

SOMMAIRE

INTRODUCTION

•4

D.A. VISUELLE

·106

UX ·174

Equipe

FICHE D'IDENTITÉ

INTENTIONS

GAME DESIGN •8

Références

TENSION LUDIQUE

GAME SYSTEM

MOTIVATION DU JOUEUR

Level Design **•**84

LEVEL DESIGN MACRO

ENEMY DESIGN

QUEST DESIGN

EXPLORATION

INTENTIONS

RECHERCHES INITIALES

Environnement

ENNEMIS

AVATAR

INTERFACES

Logo

D.A. Sonore **•**162

SOUND DESIGN

Composition

EVENT LIST

Prise de son

ONBOARDING

TESTS UTILISATEUR

•178 TECHNICAL

ENEMY SYSTEM

GRID SYSTEM

CSV Tool

Console Tool

COMPONENT BUILDING TOOL

Sprite Tool

INTRODUCTION

Cluster 71 est un projet réalisé dans le cadre de notre 3ème année de Game Design à l'ICAN, avec comme thème la collection "La Ville qui n'existait pas" de Grégory Chatonsky.

Cette collection est composée d'images mélangeant photographies d'archives et actuelles de la ville du Havre, avant et après sa destruction, respectivement en noir et blanc, et en couleurs. S'ajoutent des formes abstraites violettes générées par IA. Nous avons interprété ces oeuvres comme du "collage". De là a émergé l'idée de coller des éléments sur un avatar pour modifier ses capacités. C'était le point de départ de notre réflexion design, et la mécanique principale de notre jeu.

Au cours du premier semestre, nous

avons élaboré un noyau jouet, un "toy", autour de cette interprétation de la thématique. Nous entendons par toy un ensemble de mécaniques que le joueur peut librement manipuler sans condition de victoire ou de défaite.

Ensuite, le noyau jouet validé, nous l'avons transformé en jeu, l'orientant vers un jeu de livraison et d'exploration.



La ville qui n'existe pas (1895-1944) - Gregory Chatonsky

EQUIPE

MATTHIEU ALLE

- · Lead Game Designer
- Art Director
- Programmer

ERICK MADEC

- Producer
- Lead Sound Designer

LÉO HARDOUINEAU

- Game Designer
- Level Designer
- Environment Artist

ANTONIN GAY

Systems Programmer

MATHIS DOS

Music Composer

Sound Designer

MARIANA GUARNIZO

ELIAZ LEBERT

- Programmer
- Enemy Designer

MARIO NICOLETTI

- Game Artist
- Game Designer

FICHE D'IDENTITÉ

Pitch : Naviguez entre différentes régions et assemblez votre vaisseau.

Type de jeu : Aventure en vue top

Support : Windows

Cible : Joueurs expérimentés et curieux

Univers: Monde nuageux et mystérieux, structures en ruines à la dérive

Game Concept: Dans Cluster 71, le joueur contrôle un vaisseau dans un espace flottant qui permet des sensations de transfert d'inertie et de souplesse. Il est possible de s'approprier le controller en accrochant des "components" sur le vaisseau qui viennent modifier ses propriétés. Le joueur explore un monde nuageux parmi des ruines en suspension où il devra effectuer des livraisons. Cependant des créatures rôdent et pourraient entraver ses voyages. Dans ce monde inquiétant et mystérieux, le joueur va trouver du confort dans le vaisseau qu'il a construit lui-même.



Concept Art d'ambiance

INTENTIONS

- · Appropriation de l'avatar
- Mouvement souple et profond
- · Optimisation de l'espace
- · Monde hostile et intrigant

Nous voulions fournir au joueur la sensation de s'approprier son vaisseau au travers de l'assemblage, ce qui participe à un confort dans l'avatar qui s'oppose à l'hostilité et la froideur dans le monde.

Aussi, nous voulions solliciter la capacité du joueur à optimiser l'espace disponible, le poussant à identifier des patterns dans son vaisseau et à réfléchir. Le temps passé à optimiser l'avatar renforce l'attachement mécanique du joueur pour celui-ci.

Le mouvement se devait d'être agréable et souple, en offrant une grande possibilité de maîtrise.

Nous voulions que l'ambiance de Cluster 71 soit prenante et donne envie au joueur d'explorer, d'aller voir ce qu'il se trouve plus loin, tout en ayant une dimension menaçante qui le mette sur ses gardes. Cependant, si le monde est hostile et froid, il est possible de le surmonter et d'aller au delà.

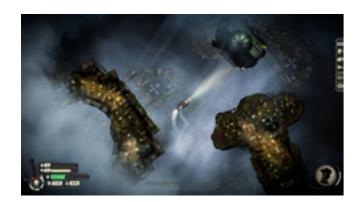
Image du jeu

GAME DESIGN

Références

MOUVEMENT

Sunless Skies a été une référence importante en début de pré-production. Ce jeu nous a aidé à établir nos 3Cs. Par la suite, nous nous somme éloignés de la proposition de Sunless Skies vers quelque chose correspondant mieux à nos intentions. Notamment un mouvement plus rapide et dynamique, et une absence de mécanique de tir.

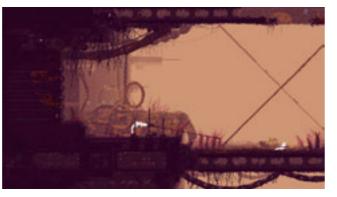


Sunless Skies - Failbetter Games, 2019

Les jeux de course anti-gravité
Wipeout et BallisticNG nous ont servi de
références de challenge pour un controller
avec de l'inertie. Ces jeux sont presque
exclusivement basés sur la prise de virage.
Sur l'évaluation de la trajectoire à prendre,
sur le dosage de la quantité de rotation
que le joueur applique à son avatar, du
moment où il entame sa rotation et de
la durée pendant laquelle il tourne. Ce
sont également des références pour le
traitement de la sensation phsyique du
déplacement de la masse de l'avatar.



Wipeout HD - Studio Liverpool, 2008



Enfin, nous pourrions également citer Rain World que nous avons exploité comme référence de possiblité d'apprentissage dans le controller. Même si notre jeu est très différent, nous reprenons le principe d'un avatar qui paraît étrange et approximatif à contrôler en début de partie, et que le joueur peut finalement maîtriser et utiliser de manière efficace. Nous avons également tenté de créer, comme dans Rain World, un controller qui présente de nombreuses subtilités et mécaniques que le joueur peut découvrir pour obtenir un avantage sur son environnement. Cette dynamique d'apprentissage nous a semblé particulièrement pertinente vis-à-vis de notre volonté de créer sur le long terme un confort et un attachement pour l'avatar dans un monde hostile et de permettre au joueur de finalement surmonter cette hostilité.

ASSEMBLAGE

Besiege a servi de point de départ pour nos réflexions sur la manière dont « coller des éléments sur l'avatar » allait se manifester mécaniquement. En effet, il permet d'alterner avec un input entre deux phases de jeu, une où le joueur contrôle une machine et une où il place des pièces sur cette machine pour déterminer ses capacités. Cependant, dans Cluster 71, contrairement à Besiege, passer en mode assemblage ne réinitialise pas l'état du niveau. Le jeu est également une référence intéressante de par son utilisation d'une forme de grille pour faciliter le placement des pièces.

Nous avons également choisi d'utiliser une grille pour faciliter le placement de nos components, mais en 2D contrairement à Besiege. De plus, cette grille n'est pas



Besiege - Spiderling Studios, 2015



seulement un outil pour le joueur, nous en avons fait une phase de jeu à part entière avec son propre gameplay: un gameplay d'optimisation spatiale inspiré des inventaires de Dredge ou Resident Evil. Le joueur a un espace fini représenté par une grille dans lequel il peut placer des objets qui occupent un certain nombre de cases. Il va chercher à agencer les objets dans cette grille de la manière la plus compacte possible. Dredge est particulièrement intéressant comme référence car il n'utilise pas une grille purement rectangulaire mais une forme plus irrégulière qui invite le joueur à identifier les objets et la disposition qui permettent de combler précisément les espaces libres.

Enfin, pour créer certains challenges avancés dans notre gameplay d'assemblage nous nous sommes intéressés aux RPGs

et jeux de stratégie qui utilisent des grilles dans leur système de combat. Ces jeux nuancent généralement la qualité du placement d'une unité en fonction de la manière dont son effet est appliqué dans l'espace, et en fonction de l'effet que les cases ont sur les unités. Dans Pit People par exemple, le Mushroom inflige des dégâts à toutes les unités adjacente, ce qui encourage le joueur à l'entourer d'autant d'ennemis possibles, tout en gardant ses propres unités loin de lui. Dans Cluster 71, nous avons créé un type de component qui renforce les components adjacents d'un type donné. Ce qui fait que le joueur cherche non seulement à les faire rentrer dans la grille, mais aussi à les placer à un endroit précis, et à penser tout son agencement pour optimiser son utilisation de ce component.



Pit People - The Behemoth, 2016

EXPLORATION

La conversion de notre jouet en jeu s'est rapidement orientée vers un système de livraison avec une notion d'exploration, pour lesquels nous nous sommes principalement inspirés des systèmes de navigation et de lecture de l'environnement de Tunic et Hollow Knight.

Tunic nous a servi de référence pour la variation du niveau de précision de la map, et la possibilité d'obtenir des informations plus précises.



TUNIC - Andrew Shouldice, 2022

Hollow Knight est une référence pour la structure du monde. Le découpage en régions, la manière dont les régions peuvent être découvertes dans différents ordres, mais les régions plus éloignées sont plus dangereuses. Ce jeu nous a également servi de référence pour les secrets, la manière dont les passages découverts changent la menière dont le joueur aborde le monde, et pour le teasing des secrets à trouver.



Hollow Knight - Team Cherry, 2017

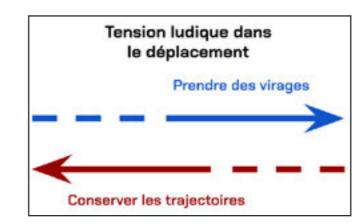
TENSION LUDIQUE

Notre noyau jouet se décompose en deux core gameplays : le déplacement et l'assemblage.

Les diagrammes ci-dessous affichent les tendances joueur en bleu et les tendances système en rouge.

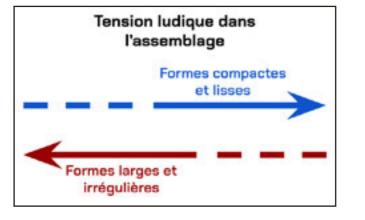
DÉPLACEMENT

Avec sa forte inertie, l'absence de gravité et sa lente rotation, l'avatar tend à conserver sa trajectoire. Alors que le joueur va chercher modifier sa trajectoire afin d'éviter des obstacles et des ennemis et d'atteindre des endroits précis.



ASSEMBLAGE

Les components que le joueur peut assembler ont été conçus pour former des excroissances, des trous et des irrégularités quand ils sont placés. Je joueur a cependant intérêt à créer un vaisseau compacte, et nous avons observé qu'il essayera souvent d'obtenir un résultat réglier, lisse, et symétrique si possible.



GAME SYSTEM 3Cs

CHARACTER

Le joueur contrôle un avatar dans un univers flottant sans gravité. L'avatar peut avancer, tourner, et surtout s'assembler avec différents components dans son environnement. Le collider, le poids, et les capacités des ces components s'ajoutent ainsi à ceux du player.

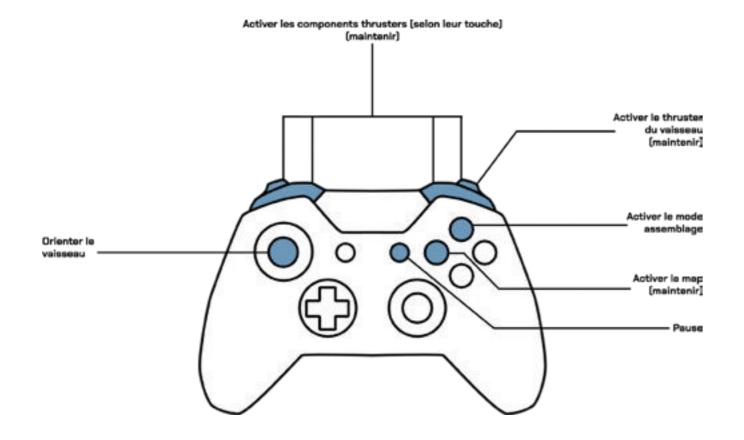
Cela provoque des changements dans les metrics du player, modulant les collisions qu'il va prendre à cause de sa taille différente, sa vitesse de déplacement ainsi que sa vitesse de rotation.

Il peut également se réassembler et désassembler, et perdre des components si ceux-ci subissent trop de dégâts.

10 - Cluster 71

Game Design

Controls (DÉPLACEMENT)



CONTROLS [ASSEMBLAGE]

Mode Outside



Permet au joueur d'ajouter des components extérieurs à son vaisseau.

Mode Inside



Permet au joueur de déplacer ou retirer les components déjà présents sur son vaisseau.

12 - Cluster 71

Game Design _

CAMERA

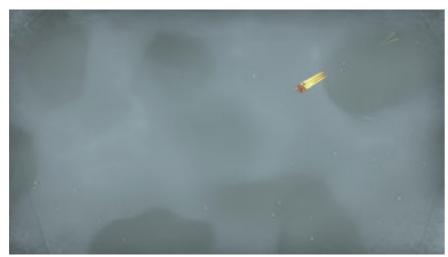
Le jeu est en vue du dessus orthographique et la caméra ne peut pas tourner.

La caméra suit un point qui se situe sur le vaisseau lorsque ce dernier est immobile.

Lorsqu'il se déplace, ce point se décale dans la direction de son mouvement. La distance de ce décalage dépend de la vitesse du joueur. La caméra dézoome également en fonction de la vitesse du joueur. Ces deux comportements permettent de renforcer la sensation de vitesse tout en donnant au joueur plus d'informations sur ce qu'il se trouve devant lui, lui permettant de réagir aux obstacles même à haute vitesse.



Vaisseau à l'arrêt



Vaisseau en mouvement

La caméra dézoome également en fonction de la taille du vaisseau du joueur. Aidant aussi à l'évitement des obstacles avec un vaisseau dont l'intertie est accrue. (voir comparaison ci-dessous)

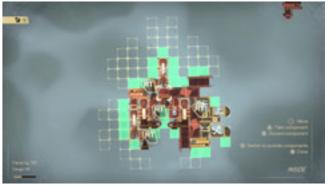




Différence de taille de caméra selon la taille du vaisseau

Enfin, lorsque le mode assemblage est activé. la caméra zoome et se recentre sur le joueur pour permettre une vue plus précise sur la composition du vaisseau. La caméra prend également la taille exacte de la zone dans laquelle le joueur peut récupérer les components présents dans le monde. Rendant clair lesquels sont accessibles et lesquels ne le sont pas. Quand le joueur active le mode « inside » de la grille (c'est à dire quand il déplace les components déjà ramassés au lieu d'ajouter des components extérieurs), la caméra zoome encore un peu plus pour traduire le changement d'état et aider le joueur à comprendre que dans ce mode il n'influe que sur les components déjà présents sur son vaisseau.





Différence entre mode Outside et Inside

14 - Cluster 71

5

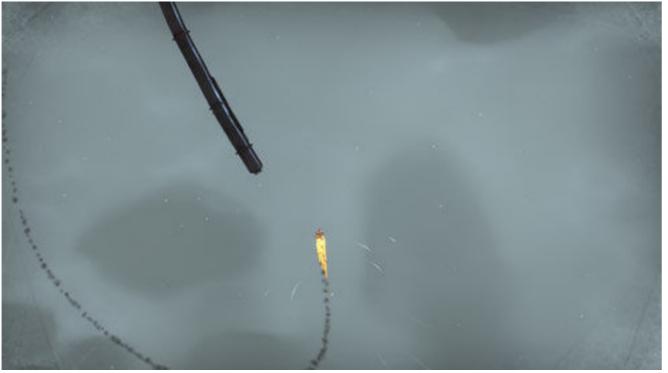
Dans notre level design, nous utilisons des zones qui appliquent un modificateur sur la taille de la caméra quand le joueur entre dedans. Elles permettent un travail de mise en scène pour renforcer l'impression de vastitude de certains espaces et l'étroitesse de certains autres. Ci-dessous, comparaison entre une zone vaste et une zone étroite.

Un effet similaire est appliqué quand le joueur est à proximité d'un de nos ennemis, le « Worm ». Mais au lieu de multiplier la taille de la caméra, cette zone-ci la fixe à une valeur précise. Ce qui permet d'assurer que la caméra ne soit ni trop serrée, pour que le joueur puisse lire les comportements de l'ennemi, ni trop large, pour que le Worm puisse paraître réellement menaçant. Le fait que la taille de la caméra soit fixe et cesse d'évoluer avec les mouvements du joueur permet aussi de créer une atmosphère oppressante dans ces moments. Enfin, la zone de l'effet étant relativement large, le joueur peut identifier le changement de comportement de la caméra et savoir qu'un Worm est à proximité avant d'être repéré.





Différence de taille de caméra selon la vastitude ou l'étroitesse de l'espace



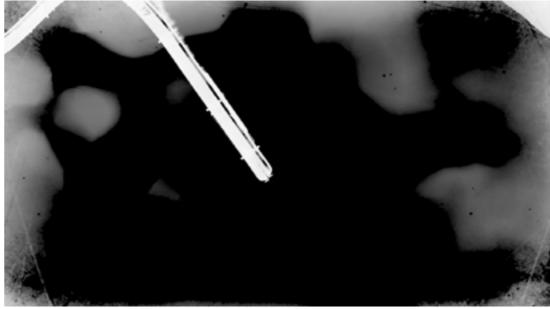
Un Worm qui a vu le joueur, la caméra est bloquée à une taille précise

Lorsque le joueur est détruit et l'écran de game over apparaît, nous effectuons on fort dézoom de caméra pour ajouter une dimension spéctaculaire à la situation tout en explicitant le changement d'état.

Tous les décalages et changements de taille de la caméra sont effectués avec un smoothing, selon une courbe logarithmique.

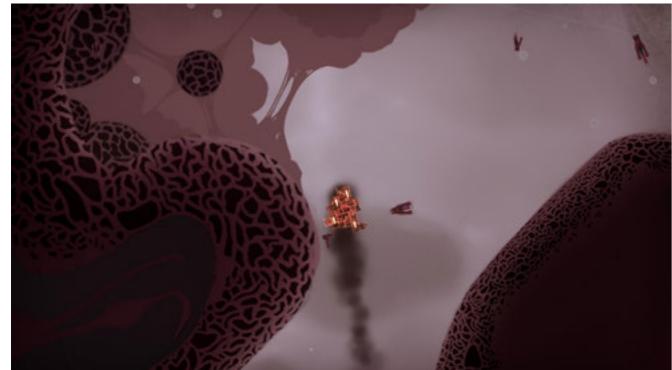
Frame de morsure du worm





En outre, plusieurs effets sont appliqués sur l'image pour renforcer le feeling de diverses situations. D'abord, quand le Worm mord le joueur, l'image est figée pendant 0.4 secondes, passe en noir et blanc très contrasté avec des couleurs inversées pour rendre son attaque plus choquante et violente.

Chittering Construct En haut - The Maw Remnants

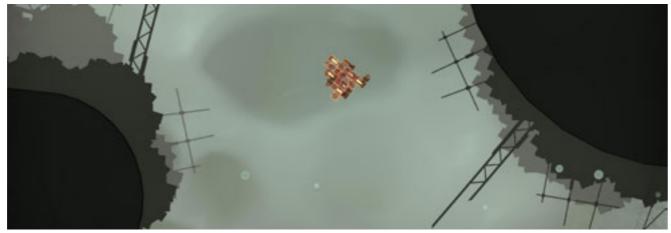






Des filtres de couleur sont aussi appliqués lorsque le joueur se trouve dans les différentes régions, ce qui participe à leur donner des identités distinctes.

De plus, nous utilisons des effets de screenshake pour accentuer divers chocs, impacts et explosions.



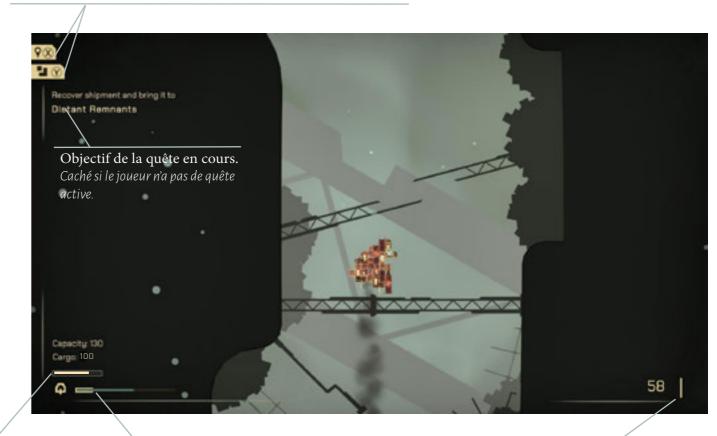
Old Herringen

Enfin. un HUD est affiché par dessus le jeu pour transmettre au joueur les informations dont il pourrait avoir besoin. Cet affichage est décomposé en plusieurs éléments d'interface qui sont affichés uniquement quand ils sont nécessaires, et cachés autrement. Ainsi, l'image n'est pas inutilement surchargée. Par exemple, la barre de vie du joueur n'est affichée que si cette donnée a été modifiée pendant les 15 dernières secondes ou si ses points de vie sont sous un seuil critique. Ci-contre, une capture d'écran du HUD complet.

Barre de capacité et cargo Cachée si la capacité du joueur est à O. La barre beige représente le cargo transporté par le joueur. La largeur du contenant gris représente sa capacité max (dépendante de ses components)

Indications d'input et d'état du mode assemblage et de la carte. Les rubans s'étendent vers la droite quand l'affichage correspondant est

les rubans s'étendent vers la droite quand l'affichage correspondant est actif. Celui de la carte se retire complètement quand la grille est active : il n'est pas possible d'ouvrir la carte à ce moment là.



Barre de vie et bouclier

La barre jaune représente le bouclier (qui se régénère avec le temps) et la barre verte les points de vie (qui ne se régénèrent pas d'eux mêmes). Il s'agit de la barre de vie vaisseau de base, et non celle de l'ensemble des components.

Vitesse actuelle du joueur.

Caché si le joueur ne se déplace pas. La taille du texte et la hauteur de la barre à droite augmentent en fonction de la vitesse.

HUD du mode Assemblage

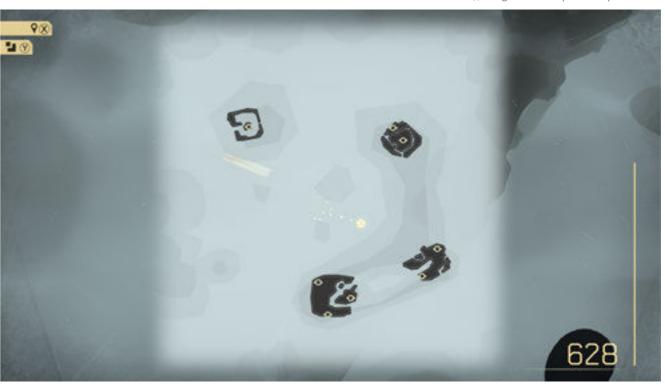
En mode assemblage, le compteur de vitesse est remplacé par une liste des actions possibles dans le mode actuel ainsi que par une indication du mode actuel (Outside sur cet exemple). Une grille apparaît sous le vaisseau, représentant l'espace disponible et les cases valides sont affichées en vert. Les cases qui rendent le placement du component sélectionné impossible sont en surbrillance rouge. De plus, les contrôles de chaque thruster sont affichés pour permettre au joueur de prendre des décisions de placement en fonction de cette information (par exemple orienter différemment un qui serait déclenché avec un autre bouton). Enfin, lors du placement d'un component, un texte apparaît temporairement par dessus pour indiquer l'effet de ce que le joueur vient d'accrocher.





Effet d'assemblage d'un component

Affichage de la map en temps réel



Le joueur peut également maintenir X pour afficher la carte du jeu. Il peut toujours se déplacer pendant ce temps là, nous avons donc rendu la carte légèrement transparente pour qu'il puisse voir ce qu'il fait. Sur la carte figurent la topologie générale du monde du jeu, la position du joueur, sa trajectoire récente, les points de livraison et les objectifs de la quête en cours.

Description des Mécaniques

AVANCER

Maintenir RT

Propulse le joueur dans la direction à laquelle fait face son vaiseau.



Poussée analogique

Maintenir plus ou moins fort LT/RT

Propulse le joueur plus ou moins fort en fonction de la pression exercée sur la touche. La taille et l'opacité de la flamme des thrusters dépendent de la pression exercée. Cette mécanique concerne le vaisseau de base, et les thrusters associés à RT et LT.



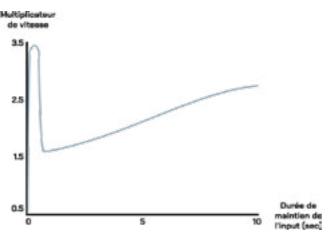
Accélération

Maintenir LB/RB/LT/RT

La puissance de propulsion d'un thruster évolue au cours de sa durée d'utilisation.

Il y a un fort pic au démarrage puis une lente montée. Il faut entre 10 et 20 secondes pour atteindre la valeur d'accélération max selon la masse du vaisseau.

La flamme des thrusters devient plus large et plus blanche plus le multiplicateur de poussée de l'accélération est élevé.



BOOST DE DÉMARRAGE

Appuyer sur LT/RT

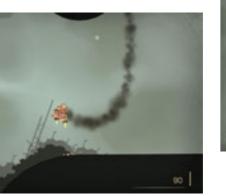
Lors du démarrage, un léger multiplicateur est appliqué sur le pic d'accélération initial en fonction de la vitesse à laquelle le joueur atteint la valeur max de l'input analogique.



TOURNER

Incliner le Stick Gauche

La rotation est une transition entre la direction actuelle de l'avatar, et la direction indiquée par le joystick. Cette rotation s'effectue plus ou moins rapidement selon le poids du vaisseau. Plus la différence entre l'angle actuel du vaisseau et l'angle de l'input joueur est grande, plus le vaiseau tourne vite.



BOOST DE VIRAGE

Maintenir LB/RB/LT/RT après avoir tourné avec le Stick Gauche

En sortie de virage, multiplie la puissance de propulsion du vaisseau pendant 0.35 sec en fonction de l'angle et de la durée du virage effectué. Jusqu'à un poussée x4. La flamme s'élargit pour traduire cet effet.



BOOST LATÉRAL

Activer des thrusters latéraux

Un thruster qui fait face à une direction décalée d'un angle compris entre 70 et 170° par rapport à la direction de la vélocité du vaisseau a sa poussée multipliée par 3.



WALL PUSH

Maintenir LB/RB/LT/RT dans la direction opposée à un obstacle.

S'il y a un obstacle à moins de 30 unités derrière un thruster, sa poussée est multipliée jusqu'à 2.5 fois. Plus l'obstacle est proche, plus ce boost est élevé. Des particules apparaissent sur le mur pour traduire cet effet, leur vitesse et quantité dépendent de la puissance de celui-ci.

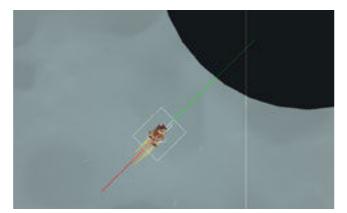


BOOST LIMITER

Accélérer vers un mur

S'il y a un mur à moins de 55 unités en face du joueur, les boosts de poussée situationnels décrit plus tôt sont réduits jusqu'à 20% de leur effet normal. Plus le mur est proche plus la réduction est forte.

Cette mécanique est une sécurité pour éviter que le joueur ne se propulse d'un coup dans un mur en activant par accident plusieurs de ces effets dans un espace exigu.



POINTS DE VIE

/

Le vaisseau de base (le « Core ») et les components ont chacuns leur ressource Points de Vie indépendantes. Si les PV d'un component tombent à 0, il est détruit. Si ceux du Core tombent à 0, tout l'assemblage est détruit et l'écran de game over apparaît. L'entièreté des points de vie du vaisseau et de ses components est régénérée une fois arrivé dans une ville.



Bouclier

Le Core et les components ont également une ressource de Bouclier qui subit les dégâts avant les points de vie, et se régénère toutes les 8 secondes d'un montant fixe si le joueur ne subit pas de dégâts. L'entièreté du bouclier du vaisseau et de ses components est régénérée une fois arrivé dans une ville.

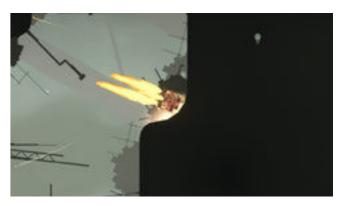
Le ratio PV / Bouclier est de 0.7 / 0.3.



Collision

Rentrer dans un mur avec une vitesse supérieure à un seuil

Rentrer dans un mur fait baisser les points de vie du joueur. Plus le joueur est rapide, et plus la trajectoire du joueur est perpendiculaire au mur, plus les dégâts subis sont élevés. Aussi, le joueur perd de l'accélération lorsqu'il collisionne.



Collision - Mur de swarmers

Rentrer dans un mur de swarmers avec une vitesse supérieure à un seuil

En plus des dégâts de collisions, rentrer dans un mur de swarmers rend les swarmers environnants agressifs. Ils vont chercher à attaquer le joueur. Plus la collision est puissante, plus le rayon de swarmers provoqués est large.



ATTAQUE PAR SWARMERS

Entrer en contact avec un swarmer agressif

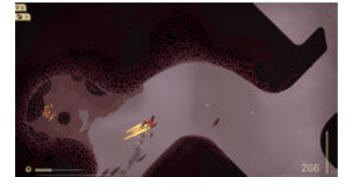
Le swarmer s'attache à un component et lui inflige des dégâts périodiques. Si les PV du component atteignent o, il n'est pas détruit mais détaché. Il ramène ensuite le component jusqu'à son nid. Le component ne peut pas être assemblé tant qu'il est en cours de transport. Plusieurs swarmers peuvent s'attacher au même component. S'ils attaquent le Core, ils n'infligent qu'une instance de dégâts (plus élevés) puis redeviennent passifs.



DÉTACHER DES SWARMERS

Dépasser un palier de vitesse alors qu'un swarmer est accroché

Au-delà d'un palier de vitesse, les swarmers auront une chance de se décoller du joueur. Une fois détachés, ils redeviennent passifs. S'éloigner suffisamment de leur nids ou rentrer dans une ville a le même effet.



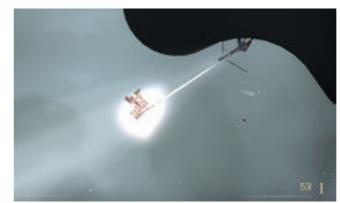
ATTAQUE PAR TOURELLE

Entrer dans la zone de détection d'une tourelle

La tourelle va tirer des projectiles dans la direction du joueur (sans prédiction de mouvement) tant qu'il est à portée.

