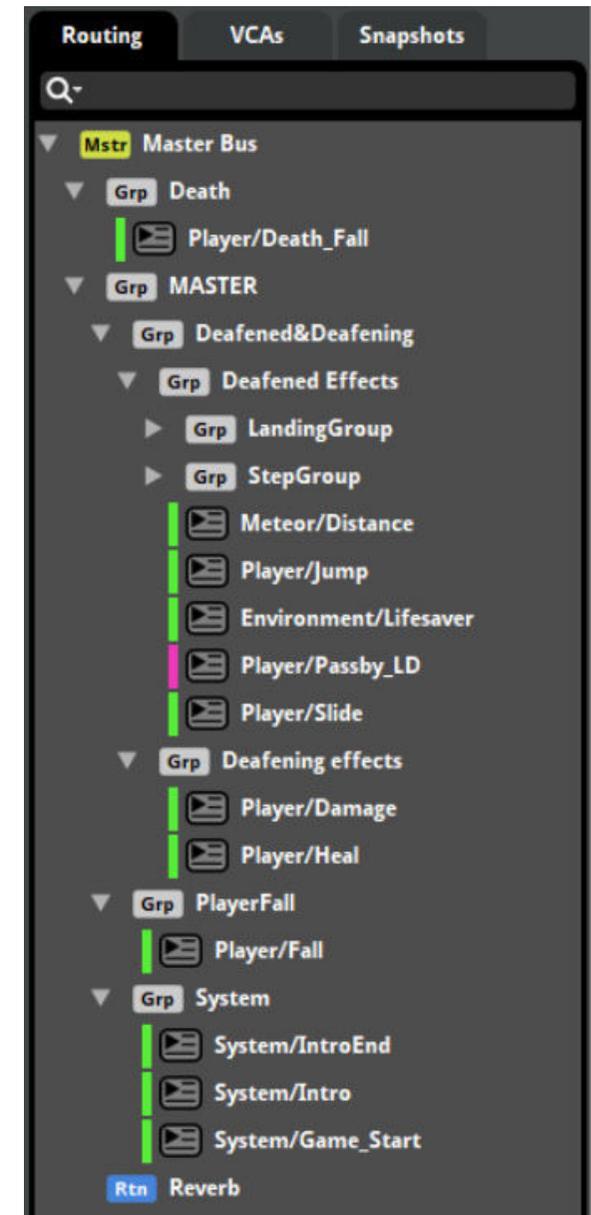
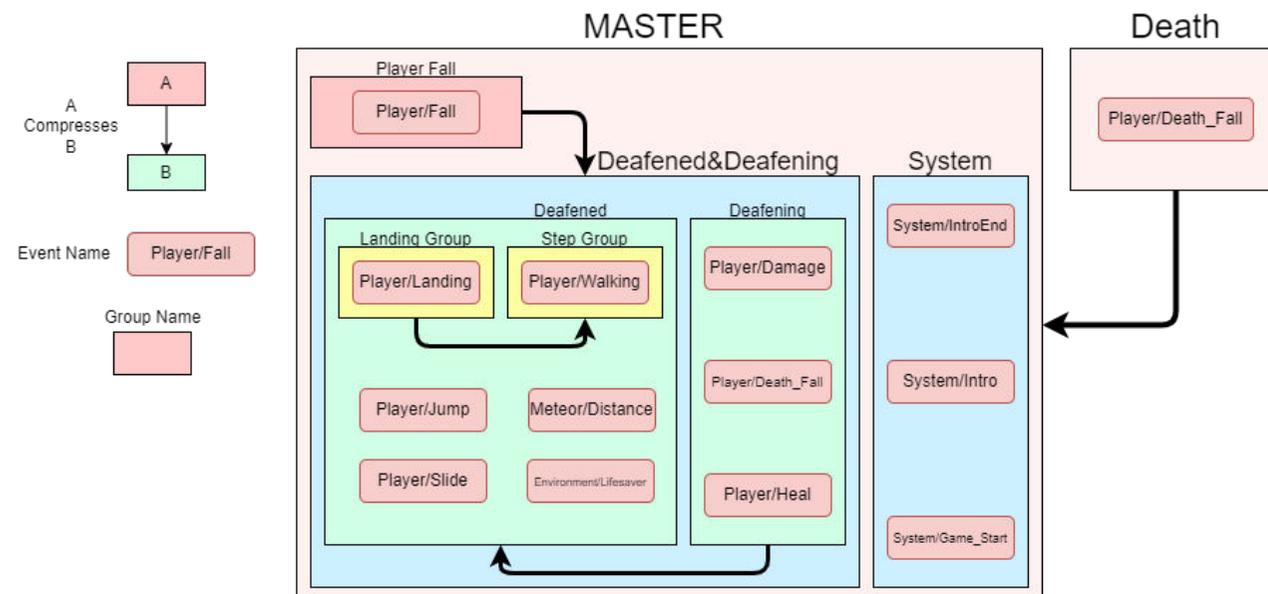
```
1 référence
public void _playLandingSound(float speed)
{
    FMOD.Studio.EventInstance _fallInstance = FMODUnity.RuntimeManager.CreateInstance("event:/Player/Landing");
    _fallInstance.setParameterByName("Falling_Speed", speed);
    _fallInstance.start();
    _fallInstance.release();
    FMODUnity.RuntimeManager.PlayOneShot("event:/Player/Landing");
}
```

```
if (other.CompareTag("Ground") || other.CompareTag("Slide"))
{
    if (_airborneTime > _airborneTimeSoundTrigger && other.transform.CompareTag("Ground"))
    {
        _soundController._playLandingSound(-(_topSpeedThisJump / _maxFallSpeed) * 10);
        //Debug.Log("play landing sound" + _airborneTime);
    }
    _airborneTime = 0;
    _topSpeedThisJump = 0;
    _deathTimer = 0;
}
```