Si un projectile touche le joueur, il le traverse en infligeant des dégâts à tous les components qu'il touche et le repousse.

La distance pour activer la tourelle est plus courte que la distance à laquelle elle oublie le joueur.

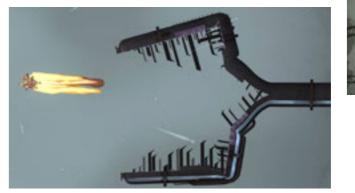


ATTAQUE PAR WORM

Être à portée de morsure du worm

Le worm va ouvrir sa gueule quand il s'approche du joueur. Si le joueur est toujours à portée après une courte durée, il mord.

La morsure inflige de lourds dégâts aux parties du vaisseau dans sa zone d'effet. Il y a un nombre maximum de parties pouvant être affectées par une morsure. Le joueur est violemment projeté et le worm effectue un rapidement mouvement de recul avant de repartir à l'offensive.



RÉVÉLER UN FAUX MUR

Traverser un faux mur avec le player

Le faux mur disparaît au contact avec le joueur, révélant un passage secret.Le faux mur réapparaît quand le joueur s'en éloigne de nouveau suffisamment. Un effet d'onde apparaît autour du joueur quand il révèle un faux mur pour la première fois.



26 - Cluster 71

Game Design _____

Nuages de Bords

Rentrer dans un nuage de bords de map

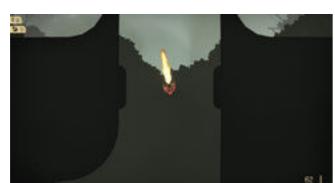
Repousse le joueur hors du nuage, vers le centre de l'espace de jeu. Lorsque le joueur est en eux, une petite spotlight permet de continuer à situer sa position.



ENTRER DANS UNE VILLE

Rentrer dans une ville

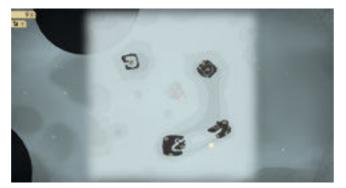
Quand le joueur entre dans la zone du point de livraision, un menu apparaît. Il permet d'accéder aux quêtes marchés de component et de map. Les points de vie des components sont entièrement régénérés. Si la ville en question est la destination de la quête en cours du joueur, la quête est terminée et il obtient ses récompenses. Sortir rouvre la porte par laquelle on peut sortir, de retour vers le jeu. Rentrer dans une ville réinitialise également les ennemis.



OUVRIR LA MAP

Maintenir X

La map s'ouvre par dessus l'écran sans mettre le jeu en pause.



ACTIVER LE MODE ASSEMBLAGE

Appuyer sur Y

Si le mode assemblage n'est pas déjà actif, le jeu se «met en pause», la caméra zoome sur le joueur et une grille d'assemblage apparaît autour, avec les contrôles affichés. S'il y a des components à proximité, la grille passe en mode Outside et le joueur peut les assembler. S'il n'y en a pas, elle passe en mode Inside, permettant au joueur de déplacer et retirer ses components déjà assemblés.



Changer le mode de grille

Appuyer sur Y quand la grille est ouverte

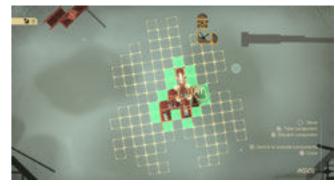
Si la grille est en mode d'assemblage Inside, passe en Outside, et inversement.



DÉPLACER LE CURSEUR

Appuyer sur les touches du D-Pad ou orienter le Stick Gauche en mode Inside

Déplace le curseur de sélection sur la grille.



SÉLECTIONNER COMPONENT

Appuyer sur A en ayant le curseur sur un component installé, en mode Inside

Sélectionne le component. Une fois sélectionné il peut être déplacé.



COMPONENT SUIVANT/PRÉCÉ-DENT

Appuyer sur LB ou RB en mode Outside

Sélectionne le component suivant ou précédent parmi ceux disponibles dans l'environnement.



Déplacer le component sélectionné

Appuyer sur (ou maintenir) une touche du D-Pad ou incliner le Stick Gauche en ayant un component sélectionné

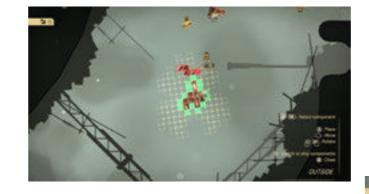
Déplace le component case par case au sein de la grille, dans les limites de celle-ci.



TOURNER UN COMPONENT

Appuyer sur LT ou RL en ayant un component sélectionné

Tourne le component respectivement de -90° et 90° dans le sens horaire, autour de son pivot.



PLACER UN COMPONENT

Appuyer sur A en ayant le component sélectionné sur une position valide

Pose le component à la position validée, ce qui augmente la taille, le poids et modifie les capacité (vitesse de rotation, capacité de cargaison, thruster supplémentaire) du vaisseau.

(Le placement est valide si le component est en contact avec au moins une case adjecente au vaisseau, sans avoir de cases superposées avec ce dernier, ni de case qui déborde hors des limites de la grille).



Appuyer sur X en survolant ou sélectionnant un component (en Inside)

Détache le component et le remet dans l'espace environnant. Tous ses effets sont retirés du joueur.





QUITTER LE MODE ASSEMBLAGE

Appuyer sur B si la grille est activée

La grille se ferme, la caméra dézoome et le temps reprend son cours. Le joueur récupère les contrôles de déplacements.



MOUVEMENT

INTENTIONS

Se déplacer est l'action que le joueur passe le plus de temps à effectuer dans Cluster 71. Il était donc crucial que cette action soit à la fois agréable et intéressante.

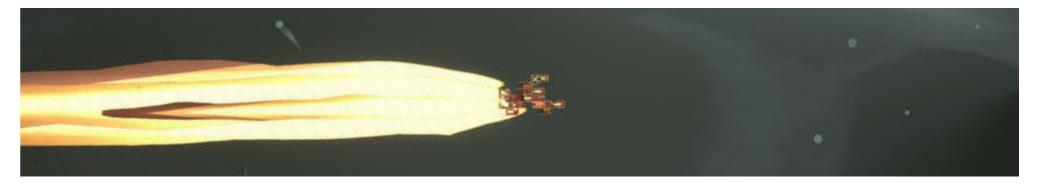
Nous voulions créer un gameplay de mouvement qui transmette la sensation physique de pousser la masse du vaisseau, et de ressentir son inertie. Un défi que nous avons dû relever en tant que designers était de créer ce mouvement glissant et basé sur l'inertie sans pour autant que le gamefeel ne paraisse mou ou qu'il ne donne l'impression de ne pas répondre aux inputs du joueur. Nous voulions que, si le joueur rate un virage, ça ne semble pas injuste,

qu'il ait le sentiment qu'il est possible de le faire mieux, et qu'il finira par y arriver.

Enfin, pour rendre ce gameplay intéressant, il nous a semblé important qu'il présente une profondeur d'apprentissage. Elle prend deux formes : d'abord la familiarisation avec le comportement et les metrics «bas-niveau» de l'avatar, tel que la manière dont il tourne, accélère et décélère. Le fait de sentir instictivement que si on incline le stick gauche à tel angle à tel moment, le vaisseau fera face à telle direction dans tant de temps. Cet apprentissage étant d'ailleurs renouvelé au cours du jeu au fur et à mesure que

le joueur modifie son vaisseau; il peut apprendre la manière dont l'assemblage modifie les propriétés du mouvement. Et d'autre part, nous avons créé un certain nombre de subtilités dans les mécaniques de déplacement qui ouvre la porte à des techiques de mouvement spécifiques que le joueur peut apprendre et maîtriser pour avoir un avantage sur son environnement.

Dans cette partie nous allons nous intéresser à la manière dont nous avons répondu à toutes ces questions.



32 - Cluster 71

Game Design

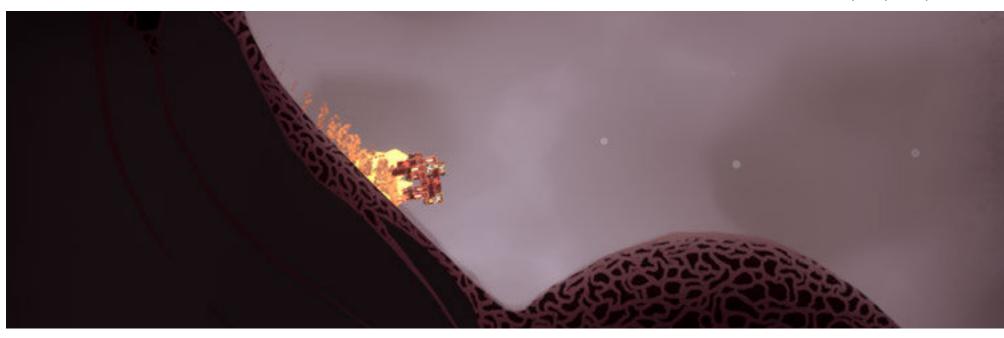
Design & Itérations

CONTROLLER & ROTATION

Dans une de nos itérations les plus anciennes, le jeu était contrôlé au clavier avec ZQSD. Z et S contrôlaient la poussée en avant et en arrière, et Q et D faisaient tourner l'avatar dans le sens anti-horaire et horaire respectivement. De plus, la vitesse de rotation était linéaire : tant que le joueur maintient l'input, l'avatar tourne de X degrés par seconde.

Ce système ne permettait pas la souplesse et l'immédiateté que nous recherchions. Nous avons notamment remarqué que ce système encourageait une approche séquentielle du déplacement, les joueurs tournaient d'abord, puis accéléraient, et il était difficile de gérer les deux en même temps avec 3 doigts. La rotation avait également un feeling rigide, et devenait assez perturbant si l'orientation de l'avatar changeait souvent.

Nous avons donc décidé de changer de controller et d'utiliser la manette plutôt que le clavier. Cela donne plus de précision dans les inputs et permet de mieux décoreller la rotation de la propulsion. De cette façon, le joueur peut beacoup plus facilement gérer les deux en même temps. Nous avons également changé le fonctionnement de la rotation en faveur d'un système où l'avatar va tourner pour faire face à la direction dans laquelle le Stick Gauche est incliné. Ainsi, quelle que soit l'orientation de l'avatar, il ira dans la direction indiquée par le joueur.



De plus, la transition de la rotation actuelle de l'avatar à la cible indiquée par le joueur n'est plus linéaire mais logarithmique et dépend de la différence entre les deux rotations. Plus la différence est grande, plus l'avatar tournera vite, et plus il se rapproche de la cible, plus la rotation est lente. Et la rotation est dans l'ensemble plus rapide. De cette manière, l'avatar réagit beaucoup plus rapidement aux commandes du joueur et le ressenti est plus dynamique. Ce système de rotation incite également un comportement très intéressant : vu que l'avatar tourne plus vite pour une rotation de plus grande amplitude, le joueur est encouragé à input un virage plus grand que ce dont il a réellement besoin éviter un obstacle afin d'être plus rapidement sur la bonne trajectoire. Mais en contrepartie, il a ensuite besoin de compenser son virage exessif. Ce gameplay basé sur le fait de rééquilibrer des rotations excessives pour une meilleure agilité nous a semblé parfaitement cohérent avec nos intentions.

Enfin, nous avons observé en testant qu'il était plus intéressant de jouer en conservant le Stick Gauche incliné au maximum et en le tournant, plutôt qu'en l'inclinant ponctuellement dans une direction. Pour encourage cette manière de jouer nous avons mis une dead zone très importante sur ce stick : il faut qu'il soit incliné d'au moins 60% pour que le jeu détecte un input.



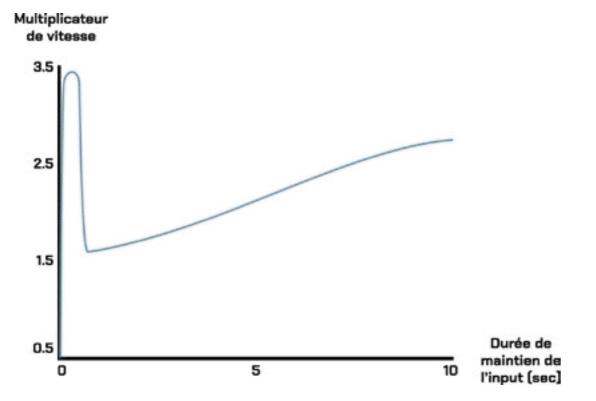
PROPULSION

Une des premières modifications que nous avons faite sur la propulsion pour rendre le déplacement plus rythmé et réactif était d'ajouter une mécanique d'accélération. Concrètement, un multiplicateur est appliqué sur la poussée du joueur, ce multiplicateur évolue selon une courbe en fonction de la durée d'activation. Si le joueur lâche l'input de propulsion, l'accélération revient rapidement à sa valeur mimale.

La courbe commence par un bref mais significatif pic au démarrage. Immédiatement après le pic elle atteint sa valeur la plus basse, puis va lentement augmenter. Le but était de donner une impulsion initale à l'avatar, toujours dans l'optique de faire en sorte qu'il réagisse rapidement, tout en donnant un avantage au fait de maintenir la propulsion activée sur une longue durée, accompagné d'une impression de montée en puissance. Lors d'une collision. l'accélération recule de la moitié de la longeur de la courbe. Cela donne le ressenti d'avoir fait une erreur tout en ralentissant l'avatar, ce qui le rend plus facile à controler pendant quelques secondes, permettant au joueur retrouver

une trajectoire stable. La courbe a connu plusieurs ajustements pour être équilibrée. Notamment la durée pour atteindre l'accélération max, qui est passé de 3 à 10 secondes au cours du développement. Précédemment, le pic initial était aussi trop puissant par rapport à l'accélération max, les joueurs n'avaient pas de raison de faire autre chose que démarrer en boucle. Nous avons donc rendu l'accélération max plus importante. Aussi, nous avons brièvement

envisagé de mettre un deuxième pic lorsque le joueur atteignait l'accélération max, mais ce pic de vitesse soudain était peu contrôlable. Nous avons cependant conservé un effet visuel à ce moment-là. Nous avons jugé que dans l'état actuel du jeu, le pic initial est utile pour changer de trajectoire ou accélérer soudainement, et l'accélération max est plus avantageuse sur de longues distances. Ce qui nous paraît équilibré.



Au début du projet, la propulsion était purement binaire, mais changer de controller et utiliser la manette nous a permis de mettre en place une poussée analogique pour le core ship et les thrusters associés aux gachettes de la manette. Cela fournit au joueur beaucoup plus de contrôle sur sa vitesse, rendant notamment les espaces serrés plus navigables (ce qui était un problème jusqu'à lors). Aussi, pour rendre les basses accélérations plus plus faciles à utiliser, un exposant 4 est appliqué sur la valeur d'input donnée par le joueur. De cette manière, toute pression exercée sur la gachette, autre que le maximum, est plus proche de o qu'elle ne l'aurait été avec une relation linéraire. Le maximum est toujours le maximum, o est toujours o, mais les valeurs basses sont plus faciles à input.

L'ajout d'une poussée analogique nous a cependant demandé de repenser le fonctionnement de l'accélération. Maintenant, l'augmentation de l'accélération n'est plus liée au fait de simplement maintenir l'input, mais de maintenir l'input à plus ou moins la même valeur. Nous détectons les différences brusques d'input, de sorte à ce que si le joueur augmente d'un coup la valeur de son input, l'accélération sera réinitialisée, l'abscisse de la courbe retourne à 0, et donc le joueur bénéficie à nouveau du pic de démarrage. Et à l'inverse, si la valeur d'input du joueur baisse d'un coup, l'accélération est également reset, mais l'abscisse ne retourne pas jusqu'à 0, mais environ 0.6, c'est à dire après le pic. Si le joueur ralentit, il n'a probablement pas envie d'un pic de poussée.

Vu que nous détections les différences de valeur d'input dans le temps (concrètement à quelle vitesse le joueur atteint une certaine valeur), nous avons eu l'idée d'ajouter un léger multiplicateur sur le pic d'accélération initiale. Un multiplicateur qui est plus élevé plus le joueur atteint rapidement l'input max. Concrètement, plus il appuie fort sur sa manette, plus il accélère fort au démarrage. Ce petit plus nous a paru intéressant pour les situations d'urgence comme les courses poursuites. Nous avons aussi trouvé que la mécanique était originale et qu'elle avait le mérite de se baser sur la physicalité de la manipulation réelle effectuée par le joueur.



36 - Cluster 71

Cependant, une des mécaniques les plus importantes dans le rythme de notre déplacement est le boost en sortie de virage. Nous trouvions que le ressenti du mouvement était trop constant et linéaire, légèrement ennuyeux en conséquence. Nous avons eu l'idée de réduire la poussée du joueur pendant les virages pour créer des variations dans sa vitesse en fonction des actions qu'il effectue. Nous y avons vu un potentiel mais ça ne marchait pas très bien en l'état. Un fort ralentissement faisait que le joueur prenait automatiquement des virages très serrés, et on se sentait limité, contenu. Alors nous avons eu l'idée de faire l'inverse : le joueur a plus de poussée lorsqu'il sort d'un virage. Nous avons d'abord essayé un boost d'une valeur et durée fixe dès que le joueur arrêtait de tourner, puis nous l'avons raffiné pour que le multiplicateur dépende à la fois de la durée et de l'amplitude de la rotation (avec



un maximum et des diminishing returns). Le joueur est toujours très légèrement ralenti pendant qu'il tourne (jusqu'à 85% de sa poussée normale), mais surtout, sa poussée peut aller jusqu'à 400%(!!) pendant une demi-seconde lorsqu'il arrête de tourner, ajoutant énormément au dynamisme au gameplay de déplacement. De plus, cette mécanique peut être utilisée

consciemment par les joueurs pour avoir un boost de vitesse à une instant donné. Elle peut même être combinée avec le pic d'accélération au démarrage ainsi que d'autres mécaniques que nous allons aborder. Tous ces bonus se cumulent de manière multiplicative, donnant un réel avantage à un joueur qui trouve le moyen d'en déclencher plusieurs d'un coup.

Un autre multiplicateur de poussée conditionnel que nous avons rajouté est le «wallpush». Nous cherchions un moyen de rendre le joueur un peu plus accessible et moins frustrant et avons eu l'idée de faire en sorte que si le vaisseau pousse contre un mur il pousse plus fort. Concrètement, s'il y a un obstacle à moins de 30 unités derrière le joueur alors qu'il accélère, sa poussée est multipliée en fonction de la proximité de l'obstacle. Cette mécanique aide un joueur qui a amorcé une manoeuvre pour s'éloigner d'un mur, quand bien même il s'y serait prit trop tard. C'est en quelque sorte l'équivalent pour notre jeu du «coyote time» des platformers. Et comme un coyote time, s'il aide les nouveaux joueurs, il peut également être exploité par un joueur expérimenté pour avoir un avantage. Un joueur peut essayer de se rapprocher des murs et de s'en servir pour se propulser plus fort. Aussi, la notion de «mur derrière l'avatar» est propre à chaque thruster. Le wallpush d'un thruster sera activé s'il y a un mur derrière celui-ci, même s'il n'y en a pas derrière le noyau de l'avatar.

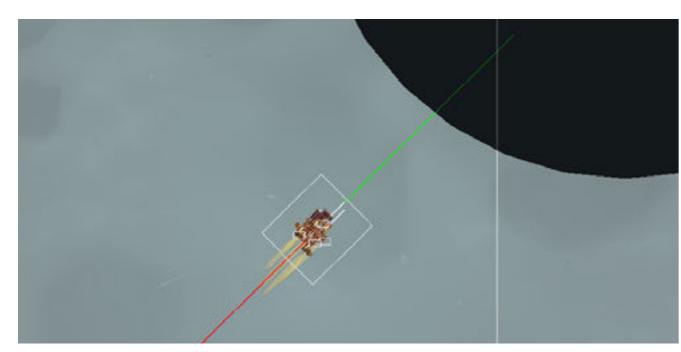
En effet, le système d'assemblage de notre jeu permet de placer des thrusters dans 4 directions et nous considérons les thrusters latéraux comme une une manière particulièrement importante d'exploiter nos systèmes. Placer des thrusters perpendiculaires à la direction du déplacement de l'avatar permet d'esquiver des obstacles dans avoir besoin de tourner ou de prendre des virages beaucoup plus



serrés en utilisant ces thrusters pour contrer l'inertie du vaisseau et l'empêcher de glisser trop loin. Cependant, nous trouvions qu'un thruster latéral avait trop peu d'impact. Surtout après l'ajout des components rotators dans le jeu, qui étaient devenu un moyen beaucoup plus efficace de rendre le vaisseau plus maniable. Alors, nous avons fait en sorte que si un thruster pousse dans une direction perpendiculaire à la direction de la vélocité de l'avatar, sa puissance est multipliée par 3. Ce qui a rendu les thrusters latéraux utiles à nouveau. Cela dit, techniquement, l'angle n'a pas besoin d'être strictement perpendiculaire. L'interval est en fait assez généreux, compris entre 60 et 170°. Ce sont les valeurs que nous avons trouvées en testant pour trouver les contextes où ce boost serait nécessaire. Cette mécanique a également eu pour effet de permettre au joueur, même sans thrusters latéraux, de repartir un peu plus facilement après un changement de direction brusque. Ajoutant au sentiment d'élasiticité du mouvement. Nous avons décidé de garder cette propriété.

Un problème a cependant émergé de tous ces multiplicateurs de vitesse conditionnels : il arrivait que le joueur en active plusieurs en même temps sans le vouloir, ce qui pouvait, dans les endroits exigus, le propulser d'un coup dans un mur contre son gré. Et s'il est intéressant dans une optique d'apprentissage de ces mécaniques qu'elles arrivent parfois par hasard, nous ne pouvions pas laisser un système qui jette le joueur dans des obstacles sans que celui-ci ne comprenne

pourquoi. Alors, nous avons mis en place une mécanique qui vérifie s'il y a un mur devant le joueur à moins de 50 unités. S'il y en a un, l'effet de tous les boosts susmentionnés est réduit jusqu'à 20% de leur effet normal. La poussée de base du joueur et l'inertie déjà accumulée ne sont pas affectées. Cette sécurité a permis d'éliminer presque complètement ce genre d'incidents sans pour autant retirer le challenge du déplacement ou limiter ce qu'il est possible de faire avec le noyau jouet.



Enfin, nous avons mentionné plus tôt que l'avatar avait une marche arrière quand nous avons commencé le projet. Elle avait été ajoutée par défaut, et nous nous sommes par la suite rendu compte que la retirer n'enlevait rien à notre gameplay, alors nous l'avons retirée. Ce qui, en outre, nous a permis de ne pas avoir besoin de régler certains problèmes de rendre le système un peu plus simple. Et puis, si un joueur désire une marche arrière, il a la possibilité de la créer lui-même grâce au système d'assemblage.

RAPPORT STATS / MASSE

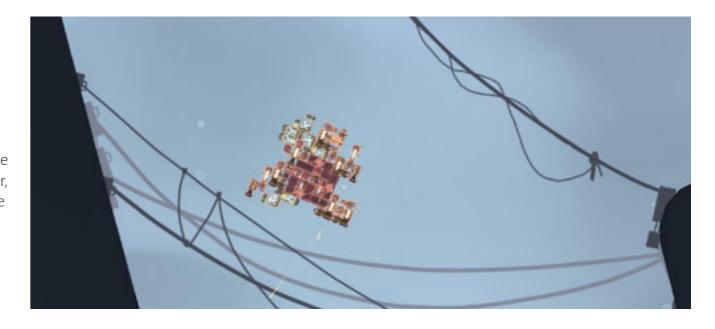
L'assemblage a un fort impact sur le comportement du déplacement. Notamment, chaque component ajouté augmente la masse du vaisseau. Nous voulions que ces changements de masse affectent réellement le ressenti du déplacement. Cependant, il nous fallait éviter qu'un vaisseau lourd ne devienne incontrôlable comme ce fut le cas à un moment lors du développement. Ainsi, 4 paramètres sont affectés par la masse de l'avatar:

- Inertie
- · Vitesse de rotation
- Poussée
- Vitesse d'accélération

Ils évoluent selon une courbe qui prend la masse de l'avatar en abscisse. La modification apportée à chaque paramètre est ensuite pondérée par un multiplicateur, nous nous sommes rendu compte qu'ils ne fallait pas qu'ils baissent tous de la même quantité. La poussée n'est donc que très légèrement affectée. Nous ne voulions pas qu'un vaisseau lourd devienne trop lent, surtout qu'avec une inertie accrue,

il aurait besoin de beaucoup de poussée pour pouvoir changer de direction. La vitesse d'accélération en revanche est très lourdement affectée par la masse. Le temps nécessaire pour atteindre la vitesse de pointe augmente drastiquement avec la masse, il est donc plus difficile d'optimiser la vitesse d'un vaisseau lourd, mais leur potentiel en terme de vitesse n'est que marginalement plus bas que celui d'un vaisseau plus léger. Il est aussi capable de transporter beaucoup plus de cargo, justifiant cette difficulté accrue.

La réduction de la vitesse de rotation et de l'inertie permet de donner l'impression de contrôler une espèce de bélier qui a du mal à changer de direction, tout en incitant à l'utilisation de rotators et de thrusters latéraux, demandant une certaine maîtrise de l'assemblage à un joueur qui souhaite un énorme vaisseau. Une maîtrise qu'il devrait avoir commencé à acquérir sur le temps de jeu qui lui a permis d'aquérir les components nécessaires pour devenir aussi grand.



40 - Cluster 71

ASSEMBLAGE

INTENTIONS

Grâce à l'assemblage, nous voulions que le joueur accorde de la valeur à son avatar. Notre raisonnement était qu'un joueur qui passe du temps et s'investi dans une réflexion pour modifier et customiser son avatar développe un attachement pour celui-ci.

Design & Itérations

«KATAMARI SNAPPING»

La notion de collage est la première à avoir émergé de la thématique donnée, laquelle nous avons d'abord itérée comme mécanique physique, inspirée notamment du jeu Katamari Damacy.

Nous avions alors mis en place un player aux déplacements classiques qui, au contact de components, s'attachait à eux comme ils venaient.

Cela permettait en plus d'avoir une superposition entre nos deux gameplays, d'assemblage et de mouvement.

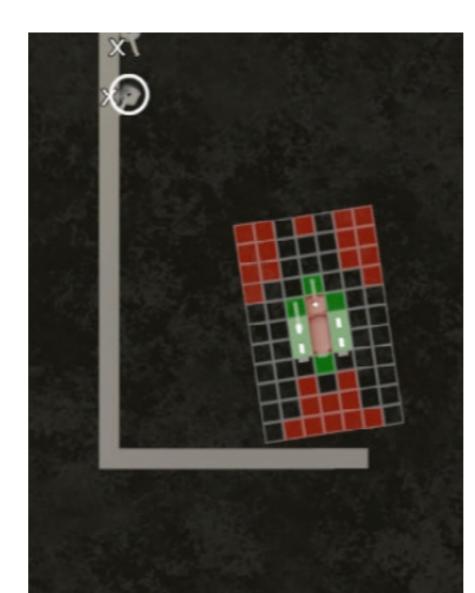


Ainsi, un thruster poussait dans la direction dans laquelle il fut assemblé

De plus, la force des thrusters était appliquée localement, donc s'ils n'étaient pas alignés avec le centre de gravité de l'avatar, ils induisaient une rotation.

Bien que des comportements en émergeaient, cette itération manquait cruellement de structure et empêchait au

joueur d'avoir un vrai contrôle sur ce qu'il assemblait, en plus de faire rapidement des vaisseaux incontrôlables ou accidentels. Une série de playtests nous a permi de confirmer que ce fonctionnement était extrêment frustrant.



Itération de Décembre, premier système de grille

GRILLE BASIQUE

Pour offrir plus de contrôle au joueur et avoir une structure de jeu plus prédictible et malléable, nous avons mis en place un système de grille pour l'assemblage, séparant ainsi nos gameplay de déplacement et de construction.

Le joueur pouvait ainsi activer la grille d'assemblage à tout moment, ce qui mettait le jeu en pause et permettait au joueur de sélectionner les components proches pour les assembler sur son vaisseau en utilisant la grille.

Ceux-ci avaient des positions prédéfinies et des rotations bloquées à des pas de 90°.

La grille était alors infinie, et les components étaient tous des carrés n'occupant qu'une seule case, cela nous permettait d'itérer avec les paramètres «irréductibles», explorer ce que la forme la plus basique de ce concept offrait pour envisager ses différentes directions.

Aussi, décoréler l'assemblage du déplacement nous a libéré de nombreuses contraintes en terme de design. Notamment, le déplacement n'était plus limité par la nécessité de précision de l'ancien système d'assemblage.

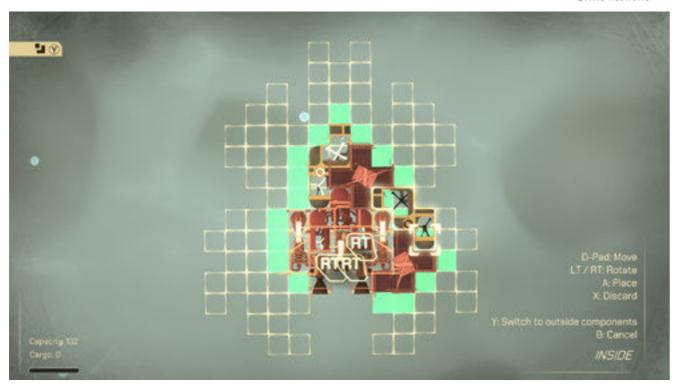
Enfin, assembler des components sur le vaisseau augmente sa masse, modifiant légèrement ses propriétés de déplacement.

GRILLE AVANCÉE ET COMPONENTS

Le gameplay d'assemblage devenu une phase de jeu à part, il nous a semblé possible et nécessaire d'en faire un vrai gameplay plutôt qu'une fonction purement utilitaire. Surtout si nous voulions que le joueur accorde de la valeur au vaisseau qu'il a construit, le processus de cette construction allait devoir être intéressant et challengeant.

Ainsi, nous avons repris la structure des inventaires de jeux comme Resident Evil ou Dredge, où l'inventaire est manifesté par une grille de taille limitée et une forme définie, dans laquelle il faut faire rentrer un nombre limité d'objets avec eux même des tailles et formes définies.

S'inspirant de ces structures, nous avons ainsi rendue la grille finie, avec une forme propre (cf. Pattern de la grille), et avons élaboré un set de formes de component (cf. Patterns des components).



De plus, nous avons raffiné l'évolution des paramètres de déplacement de l'avatar en fonction du nombre de components assemblés. Un vaisseau avec plus de components aura plus d'inertie, et mettra plus de temps à atteindre une vitesse de plus basse, et tournera moins vite.

Avec cette itération nous avons également achevé de définir nos types de components.

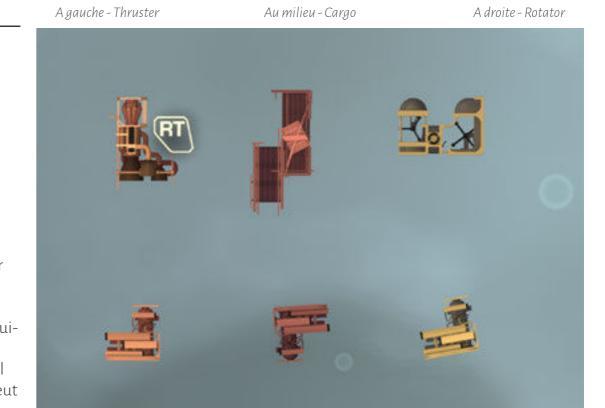
COMPONENTS

Le joueur a accès à 4 types de components dans Cluster 71 :

- Thrusters
- Cargos
- Rotators
- Amplifiers

Chacun a une fonction différente, augmente la masse du vaisseau et prend de l'espace sur la grille : le joueur va donc devoir prendre des décisions pour trouver un équilibre dans la quantité de chaque type qu'il accroche à son vaisseau. Il peut également adapter la composition de celuici en fonction des activités qu'il compte entreprendre dans le jeu. Par exemple, s'il souhaite explorer une nouvelle zone, il peut ne pas emporter de Cargos components, ceux-ci n'étant utiles que pour réaliser des quêtes de livraison.

Chaque type de component a des couleurs et des codes graphiques différents pour permettre une meilleure identification de leurs fonctions par les joueurs. Les Amplifiers quant à eux sont de la couleur du type qu'ils affectent.



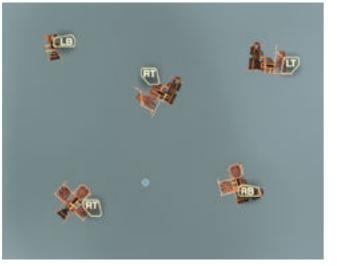
En bas - Amplifiers

THRUSTERS

L'effet des Thrusters, contrairement aux autres components, est déclenché par un input. Quand activés, ils appliquent une force sur le joueur, le poussant dans la direction à laquelle le thruster fait face. Ils permettent au joueur d'augmenter la vitesse de son vaisseau, mais aussi de lui donner la possibilité de se déplacer dans d'autres directions qu'en face de lui.

Ils peuvent être associés à différents inputs: LB, RB, LT, RT. Et le bouton qui leur est associé est affiché à côté d'eux quand ils sont dans l'enivironnement ou en mode assemblage. Ainsi, le joueur peut créer un vaisseau qui contient des Thrusters liés à divers boutons ce qui lui donne la possibilité de ne pas tous les activer en même temps et de leur donner différentes fonctions. Cette décision permet de la variété et de la créativité dans l'usage de ces

Thrusters avec différents inputs



components. Nous avous décidé de donner 4 inputs possibles pour ne pas limiter le joueur. 4 c'est potentiellement un par direction, ou encore un pour aller à gauche, un à droite, et deux pour devant, donnant au joueur plus de contrôle sur la force appliquée dans cette direction.

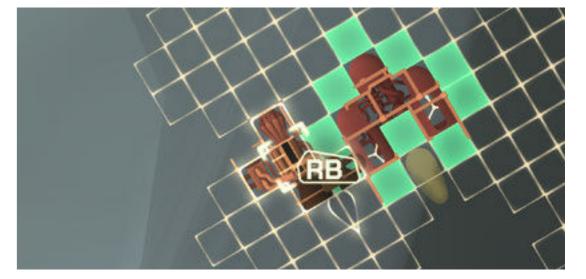
L'input est une propriété d'un Thruster et ne peut être changé. Quand un Thruster est trouvé dans l'environnement, le bouton qui lui est associé est déterminé de manière procédurale. Les chances sont les suivantes :

RT:35%LT:15%LB:25%RB:25%

S'il est acheté, le joueur peut choisir le bouton. Le but est que le joueur ait à prendre des décisions quand il trouve un component, qu'il prévoit les manières dont il pourrait l'utiliser en fonction du contrôle qui lui est associé. Ce fonctionnement crée aussi des situations où le joueur veut un Thruster spécifique, et va donc partir explorer plus pour le trouver, ou effectuer des livraisons pour obtenir les fonds nécessaires pour se le procurer. Il est encouragé à interagir avec les autres systèmes du jeu.

Enfin, les Thrusters présentent un challenge d'assemblage supplémentaire par rapport aux autres types car le joueur ne cherche pas seulement à les faire rentrer dans la grille, mais aussi à les orienter d'une manière spécifique.

Pour permettre une meilleure lisibilité de leur orientation, une icône de flamme est présente à l'arrière de leur outline de selection en mode assemblage, et les échappements sur leurs modèles sont plus sombres que le reste.





Poussée avec des thrusters installés dans deux directions différentes

Game Design

Cargos

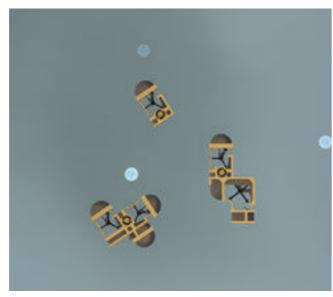
Les components de type Cargo permettent au joueur de transporter des cargaisons, et donc d'effectuer des livraisons: un des objectifs principaux du jeu. Chaque Cargo augmente la capacité de cargaison du joueur.

Ces components n'ont pas d'autre effet et n'apportent rien qui facilite le déplacement du joueur : au contraire, ce sont les components les plus lourds. Ainsi, le joueur doit en permanence équilibrer son nombre de components Cargo avec les autres types de components de sorte à conserver un vaisseau agile tout en augmentant sa capacité de transport pour engranger des récompenses plus grandes. Un vaisseau plus lourd demande plus de maîtrise et rend les livraisons plus risquées, le joueur doit donc doser la quantité de risque qu'il est prêt à prendre.

ROTATORS

Les Rotators, comme leur nom l'indique, augmentent la vitesse de rotation du vaisseau. Tout comme les Cargos, leur effet est passif et permanent tant que le component est accroché au joueur. Cargos





Rotators

Si placer des Thrusters latéraux est un moyen de rendre le vaisseau plus agile en lui donnant plus de directions de mouvement et un moyen de glisser moins loin dans les virages en compensant son inertie, les Rotators sont une approche différente pour répondre à cette question. En permettant au joueur de changer de direction très vite, ils ouvrent la porte à une autre possibilité de changement de trajectoire : avoir une forte poussée en forward, et tout le temps se propulser dans une nouvelle direction pour éviter les obstacles. Ce style de déplacement où le joueur a presque l'impression de rebondir entre les murs sans jamais les toucher, en contrant son inertie à chaque fois, nous a semblé très intéressant.

Et même sans se spécialiser dans le Rotator de manière aussi extrême, c'est un type de component qui permet de compenser la rotation lente des vaisseaux lourds, laissant les joueurs régler leur vitesse de rotation pour obtenir quelque chose qui leur est agréable.



AMPLIFIERS

Les Amplifiers sont plus complexes à utiliser que les autres types de components et ont été pensés comme un challenge d'assemblage avancé pour les joueurs expérimentés. En conséquence de quoi ils ont le potentiel d'être très avantageux pour l'optimisation du vaisseau, mais sont plus difficiles à obtenir.

Chaque Amplifier affecte un type de component donné (Thruster, Cargo ou Rotator), il augmente de 42% les stats de tous les components adjacents du type concerné

Les cases adjacentes à un Amplifier sont affichées en jaune orangé plutôt qu'en vert pour expliciter où son effet est appliqué. Pour qu'un component bénéficie de l'effet d'un Amplifier il suffit qu'il soit sur une de ces cases.

En terme de gameplay, le but de l'Amplifier est d'inciter le joueur à placer des components à des endroits précis plutôt que n'importe où sur la grille. Plus précisément, pour utiliser efficacement un Amplifier, il doit l'entourer d'autant de components du bon type que possible. Ce qui le pousse à observer avec soin les formes des components et l'espace disponible et à repenser l'agencement de son vaisseau pour tirer parti au mieux de l'Amplifier.

A partir de 3 components affectés, nous considérons que l'Amplifier est vraiment avantageux, mais il est théoriquement possible d'aller jusqu'à 7. Il devient cependant très difficile de construire autour de ça.



En jaune, zone d'effet de l'amplifier de thruster

RATIONALISATION

PATTERNS DES COMPONENTS

Nous avons tenté de lister, d'abord, les trous et bosses qui se forment naturellement avec la simple pose d'une forme « L 2x3 » et avons listé les patterns (à petite échelle) redondants.

Ainsi, à partir de ces patterns, nous avons commencé à chercher des formes de components intéressantes. Nous avons établi les directives suivantes pour faciliter ce processus :

- 5 cases ou moins
- Placer un component doit faire émerger des nouveaux trous et bosses (pour rendre difficile de créer des formes lisses)
- Patterns de formes identifiables dans les components et les vides créés
- Plusieurs choix de placement (rotations...)

Nous avons aussi envisagé de réaliser un tableau des outcomes possibles des interactions entre tous les patterns de trous et de bosses identifiés. Nous avons cependant réalisé que ce n'était pas réalisable et nous nous en sommes tenus à l'approche empirique décrite ci-dessus.



Il était important que le placement d'un component forme une nouvelle irrégularité pour entretenir la tension ludique de l'assemblage. Chaque trou comblé crée un nouveau trou à combler, une nouvelle forme vide que le joueur peut retrouver dans la forme des components qu'il place. Un lien qu'il peut faire entre espace libre et objet à placer.

Le component rectangulaire de 2x1 cases a été pensé comme une sorte de joker, un component significativement plus facile à placer qui permet de combler les trous restants, de corriger les erreurs du joueur.

Un 1x1 n'aurait présenté aucun challenge

En haut à gauche, la forme établie du vaisseau de base. A gauche en couleur, les formes de components trouvées. A droite, en noir, les patterns de trous et de bosses.

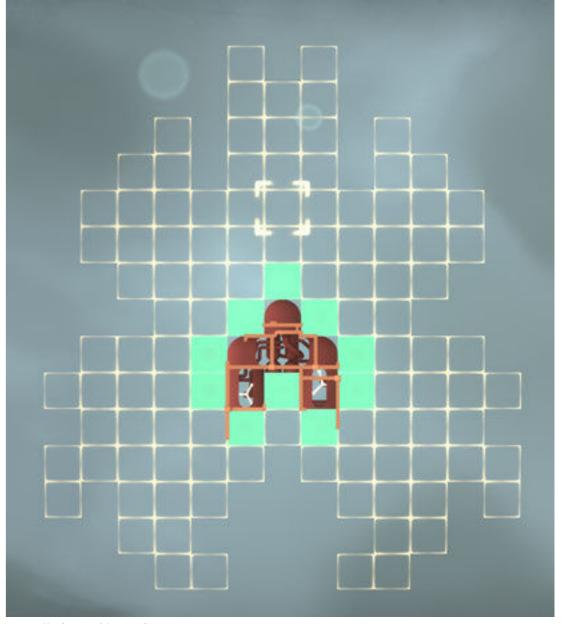
PATTERN DE LA GRILLE

Inspirés notamment par Dredge, nous avons décidé de faire une grille qui ne soit pas strictement rectangulaire.

Nous avons effectué plusieurs itérations jusqu'à obtenir cette forme qui permet

- D'éviter des construction trop monolithiques
- De faire augmenter la difficulté de placement des components alors qu'on s'éloigne du centre
- D'avoir très tôt des obstacles de formes spécifiques qui invitent à placer des components précis pour les combler
- Qui est très difficile à remplir entièrement et invite les joueurs assidus à essayer différentes permutations pour optimiser leurs placements

Nous voulions aussi que, si remplie entièrement, elle conserve un aspect intéressant esthétiquement.



Grille d'assemblage vide

BALANCING DES COMPONENTS

Category	Shape	Nb of Cells	Diff. Mod.	Mass / Cell	Indicative Mass	Final Mass				Base HP	Extra HP / cell	Indicative HP	Final HP	Aug.	Avg. Av
		10					Thrust	Handing							
Core	V3x3	6				2.7	48	30					10	5	
		100	8	12	3		Capacity / cell	Indicative Cap.	Final Cap.		1			1	
Cargo	2x1	2			0.2	0.2		20	20			8.8	1		
	L2x3	4	1.18		0.4	0.34	ı	40	47			9.6	11	1	
	U3x2	5	1.26		0.5	0.4	10	50	63			10	1:		
	W3x3		1.39	0.1	0.5			50	70		0.4	10	14	11.8	5
	52x4	5	1.3		0.5	0.38		50	65		1972	10	1:	9	
	Y3x3		1.32		0.5	0.38		50	56			10	13		
	X3x3	5	1.31		0.5	0.38		50	66			10	1:		
					8		Thrust / Cell	Indicative Thrust	Final Thrust						
	2x1	2	1.03		0.05	0.05		. 26	27		0.7	15.4	16	5	
	L2x3	4	1.24		0.1		1	52	64			16.8		19.85 15	
	U3x2	5	1.27		0.125	0.1		65	83			17.5	22		
Thruster	W3x3	5	1.37		0.125	0.09		65	89	14		17.5	24		
	52x4	5	1.38		0.125	0.09		65	90			17.5			15.57
	Y3x3	5	1.33		0.125			65	86			17.5			
	X3x3	5	1.25		0.125	0.1		65	81			17.5	22		
		15 3		8	3		Rotat/Cell	Indicative Rotat	Final Rotat						
	2x1	2	1		0.16	0.16		4.5	4.6		0.5	12		5 7 9 15.57	
	L2x3	4	1.18		0.32	0.27	1	9	10.6			13	15		
	U3x2	5	1.20		0.4	0.32	ı	11.25	14.2			13.5	17		i7
Rotator	W3x3	5	1.39	0.08	0.4	0.29	2.25	11.25	15.6	11		13.5	11		
	S2x4	5	1.3		0.4			11.25	14.6	20		13.5	4		
	Y3x3	5	1.32		0.4			11.25	14.9			13.5			
	X3x3	- 5	1.31		0.4	0.31		11.25	14.7			13.5	11	3	
Amplifier	3	4 7	1.41	0.1	0.7	-	L.			10		17			
Milpinier	Elb	3	1.12	1001	0.3	0.27	8			10	*	13	15	15	5

Tableau de balancing des différentes statistiques des components

Avec trois types principaux de components qui interagissent avec les capacités du joueur et sept formes différentes qui ne sont pas toutes aussi difficiles à placer dans la grille, nous avons fait face à plusieurs questions d'équilibrage.

Chaque component est défini par un nombre de cases occupées, une masse, une stat qui lui est propre (poussée, capacité, etc) et des points de vies. Chacune de ces valeurs est calculée selon une valeur par case et par type. Par exemple, les cargos ont une masse de 0.1 par case, les thrusters 0.025. Ces valeurs brutes sont ensuite pondérées par un «Difficulty Modifier». Un multiplicateur qui dépend de la

difficulté de placement de la forme du component. Le but était de faire en sorte qu'un component plus dur à placer soit plus avantageux. Notamment pour que le joueur ne se rabatte pas exclusivement sur les components les moins challengeants.

Pour définir le Difficulty Modifier, nous avons établi plusieurs critères auxquels nous avons associé des coefficients dans leur importance dans la difficulté de placement :

- Dimensions
- Nombre de cases
- Nombre d'extrémités
- · Nombre de coudes.

Coeff->	4		1		3		5				
	Size		Cells		Ends		Elbows		Difficulty modifier	Adjusted Diff. Mod.	
2x1	2	0	2	0	2	0	0	0	1		
L2x3	6	0.5714285714	4	0.6666666667	2	0	1	0.3333333333	1.36	1.18	
U3x2	6	0.5714285714	5	1	2	0	2	0.6666666667	1.51	1.26	
W3x3	9	1	5	1	2	0	3	1	1.77	1.39	
S2x4	8	0.8571428571	5	:1	2	0	2	0.6666666667	1.6	1.3	
Y3x3	9	- 1	5	1	3	0.5	1	0.3333333333	1.63	1.32	
X3x3	9	1	5	1	4	1	0	0	1.62	1.31	

Tableau de calcul du modificateur de difficulté

Game Design

Ajouts au calcul du modificateur de difficulté spécifique aux thrusters

Pour les thrusters, nous avons eu besoin de calculer une variante du Difficulty Modifier pour prendre en compte la notion d'orientation de ces components. Le joueur ne cherche pas seulement à les faire rentrer dans la grille, mais aussi à leur donner une rotation précise. Ce qui rend comparativement plus simples à placer les components qui ont une symétrie sur un ou deux axes.

Ci-contre, le Difficulty Modifier pondéré pour les thrusters.

De plus, nos tableaux sont dynamiques, modifier par exemple le poids d'un des paramètres du difficulty modifier va automatiquement modifier les stats de tous les components. Et nous avons également développé un outil pour récupérer les valeurs de nos feuilles d'équilibrage et les intégrer dans le jeu afin d'itérer plus facilement (Cf. CSV Tool).

3		5			
Orientations		Length		Thruster Difficulty Mod.	Adj. Thrust. Diff. Mod.
2	0.3333333333	2	0	1.05	1.03
4	1	3	0.5	1.48	1.24
3	0.666666667	3	0.5	1.53	1.27
4	1	3	0.5	1.74	1.37
4	1	4	1	1.75	1.38
4	্ৰ	3	0.5	1.65	1.33
1	0	3	0.5	1.5	1.25

Un problème que nous avons dû traiter est le rapport entre la poussée et la masse apportées par les components et celles du core. Dans une version antérieure du jeu, attacher ne serait-ce qu'un thruster sur le core rendait l'avatar extrêment rapide, au point de ne pas être controllable. Il redevenait jouable si on rajoutait des cargos. Pour conserver un avatar équilibré en toutes situations, nous avons augmenté significativement la masse du core, pour que l'ajout de components ne fasse pas une différence aussi drastique. Nous avons également réduit l'écart de masse entre les différents types pour qu'il ne soit pas nécessaire d'avoir des cargos pour que le vaisseau soit équilibré.

Enfin, nous avons égulibré les points de vie des components de sorte à ce que les components les plus importants pour la survie du joueur soient les plus résistants. En effet, nous avons constaté que quand un joueur est pourchassé par un ennemi, si l'ennemi détruit ses thrusters, le joueur n'a simplement plus de moyen de s'échapper et doit attendre sa destruction. Nous avons donc rendu les thrusters significativement plus résistants. Les cargos eux sont plus fragiles de sorte à ce qu'ils soient détruits en premier. Un vaisseau sans cargo devient plus léger, et donc a plus de chances d'échapper à son poursuivant.

Base HP	Extra HP / cell	Indicative HP	Final HP	Avg.	Avg. Avg
			15		
		8.8	9	1	
		9.6	11		
		10	13		
8	0.4	10	14	11.85	
	1000000	10	13		
		10	13	1	
		10	13		
		45.4	40		
		15.4	16 21	ı	
		16.8 17.5	21		
14	0.7	17.5	24		. +
14	0.7	17.5		19.00	15.571
		17.5		ı	10.01
		17.5	0.000		
		12	12	ı	
		13	15		
		13.5		1000	
11	0.5	13.5	19		
		13.5	18		
		13.5	18		
9	1	13.5			
10	1	17			
		13	15	15	

Section d'équilibrage des points de vie, par forme et par type de component

EXPLORATION

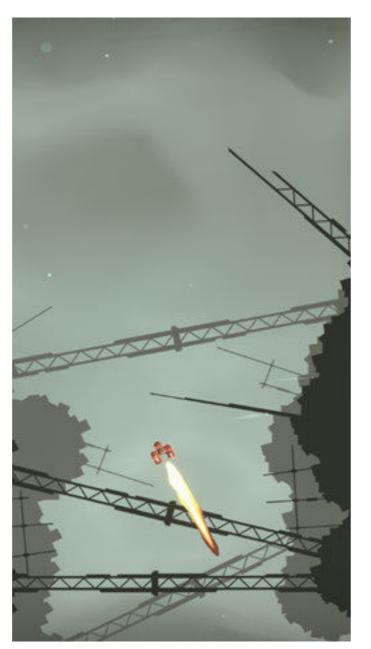
RECHERCHE ET RÉFÉRENCES

Nous avions dans nos intentions d'avoir un système d'exploration, mais n'avions aucun outil ou élément le nourrissant dans notre toy. Nous avons décidé de produire une recherche et analyse sur les modes d'exploration de différents jeux.

Ainsi, nous avons listé 14 jeux avec des 3C, exploration et gameplay variés. Pour chacun d'entre eux, nous avons établi le fonctionnement de base du jeu, du système de navigation, de la carte et de l'exploration. Nous en avons également listé les points positifs et négatifs au sein de l'expérience de jeu, et projetés dans notre toy. Aussi, nous avons procédé à l'analyse de l'exploration présente dans notre toy en l'état pour en lister les points sur lesquels il nous fallait travailler. Pour chacun de ces points, nous avons listé les solutions présentes dans les jeux analysées qui pourraient correspondre à nos problèmes.

La liste de jeux est comme suit

- The Legend of Zelda: Breath Of The Wild (2017)
- Dredge (2023)
- Tunic (2022)
- Hollow Knight (2017)
- Outer Wilds (2019)
- Rain World (2017)
- Death's Door (2021)
- Hyper Light Drifter (2016)
- · Horizon Zero Dawn (2017)
- Barotrauma (2019)
- Chants of Sennaar (2023)
- Death Stranding (2019) Project Zomboid (2013)
- Subterrania (1993)



ABSENCE DE BUILD-UP

Problème: Le joueur pouvait croiser une créature sans s'y attendre, cela créait certes de la surprise, mais celle-ci n'était pas bien exploitée car rien n'annonçait un danger que le joueur pouvait appréhender.

Solutions:

- Faire varier l'ambiance sonore et/ou visuelle lorsqu'un ennemi est proche mais pas encore visible par le joueur (ombre des Vultures dans Rain World, fenêtres/portes attaquées par les zombies dans Project Zomboid).
- Indiquer la présence (plus ou moins précise tant dans la nature que la localisation) d'un danger sur une map ou l'UI, ou via des éléments narratifs (indication des ennemis sur la map dans Tunic, squelettes et restes d'anciennes créatures dans BotW).



Vultures - Rain World

CURIOSITÉS À EXPLORER

Problème: Rien ne poussait le joueur à se balader, explorer les environnements, sinon la direction indiquée de la livraison à déposer.

Solutions

- · Attiser la curiosité via des éléments sur la/les maps.
- · Montrer des objets de l'autre côté d'un mur fin, inaccessibles (Passages secrets ou verrouillés dans Hollow Knight).
- (En lien avec la map) Exploration permettant de cartographier les zones traversées.
- Présence de différentes régions marquées par leurs ambiances ou des comportements différents à adopter (Planètes dans Outer Wilds, régions dans Rain World et Hollow Knight).

Exploration d'Old Herringen

56 - Cluster 71 Game Design

VISION RESTREINTE

Problème: Le joueur peut rapidement atteindre de grandes vitesses, mais le format de notre caméra nous limite dans les éléments que l'on peut montrer (via un dézoom et un décalage), ce qui devient frustrant et limitant dans les déplacements, d'autant plus en présence de dangers (obstacles/ennemis).

Solutions

- Redéfinir la vitesse de déplacement.
- Limiter la range des ennemis à ce que le joueur peut voir.
- Amortir la collisions contre certains murs via des éléments type nuage.
- · Redéfinir la range de la caméra



Dézoom avec une musique qui se joue - Death Stranding

ACCÈS À LA MAP

Problème : Pas de map, ni de projection dans son utilisation in-game.

Solutions:

- Disponible en permanence sans aucune restriction, mettant le jeu en pause.
- Disponible en permanence mais avec des restrictions: temps d'activation, temps continuant de s'écouler, attirant les ennemis, etc.
- Limiter l'accès à une/aux maps à certains endroits.

INDICATIONS

Problème: Ce que nous avions fait pour indiquer au joueur les objectifs n'était que la première option qui nous était venue, c'est-à-dire indiquer où aller, ce qui était très limité.

Solutions:

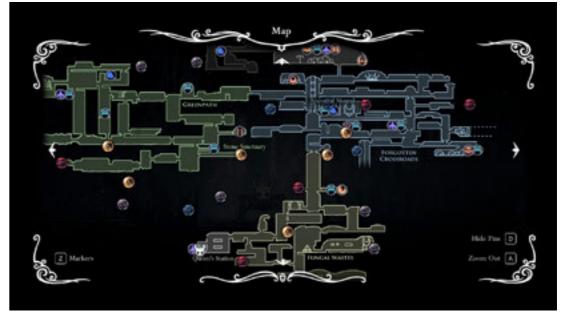
- Etablir un système de waypoints.
- Point d'arrivée indiqué sur la map/UI.

ANNOTATION DE LA MAP

Problème: En projection d'une map, nous nous demandions si l'exploration n'était pas également en partie dans l'appropriation du joueur de sa map.

Solutions:

- Placer des pins (plus ou moins indicatifs / abstraits).
- Placer un ou plusieurs waypoints.



Map avec des indications que le joueur a achetées, et des pins abstraits que le joueur pose comme il veut - Hollow Knight

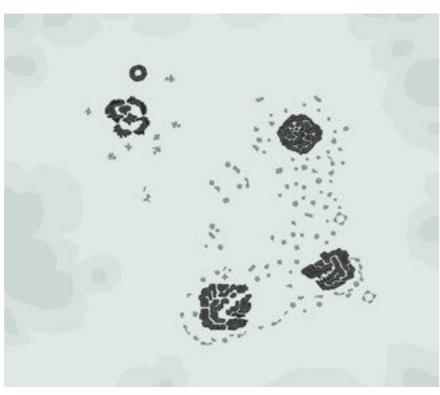
SYNTHÈSE DES CHOIX RÉALISÉS

INDICATIONS ET ACCÈS À LA MAP:

Indiquer une topologie vague ainsi que la position du joueur, accessible tout le temps et en temps réel.



Le joueur peut compléter cette map en achetant des informations, qui précisent la topologie des différents archipels.



Map complétée, régions et astéroïdes détaillés

ANNOTATIONS DE LA MAP:

Indiquer les différents points de livraison, de récupération et la position du joueur sur la map ainsi que sa trajectoire récente.



CURIOSITÉS À EXPLORER :

Formes intrigantes dans la carte, landmarks à petite échelle, et ambiances/ ennemis variés selon la zone dans laquelle le joueur se trouve.



Indice qu'il y a un passage secret quelque part, dans The Maw

ABSENCE DE BUILD UP:

Mettre un trigger qui déclenche une musique menaçante à distance de la créature, puis une de danger lorsqu'elle a repéré le joueur.

Faire que le worm génère des particules quand il racle contre des obstacles. Ces particules vont relativement loin et survivent assez longtemps, le joueur peut donc les voir même si le worm est hors de l'écran. Un son accompagne également le frottement.

Fournir une forme «étrange» et presque environnante, qui change drastiquement lors de la phase d'agression.

Aussi, bloquer la taille de la caméra pendant la course poursuite pour augmenter la tension de la rencontre.



Attaque de worm dans Old Herringen



Outer Wilds, référence de Full Access

STRUCTURE

La structure de Cluster 71 est inspirée de jeux d'exploration comme Outer Wilds ou Myst, dits «Full Access».

Cela implique que l'entièreté du jeu est disponible dès le début, incluant la fin (si le joueur sait la déclencher).

Ca nous permet d'approfondir la notion d'exploration; de ne pas bloquer des accès au joueur et lui permettre de voyager même si ce n'est pas un objectif de mission, et hypothétiquement créer un effet de surprise s'il essaye de sortir sans que la quête finale l'informe de cette possibilité (cf. Game's End).

Boucles de prédiction

Effectuer un virage

Prédiction :

- Le couloir dans lequel l'avatar se déplace présente un coude.
- · L'avatar va tout droit, à une vitesse constante.

Décision :

- Évaluer à quel moment entamer une rotation.
- Évaluer de quel angle effectuer la rotation.

Action :

- Appuyer moins fort sur RT
- Tourner dans l'angle du virage

Correction :

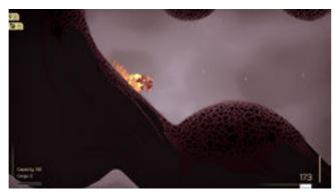
- Ajustement de l'angle de rotation
- Activation plus violente des thruster

Apprentissage:

- Meilleure appréciation des timings et des dosages d'input
- Meilleure compréhension de l'inertie de l'avatar à cette taille







ASSEMBLER UN ROTATOR DE FORME W

Prédiction:

- La grille est ouverte, plusieurs components qui ont déjà été assemblés et il y a un component rotator en forme de W à proximité qui pourrait être attaché.
- Ce component serait utile au joueur car il a besoin de tourner plus vite.

Décision:

- Identifier un pattern de forme du component dans les patterns de forme des espaces libres de la grille
- Choisir une position pour le component qui permet de l'orienter dans la bonne direction tout en créant un assemblage compact qui ne laisse pas de trous impossibles à remplir

Action:

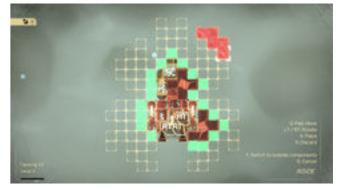
- · Déplacer le component jusqu'à la position choisie.
- Faire tourner le component pour lui donner l'angle désiré.

Régulation:

• Si le component a été déplacé trop loin, ou tourné de trop, ajuster.

Apprentissage:

- Comportement du nouvel assemblage.
- Impact des components sur la structure de l'avatar (par rapport à une autre conformation, comparer image 1 et 2).









64 - Cluster 71

Game Design

BOUCLE DE GAMEPLAY (GUARDIOLA)

Notre boucle de gameplay se sépare en une partie In qui communique avec deux parties Out, une par gameplay core de notre toy.

La première partie Out est celle des déplacements, intervenant comme obstacle cognitif premier à chaque action du joueur, mettant en œuvre toutes ses connaissances sur les capacités de son vaisseau à un instant T pour naviguer correctement.

La seconde partie est celle de l'assemblage, servant les déplacements, et posant un challenge dans le joueur dans une seconde mesure; celle de l'optimisation de l'assemblage, tant dans l'espace, que dans les capacités et contre-coups, des components.