Audio Compressing

Le compressing audio ou audio ducking est une méthode utilisée sur plusieurs sons, par le biais de sidechains et d'effets Compressor dans le mixer de FMOD il est possible de limiter l'output de certains sons lorsqu'un son appartenant à un groupe de sons définis est joué.

Par exemple, le son de chute comprime les autres sons liés au gameplay pour renforcer le sentiment de chute et du vent qui nous assourdit durant cette dernière.



COMPTES RENDUS

RETROSPECTIVES

Outils employés

Difficultés notables

Retours de chaque membre sur le projet

Outils employés



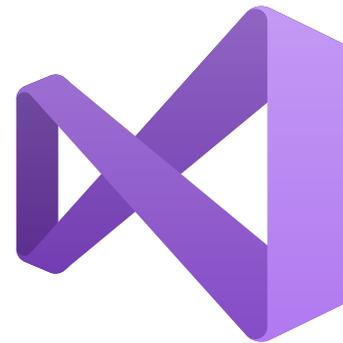
 **Trello**



REAPER
Digital Audio Workstation

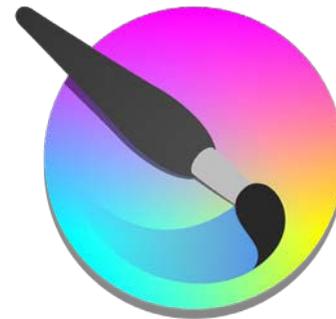


git




Soundly

fmod[®]



Difficultés notables

Les challenges de développement sont principalement provenus du besoin d'établir des processus de production spécifiques qui nous permettaient de remplir les nécessités du projet.

Par exemple, nous avons du entreprendre un processus de création de 3D par pattern plutôt que par bloc de LD, ce qui va à l'encontre des logiques de production réutilisable. Cependant, cela était nécessaire pour donner à chaque structure sa personnalité, et une véritable lisibilité.

Egalement, l'intégration d'un grand nombre de features liées au character controller ont été un challenge à cause des inconsistences de collisions calculées par le moteur physique de Unity, à grande vitesse. Le personnage passe encore à travers certains mesh colliders non convex.