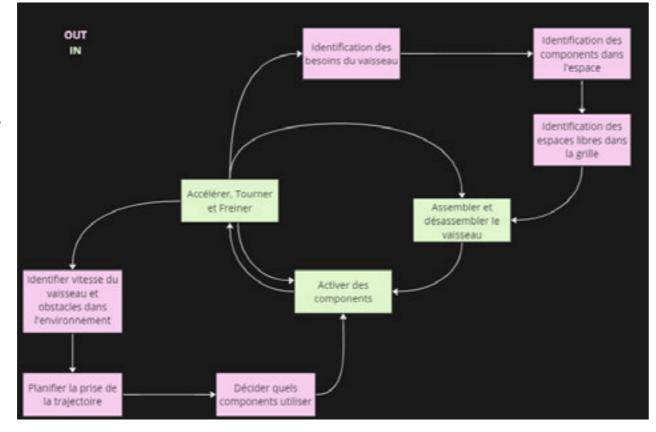
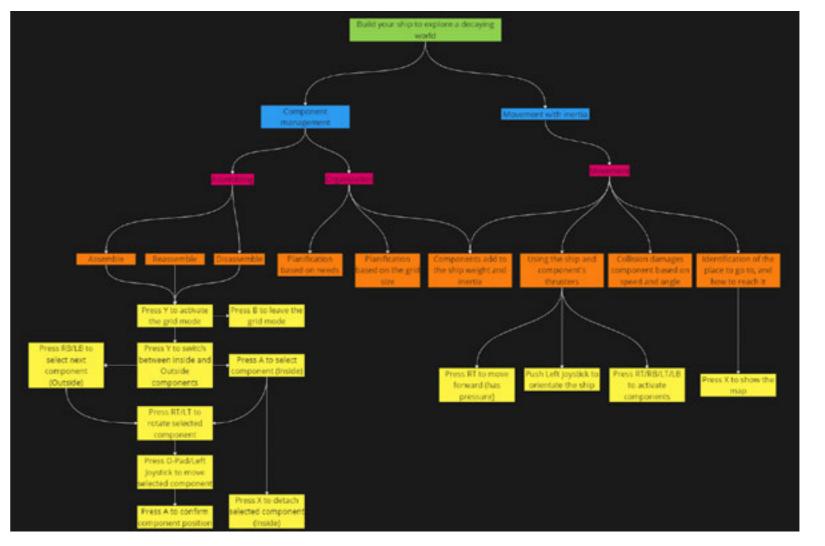


DIAGRAMME DE VENTRICE

Ce schéma de Ventrice retrace la structure de nos core mécaniques.



66 - Cluster 71

Game Design

MOTIVATION DU JOUEUR

Boucles OCR

Court terme

PLACER UN THRUSTER

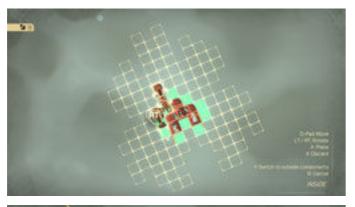
Objectif: Ajouter un thruster à l'avatar

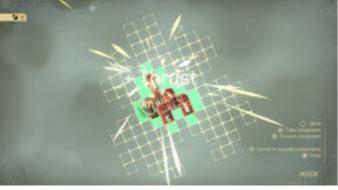
Challenge:

- Choisir la direction dans laquelle utiliser le thruster, prévoir de quelle manière le joueur compte s'en servir
- · Identifier le pattern de forme du component

Reward:

- Amélioration des capacités de déplacement et de la durabilité de l'avatar
- Feedbacks satisfaisants





Après avoir placé un component, des nouveaux espaces à combler sont formés, et il y a un nouveau déséquilibre dans la composition du vaisseau (ex: un thruster augmente la vitesse de déplacement du joueur, ce qui crée donc un besoin de rotators pour conserver un vaiseau maniable). Le joueur est donc encouragé à aller explorer ou effectuer des quêtes pour répondre à ces nouveaux besoins.

Accélérer plus fort en ligne droite

Objectif : Bénéficier du boost de virage en conservant une trajectoire rectiligne et le multiplier avec le boost de démarrage.

Challenge

- Synchroniser et effectuer dans le bon ordre une séquence d'actions (lâcher l'accélérateur, effectuer une rotation tant que l'avatar est toujours déplacé par son inertie, lâcher l'input de rotation et accélérer à nouveau)
- Orienter l'avatar dans une direction dépourvue d'obstacles
- Appuyer fort sur RT

Reward

- · Gain de vitesse significatif
- VFX des thrusters beaucoup plus grand
- Dézoom plus important de la caméra

En effectuant cette manoeuvre, le joueur va souvent être propulsé dans la direction d'un obstacle, ce qui lui demande d'utilser à nouveau cette mécanique pour l'éviter.







68 - Cluster 71

69

EXPLORATION D'UN PASSAGE SECRET

Objectif: Récupérer le component qui a l'air coincé

Challenge:

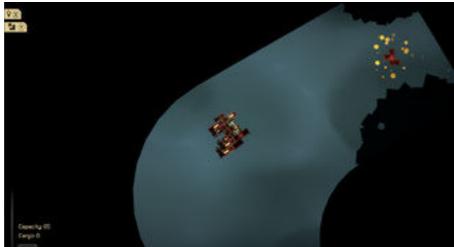
- Se repérer sur l'archipel
- Identifier le faux mur sur la longueur de l'îlot

Reward:

- Obtention d'un nouveau component
- Musique de déclenchement de faux mur

Après avoir découvert l'une de ces zones secrètes, le joueur va essayer d'en chercher partout, et apprendre à identifier les faux murs.





FUIR UN WORM DANS UN ARCHIPEL

Objectif: Perdre l'attention du worm

Challenge:

- Entrer dans un environnement sinueux
- Zigzaguer entre les éléments du décors pour ne pas être ralenti
- Se déplacer pendant une certaine durée en faisant peu d'erreurs pour que le worm ne rattrappe pas le joueur

Reward

 Possibilité de continuer les activités normalement





70 - Cluster 71

Game Design

71

Long terme

Mission de récupération

Objectif: Récupérer des components de quête et le ramener à un point de livraison

Challenge

- Planifier sa trajectoire entre le départ, les points de passage et l'arrivée selon les zones à parcourir et l'avatar du joueur
- Manoeuvrer l'avatar
- Assembler les component de récupération au sein de l'avatar, en minimisant leurs risques d'être cassé
- Prendre en compte le nouveau poids de l'avatar pour se rendre au point d'arriver
- S'adapter aux difficultés engrangées par les rencontres des différents agents

Reward

- Obtention de monnaie pour acheter de nouveaux components ou des éléments de la map
- Nouvelle Quête disponible





QUITTER LE CLUSTER

Objectif: Accumuler assez de vitesse pour sortir

Challenge

- Accumuler suffisamment de thrusters
- Assembler un vaisseau optimisé pour la vitesse
- Maîtriser les mécaniques de mouvement qui permettent de gagner beaucoup de vitesse

Reward:

- · Musique de fin épique
- Ambiance visuelle
- Illustration de fin

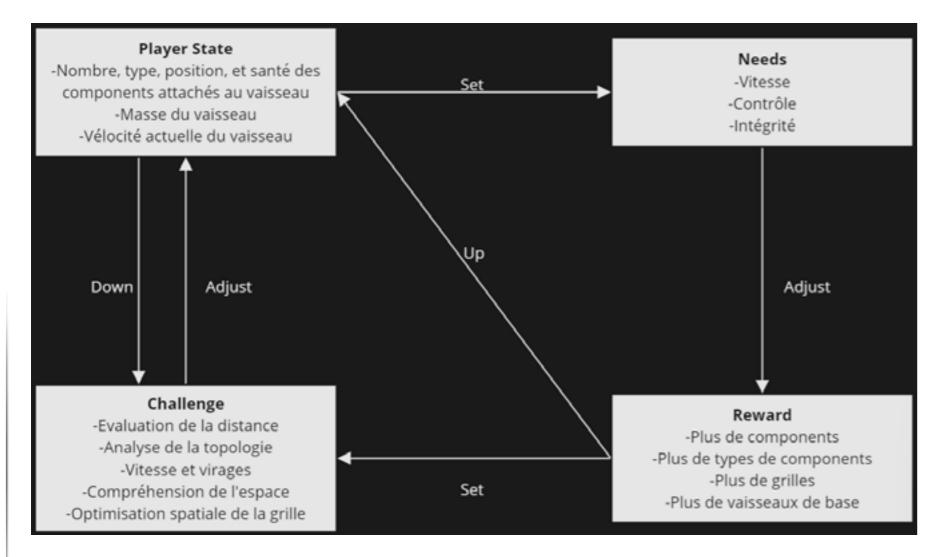




72 - Cluster 71

Game Design ______

Boucle PNRC



74 - Cluster 71

Game Design

SIGNES & FEEDBACKS

Астіон	Effet Visuel	Animation	Caméra	Effet Sonore	Effet Hap- tique
Prendre de la vitesse	Barre de vitesse se remplit et compteur augmente	Compteur de vitesse grossit	Se décentre vers la direction de la vitesse et s'élargit	Bruit de vent	Vibration (à un certain palier)
Avoir un gros vaisseau	/	/	Dézoome	/	/
Appuyer doucement sur l'accélérateur	Petite et fine flamme et translucide	Animation de flamme	/	Bruit d'échappement doux	/
Appuyer fortement sur l'accélérateur	Grosse flamme en sortie de thruster	Effet similaire à une explosion sur la flamme	/	Bruit de flamme qui s'embrase et d'échappement	Vibration au pic d'accélération
Prendre un virage	/	/	/	Tremollo dans le bruit d'échappement	Vibration en sortie
WallPush	Particules sur le mur, plus ou moins denses selon distance	/	/	Bruit de flamme contre une surface	Vibration
Rentrer doucement dans un mur	Petites particules continues au point d'impact	/	/	Bruit d'étincelle	/

Action	Effet Visuel	Animation	Caméra	Effet Sonore	EFFET HAP- TIQUE
Subir des dégâts	Barre de vie descend	/	Tremblement	Bruit de froissement mécanique	Vibration
Rentrer fortement dans un mur	Effet d'étincelles et lumière à l'endroit de l'impact	/	/	Son de choc métallique qui évolue en fonction de la violence	/
Rentrer très fortement dans un mur	Effet de murs cassé qui se propagent et disparaissent	/	/	Son de choc métallique qui évolue en fonction de la violence	/
Rentrer dans un faux mur	Ondes jaunes	Mur se "dissout"		Musique change vers un thème mystérieux et son de disparition brumeux	
Rentrer dans un mur à swarmers	Effet de cercle qui grossit	Particules qui se tortillent en s'éloignant du point d'impact	/	Son de choc avec une cloche en plus	/
Swarmers deviennent agressifs	Swarmers se déplacent plus rapidement, et vers le joueur	Dents de swarmers se dégainent	/	Son de déplacements des swarmers spécial	
Se faire attaquer par des swarmers	Petites particules locales et swarmers attachés au player	/	/	Bruit de froissement de métal	Petites vibrations

Game Design .

Action	EFFET VISUEL	Animation	Caméra	Effet Sonore	Effet Hap- TIQUE
Rentrer dans la zone de navigation d'un worm	/	/	/	Musique spécial de tension	/
Se faire repérer et pourchasser par un worm	Worm te poursuit	/	Caméra atteint une position fixe	Musique spécial de poursuite	Moyenne vibration
Se faire mordre par un worm		Ouverture, vibration puis fermeture de la bouche du worm	Frame Stop avec une inversion des couleurs et un passage au noir et blanc pendant moins d'une seconde	Son de morsure métallique et arrêt momentané de la musique, puis reprise	Vibration
Worm rentre dans un mur	Particule de mur qui se déchiquette	/	/	Son de mur qui se fait détruire	Vibration
Semer le worm	/		/	Retour a la musique d'exploration avec instrument piano et crie du worm	
Se faire repérer par une tourelle radiale	Lumière s'allume et suit le joueur	Tourelle se déploie	/	/	/
Se faire repérer par une tourelle sentinelle	Lumière suit le joueur	Tourelle se déploie	/	/	

77

Астіон	Effet Visuel	Animation	Caméra	Effet Sonore	EFFET HAP- TIQUE
Tir par une tourelle	Clou propulsé en direction du joueur	Animation de tir de la tourelle	/	Bruit de tir méchanique	/
Se prendre un tir de tourelle	Zone d'impact sur le joeur	/	Tremblement	Bruit d'impact	Vibration
Perdre 60% de la barre de vie de components ou du vaisseau core	Particules de fumée sur les component blessés	/	/	/	/
Perdre un component	Disparition du component	Particules de destruction	Tremblement	Bruit d'explosion métallique	Vibration
Mourir	Ecran noir avec texte faisant référence à la cause de la mort	/	/	Bruit d'explosion spécial	/
Rentrer dans un espace étroit	/	/	Zoom de caméra	/	/
Rentrer dans un espace vaste	/	1	Dézoom de caméra	1	/

Game Design ___

Астіон	Effet Visuel	Animation	Caméra	Effet Sonore	Effet Hap- TIQUE
Rentrer dans un nouvel archipel	Titre de l'archipel apparaît un moment	Changement d'opacité du texte, plus lent à la première arrivée (8 secondes), puis plus court (3 secondes)		Musique d'introduction de l'archipel suivi de musique d'exploration de l'archipel	
Afficher la map	Map apparait	Map descend du haut	/	/	/
Activer la grille	Grille apparaît autour du joueur	Opacité de la grille augmente	Zoome sur le joueur	Son d'activation du mode et stop de musique. Un son d'ambiance débute	1
Passer en mode Inside	Apparition du curseur	/	Zoome + sur le joueur	/	/
Passer en mode Outside	Disparition du curseur	/	Dézoome	/	/
Déplacer le curseur (Inside)	Le curseur se déplace	Rapide scale du curseur à chaque mouvement	/	Son d'UI	/
Détacher component (Inside)	Le component n'est plus allumé et est fortement éloigné du joueur		/	Son de détachement métallique	Petite vibration

Action	EFFET VISUEL	Animation	Caméra	EFFET SONORE	EFFET HAP- TIQUE
Sélectionner component suivant/ précédent (Outside)	Change le component apparaissant dans la sélection de la grille	/	/	Son de sélection sourd	/
Tourner component	Tourne le component à 90° ou -90°	/	/	Son de rotation	/
Déplacer component	Déplace le component sélectionné dans la grille	/	/	Son numérique aigu	/
Déplacer component hors de contact du vaisseau (non posable)	Contour de sélection rouge autour du component	Animation de barres rouges par dessus le component	/	Son d'erreur	/
Déplacer component dans vaisseau ou hors de grille (non posable	Carrés rouges sur les cases empêchant la pose	/			
Poser Component	Le component n'a plus sont contour de sélection	/	/	Son de pose (scale sur la taille du vaisseau)	Vibration

Game Design _

Астіон	Effet Visuel	Animation	Caméra	Effet Sonore	Effet Hap- TIQUE
Quitter mode assemblage	La grille d'assemblage disparaît	Opacité de la grille diminue jusqu'à zéro	Dézoome	Son de désactivation du mode et stop du son d'ambiance. La musique d'exploration reprend	
Carcasse	Particules derrières le component	/	/	/	/
Quête sélectionnée	Sprite de sélection autour de la ville où aller	/	/	/	/
Ville de quête	/	Effet de cercle qui grandit depuis le centre de la ville de la quête	/	/	/
Rentrer dans une ville	Transition vers scène de ville	Porte s'ouvrent	UI ville apparaît	Musique spécial de ville	/
Accepter une quête dans une ville	Popup qui informe de l'acceptation de la quête	HUD de quête qui apparaît quand on sort de la ville	/	Son d'acceptation de quête	/
Appuyer sur un bouton retour dans la ville	Changement de menu	/	/	Son d'annulation	/

Action	EFFET VISUEL	Animation	Caméra	Effet Sonore	EFFET HAP- TIQUE
Acheter un composant	/	/	/	Son de choque métalique	/
Acheter une carte	/	/	/	Son de froissement de papier	/

Game Design _

Rewards

REWARD TYPE	Тімер	Transient	PERMANENT	Consumable
Access	/	/	Dans le tutoriel, accrocher un component pour ouvrir la porte	/
FACILITY	Accélération en sortie de virage	Carcasse	Débloquer les éléments de map	
Sustenance	Boost de vitesse en sortie de virage	Nouveaux components	/	/
GLORY	1	1	1	1
Sensory Feedback	Visuels d'accélération des réacteurs	Changement de couleurs par zone	Loupiottes/effet de confort	/
Positive Feedback	Capacités gagnées à la pose d'un component Texte d'arrivée dans une zone		Compteur et barre de vitesse	/

Map de notre monde, divisée en 4 archipels compacts

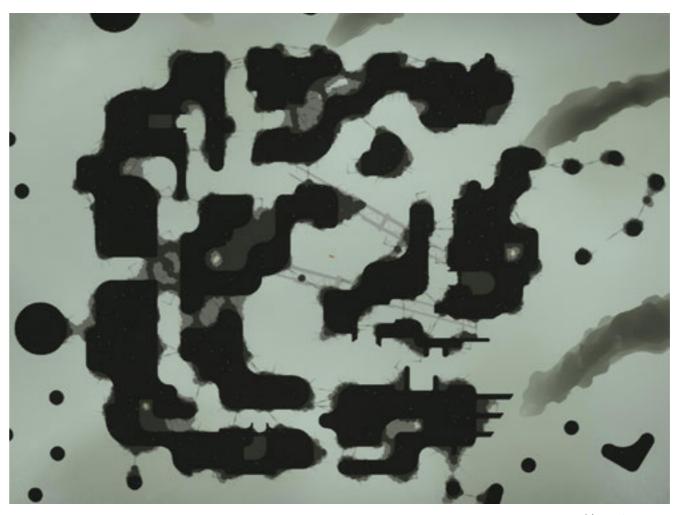
Level Design

LEVEL DESIGN MACRO

Nous souhaitions faire plusieurs régions entre lesquels le joueur devrait se déplacer pour effectuer des livraisons.

Il y a 4 régions différentes : Old Herringen, The Maw, Chittering Construct, Distant Remnants.

OLD HERRINGEN



Old Herringen

C'est la première région du jeu, qui contient 3 points de livraison.

Cette région est la plus grande de la carte, remplie de longs couloirs et de grands virages pour permettre au joueur d'apprendre et d'expérimenter avec les contrôles du jeu.

Elle sert de région principale d'early game.

Les seuls ennemis présents sont des tourelles, en petite quantité.

84 - Cluster 71

Level Design

THE MAW

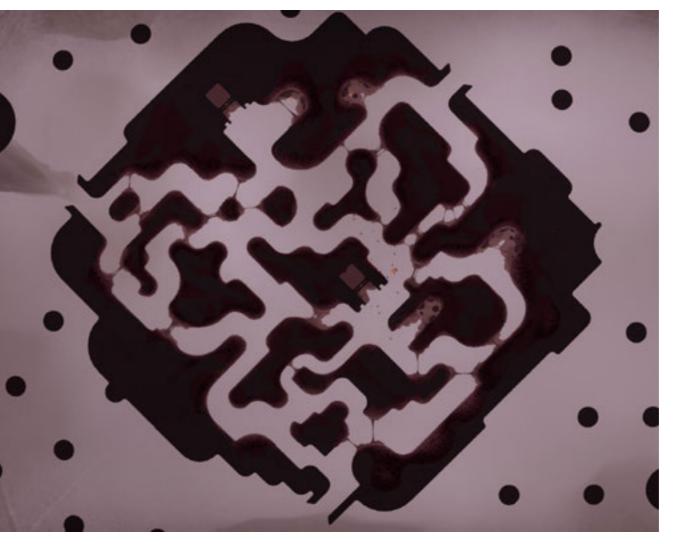


The Maw

C'est la deuxième région rencontrée par le joueur, elle est plus petite qu'Old Herringen et contient 2 points de livraison.

Cette région met en valeur les tourelles, ennemi introduit dans Old Herringen. Il y a deux types de tourelles, les tourelles Radiales, qui tirent sur le joueur dès qu'il entre dans leurs champs de détection, et les tourelles Sentry qui tirent dès que le joueur entre dans leurs cônes de vision.

CHITTERING CONSTRUCT



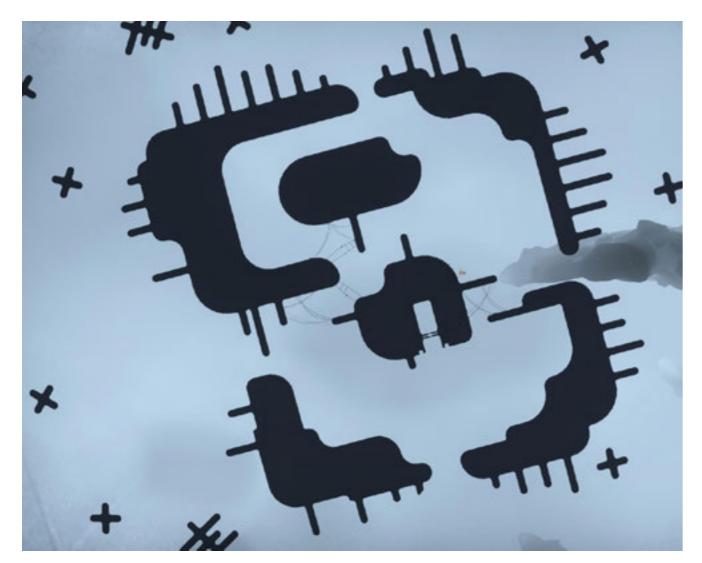
Chittering Construct

Il s'agit de la troisième région que rencontre le joueur. Elle contient 2 points de livraisons.

Grâce à son architecture labyrinthique, cette région met en valeur les Swarmers. Ces ennemis fonctionnent en meute et foncent sur le joueur dès qu'il rentre trop violemment dans un mur de Swarmer. La structure de la région pousse le joueur à se perdre dans les couloirs, et donc à passer plus de temps dans les couloirs, ce qui augmente les chances que le joueur touche un mur.

86 - Cluster 71

DISTANT REMNANTS



C'est la dernière région du jeu, elle est très petite et propose peu de couloirs et d'espace permettant au joueur de se cacher.

De plus, il n'y a qu'un seul point de livraison. Cet archipel, très ouvert, permet aux worms (notre dernier type d'agent hostiles) de naviguer librement autour et à l'intérieur de la région. Son tracé en forme de huit avec des croisements étroits est pensé pour permettre des coursepoursuites entre les worms et le joueur où ce dernier pourrait tenter de prendre le risque d'emprunter un virage difficile à négocier pour mettre plus de distance entre lui et ses poursuivants.

Position des archipels

Pour déterminer la distance séparant les différentes régions nous avons calculé la vitesse de déplacement d'un vaisseau moyen.



Nous avons déterminé que le vaisseau se déplace à environ 168 km/h (46m/s). A partir de ce résultat nous avons calculé la distance entre les régions en fonction de la durée souhaitée.

Nous sommes donc parvenus au résultat ci-contre :

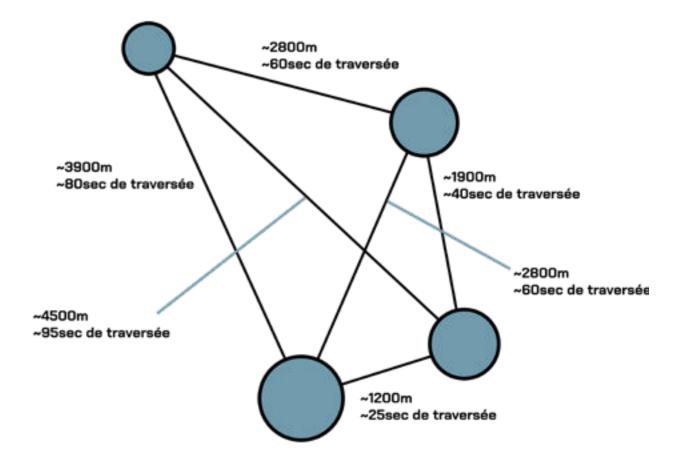
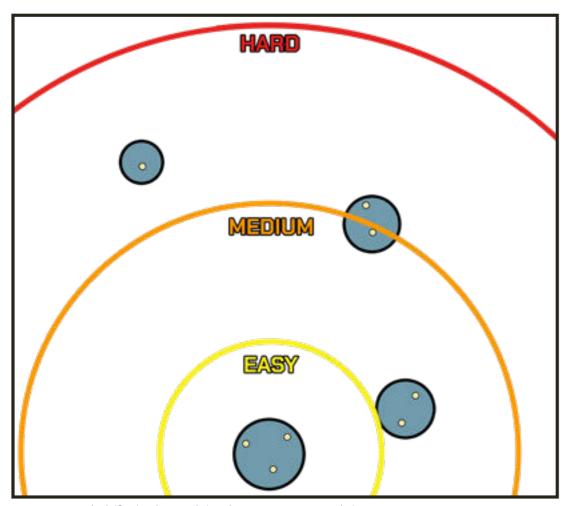


Schéma prématuré de notre structure du LD, avec distances et durées de traversée à vol d'oiseau

Distant Remnants

Level Design



Enfin, pour déterminer la progression de la difficulté, nous avons défini la première région comme étant la plus simple, et plus le joueur s'éloigne plus la difficulté augmente (la difficulté se traduit par l'augmentation du nombre d'agents hostiles, la complexité des parcours et le nombre d'endroit où se «cacher»).

Heat Map de difficulté d'accéssibilité des régions, et points de livraison par région

Level Design

ENEMY DESIGN

" or A. Back, "The DESCRIPTION OF THE

Nous avons conçu nos agents en ayant deux directives principales. La première était de leur donner une «double nature» où l'on aurait une approche d'abord étrange assez passive suivie d'une sensation de choc et de répulsion lors de leur phase agressive. La seconde était de concevoir leur comportement autour des challenges de déplacement que l'on voulait qu'ils induisent au joueur.



Tourelle Radiale

Nous voulions inciter le joueur à rester à la fois en mouvement, et à appréhender «comment entrer ici».

Le joueur qui rentre dans un couloir ou un passage où sont présentes ces tourelles doit alors réagir vite et continuer de se déplacer, au risque de se faire toucher par les tirs.

Ces couloirs sont parfois facilement praticables, la solution étant alors de rapidement passer en ligne, mais d'autres sont plus escarpés et nécessitent alors d'abord une évaluation du joueur de «suisje capable de passer par là?», et s'il choisit de ne pas faire demi-tour, être capable de traverser la zone.



Joueur attaqué par une tourelle radiale, The Maw

TOURELLE SENTRY

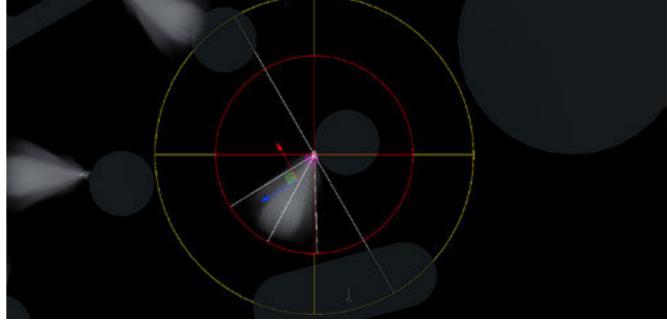
Un challenge que nous voulions créer via les tourelles mais que les radiales ne faisait pas était une capacité «d'esquiver» mais pas dans le contexte d'action d'un joueur déjà repéré, comme une légère phase d'infiltration centrée sur du dosage et du timing.

Contrairement aux radiales qui se déclenchent quand le joueur est à proximités, les sentry ont ainsi un cône de détection qui balaye une zone périodiquement.

Pour éviter que les vaisseaux lourds ne soient trop pénalisés, leur balayage est lent et leur portée de détection courte; mais ils sont plus violents.

Aussi, un vaisseau «rapide» peut se faire punir plus facilement; les tirs des sentry sont rendus plus rapides, fréquents, et font plus de dégâts.





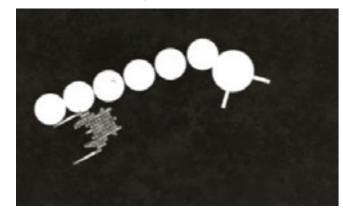
Gizmo de tourelle sentry

Worm

Le worm a été conçu pour être une forme de danger très pesante dans l'espace entre les régions, renforçant l'impression de confort d'être entre des murs.

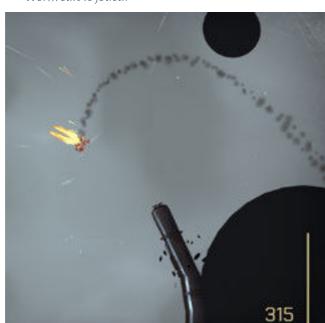
Il nous a permis de créer des situations de course-poursuites prolongées, incitant le player à le semer en se faufilant dans des endroits exigus; les champs d'astéroïdes de l'espace, mais aussi et surtout les murs des régions elles-mêmes.

Dans sa double nature, nous le voulions particulièrement marquant; là où notre première version ressemblait plus à une chenille mâchant en bouche, ce qui rendait son hostilité trop «évidente».

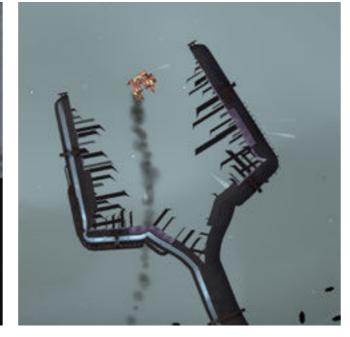


Ancien Worm, au 1er Semestre

Worm suit le joueur



Nous lui avons alors donné une forme beaucoup plus intrigante, lisse, qui ne laisse pas penser au joueur qu'une énorme mâchoire est prête à l'engloutir, jusqu'au moment où il la voit. Worm se prépare à mordre le joueur



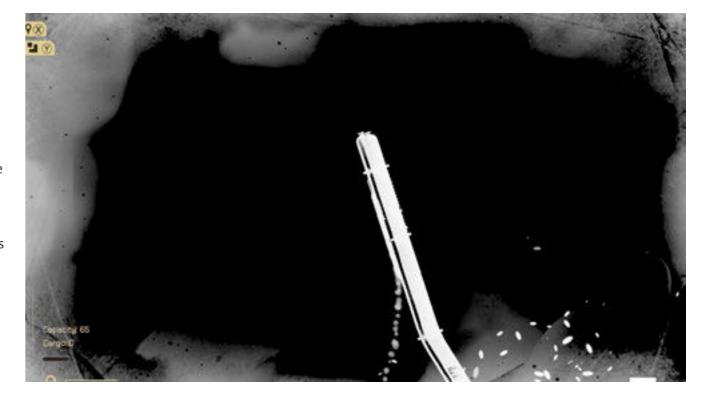
Pour que la morsure ne pénalise pas trop le joueur et ne l'enfonce dans un effet de boule de neige négatif, nous avons ajouté à la fois un temps de rechargement à sa morsure, ainsi qu'une projection appliqué au joueur; lui permettant de repartir se faufiler entre les éléments du décors pour semer le worm. Dans la même optique, le worm effectue également un mouvement de recul avant de repartir à l'assaut.

Le comportement du worm est basé sur un système d'états :

- Son état de base, le mode «wander» est de se déplacer entre des points aléatoires dans une zone de patrouille large mais délimitée. Si le joueur entre dans un certain rayon autour de sa tête ou entre en collision avec lui (son rayon de détection ne va pas jusqu'au bout de la queue), le worm passe en mode chasse.
- En mode chasse, le worm va poursuivre le joueur et tenter de se rapprocher suffisamment pour le mordre. Dans cet état, le worm accélère tant qu'il se déplace en ligne droite et a une ligne de vue directe sur le joueur. En revanche, il ralentit rapidement dans les virages. Si le joueur s'éloigne suffisamment, ou si le worm perd la ligne de vue pendant plusieurs secondes, il va émettre un son distinctif et passer en mode «stalk». Le but est d'encourager le joueur à effectuer des virages efficaces (un des piliers de notre gameplay de mouvement) entre des obstacles et à ramener son poursuivant jusqu'à une des régions du jeu pour pouvoir le semer plus facilement.
- En mode stalk, le worm n'abandonne pas tout de suite la poursuite du joueur, il va aller jusqu'à l'endroit où il a perdu le joueur de vue et patrouiller dans cette zone pendant quelques temps, à la fin de cette durée, s'il n'a pas trouvé le joueur, il repasse en mode wander et retourne dans sa zone de patrouille initiale.

• Lorsque le worm est en mode attaque, il ouvre ses mâchoires et si le joueur est toujours à portée après une courte durée, il mord. Sinon il les referme plus lentement. Après avoir mordu, il se déplace vers la direction opposée au joueur pendant une courte durée, pour éviter que celui-ci ne se retrouve coincé entre le worm et un mur par exemple.

Si la morsure touche sa cible, il y a un court effet de hitstop et l'image passe en noir et blanc, très contrasté et en négatif pendant cette durée. Le but est de renforcer la violence et le spectacle de l'attaque.

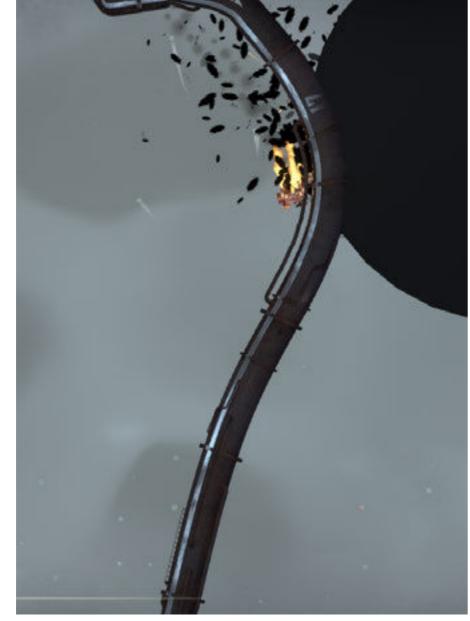


94 - Cluster 71

Level Design

Pour que le worm paraisse plus menaçant, nous avons pris le parti de ne pas lui faire suivre une trajectoire parfaitement exacte : il va racler contre les murs, entrer en collision avec eux, pendant qu'il poursuit le joueur. Des particules et sons inquiétants accompagnent ces chocs. Nous voulions donner l'impression d'une créature agressive et désespérée qui est purement concentrée sur sa proie, au mépris de sa propre sécurité et des obstacles autour.

Le worm va aussi chercher à mordre les components qui ne sont pas accrochés au joueur. Ce qui lui donne un air un peu plus vivant, tout en donnant au joueur la possibilité de le distraire en détachant des parties de son vaisseau.



SWARMERS

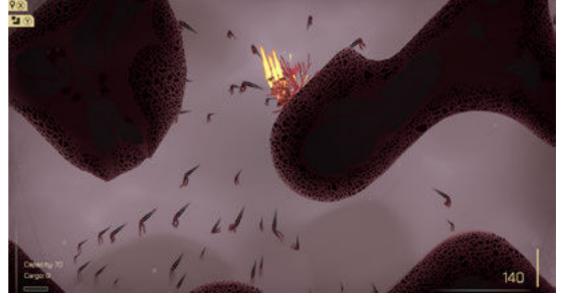
Nos collisions étaient négatives, mais d'une seule manière et directe; nous trouvions intéressant de mettre une conséquence différée à une collision non maîtrisée, ce qui a donné Chittering Construct et ses endémiques Swarmers.

Normalement passifs, ils arpentent la structure labyrinthique et difficilement arpentable de Chittering Construct.

Le joueur qui commet une erreur et se heurte violemment à un mur est alors surpris lorsque ces éléments qui semblent être environnementaux deviennent subitement agressifs et le pourchassent avec leurs dents insoupçonnées.

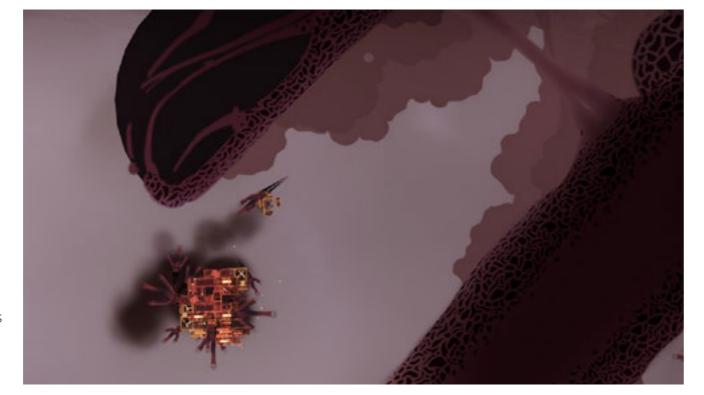
Le joueur qui essaye alors de fuir et détacher les swarmers qui le chassent va avoir d'autant plus de risque de se heurter à nouveau à un mur, provoquant d'autres swarmers qui s'ajoutent à la meute, provoquant un effet boule de neige où c'est la réaction exagérée du joueur qui est l'erreur





Pour mettre l'emphase sur le côté labyrinthique, les swarmers ne détruisent pas les components mais les arrachent et les emmènent; cela peut induire l'idée au joueur de «ils les emmènent quelque part» s'il le voit, et de chercher ce quelque part. Le but est de créer une forme de quête secondaire. Si le joueur par exemple le joueur passe par Chittering Construct pour effectuer

Aussi, cet effet de non-destruction permet de pallier à l'effet boule de neige punitif car il ne perd concrètement pas ses components, sauf si il panique et les casse en rentrant dans un mur en fuyant les swarmers.



Quest Design

Pour les quêtes de notre jeu, nous avons imaginé 4 types de quêtes. Mais nous avons décidé d'en garder seulement 2. Lorsque le joueur se trouve dans un archipel, il a le choix entre 3 quêtes qui lui sont proposées. Chaque quête est définie de manière procédurale.

Pour chacune des quêtes nous avons défini les paramètres exploitables pour faire varier la difficulté.

Quêtes de Livraison

Principe: Le joueur part d'un point de départ (point de livraison sur lequel il se trouve) et doit livrer une certaine quantité de ressources à sa destination (un autre point de livraison).

Paramètres permettant de faire varier la difficulté :

- · Emplacement du point de départ
- Contrainte de temps
- · Distance entre les points de livraison
- Quantité de ressources à transporter
- Points de livraisons dans le même ou dans un autre archipel

Quête de récupération

Principe: Le joueur doit se rendre des un point précis pour récupérer des components spéciaux et l'amener à un point de livraison.

Paramètres permettant de faire varier sa difficulté :

- Emplacement du point de départ
- · Nombres de points de récupération
- Emplacement du/des point(s) de récupération
- Distance entre ces points
- Contrainte de temps



Quête de course poursuite

Principe : Le joueur doit poursuivre un élément et le rattraper.

Paramètre permettant de faire varier la difficulté :

- · Emplacement du point de départ
- · Fonctionnement de la cible
- · Distance limite de poursuite
- Endroit de la poursuite (dans un archipel, dans le vide, entre les ennemis...)

Cette quête n'a pas été conservée car nous avions besoin de créer de nouveaux systèmes pour la rendre utilisable, et nous avons préféré nous concentrer sur les systèmes déjà existants.

QUÊTE D'APPÂT

Principe : Le joueur doit attirer un monstre dans un piège.

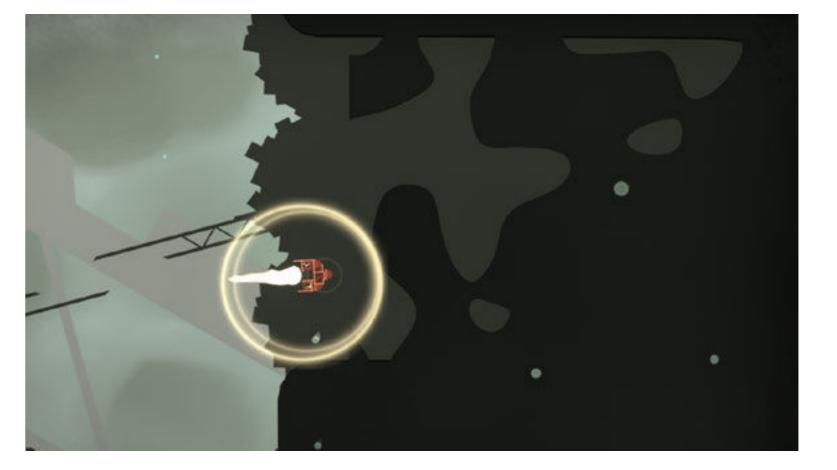
Paramètres permettant de faire varier la difficulté de la quête :

- · Point de départ
- Type de cible
- Distance à parcourir
- Point d'arrivée

Comme pour la quête de course poursuite, cette quête n'a pas été conservée car nous avions besoin de créer de nouveaux systèmes pour la rendre utilisable, et nous avons préféré nous concentrer sur les systèmes déjà existants. De plus, hormis le worm nous n'avions aucun ennemis à utiliser pour cette quête.

EXPLORATION

Pour inciter le joueur à explorer, se déplacer dans l'espace de jeu, nous avons créé plusieurs ingrédients de LD donnant au joueur l'envie d'explorer et de retourner sur ses pas.



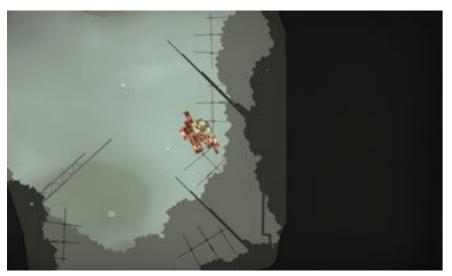
100 - Cluster 71

Level Design

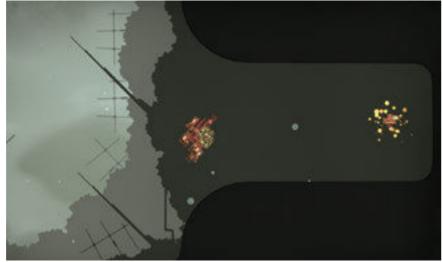
101

Murs intangible

Ces murs recouvrent des passages dissimulés, lorsque le joueur entre dans le trigger d'un mur intangible, celui-ci disparaît pour laisser apparaître le chemin caché.



Certains de ces murs cachent des raccourcis, mais la grande majorité permet de trouver des components cachés.





CARCASSES

Les carcasses sont des components générés aléatoirement dans l'espace de jeu. Les carcasses sont disposées à des endroits précis, mais les components y sont générés de manière aléatoire, on peut y trouver un thruster, un rotator, un cargo, ou même un amplifier.

Les taux d'apparition des carcasses sont variables et sont rechargés à chaque fois que le joueur quitte un archipel, afin de donner envie au joueur d'explorer et de trouver les carcasses dissimulées dans l'espace de jeu.

102 - Cluster 71

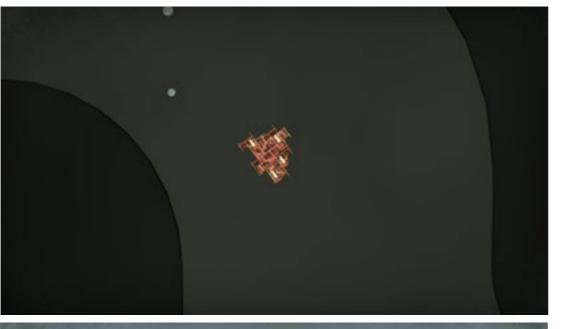
Level Design

TRIGGERS DE CAMERA

Les triggers de caméra sont des zones qui augmentent ou réduisent la taille de la caméra. Faire varier la taille de la caméra permet de faire de la mise en scène environnementale. En réduisant la caméra dans des milieux étroits pousse le joueur a ralentir, tout en lui donnant une meilleure vision sur les obstacles alentour.

À l'inverse, dans des endroits ouverts, nous augmentons la taille de la caméra afin de donner un sentiment de grandeur, en plus de permettre au joueur d'apercevoir des zones potentiellement non explorées.

Ces triggers de caméra permettent aussi de répondre à une problématique que nous avions rencontrée lors de tests utilisateurs. Lorsque les joueurs allaient dans des grands espaces, il pensait que c'était la fin de l'espace de jeu, mais dès lors que nous avions disposé les triggers de caméras, les joueurs ne se posaient plus la question.





104 - Cluster 71

Level Design

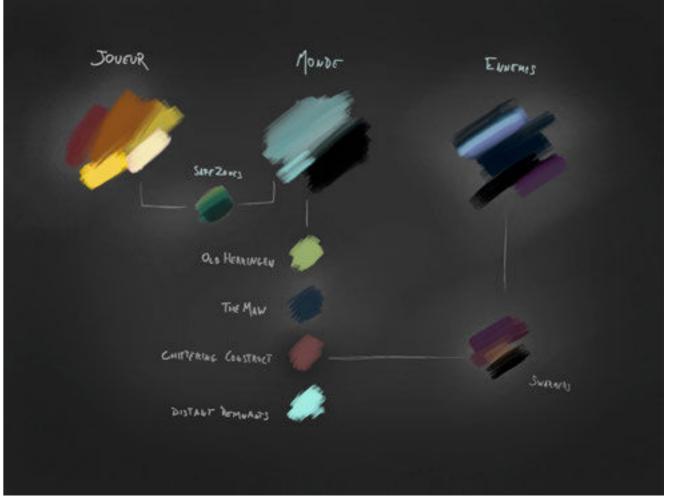
105



INTENTIONS

Le ressenti que nous cherchions à transmettre dans Cluster 71 est celui d'un joueur qui se sent minuscule et insignifiant dans un environnement vaste, imposant, intimidant et mystérieux. Un environnement dangereux également. Mais, à force d'apprentissage des mécaniques et de l'espace, et du raffinement de son avatar, il finit par être capable de naviguer librement dans ce monde hostile et de le surmonter.

Ainsi le jeu prend place dans un espace indéterminé et nuageux, où d'immenses ruines sont en suspension. Il n'est pas exactement clair s'il y a un haut ou un bas et on soupçonne que le Cluster 71, l'espace du jeu, est contenu dans un gigantesque nuage ou tempête. Le joueur lui-même est également plus ou moins en ruine, fait de ferraille rouillée et de vielle machinerie. Et des créatures arpentent le Cluster pour lui rappeler que l'hostilité n'est pas seulement visuelle.



Nuancier du jeu

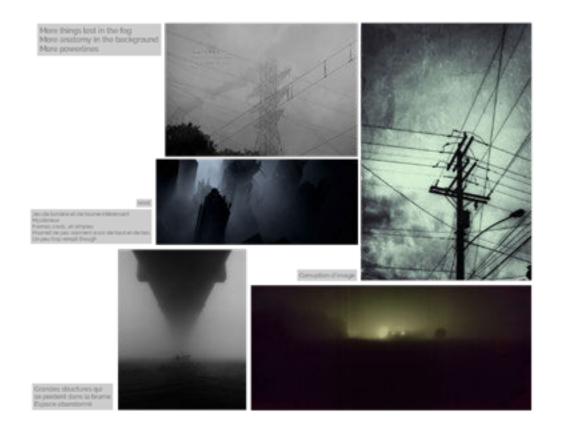
RECHERCHES INITIALES

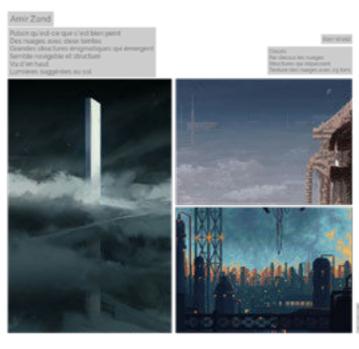
Moodboards

Avant de commencer la production des concepts et des visuels du jeu nous avons effectué une phase de recherches préliminaires pour explorer diverses directions possibles et solidifier nos intentions.

Ces recherches ont avant tout pris la forme d'un vaste moodboard qui ordonnait une variété de références basées sur nos intentions naissantes à cette étape du développement (page de droite).

Sur cette page, une synthèse de ce que nous avons retenu après ces recherches. Nos intentions ont bien entendu continué à évoluer au cours du projet, mais tel fut notre point de départ.





Extrait de notre moodboard de recherches préliminaires. La position d'une image dans l'espace du moodboard est liée au genre d'idées qu'elle représente.

108 - Cluster 71

Visuel

ÉTUDES DE NUAGES

Ayant établi que nous voulions créer un espace nuageux, nous avons aussi utilisé cette phase de recherche pour pratiquer le dessin de nuages en essayant de reproduire et de modifier diverses références.

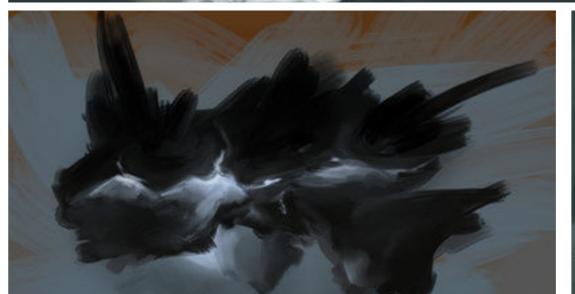
Cette pratique nous a permi d'avoir une meilleure compréhension du graphisme des nuages, pour pouvoir ensuite mieux les styliser dans la suite du développement.







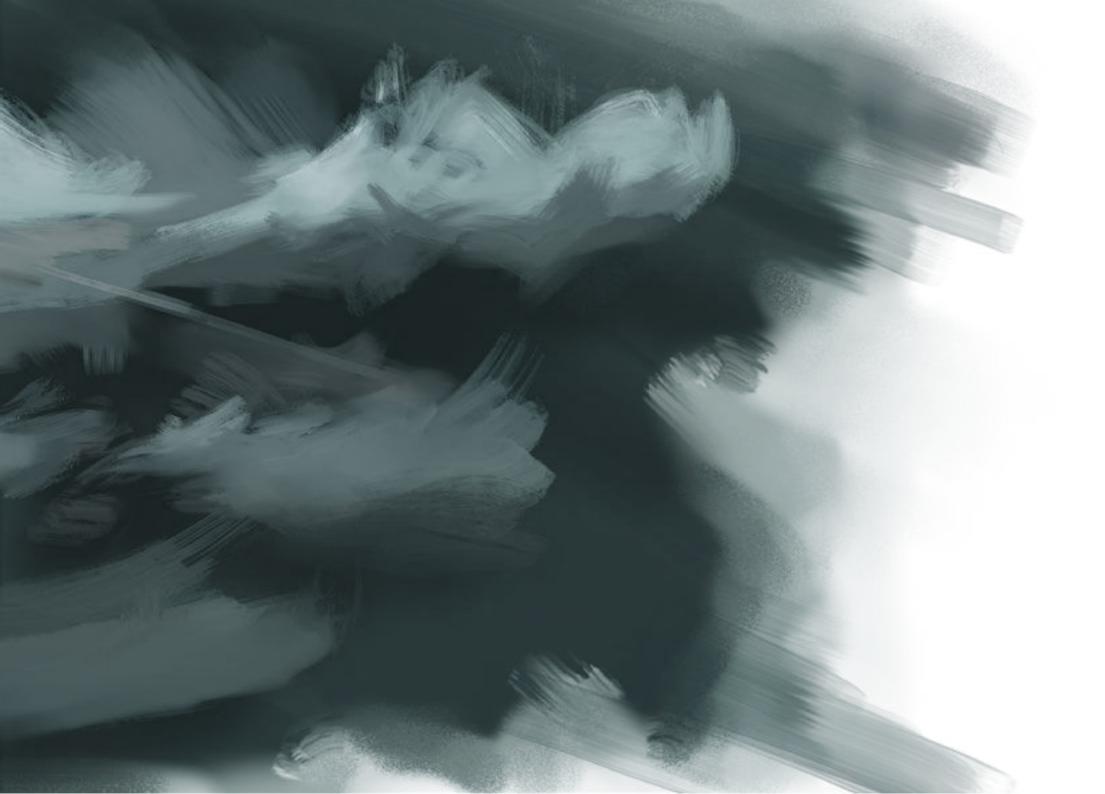






Diverses études graphiques de nuages préliminaires

110 - Cluster 71



ENVIRONNEMENT

L'environnement de Cluster 71 a été conditionné par notre gameplay de déplacement : sa forte inertie et faible friction implique un milieu où l'avatar est en suspension. Nous avons évoqué l'espace et les abysses.

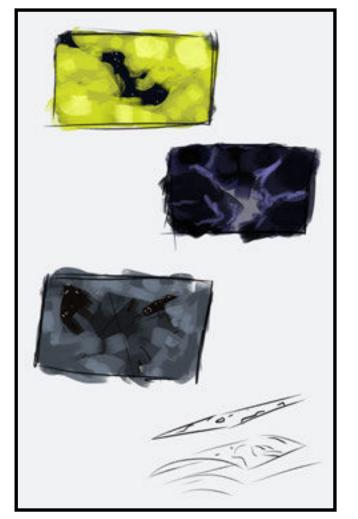
Ces dernières ont été abandonnées car jugées trop difficiles à traiter, même si leur forte aura de mystère était intéressante. Leur influence se sent cependant encore un peu dans le jeu.

Cela dit, l'espace tel qu'il est généralement envisagé ne nous semblait pas parfaitement pertinent non plus. En plus d'être très peu original, il renvoie trop facilement à un idéal de conquête et de grandiose, ce qui n'est pas cohérent avec notre jeu où le joueur est écrasé par son environnement (au moins dans un premier temps). De plus, un contexte trop commun aurait atténué l'impression de mystère et de perte de repères que nous recherchions.

Ainsi nous avons opté pour quelque chose d'un peu différent : un milieu toujours en apesanteur, mais plus proche du nuage et de la tempête.

Il était aussi important pour nous que l'orientation du monde ne soit pas évidente. Ainsi, le sol n'est jamais visible dans le jeu. Ce choix contribue à l'étrangeté du contexte. Nous avons également donné aussi peu d'indications d'échelle que possible dans la même optique.

Croquis exploratoires très anciens, basés exclusivement sur le concept d' «espace avec des nuages».

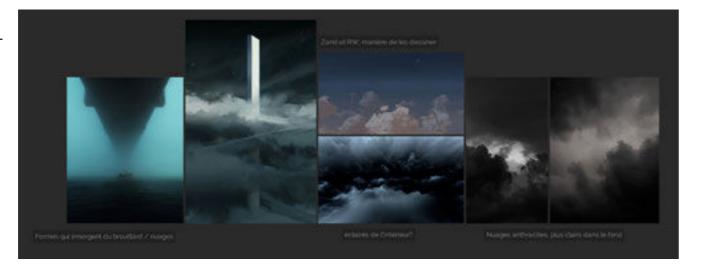


Nuages

Comme nous l'avons évoqué, les nuages sont une partie importante de l'identité visuelle de notre jeu. Leur design a été en grande partie établi par le concept art d'ambiance en bas à droite de cette page. C'est avec lui qu'a émergé l'idée de ces nuages aux formes étrangement denses, solides et rectilignes, qui rend leur nature exacte légèrement incertaine, tirant presque sur le minéral. Ce traitement permet d'allier l'aspect flottant et mystérieux à l'apparence imposante et intimidante que nous recherchions pour l'environnement.

Nous avons ensuite fait une phase de pré-production sur ces nuages pour trouver des moyens de les réaliser techniquement dans le jeu. Nous avons finalement mis au point un shader qui utilise du vertex displacement et des noise maps custom pour créer ces formes particulières et les animer

(voir page de droite)



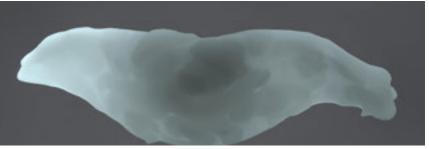


Ci-contre, le premier résultat convaincant de nos tentatives de reproduction des nuages du concept art d'ambiances. Ci-dessous, la version finale, présente dans le jeu. Cette dernière est plus optimisée et a suivi l'évolution de nos intentions au cours du projet.

En effet, plus tôt dans le développement, nous envisagions quelque chose de plus sombre avec beaucoup moins d'espoir, menant donc à ces nuages plus noirs et menaçants. Notre monde maintenant est froid et étrange, mais surmontable. De plus, des nuages trop sombres se confondaient avec les éléments solides du level design.









Visuel

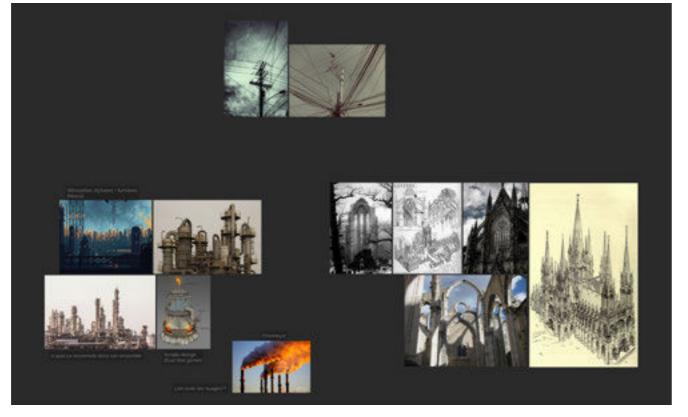
STRUCTURES

Autre que les nuages, notre environnement contient des structures solides définies par le level design. Cet aspect du jeu a connu un changement de direction en cours de route.

CONCEPT ABANDONNÉ

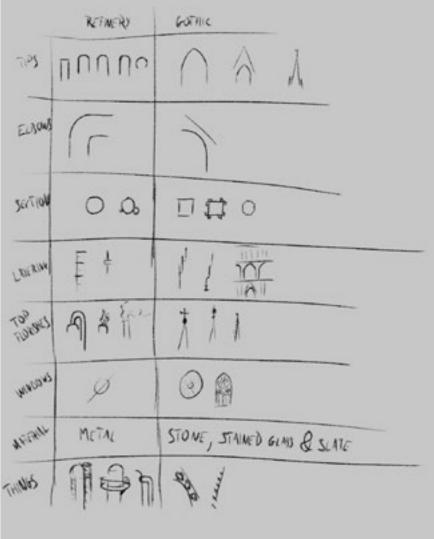
En effet, nous avions d'abord envisagé de représenter ces structures par des ruines architecturales, formées d'un mélange de deux esthétiques : des raffineries pétrochimiques, et de l'architecture gothique. Cet idée a été abandonnée bien qu'intéressante car elle demandait trop de travail, à la fois en terme de conception et de production. D'une part, bien que le concept soit intéressant, obtenir des designs vraiment satisfaisants était difficile, surtout si leur forme devait suivre le level design. Et d'autre part, nous voulions modéliser des structures pour tous nos éléments de LD. Ce n'était pas réalisable.

Cependant, nous n'avons abandonné ce concept qu'assez tard, et avons eu le temps de l'explorer. Nous allons passer en revue le résultat de ces recherches avant de nous intéresser au jeu final.



Pour explorer cette association d'industriel et de gothique, nous avons essayé d'identifier les formes caractéristiques de ces deux esthétiques. Nous voulions en isoler les codes respectifs pour ensuite pouvoir les associer et les combiner de manières nouvelles.

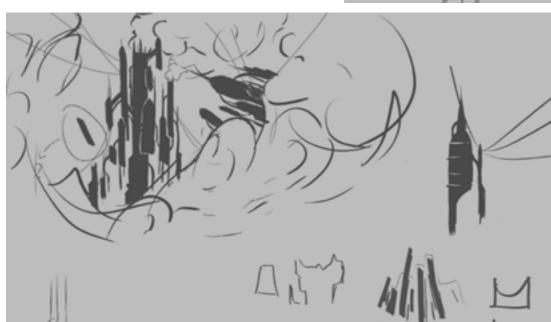


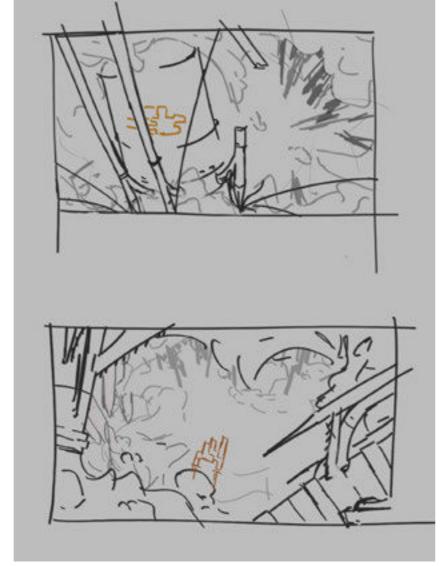


116 - Cluster 71

Sur cette page, les derniers croquis que nous avons produits pour explorer des idées de mises en scène et d'architectures dans cette idée d'architecture que nous avions baptisé «pétro-gothique», puis mis de côté.







ENVIRONNEMENTS FINAUX

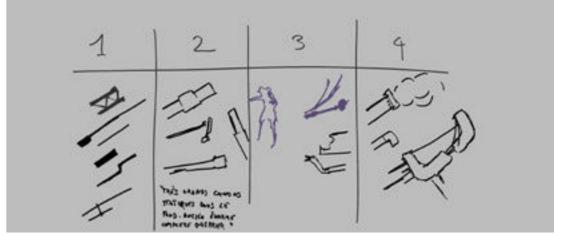
Après notre reboot d'environnement, nous avons décidé de repartir du level design déjà en place, et notamment d'exploiter les murs placés comme silhouettes noires qui se détacheraient clairement du fond devenu plus clair.

Ensuite, pour enrichir ces formes primaires, nous avons préparé des sprites que nous avons positionnés «derrière» le LD, avec des tons plus clairs. Ils permettent de créer des formes plus intéressantes tout en donnant une impression de profondeur. Chaque archipel a ses propres sprites.

A cela s'ajoute un changement de colorimétrie qui s'opère pour renforcer l'identité l'archipel, changement qui s'applique également à l'affichage de la carte. .







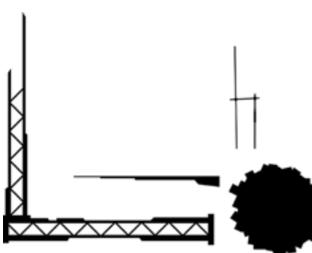
Premiers croquis des formes de sprites de chaque région

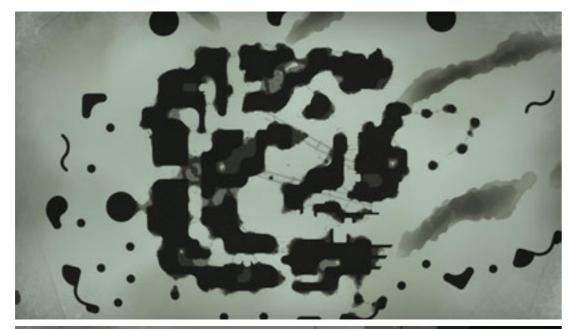
A gauche, une illustration qui a servi de référence pour l'usage de perspective atmosphérique sur des silhouettes pour traduire une profondeur ainsi que pour la forme de certains sprites.