Afterword des développeurs: Marcel

The Pit est un projet dont le concept narratif me tient à cœur. Je suis très heureux de l'avoir vu réalisé avec tant d'énergie de la part de mes collègues, et suis principalement satisfait par le rendu que nous avons accomplis.

Je dis principalement uniquement car le potentiel itératif du jeu, et l'attachement que j'ai développé pour lui me donnent envie de continuer à le retravailler jusqu'à en faire un produit commercial, pitchable.

Le personnage du nettoyeur de puits, et l'univers dans lequel il évolue ont pris vie grâce à un travail de groupe dynamique où chaque membre a réellement porté une part essentielle de la charge.

Je souhaite une bonne continuation à mes collègues de ce projet, et au projet lui-même : peu importe quelle forme il prendra si on revient se pencher dessus pour le retravailler.

Afterword des développeurs: Mathieu

Travailler sur ce projet a été une expérience enrichissante, j'ai appris beaucoup de nouvelles manières de faire sur Unity et ai pu affiner mes capacités par rapport à l'implémentation FMOD.

J'ai énormément aimé travailler au sein de cette équipe, nous avons pu mettre en place une bonne ambiance, couplée à une organisation bien ancrée dès le début du projet, nous n'avons jamais eu de grand flou sur ce que nous devons faire.

Nous avons réussi à garder un cap que nous nous étions fixé le 1er jour de ce projet et malgré le fait que nous n'avons pas exactement atteint toutes nos attentes (qui étaient clairement irréalisables avec le temps qu'il nous était accordé), je suis extrêmement satisfait et fier du résultat.

Ce jeu est clairement le plus aboutit de tous ceux sur lesquels j'ai eu l'occasion de travailler et cela me fait chaud au coeur de savoir que j'ai participé activement et avec un rôle majeur à son développement et, je l'espère, son succès auprès de nos collègues et professeurs.

Pour finir, j'aimerais remercier mon groupe ainsi que tous les professeurs qui nous ont suivi (de près ou de loin) sur ce projet car leur aide a été précieuse afin d'atteindre le résultat que nous pouvons aujourd'hui présenter.

Afterword des développeurs: Nicolas

The Pit a été pour moi un projet d'une ampleur que je n'avais jamais eu l'occasion de produire auparavant. Et ce fût avant tout une expérience positive et enrichissante.

En effet, travailler avec une équipe à l'écoute et avec une communication quasi irréprochable nous a permis d'avancer sur ce projet bien plus vite que je ne l'aurais imaginé.

Bien que la 3D soit, pour moi, un des domaines dans lesquels je me sens le plus à l'aise, ce projet a su repousser mes limites, que ce soit en terme de créativité ou de technique.

Chaque membre a pu mettre sa patte graphique et je tenais à y participer aussi.

Je remercie énormément mon équipe ainsi que toutes les personnes qui ont pu m'aider lors de la réalisation de ce projet qui me tient à coeur.

Afterword des développeurs: Rémi

Travailler dans ce groupe durant ce semestre fût un vrai plaisir, l'ambiance dans l'équipe est restée très bonne tout du long et j'ai trouvé que pu maintenir un bon niveau d'équilibre entre le professionnel et l'amical durant ce projet.

J'ai trouvé l'expérience particulièrement enrichissante car elle m'a permis de découvrir dans une plus grande ampleur un domaine que je n'affectionnais pas spécialement, la programmation, et d'en approfondir un autre avec lequel j'avais déjà une certaine affinité, le sound design. Travailler avec une équipe aussi réactive et passionnée m'a vraiment plu et je me retrouve satisfait de notre travail. Même si notre vision a changée depuis les premiers prototypes je trouve personnellement que nous avons su faire les bons choix, tant en terme de scope quand terme de choix de design et je suis satisfait de la version du jeu sur laquelle nous terminons ce semestre.

Ce projet fût rafraîchissant car terminer sur un projet que l'on trouve amusant à jouer et à créer, personnellement ce fût le projet sur lequel j'ai pris le plus de plaisir à travailler.

Merci.