118 - Cluster 71

OLD HERRINGEN

C'est la première zone du jeu et la moins dangereuse, l'objectif de cette région était de créer un environnement semblant être l'héritage d'une époque passée, dans lequel tout serait désorganisé. Elle embrasse d'une certaine manière le joueur, dans ses murs comme dans ses décors qui enveloppent souvent ce qui se trouve dedans. Pour refléter son côté «époque passée», nous avons accentué un aspect de ferraille et décharge, accompagné d'une colorimétrie dans les jaunes et verts pour donner à la fois un ressenti un peu plus confortable et chaleureux que le reste du jeu, mais aussi une impression de lieu ancien, presque en décomposition.

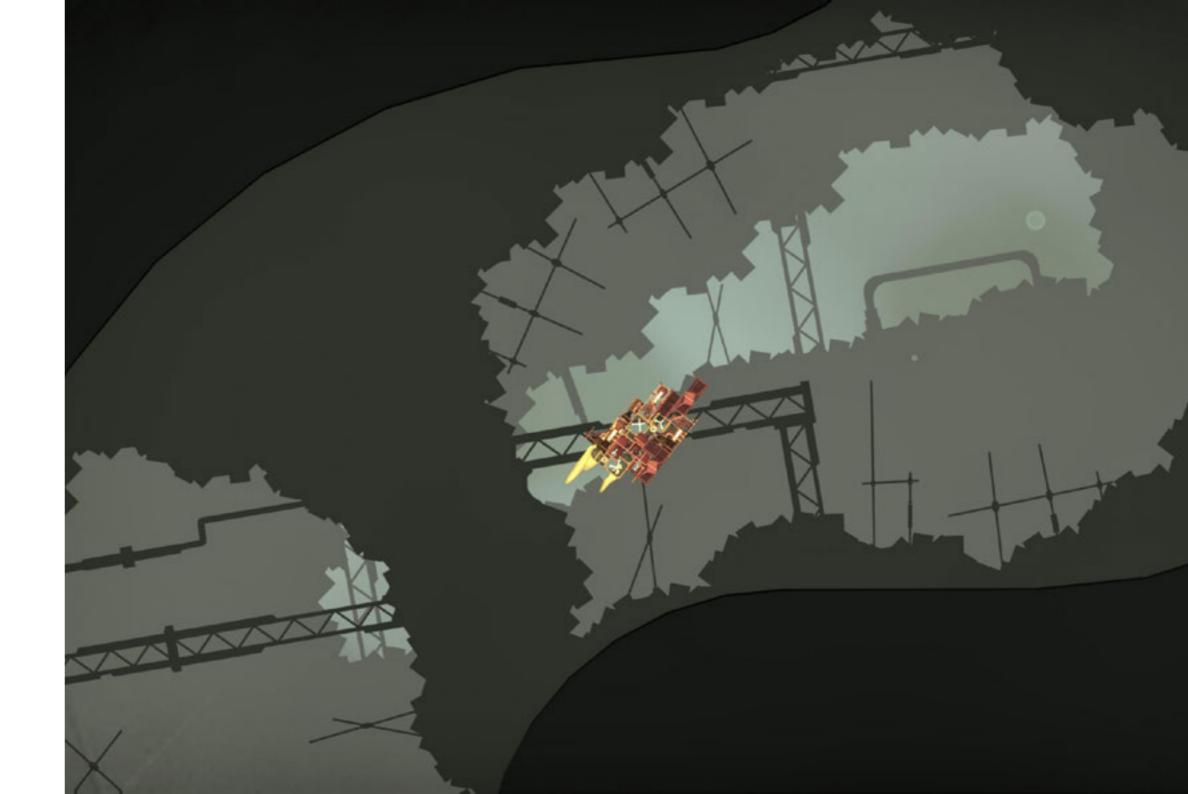






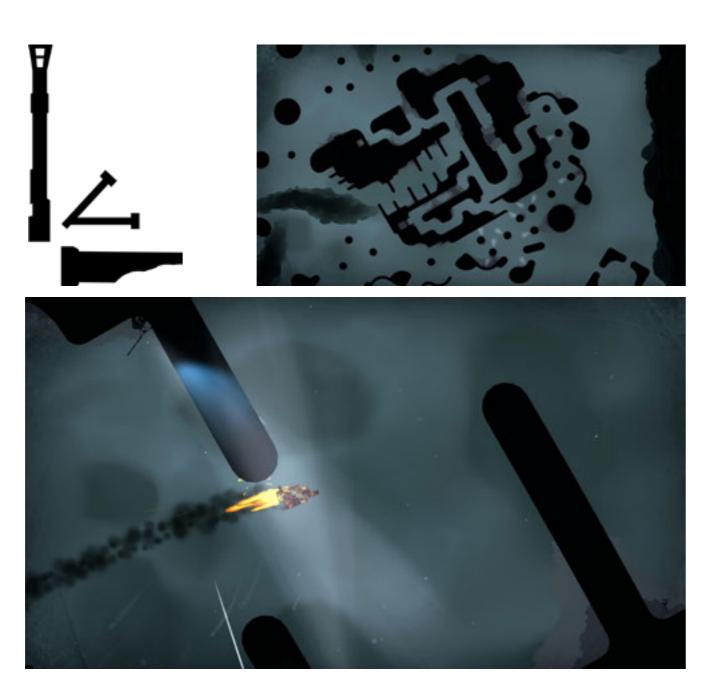






THE MAW

Seconde région que visite le joueur, elle présente des dangers en plus grande densité, sous la forme des tourelles. Celles-ci émettant des lumières, en plus de la sensation d'oppression que nous voulions faire ressortir de The Maw, nous l'avons plongée dans une sorte de nuit. Ses couleurs sombres et contrastées poussent aussi le joueur à être vigilant à ses mouvements et environnements dans lesquels à la fois se cachent des tourelles, mais aussi des décors qui peuvent être trompe l'œil au sein des piles de déchets, plus organisés que dans Old Herringen. Certains de ses décors rappellent aussi des canons pour faire le lien avec les tourelles. On peut imaginer que la région a eu une fonction militaire dans un lointain passé.

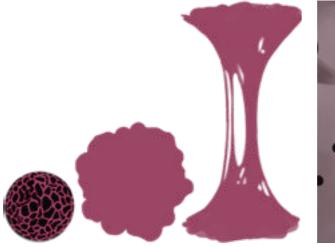


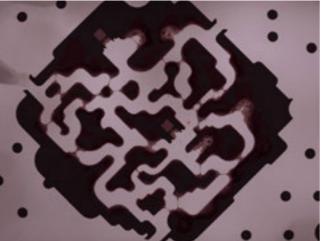


122 - Cluster 71

CHITTERING CONSTRUCT

Région des endémiques swarmers, et abordant une forme labyrinthique, nous voulions que le joueur qui la parcourt ressente viscéralement qu'il n'entre pas que dans une région, mais dans un nid, dans le ventre de la bête. Cette région est organique, croulant de swarmers et de leurs nids en trous, parcourant et dévorant les murs-mêmes. Pour en faire sentir la sorte de corruption, nous avons à la fois opté pour des couleurs très vives et cramoisies, et des formes gluantes, dont les formes rappellent un mélange entre de la chair que l'on dévore et des nids d'amphibiens. Pour les formes, nous avons pris comme exemple les formes de coraux, et de nids étranges (comme on peut en voir dans le jeu Carrion, ou dans le webcomic Un jour à Séoul). Aussi, plus on se rapproche du centre de la région, plus la densité de swarmers est forte; chose que l'on a retranscrit par une densité aussi croissante de nos sprites jusqu'à certaines zone presque entièrement recouverte de l'infection propre à la région.





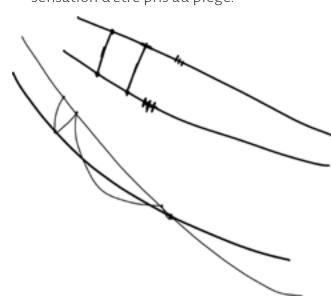




124 - Cluster 71

DISTANT REMNANTS

Cet archipel est le plus éloigné du jeu et est constamment entouré de worm, pour renforcer ce côté "lointain" et cette idée de danger permanent, nous lui avons donné des couleurs très froides, pour donner l'impression que c'est un endroit peu accueillant, dans lequel le joueur n'as pas envie de rester longtemps. Les sprites disposés dans l'espace de jeu sont des câbles, qui semblent désorganisés et emmêlés les uns avec les autres, pour créer une impression d'environnement déchiré en plusieurs morceaux, donnant la sensation d'être pris au piège.









Visuel





VIDE

Là où nous voulions que nos archipels paraissent denses et riches d'appréhension du joueur, nous voulions un espace entre ceux-ci très large, qui retranscrive une vastitude et une sorte de calme en apesanteur. Cet espace est assez neutre en couleurs, tirant vers les gris clairs légèrement bleutés, présentant des silhouettes lisses, disparates, et des formes de nuages très larges avec une fog pour augmenter également la profondeur de cet espace.

Il y a également des particules en suspension dans cet espace. Elles ont à michemin entre des bulles et de la poussière, avec pour but de renforcer l'impression de mystère renvoyée par l'endroit, et d'appuyer son aspect flottant. Ces particules sont présentes sur plusieurs plans, avec un effet de parallaxe, ce qui accentue la profondeur. Enfin, ces particules permettent également de mieux lier l'avatar au reste du monde. Lorsqu'il se déplace, les particules environnantes sont poussées hors de son chemin et laissent des traînées derrière lui.

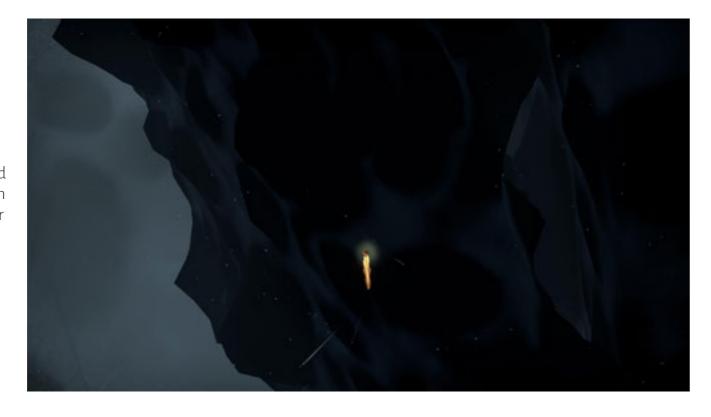


Visuel



Extérieur

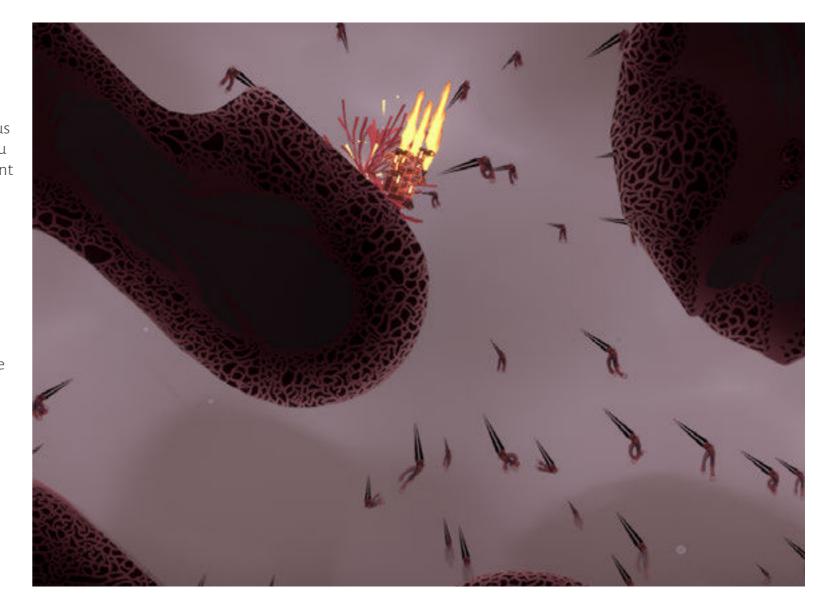
Les bords de notre espace de jeu «de base» sont composés d'immenses nuages à hauteur du joueur, lequel peut essayer de pénétrer mais sera repoussé en arrière. Cependant, si le joueur a le vaisseau nécessaire, ou est en train de réaliser la quête finale, il peut passer au-delà de ces nuages avec suffisamment d'efforts. Quand il sort de l'autre côté, ainsi que pendant son passage des nuages, il nous fallait renvoyer la sensation de violence de l'extérieur de l'espace de jeu, ce que l'on a fait par à la fois un traitement de la colorimétrie très sombre, et par l'ajout d'un dense effet de brouillards qui défile. Cela renforce la sensation de «endroit où nous ne sommes pas censés être mais dans lequel on se démène pour arriver», que nous voulions rendre à cet endroit.



ENNEMIS

Le design de nos ennemis est défini par la traîtrise et la surprise. Tous à leur manière révèlent au dernier moment qu'ils sont autre chose que ce qu'ils semblaient être à travers un brusque changement d'état. Et la plupart d'entre eux prennent une dimension horrifique, suscitant la peur ou la répulsion.

Nous avons estimé que c'était un moyen efficace de renforcer l'impression d'hostilité du monde.

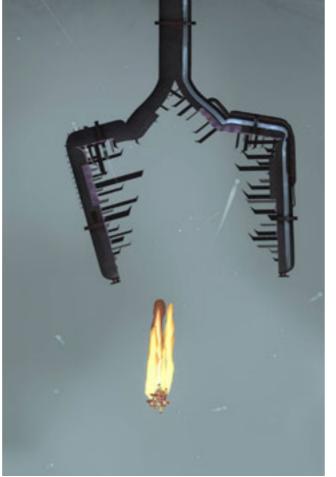


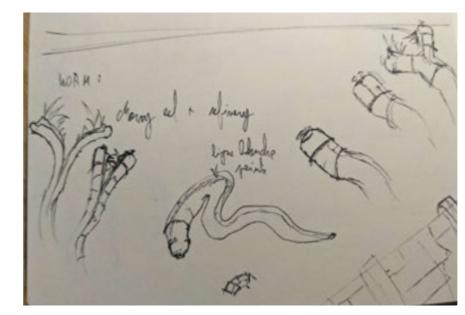


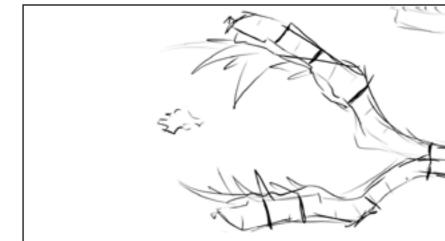
C'est le premier ennemi que nous avons conçu, et c'est avec lui qu'est apparue cette idée de brusque changement d'état vers quelque chose de repoussant.

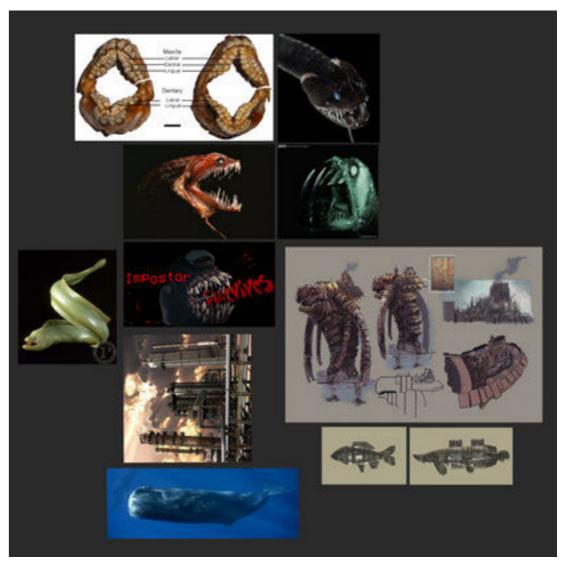
Nous l'avons envisagé comme une vaste créature de nature indéterminée qui, au premier abord, semble être faite des mêmes genres d'objets que les ruines de l'environnement. Le fait qu'un morceau d'environnement se déplace et se torde est censé intriguer le joueur et attirer son attention, le mettant dans les bonnes dispositions pour la surprise qui va suivre.

Le worm va pourchasser le joueur sans relâche et s'il se rapproche suffisamment de lui, il va s'ouvrir en deux pour révéler des machoires disproportionnées bordées de dents immenses.









132 - Cluster 71

Visuel

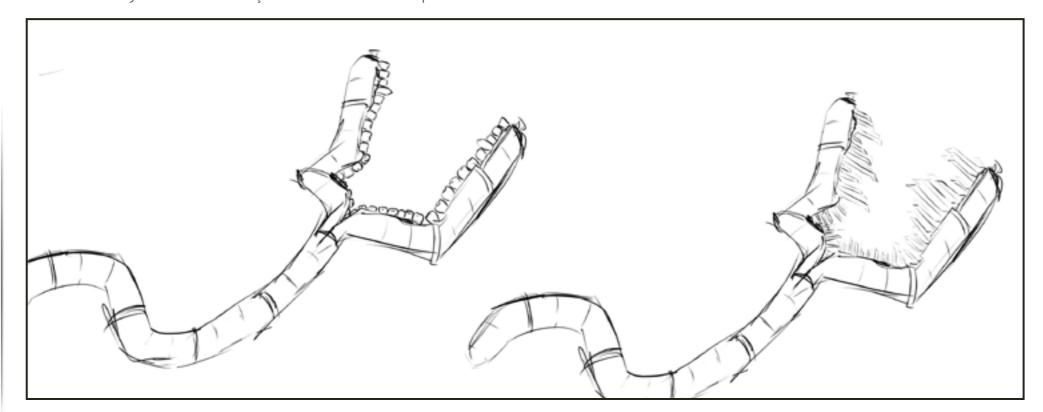
133

Nous avons rencontré un dilemme durant la phase de design de cet ennemi : les dents.

Nous avons hésité entre lui donner des dents presque humaines et des longs crocs de poisson abyssal. Les premières ont l'avantage d'être particulièrement dérangeantes et donc à même d'accentuer le choc et le dégout lorsqu'il ouvre la gueule. Les dents abyssales sont menaçantes d'une manière plus classique et évidente.

Nous avons finalement retenu ces dernières car les dents humaines, bien que profondément répugnantes lors de la première rencontre, allaient perdre leur qualité inattendue sur le long terme. Aussi, si nous avions choisi celles-ci, nous aurions dû reprendre cette thématique de body horror dans le reste du jeu pour qu'elle ne fasse pas tâche.

Page de droite : Concept art final du worm



134 - Cluster 71

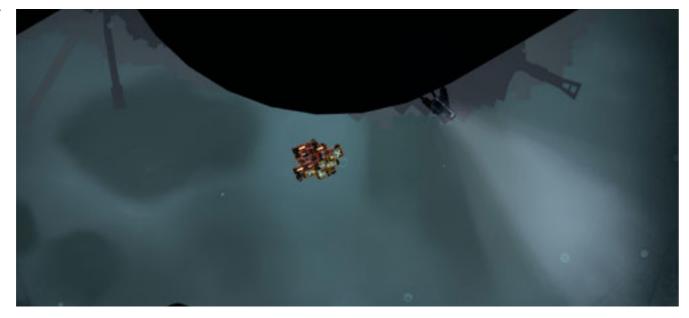
Visuel



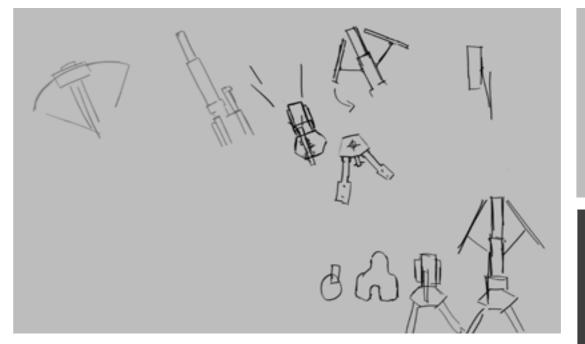
SMALL RIMEISCHAPTER

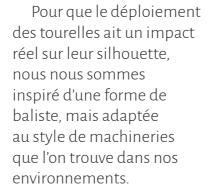
Tourelles

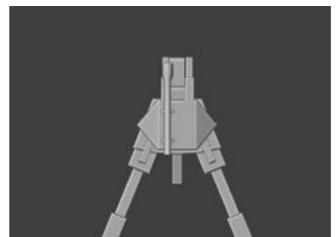
Les tourelles étant notre seul ennemi qui est à la fois strictement mécanique et statique, il a été difficile de leur donner le même genre de changement d'état inattendu qu'aux autres ennemis. Nous avons finalement décidé de les faire se déplier d'un coup lorsqu'elles repèrent le joueur, presque comme si elles bondissaient. Les tourelles de type radial allument également leur spot une fois que le joueur est à portée.

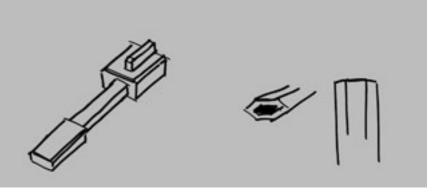


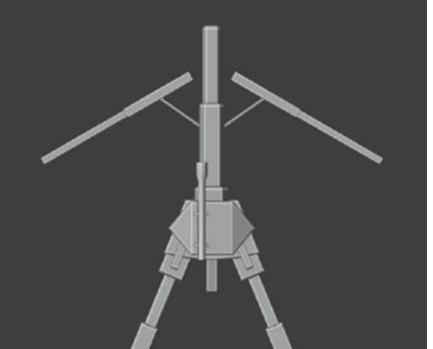












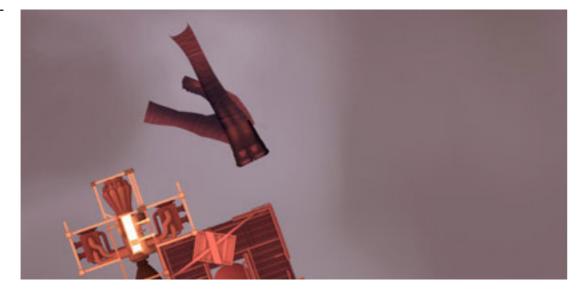
136 - Cluster 71

SWARMERS

Il s'agit de notre troisième et dernier type d'ennemis. C'est un essaim de petites créatures, normalement passives, qui vont foncer sur le joueur pour démanteler son vaisseau si celui-ci entre en collision avec leurs nids qui couvrent les murs de Chittering Construct, leur région de résidence.

Nous voulions que ces créatures inspirent le dégoût et donne une impression de masse grouillante. Ce ressenti est notamment renforcé par leur déplacement par accoups et par les appendices en forme de rubans mous qui terminent leur corps.

Leur traîtise réside dans la manière dont ils vont brusquement passer d'un état où ils èrent lentement de manière inoffensive et un état où ils se retournent tous vers le joueur et le chargent frénétiquement. En même temps, ils déploient deux longues dents qui modifient complètement leur forme pour leur donner l'air beaucoup plus aggressif, et vont tenter d'empaler les components du joueur avec.

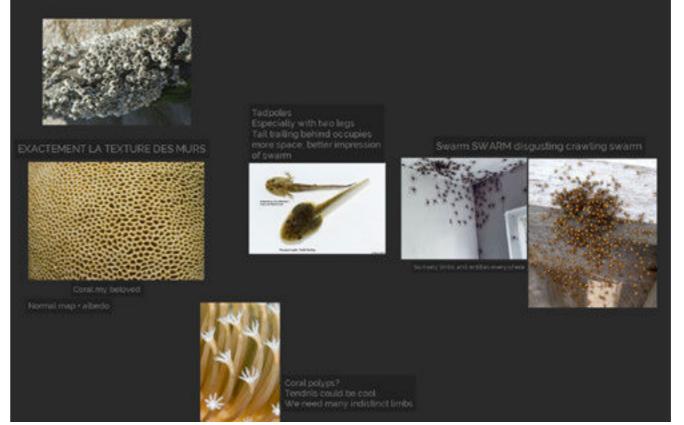




Pour cette impression grouillante, nous avions identifié qu'il était important qu'il y ait beaucoup de détails et de mouvements en simultané.

Aussi, pour donner une identé plus forte à leur région et mettre l'emphase

sur la manière dont ils l'ont colonisée, les swarmers ne reprennent pas le schéma de couleur des autres agents hostiles de notre jeu. Ils sont à la place teintés d'une sorte de magenta sombre. Qui vient appuyer l'ambiance très viciée et organique de la région.

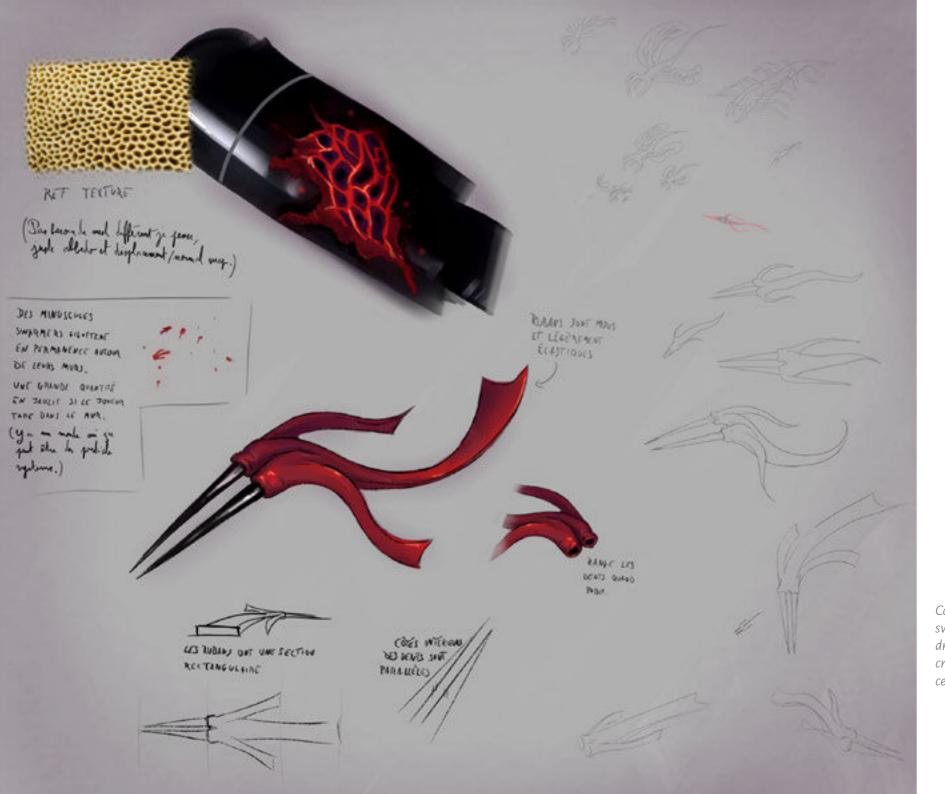




138 - Cluster 71

Visuel

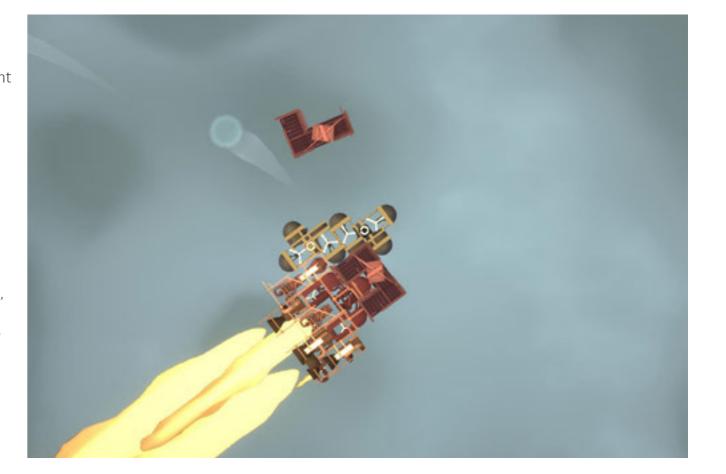
139



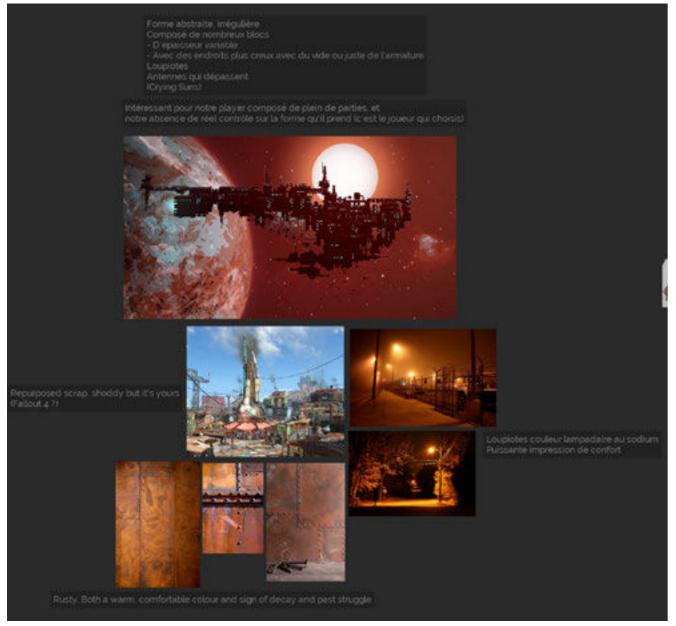
AVATAR

L'avatar devait être à la fois brinquebalant et confortable. Il devait retranscrire une impression de fragilité face à cet immense monde hostile, tout en étant un «chez lui» que le joueur pourra s'approprier. De plus, pour être cohérent avec la mécanique, il devait être un objet qui peut se propulser dans un milieu flottant et sur lequel il est possible d'attacher et retirer des parties pour modifier ses capacités.

Nous avons choisi d'en faire un vaisseau, la machinerie correspondant le mieux à nos mécaniques. Et une machine peut être rouillée. La rouille accomplit beaucoup pour la réalisation de nos intentions : elle apporte à la fois des couleurs douces et chaudes et évoque la décrépitude et les épreuves rencontrées.



Concept art final des swarmers avec sur la droite un historique de croquis qui ont mené à ce résultat.



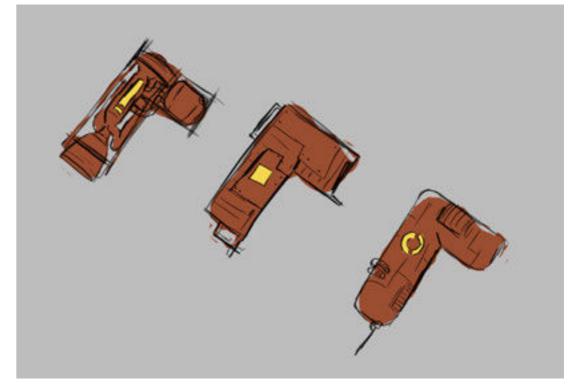
COMPONENTS

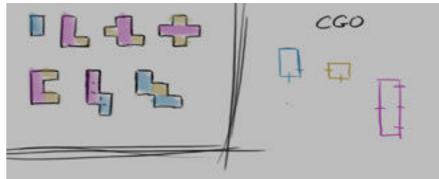
Notre avatar est composé d'un vaisseau central que nous appelons le core ainsi que de components de quatre types qui doivent pouvoir être distingués les uns des autres. De plus, les components peuvent avoir une grande variété de formes différentes, définies par le game design, ce qui a eu un impact sur la manière dont nous les avons produit.

L'illustration à droite est une de nos premières tentatives pour définir le traitement graphique de chaque type de component (nous n'en avions que trois à l'époque).

Nous avions également envisagé de donner à chaque type de component une petite lumière d'une forme précise qui indiquerait leur fonction. Nous ne les avons cependant pas entièrement conservées car elles rendaient l'ensemble trop chargé. Les retirer de certains types a permis de les rendre plus distincts les uns des autres.

Plutôt que de créer un design pour chacune des sept formes possibles pour chaque type de component, nous avons décomposé ces formes en sous-parties qui peuvent être agencées de différentes manières pour créer toutes les formes dont nous avions besoin tout en conservant une certaine variété. Cette méthode nous a permis de gagner beaucoup de temps sur la production.





Décomposition des formes de components en briques fondamentales.

Moodboard général de l'avatar

142 - Cluster 71

THRUSTERS

Les thrusters sont les components de propulsion du joueur. Ils ont été inspirés avant tout par des moteurs de fusée, contenus dans une armature carrée.

Ils sont caractérisés par une couleur plus orangée que les autres et des formes biscornues très aérées.

Les components ont aussi des morceaux d'armature qui dépasse de leur forme pour suggérer qu'ils peuvent être accrochés avec d'autres.







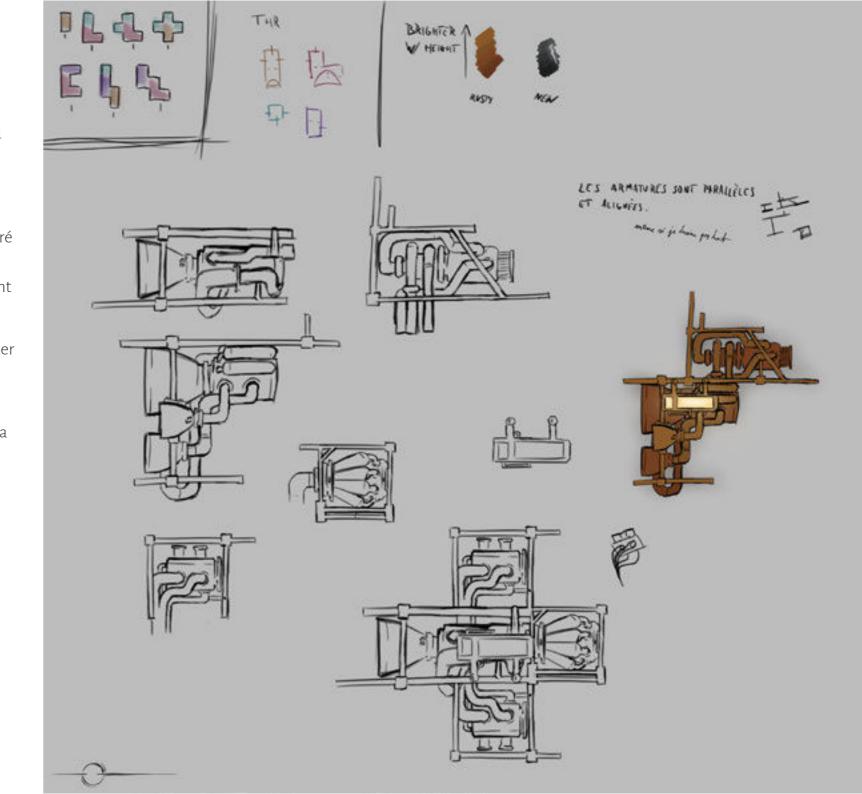




Le shader des components dans le jeu rend leur couleur plus claire en fonction de la hauteur des vertex. Le but était de rendre la profondeur lisible malgré la 2D.

Aussi, les thrusters ont demandé plus de sousparties que les autres types pour s'accommoder à la position précise de leurs échappements.

Contrairement aux autres, leur orientation a une importance dans le gameplay.

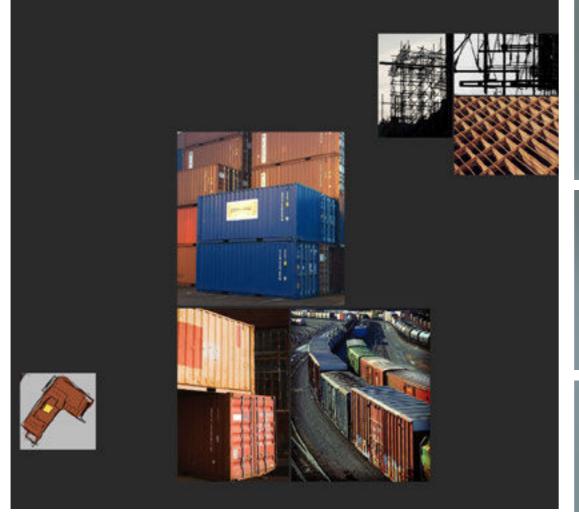


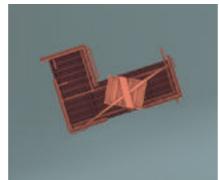
Cargo

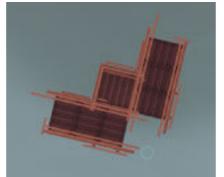
Les components cargos permettent au joueur de transporter des cargaisons, ce sont aussi les components les plus lourds. Ils sont inspirés de containers et autres grosses boîtes.

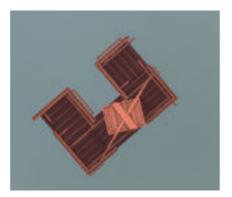
Ils sont caractérisés par des formes pleines et rectilignes et une couleur plus sombre et plus bordeaux.

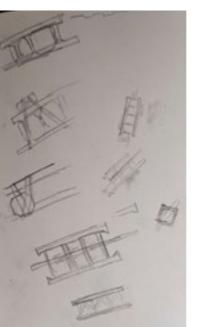
Ils ont également des armatures comme les thrusters mais plus larges et complexes.



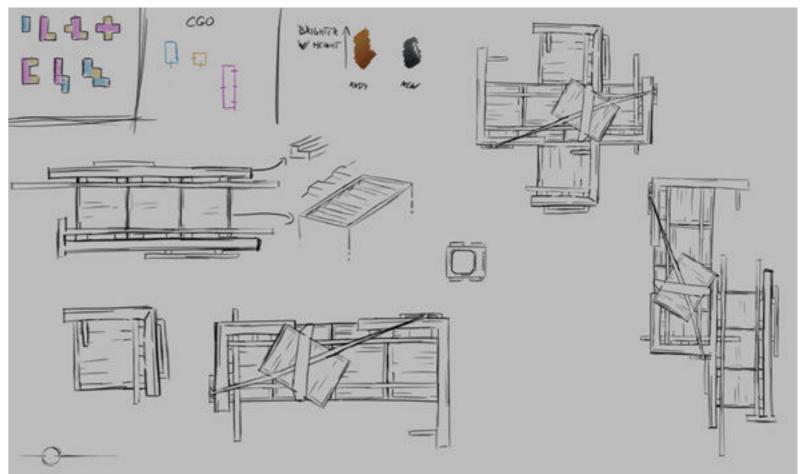










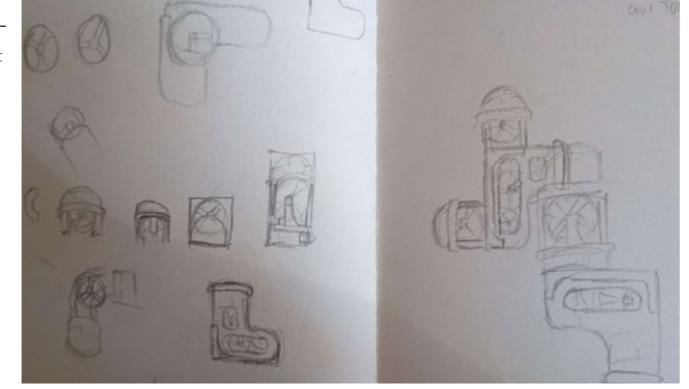


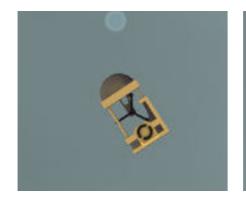
ROTATORS

Ces components servent à augmenter la vitesse de rotation de l'avatar.

Ils sont caractérisés par des formes arrondies et surtout par la présence de mécanismes rotatifs qui tournent lorsque que le joueur effectue une rotation. Dans le jeu, nous avons fait en sorte que ces mécanismes deviennent d'un jaune brillant quand ils sont assemblés, pour rendre leur rotation plus facile à voir.

Ils ont également une coloration grisée et jaunie qui les différencie des autres.



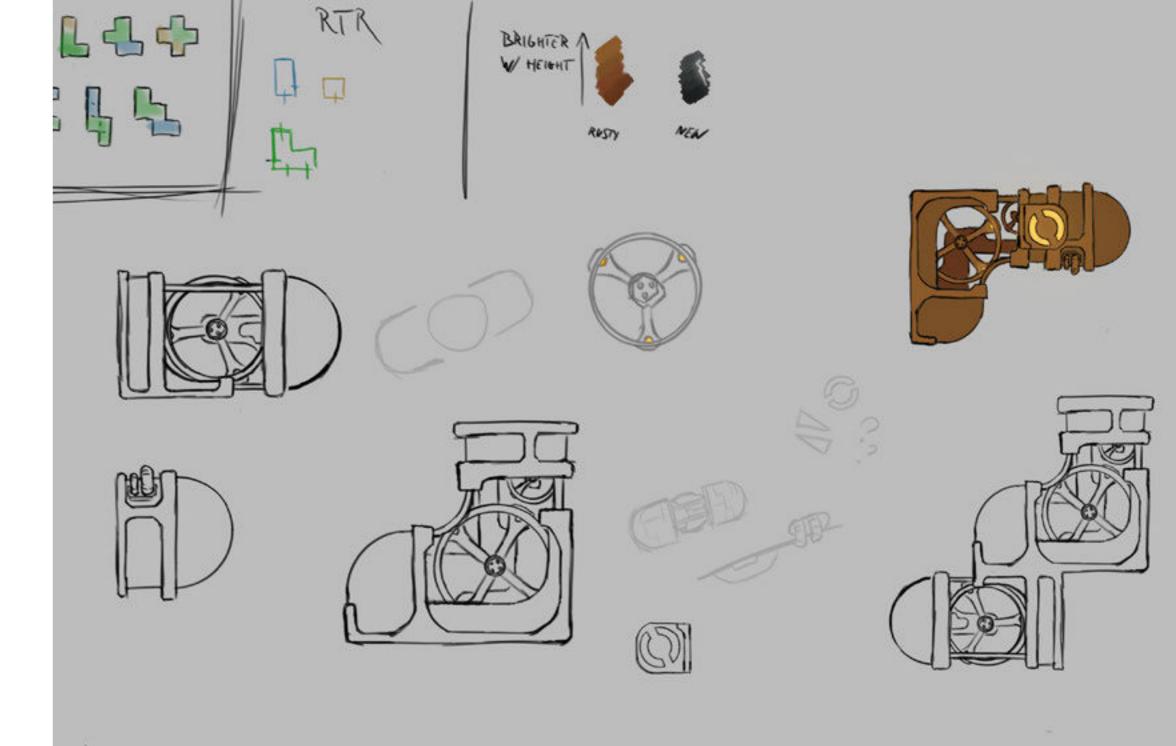










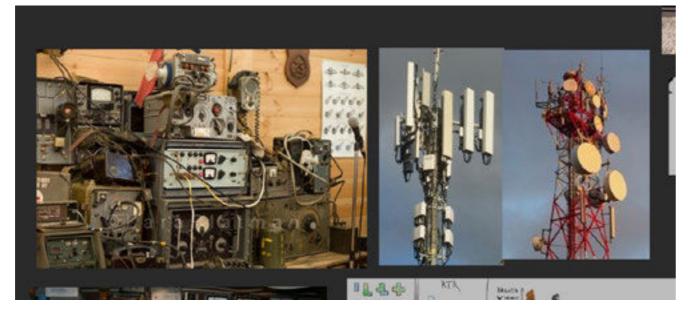


AMPLIFIERS

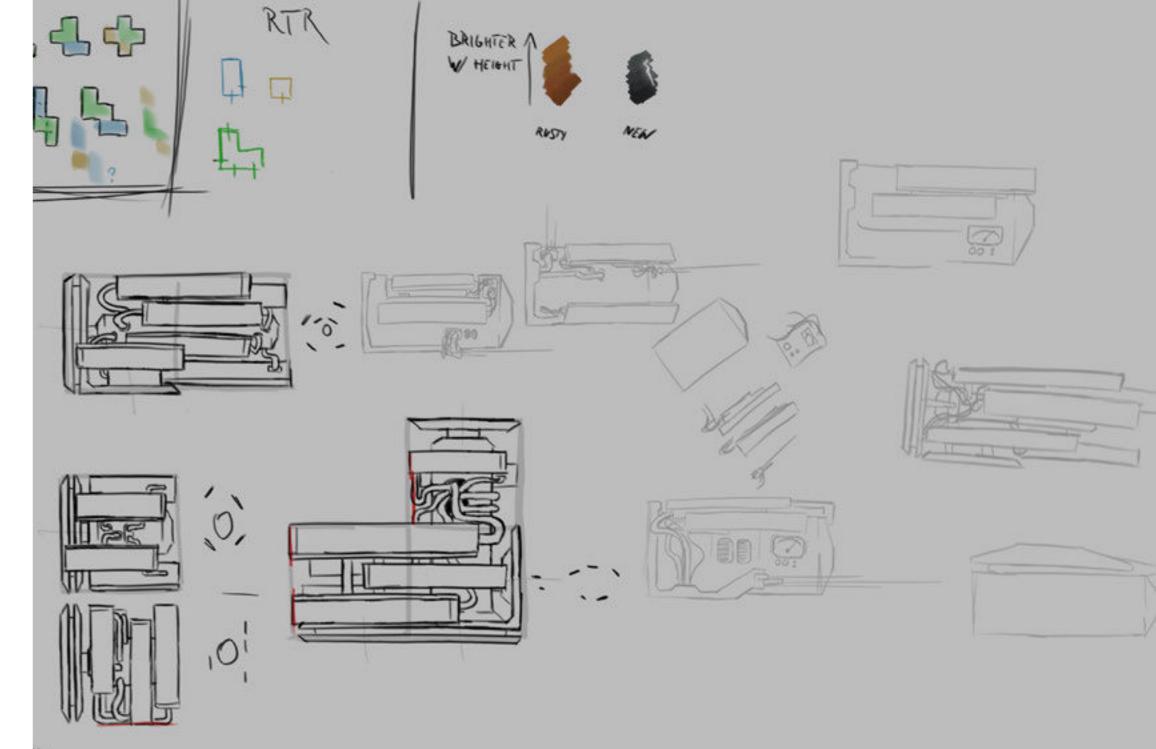
Les Amplifiers servent à augmenter les capacités des components adjacents d'un type donné. Cette fonction assez complexe et abstraite les a rendu particulièrement difficile à représenter. Nous avons finalement eu l'idée d'objets qui rayonnent au vu de leur capacité à affecter les components adjacents et nous nous sommes intéressé à des antennes et à du matériel radio. Ce dernier ne s'est pas avéré concluant mais les antennes ont servi de base au design final.

Les Amplifiers n'ont pas de couleur propre mais utilisent le matériau du type de qu'ils affectent.

De plus, même si dans le jeu, pour des raisons d'équilibrage, ils ont finalement tous leur forme de coude caractéristique, nous avions initialement conçu leurs visuels pour qu'ils s'adaptent à toutes les autres formes.

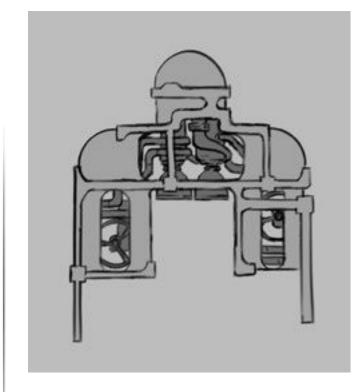


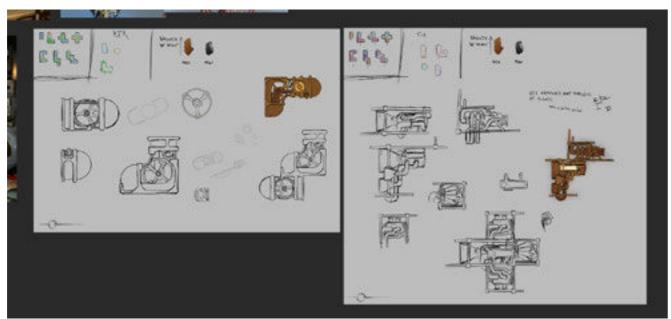


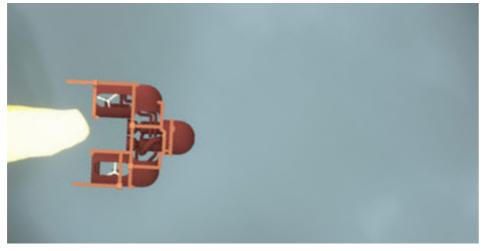


Core

Le core étant capable de tourner et de se propulser, son apparence a été pensée comme un mélange des codes graphiques des thrusters et des rotators. En conséquence de quoi nous l'avons conçu en dernier







INTERFACES

Notre jeu contient plusieurs menus et interfaces, dont certains doivent afficher une assez grand quantité d'informations. Nous avons cherché rendre ces interfaces aussi épurées et aérées que possible, pour que le jeu et son ambiance restent à l'honneur.

Dans leur esthétique, les interfaces sont plutôt liées au joueur, reprenant le jaune pâle qui fait partie de sa palette et utilisant la police Chakra Petch.

Cette dernière, qui sert également pour le titrage de ce document, est une linéale très simple et assez grasse qui a la particularité d'utiliser très peu d'arrondis. Avec à la place des formes rectilignes biseautés. Elle nous a paru allier un aspect mécanique, une simplicité, et des formes peu agressives, ce qui nous a semblé tout à fait pertinent pour notre jeu. Nous utilisons sa version SC avec un large espacement entre les caractères dans les titres à la fois du jeu et du GOD. Car les traits intéressants de cette typographie sont plus marquées sur les capitales. Le large espacement



permet d'accompagner la vastitude et le vide du monde du jeu, et d'être cohérent avec l'aspect épuré des interfaces.

Enfin, un choix récurrent dans nos interfaces est de guider le regard vers

le coin haut droit de l'image et / ou de le laisser ouvert. Ainsi, nous rendons omniprésent le message implicite que le joueur peut toujours avancer et s'élever.

Nous avions besoin d'un HUD pour transmettre au joueur diverses informations nécessaires au gameplay. Cependant, pour le maintenir aussi clair et sobre que possible, nous avons fait en sorte que les différentes parties de l'affichage soient indépendantes et puissent être cachées ou montrer au besoin. Si, par exemple, la barre de capacité n'est pas nécessaire parce que le joueur n'a aucun component cargo, alors elle glissera rapidement hors de l'écran.

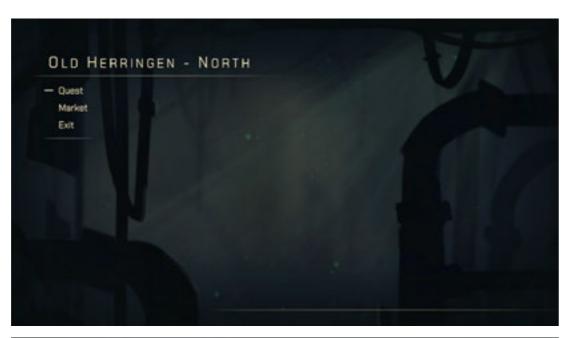




Le menu pause devait être particulièrement simple et permettre de toujours voir l'environnement derrière. Même si un filtre sombre vient indiquer le changement d'état et assurer la lisibilité du texte. Le HUD est également caché quand celui-ci est actif.

Dans les différentes pages du menu de point de livraison, nous avons veillé à laisser des marges suffisamment larges avec les bords de l'écran pour que toutes les informations soient bien en vue. Et tout comme dans les autres interfaces mais de manière plus prononcée ici, nous utilisons ces lignes douces pour apporter de la structure à la composition sans l'alourdir de trop.

Aussi, nous avons ajouté un background légèrement flou et peu contrasté qui vient suggérer une sorte de hangar ou salle des machines et créer une ambiance feutrée que l'on ne retrouve pas dans le reste du jeu sans pour autant distraire le regard des informations importantes. La couleur verte permet de conserver le mystère et une partie de la froideur du monde, mais une froideur atténuée et qui semble plus positive. C'est un endroit où le joueur es en sécurité. Il y a d'ailleurs dans cette illustration un rayon de lumière qui vient d'en haut à droite. Enfin, l'illustration est légèrement animée. Les point de vue bouge légèrement et les différents plans se décalent avec un effet de parallaxe. La lumière et les poussières qu'elle éclaire







varient en intensité.

L'écran de game over devait à la fois rendre grandiose la destruction du joueur, tout en appuyant le fait qu'il a été détruit. Les barres noires accomplissent ces deux tâches, apportant un effet cinématique tout en écrasant l'image, contenant le regard et concentrant l'attention sur le vaisseau détruit et la manière dont il a été détruit. En effet, le texte dépend de ce qui a éliminé le joueur.

Dans le jeu, les lettres de ce texte et les barres qui l'encadrent s'éloignent lentement les unes des autres. évoquant des débris à la dérive et accentuant l'aspect dramatique.

Logo

Notre logo avait deux objectifs: être reconnaissable, et transmettre un ressenti qui soit cohérent avec celui du jeu.

Il y a plusieurs choses que ce logo communique. D'abord, cette silhouette composée de deux triangles qui se font face avec un vide entre les deux est évoque la notion d'hostilité du monde. On passe entre des formes pointues, inquiétantes et menaçantes, mais on peut passer. Cette possibilité de passer est importante, elle est en accord avec la capacité de surmonter l'hostilité de cet environnement et de passer outre, un concept omniprésent dans notre gameplay, et mis en valeur par la conclusion du jeu. De plus, elle a l'avantage de créer une forme de logo plutôt originale.

Ensuite, les formes des lettres, fines et pointues, font écho aux sprites de l'environnement, et traduisent la menace latente de celui-ci, mais sans pour autant donner une impression de puissance brute qu'aurait transmit un texte plus gras.

Dans le même esprit, le logo est

158 - Cluster 71

Visuel

légèrement irrégulier. Quelque chose de trop strict, rectiligne et uniforme transmettait plus une sensation de fierté et de droiture, qui rappelait par exemple un insigne d'aviation militaire, ce qui ne correspondait pas à notre jeu.

Un espacement non uniforme entre les lettres permet aussi au regard de circuler plus facilement dans l'image, si la densité était la même partout, nous ne saurions pas où regarder et serions coincés, là il est possible de bouger.

A cela s'ajoute beaucoup de vide dans le logo, le but étant d'exprimer la vastitude, la solitude, l'exploration et l'aspect disloqué de notre monde.

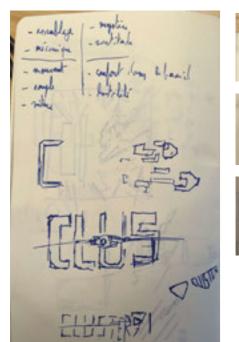
Certaines formes dans le logo sont plus grasses que le reste. Elles servent également à le rendre moins uniforme, créer une hiérarchie de l'information pour que l'œil ait des points d'ancrage et ne se disperse pas trop. Les formes plus épaisses rappellent des flèches qui pointent en haut, ou en haut à droite. Le mouvement vers le coin haut droit de l'image est récurrant dans notre jeu et notamment dans nos interfaces. Notre HUD ainsi que notre menu principal laissent du vide en haut à droite. Un côté droit ouvert, c'est la possibilité d'avancer, c'est qu'on n'est pas coincé, et de l'espace en haut c'est

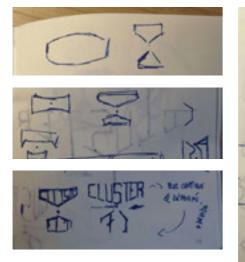
l'ascension. Et la flèche du 71 (la première que l'on voit) qui pointe vers le haut a aussi l'avantage de guider le regard vers le reste du logo.

Enfin, le texte est volontairement abstrait, destructuré, dur à déchiffrer. Il fait echo au mystère, à l'incertitude et à la découverte qui sont des éléments centraux dans l'ambiance de notre jeu.

ITÉRATIONS

Sur les deux pages suivantes, un historiques des étapes de la création de ce logo.















D.A. Sonore

Sound Design

INTENTIONS

Nos intentions principales pour Cluster 71 étaient de faire comprendre au joueur qu'il est dans un monde hostile et que l'avatar du joueur n'est composé que de morceaux de métal, soudés les uns aux autres. Pour cela, nous avons utilisé de nombreux sons métalliques de toutes sortes, allant de la poubelle au grillage en passant par des voitures en crash test. Concernant l'environnement du joueur, la partie musicale se découpe en deux points: d'abord les phases d'exploration, que nous voulions assez calmes, mais qui retranscrivent toujours le côté étrange de l'environnement. Puis l'arrivée dans chaque nouvelle région, et le côté dangereux de la rencontre avec n'importe quelle créature.

RECHERCHE SONORE

Son de choc : nos recherches sonores ont porté en partie sur les sons de choc de l'avatar dans l'environnement. Inspirés par divers jeux, dont GTFO, l'idée d'avoir un son métallique et résonnant nous a semblé possible. À partir de là, nous avons itéré sur des bruits d'impact métallique sur des portes et d'autres objets comme des poubelles, pour finalement nous rendre compte que nos efforts n'étaient pas situés au bon endroit. Le son de base était bon. mais c'était le traitement appliqué qui ne l'était pas. Le manque de réverbération dans la réalisation nous a paru flagrant quand, après quelques modifications sur Reaper, puis Fmod, le son enregistré devenait beaucoup plus impressionnant.



Son de tir: Dans les anciennes versions du jeu, le joueur pouvait avoir des canons et autres mitrailleuses, qui lui permettaient d'éliminer différents types d'ennemis. Nous avons donc essayé d'orienter nos recherches sur de multiples types de canons, futuristes ou réalistes, modernes ou anciens, et celui qui semblait le plus intéressant pour notre jeu était le moderne. Après itération et avoir trouvé de multiples inspirations, un son convenable de canon de bateau modifié nous semblait être le son parfait pour notre jeu, mais nous avons malheureusement dû retirer tout système de tir de notre jeu pour des raisons de scope.



Son d'assemblage: Pendant la phase de prototype, les placeholders de nos sons d'assemblage étaient assez basiques, un petit son de choc métallique, rien de plus. Mais nous trouvions que cela n'était pas suffisant, et nous avons itéré dessus et fini par faire de vraies recherches sonores sur le sujet afin de trouver un son parfait pour notre jeu. En premier lieu, nous avons remarqué que notre son manquait

d'ampleur. De plus, l'évolution n'était pas assez marquée dans le son du premier semestre. À partir de là, nous avons réalisé des ajouts de pistes pour marquer l'évolution du joueur dans l'assemblage. Des sons de choc à forte résonance, des bruits de chaîne, des pistons à vapeur, nous avons fini par ajouter beaucoup de choses pour pouvoir obtenir les sons finaux qui nous satisfont pleinement.



162 - Cluster 71

103

Son de moteur : Pendant la première phase de développement de notre jeu, je suis parti sur un son de moteur assez futuriste, en me basant sur des sons avec un très fort trémolo. Mais une fois la direction artistique définie, je me suis rendu compte que le son que j'avais réalisé n'était en rien approprié. À partir de là, j'ai commencé à rechercher quel genre de son pouvait être intéressant pour notre jeu. J'ai d'abord essayé des sons de moteurs de voiture, avec des ruptures et passages de vitesse, mais même modifiés, cela ne correspondait pas du tout à l'ambiance de notre jeu. J'ai donc continué les recherches pour finalement partir sur quelque chose de simple, des sons rappelant les fusées au décollage.



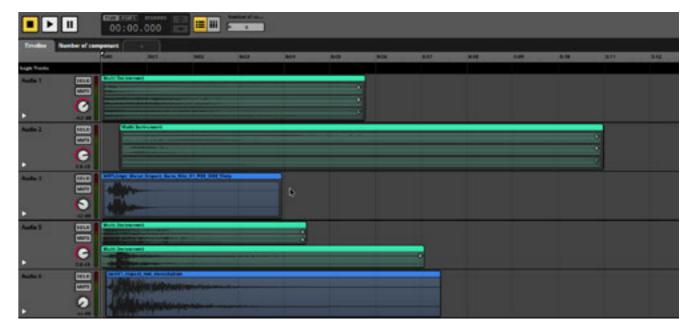
Son d'éclair: Dans les premières versions du jeu, nous avions encore des nuages qui pouvaient gêner la progression du joueur. Nous avons donc réalisé en premier lieu un snapshot pour faire un effet sur les sons ambiants, mais ce n'était pas suffisant, et après quelques recherches sur différents jeux, j'ai fini par produire une ambiance quand le joueur arrive proche d'un nuage

et finit par y entrer. Cette ambiance se base sur des éclairs et un drone. Mais le système de nuages n'a pas été conservé pour le jeu final, et le son a été recyclé pour les nuages en bordure de carte.



Système de clusters

Le son d'assemblage doit transmettre l'évolution de la taille du joueur, ainsi que ses gains de puissance et possibilité d'action. N'avoir qu'un seul son lors de l'assemblage ne nous permettait pas de retranscrire ce que nous voulions. Nous avons donc opté pour un système de cluster qui utilise plusieurs sont qui évoluent en fonction du nombre de components actuellement accrochés au joueur. L'évolution va du volume de la piste, au nombre de piste jouées, jusqu'à la résonance du composant via la reverb.



Le paramètre de "Number of components" correspond au système de clusters

MOUVEMENT

Un de nos principaux components est le "thruster" ou réacteur, permettant au joueur d'accélérer dans une direction.

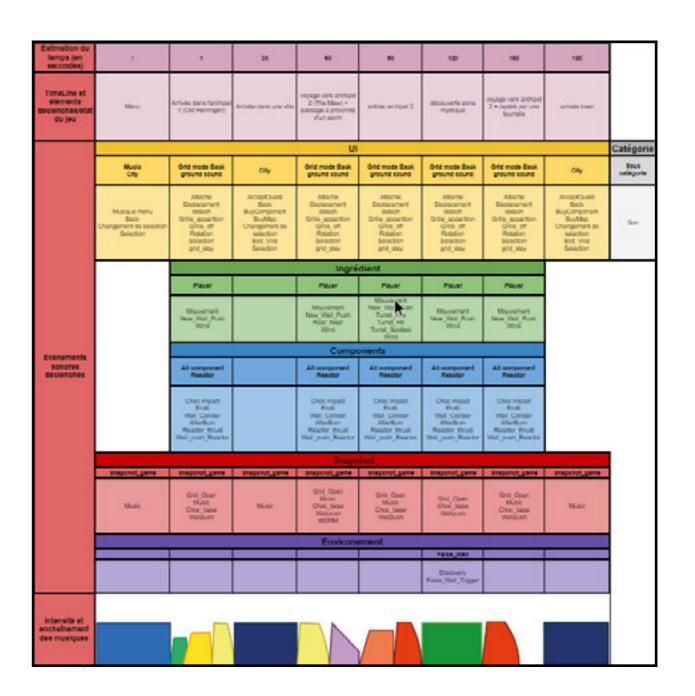
Aussi, le joueur a un réacteur de base sur son vaisseau, ainsi qu'un inertie qui peut ne pas être corrélée ni à l'action de son réacteur de base, ni de ses composants.

Nous avons donc séparé ces trois sons; le système d'accélération du réacteur de base, le système d'accélération (et de cluster) des thrusters ajoutés, et la vitesse du joueur.





Différents thrusters



SOUNDTIMELINE

Musique menu

Exploration

Thème entrée archipel 1

Thème archipel 1

Ville

Tension Worm

Thème entré archipel 2

Thème archipel 2

Zone mystique

Composition

Notre première inspiration pour la musique de jeu fut celle d'Outer Wilds, notamment l'utilisation de la guitare acoustique pour représenter un espace mal agencé mais chaleureux. Le compositeur est donc parti d'une guitare acoustique répétant un motif pour construire et composer la musique de menu du jeu. L'équipe a tout de suite apprécié cette musique, et de là un dialogue s'est installé entre le chef de projet et lui. Ensemble, ils ont pu discuter de la composition mais aussi et surtout de son incrustation dans le ieu.

Toutes les musiques se situent globalement dans la même gamme, ce qui nous a permis d'éviter des risques de faux raccords, entre différentes zones. Le plus

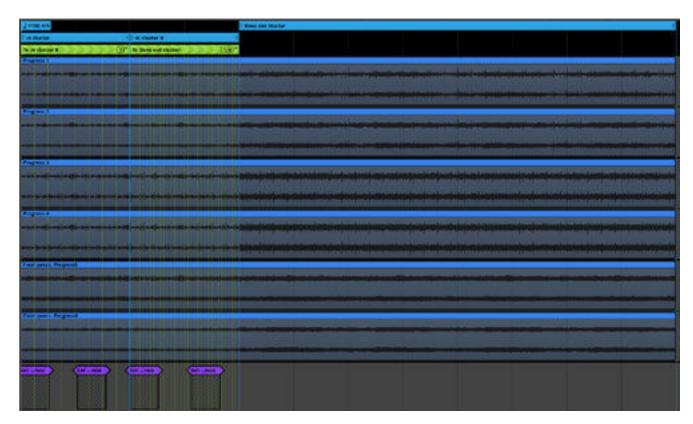


complexe a été de trouver les sonorités qui collent parfaitement à ce qu'on voit à l'écran. Prenons par exemple Chittering Construct, archipel riche en Swarmers qui sont de petites créatures s'agrippant au joueur ce qui donne un effet d'étouffement L'objectif de leur musique à alors été de reproduire cet effet. Pour ce faire, nous avons utilisé des sons dissonants, poussant l'auditeur dans un certain inconfort, effet amplifié sur Fmod avec d'autres mélodies jouées aléatoirement.

Pour la musique d'ambiance du jeu, nous souhaitions créer une sensation de confort et éviter la redondance. 3 mélodies différentes pour chacun des 5 instruments jouées ont été ainsi créées, découpées, et intégrées de manière à ce qu'elles

s'enchainent aléatoirement dans Fmod.

Pour finir, la musique de la dernière quête du jeu est la plus complexe techniquement, et la plus longue du jeu. La principale inspiration en fut la musique du jeu Celeste. Nous avons ainsi découpé cette musique finale en 6 parties légèrement superposées, de façon à pouvoir accompagner la progression du joueur dans cette quête.



L'OST du jeu contient finalement 2 musiques par archipels, 2 pour le worm, 2 pour les swarmers, 1 musique de ville, 1 musique d'exploration qui contient des variations selon les situations, ainsi qu'une musique exclusive à la quête finale. Intégration FMod de la musique de fin, chaque piste est une phase différente de celle-ci

EVENT LIST

Catégorie	Group	Event	Intégré	Son final	PlaceHolder
· ·		Entré dans la zone d'exploration	☑	✓	☑
		Enter in danger zone	\checkmark	~	$\overline{\mathbf{Z}}$
		Poursuite	\checkmark	$\overline{\mathbf{Z}}$	\sim

Dans cette partie de la liste des événements, on peut voir la catégorie et le groupe qui servent à trier les événements suivants. «Événement» sert à retrouver le nom non-officiel de l'événement. Ce nom est souvent donné par le groupe quand on doit parler d'un type de son en particulier. La liste intégrée sert à avoir un traceur visuel. Il en va de même avec «Son final» et «Placeholder». Grâce à cela, il est facile de savoir sur quoi je dois travailler en priorité et quelles sont les étapes que je dois réaliser concernant chaque son.

Fmod Event name	Sound description	Caption/Trigger condition (human readable)
Jeu	Musique calme d'ambiance	Partir d'une ville, ou après une entré d'archipel
Jeu	fog horn qui mette une tension	Le joueur entre dans la proximité du monstre
Jeu	Musique de poursuite	Le joueur se fait poursuivre par le monstre

Ici, la rubrique «Fmod Event name» correspond au nom officiel du son dans le Fmod du projet. Grâce à cela, je peux facilement retrouver n'importe quel son et le modifier. C'est particulièrement pratique quand le Fmod devient très

grand. La rubrique «Sound description» m'a surtout servi lorsque j'ai voulu créer mes placeholders et décrire exactement le son attendu lorsque j'ai sous-traité certains des sons du jeu. «Caption/trigger (H R)» permet de savoir comment le son doit être déclenché, ce qui m'a permis de travailler main dans la main avec les développeurs pour réaliser l'intégration la plus parfaite possible.

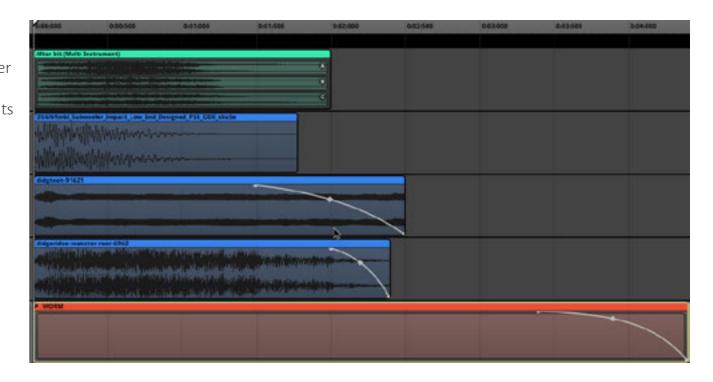
Trigger conditions (programming)	Event type	Loop	Evolution
phase = 0	2D	no	fade out vers un drone qui loop
phase =1	2D	yes	non
phase = 2	2D	yes	non

La rubrique «Trigger condition (P)» a été remplie plus tard, une fois la programmation réalisée, et aide les développeurs à trouver la source de différents problèmes quand il y en a. «Event type» permet de traquer les événements avec de la spatialisation. «Loop» permet de traquer les événements qui se jouent en boucle, c'est-à-dire souvent les musiques.

«Evolution» est le paramètre le plus important car grâce à lui, je peux garder en tête la moindre raison pour laquelle un son devrait évoluer, et grâce à cela, faire une bien meilleure intégration dans le Fmod.

Prise de son

Nous avons opté pour une prise de son au micro, et j'ai donc enregistré les voix de mes camarades afin de pouvoir les modifier dans Reaper. J'ai demandé à chacun de réaliser des cris, grognements et raclements de gorge. La majeure partie du travail s'est faite sur Reaper, où j'ai rendu chaque son beaucoup plus profond et métallique, et ajouté de la réverbération. Par la suite, j'ai intégré ces sons dans Fmod, pour ensuite ajouter quelques effets supplémentaires, faire évoluer le son en fonction de la distance par rapport aux joueurs. Grâce à cela, nous avons réussi à obtenir le son de notre ennemi principal, le worm.



UX

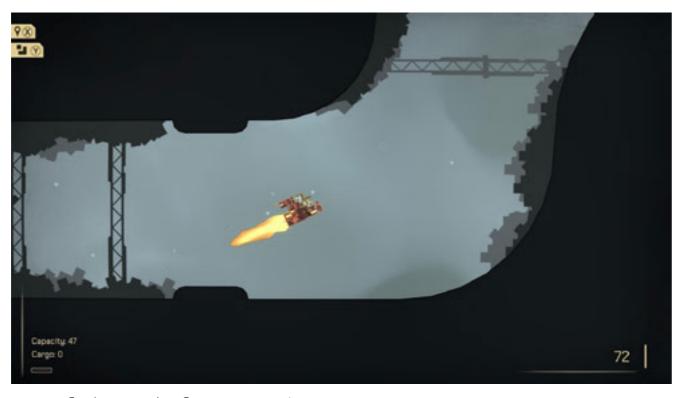
ONBOARDING

Notre jeu contient beaucoup de systèmes et de règles, il nous fallait donc mettre en place un tutoriel pour les introduire et clarifier au joueur.

Aussi, nous avons dû réadapter des conventions (dans le gameplay) pour convenir plus à nos systèmes, ce qui nous a induit la nécessité de réintroduire ces principe «connus», modifiés chez nous.

Ainsi, nous ouvrons le jeu sur une phase safe, indiquant d'abord une suite de touches au joueur s'enchaînant, introduisant d'abord notre système de grille en détails.

Ensuite, nous enseignons la rotation, les déplacements et l'activation du component précédemment installé; cela permet de faire la connexion entre nos deux systèmes de gameplay.



Enfin, le tutoriel se finit sur une quête automatique, demandant au joueur de se rendre à un point de ville (le plus proche du tutoriel).

TESTS UTILISATEUR

Nous souhaitions originalement rester aussi minimaliste que possible, quitte à mettre très peu d'informations sur nos interfaces.

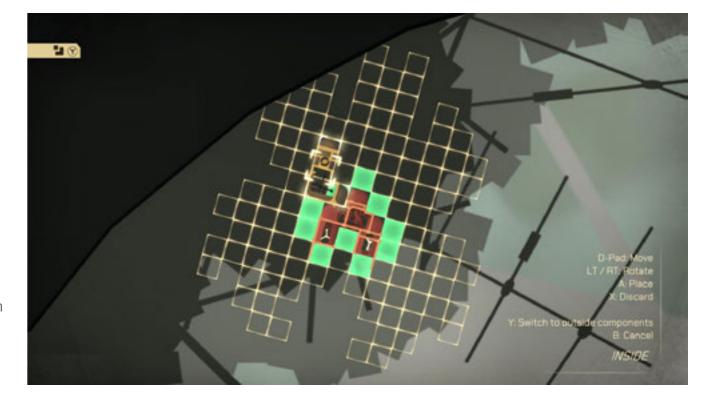
Cependant, nos différents tests UX nous ont montré que le peu d'éléments présentés était loin d'être suffisant pour comprendre la complexité de la grille.

D'ici, nous avons fini par indiquer plus clairement beaucoup de signes qui manquaient.

FORME DES COMPONENT ET DE LA GRILLE

Pour faciliter l'identification des formes de component, nous avons ajouté un contour à ceux-ci qui s'affiche lorsque le joueur sélectionne les components.

Nous avons appliqué ces mêmes changements à la grille, ce qui permet d'en voir les limites et de la démarquer du reste de l'environnement.



174 - Cluster 71

175

SENS DES THRUSTER

Une erreur souvent revenue fut autour de la pose des thruster components. Ceuxci ont un «sens» de propulsion qui n'était apparemment pas suffisamment bien véhiculé.

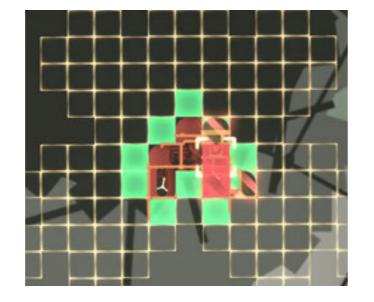
Pour en faciliter la lisibilité, nous avons à la fois modifié l'apparence des sorties de flammes en noir, et avons ajouter un symbole au niveau de la sortie lorsque ces derniers sont sélectionné (et donc installés/désinstallés/déplacés) en mode Assemblage.



Possibilité de poser

Il n'était aussi pas clair de savoir si l'on pouvait ou non poser un component à un emplacement. Bien que cela pouvait s'expliquer alors du manque de feedback sonore, il fallait aussi un retour visuel pour que le joueur puisse rapidement non seulement qu'il ne peut pas poser un component, mais aussi éventuellement pourquoi.

Ainsi, nous avons modifié la sélection qui vire au rouge avec un overlay animé de barres rouges également lorsque le component n'est pas posable, et les cases qui sont en causes (soit lorsqu'elles sont hors de la grille, soit lorsqu'elles sont superposées à un component déjà posé) sont d'un rouge encore plus intense.



Utilité de ce que l'on pose

Enfin, une confusion planait autour de ce qu'on installe, dans le sens utilitaire.

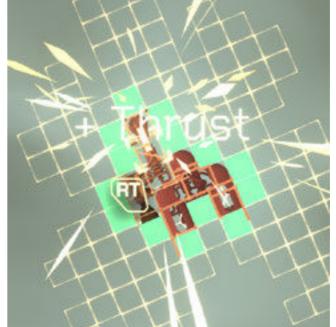
Seul les thrusters ont toujours été compris comme des outils permettant, activés, d'augmenter la vitesse de propulsion.

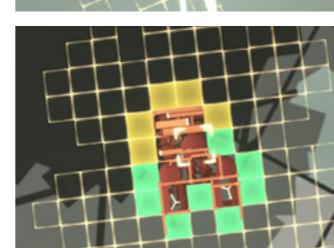
Ainsi, nous avons ajouté pour les rotator (les moins compris) des roues qui tournent lorsque le joueur tourne, en fonction de sa vitesse de rotation.

Aussi, un texte apparait lors de l'assemblage pour indiquer ce que le joueur a gagné.

Ensuite, nous avons retiré les lumières des cargos, et remplacé leurs feedbacks par une jauge en bas de l'UI se gonflant le plus on porte de cargaison, avec un chiffre augmentant et grossissant l'accompagnant

Enfin, pour tous les components, nous avons ajouté un texte qui apparaît lorsque l'on pose un component indiquant la statistique qu'il a apportée (Handling, Capacity, Thrust), et des couleur distinctes pour les différencier encore plus facilement en groupe de components.





Ces couleurs nous ont également servi pour le 4ème type de component, les effectors, lesquels boostent les capacités des components adjacent de leur couleur correspondante (les cases sont elles indiquées par des cases jaunes).





177

TECHNICAL

ENEMY SYSTEM

Worm

Le worm a 4 états différents, gérés par une state machine. Ces états sont les suivants :

WANDER

Le worm a une zone dans laquelle sont aléatoirement définis des points vers lesquels il se dirige. Lorsqu'il atteint un point, ou après un certain temps écoulé, un nouveau point est généré.

CHASE

Cet état est déclenché soit quand le worm détecte le joueur dans son champ de vision, soit quand ce dernier le touche physiquement. A ce moment, il commence à suivre le joueur, et commence à accumuler lorsqu'il poursuit le joueur sur une ligne droite.

Aussi, la portée de détection du joueur est légèrement influencée par la taille de



la caméra du joueur, pour aller dans les intentions de «teasing», permettant au joueur d'apercevoir le worm peu avant que celui-ci ne le perçoive.

ATTACK

Lorsque très proche du joueur, le worm passe en mode attaque. Sa mâchoire s'ouvre en direction du joueur et il tente de se rapprocher suffisamment du worm pour le mordre. Il conserve les propriétés de déplacement du chase.

Lorsque le worm est assez proche du joueur pour le mordre, il la garde ouverte un court moment, et la referme soit après ce délais, soit si le joueur est sur le point de sortir de sa portée.

Le worm peut mordre même si sa mâchoire n'est pas complètement ouverte, et se «précipiter».

Après avoir mordu, il y a un cooldown de quelques secondes à la morsure, pendant lequel il va tourner autour du joueur avant de pourchasser le joueur.

STALK

Quand le worm perd de vue le joueur, par la distance ou par une vue obstruée, il rentre dans ce mode quelques secondes. Ses déplacements deviennent ceux du wander, mais il conserve des portée de détection plus élevées, et tourne autour de la dernière zone où il a aperçu le joueur la dernière fois. Une fois les secondes écoulées, il repasse en mode wander et retourne dans sa zone de base.

ANIMATION

Le worm utilise aussi un système d'animation procédurale pour que ses mouvements s'adaptent mieux à son comportement. Sa physique est gérée par une succession de sphères qui se suivent les unes les autres et servent de pantin pour ses mouvements de serpents. Et sur ce squelette nous intégrons un mesh riggé qui en suit les mouvements.



178 - Cluster 71

Technical

SWARMERS

Murs de swarmers

Les swarmers sont instanciés au sein d'une région, et sont individualisés.

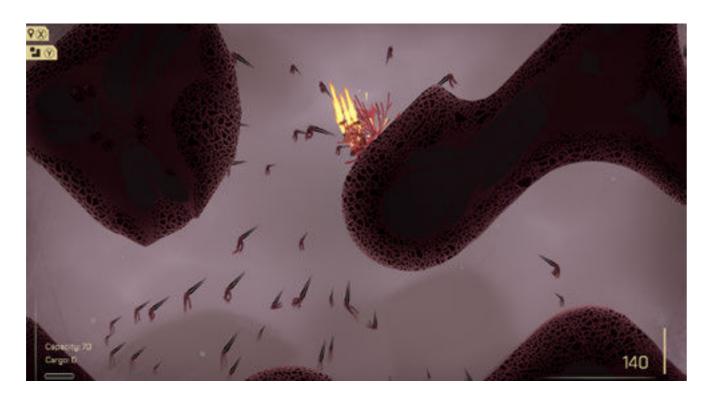
Lorsque le joueur entre en contact avec eux, les murs envoient un signal avec la force et la position de l'impact. Les régions vérifient ensuite si les swarmers qui leur sont liés sont à une distance suffisante (basée sur la force et la source de l'impact) pour en déclencher le mode chase.

Les comportements des swarmers sont également gérés par une state machine, leurs 4 état étant ceux-ci :

WANDER

Le swarmer se déplace de manière non continue, en tics d'impulsions ajoutées aux rigidbody toutes les X secondes, X contenant une marge d'aléatoire et variant selon l'état du swarmer.

Il se déplace dans des directions aléatoires à chaque tic, au sein de la région. S'il sort de la région, il utilisera le navmesh pour se diriger vers son nid de nouveau.



S'il est trop proche d'un autre swarmers, il se déplacera dans la direction opposée, permettant de les répartir assez équitablement au sein de leur zone.

CHASE

Provoqué par le choc du joueur dans un mur de Swarmer proche. Il se déplace plus rapidement, et en direction du joueur. Il passe à nouveau en Wander si le joueur s'éloigne suffisamment de lui ou de sa région.

Si le swarmer touche un component, il s'y accroche et passe en mode attached.

ATTACHED

Le swarmer perd son rigidbody et devient enfant du component auquel il est attaché. Il inflige des dégâts à chaque instance où il devait de base se déplacer en mode chase.

Aussi, le swarmer va régulièrement check la vitesse du rigidbody auquel il est attaché. Si cette vitesse est au-delà d'une certaine valeur, le swarmer a une chance de se détacher, qui augmente selon à quel point la vitesse dépasse la valeur seuil.

Si le swarmer réduit le component à 0 PV, il passe en mode return.

Si le component auquel il est attaché est détruit par autre chose (collision, worm), il est téléporté à son nid et repasse en mode wander.

RETURN

Un component réduit à 0 PV par un swarmer n'est pas détruit, mais est passé à 1 PV et détaché du joueur. Le rigidbody du swarmer reste désactivé, mais recommence à faire des impulsions de déplacement toute les X secondes, mais la vélocité est

appliquée au component.

Le swarmer essaye ainsi de rapporter le component détaché jusqu'à son nid. Une fois le component ramené au nid, il le lâche et repasse en mode wander.

PERFORMANCES

Dans une optique d'optimisation, la région connaît le point de spawn de tous les swarmers. Quand le joueur s'approche à une certaine distance du point d'un

swarmer, celui-ci spawn. Quand le joueur s'éloigne de cette distance, le swarmer est despawn. Ainsi, le comportement des swarmers n'est pas calculé si le joueur est hors de la région. De plus, les line renderers qui servent à faire les tentacules des swarmers ainsi que le script qui les gère sont désactivés s'ils ne sont pas à l'écran.



180 - Cluster 71

Technical ———

GRID SYSTEM

Très tôt lors du prototypage, un besoin de pouvoir assembler avec précision s'est fait sentir auprès des testeurs et du reste de l'équipe. Nous avons donc commencé le développement d'une suite d'outils pour donner une grille d'assemblage à l'avatar, et pour pouvoir créer des components qui peuvent s'attacher à ce dernier.

L'avatar du joueur contient la première pièce de ce système, le ShipGridManager, qui s'occupe non seulement des entréessorties de la grille, mais aussi du fonctionnement du mode Assemblage. Ce script possède plusieurs UnityEvents et méthodes pour changer l'état de la grille, permettant aux différentes fonctionnalités du jeu de s'imbriquer avec ce système.

Les components peuvent rentrer dans la grille du joueur grâce au GridObject, qui permet de définir les différentes cases qu'il prendrait sur la grille. L'avatar possède aussi un GridObject pour définir les limites et les différentes cases déja prises par le vaisseau, ce qui en fait un cas particulier.

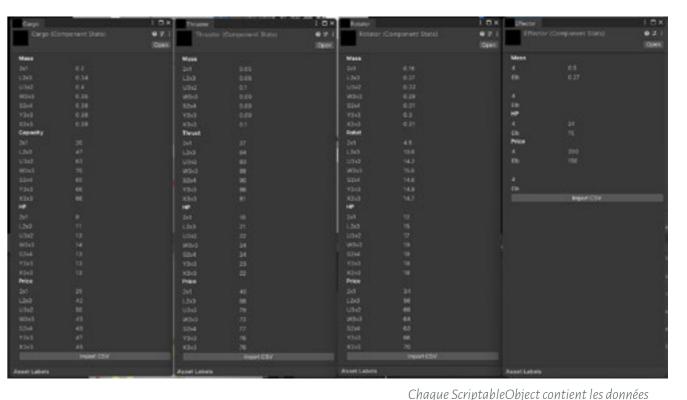


CSV Tool

Peu de temps après la phase de préproduction du jeu, nous avons réalisé que le balancing de nos components, ainsi que la quantité de variables associées, demanderait l'utilisation d'un tableur. Cependant, importer manuellement les résultats du tableur dans le moteur de jeu s'est avéré être une tâche longue et rébarbative. Pour palier à ce problème, un outil pour importer les données du tableur simplement s'est imposé, d'où la création de notre CSV Tool.

Cet outil importe des données d'un tableur au format .csv (Comma-Separated Values), les organise par paramètres et les stocke dans un ScriptableObject qui peut ensuite être utilisé comme variable directement dans un component. L'outil contient aussi des fonctions utilitaires pour trouver des paramètres simplement.

Avant le développement de cet outil, changer les variables d'un type de component pouvait prendre plusieurs minutes. Grâce au CSV Tool, nous avons pu itérer sur le balancing de nos composants de façon extrêmement rapide.



relatives à un type de composant. Les données peuvent donc être récupérées en donnant un Identifiant (ex. Y3x3) et un paramètre (ex. Capacity).

CONSOLE TOOL

Vers la fin de la production du jeu, l'équipe de développement a fait part d'un besoin d'outil pour pouvoir "tricher" en jeu - certains exemples de triche incluant "pouvoir se donner de l'argent", "pouvoir naviguer dans les murs", "avoir une très grande vitesse de déplacement"... Ce besoin à été exprimé pour pouvoir itérer plus rapidement sur l'entièreté du projet, par exemple sur le balancing, ou en situation de présentation de jeu.

Pour ce faire, nous avons développé une console, inspirée par les jeux Source Engine, Minecraft ou encore The Elder Scrolls V: Skyrim, permettant d'exécuter certaines commandes réservées aux développeurs.

La console apparaît lorsque l'utilisateur appuie sur la touche F3, et indique une liste des différentes commandes possibles, ainsi qu'une façon d'obtenir des informations sur chaque commande et son utilisation. Cela permet à n'importe qui dans l'équipe d'apprendre comment utiliser l'outil sans avoir à demander aux développeurs.

Chaque commande reflète un besoin réel pour toute l'équipe, et est développée



Aperçu de la console in-game et de l'utilisation d'une commande

compréhension et d'utilisation, tout en approchant les problèmes de la façon la plus généraliste possible. Par exemple, la commande "teleport" permet à l'équipe de placer l'avatar en plein milieu d'un GameObject. Faire une liste de GameObjects pour se téléporter n'étant pas viable dans la plupart des cas, nous

avons créé la commande "waypoint" pour pouvoir établir une liste simple de

en gardant à l'esprit la facilité de

marqueurs sur l'espace de jeu, tout en gardant la fonction généraliste de "teleport".

La plupart des commandes possèdent aussi des raccourcis (ex. "waypoint" -> "wp") permettant aux utilisateurs aguerris de la console de faire plus de choses plus rapidement. Ces raccourcis sont seulement montrés lorsque l'utilisateur demande de l'aide sur la commande en question, pour ne pas saturer l'utilisateur d'informations inutiles lors du premier lancement de l'outil

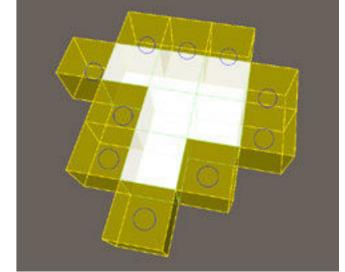
Le développement d'une nouvelle commande à été rendu très simple, chaque commande étant un type de classe instancié lors de son appel, et ayant accès à l'entièreté des fonctions Unity grâce à un GameObject servant de liaison accessible n'importe ou dans le code.

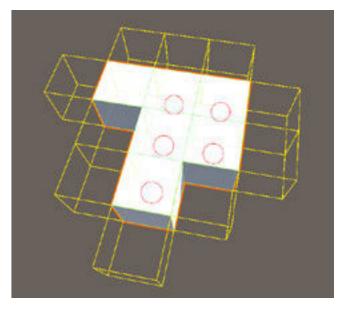
COMPONENT BUILDING TOOL

Afin de faciliter la création des components et de pouvoir itérer rapidement avec différentes formes et propriétés, nous avons créé un outil en deux parties.

Dans un premier temps, nous avons l'éditeur qui permet de modifier la forme du component. Grâce à des boutons placés sur le bord de la forme, nous pouvons rajouter des blocs à la forme. Une fois placé, un algorithme vient créer les nouvelles bordures de la forme en vérifiant les points adjacents à chaque bloc. En appuyant sur shift, nous pouvons détruire les blocs du component. Des boutons viennent alors apparaître sur les blocs qui forment le component.

En appuyant sur ces boutons, le bloc disparaît et un algorithme de «propagation» vient vérifier que tous les blocs restants sont connectés entre eux et détruit ceux qui ne le sont pas.







Dans un second temps, une fenêtre permet de voir les components présents dans la scène et d'éditer leurs propriétés rapidement. Sur la gauche, la fenêtre affiche tous les components présents ainsi que leur type. Sur la droite, nous pouvons éditer les détails du component. Cette fenêtre a pour but de remplacer l'éditeur lors des tests, en montrant uniquement les informations importantes pour le component.

186 - Cluster 71

Technical

Sprite Tool

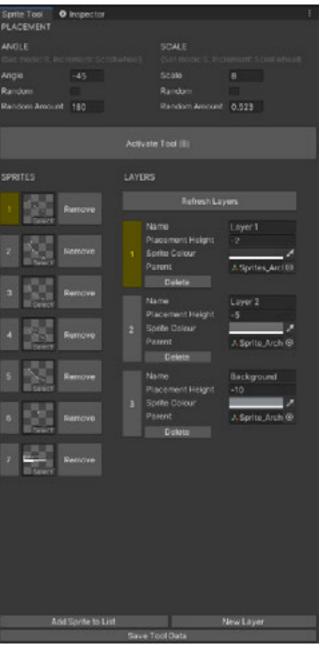
Enfin, nous avous également crée un tool pour faciliter le placement des sprites d'environnement. En effet, une grande quantité de sprites doivent être placés, il y de plusieurs sprites différents, et ils sont placés sur différents plans. Les sprites des différents plans n'ont pas la même couleur et la même hauteur.

Ainsi, le tool permet de préparer une liste de presets de sprites et de layers qui contiennent les informations de couleur, de hauteur ainsi que le parent des sprites à placer. Lorsque le tool est actif, si l'utilisateur clique dans la fenêtre scène, le sprite sélectionné sera placé à l'endroit du clic et selon les paramètres du layer sélectionné.

Il permet également de set un angle et une scale pour les sprites placés, avec la possibilité d'activer un random à l'interval réglable sur ceux-ci.

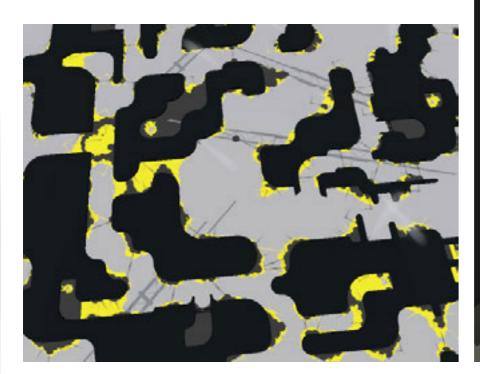
Pour augmenter le gain de temps apporté par l'outil, il nous a semblé important de créer des raccourcis clavier. L'utilisateur peut donc sélectionner les sprites dans la liste avec 1, 2, 3... et les layers avec Shift + 1, 2, 3... Le tool est toggle avec B et le joueur peut appuyer sur R ou S puis bouger la molette de la souris pour régler respectivement l'angle et la scale des sprites placés.

Aussi, pour rendre l'outil plus clair, nous avons intégré des feedbacks simples dans la tool window. Les boutons de sélection des sprites et layers, le bouton d'activation du tool et les champs de l'angle et de la scale brillent en jaune quand ils sont actifs. De plus, une preview translucide du sprite qui sera placé suit le la souris du joueurs lorsque l'outil est actif. Indiquant sa taille, sa rotation, sa couleur et sa hauteur.



Enfin, en récupérant le parent des sprites d'un layer donné, le tool permet également de changer la couleur de tout un layer d'un coup. Donnant la possibilité aux environment artists d'itérer très rapidement et facilement.

En somme, la création de ce tool a permi de multiplier significativement la vitesse de production de l'environnement. Sans lui, nous n'aurions jamais pu finir cette passe de visuel sur toutes les régions.





188 - Cluster 71

Technical



