

---

GAME  
DESIGN  
DOCUMENT

LODRASQE



# SOMMAIRE

INTRODUCTION.....	3
Équipe.....	3
Intentions.....	4
Concept.....	4

GAME DESIGN.....	6
Intentions globales.....	6
Règles d'or.....	8
Références.....	9
3Cs.....	11
Conditions de victoire/défaite.....	13
Macro objectif.....	13

Mécaniques de jeu.....	14
Armes.....	16
Boss.....	20
Boucles OCR.....	34
Onboarding.....	38
Design de la difficulté.....	44

LEVEL DESIGN.....	51
Intentions globales.....	51
Déroulés des niveaux.....	52

DÉVELOPPEMENT.....	57
Outil de production.....	57
Problèmes techniques rencontrés.....	59
CHARTE GRAPHIQUE.....	61
Intentions.....	61
Processus de recherche.....	62
Moyens techniques.....	64
Animations.....	66
Boss.....	68
Blocs et Cœurs.....	74
Environnement.....	77
Armes.....	85
VFX.....	88
Couleurs.....	94
Visuels du jeu.....	96
CHARTE SONORE.....	98
Intentions.....	98
Mécanique sonore.....	99
Références et créations sonores.....	99
Sound List.....	102
Explications des events sonores FMOD.....	106
Timeline de gameplay.....	110
FUTUR DU PROJET.....	111

# INTRODUCTION

## ÉQUIPE



**Pierre-Louis  
BOURRY**

Programmation  
Game design  
Sound design



**Anastasia  
KASYANIK**

Direction artistique  
Graphismes  
Level design



**Thomas  
LOURTIES**

Game design  
Documentation  
Level design



**Cédric  
GRIMONPONT**

Gestion de projet  
Documentation  
Sound design



**Agnès  
PRATS**

Gestion de projet  
Game design  
Documentation

■ Dans *Lodrasqe*, le joueur affronte une entité composée de centaines de Blocs. Avec ses deux armes à feu, il a la possibilité de détruire les points vitaux protégés par lesdits Blocs, dispersables grâce aux tirs. Cependant, ils auront tendance à revenir vers leur position initiale.

# INTENTIONS

---

L'intention de base de notre jeu est d'exploiter un système de répulsion/retour d'objets.

Une fois le jouet conçu, nous voulions créer un jeu de combat de boss en série, le tout en arène, avec un haut niveau de compétence requis et une expérience de jeu frénétique.

## CONCEPT

---

### SUJET

*Lodrasqe* est un projet de 3e année de game design à l'ICAN. La tâche étant de concevoir un projet de jeu, jouable et présentable, sur la période d'octobre 2019 à fin mai 2020.

Le sujet de lancement de projet est "Trace". Différents points ont été étudiés : la trace polluante du jeu vidéo, les traces sonores, les traces de changement de la matière, la

trace des médias, ou encore les traces autour de l'esprit. En parallèle de cela, des idées ont émergées, axant l'action de tracer une trajectoire et de revenir sur ses pas.

À la suite de plusieurs tests, l'élément principal retenu est l'effet de revenir sur ses traces, revenir en arrière/d'où on vient, à son point d'origine.

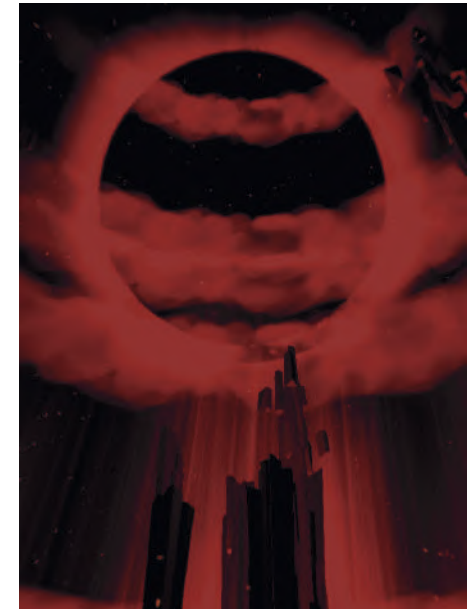
## PITCH

*Lodrasqe*, une créature ancestrale, endormie depuis des millénaires. Vous devrez tout faire pour la détruire, même si cela vous en coûte la vie. Aidez-vous de vos armes surpuissantes pour mener à bien des combats à la fois dans les airs et au sol. Préparez-vous à une expérience exigeante qui mettra vos nerfs à rude épreuve...



## FICHE TECHNIQUE

NOMBRE DE JOUEURS	_____	1
PLATEFORME	_____	PC
GENRE	_____	Action
TYPE DE JEU	_____	FPS
PUBLIC CIBLE	_____	Amateurs de FPS (hardcore gamers), joueurs complétionnistes



# GAME DESIGN

## INTENTIONS GLOBALES



### RAPPORTS A LA TRACE

Avec pour thème "Trace", nous avons choisi de nous concentrer sur les actions du joueur. Lorsque celui-ci réussit à toucher une partie de l'ennemi, les Blocs qui le composent sont expulsés, puis reviennent à leur position initiale autour de l'entité. Le projectile que crée le joueur a une réaction visuelle et physique immédiates. Les actions de ce dernier provoquent des traces visibles, et les Blocs se suivent pour rester fidèles à la composition de l'entité.



## INTENTIONS D'EXPERIENCE

Nous souhaitons un rythme rapide et un challenge simple à comprendre, mais difficile à surmonter. Nous voulions que le joueur soit tout le temps en alerte et en mouvement face à une entité qui le poursuit. À la suite de la création de la mécanique des Blocs expulsables revenant toujours vers leur position d'origine, nous avons décidé d'orienter le jeu vers une confrontation à un ennemi unique à chaque phase de jeu. Une entité puissante, très grande par rapport au joueur, d'une taille comparable à celle d'un boss de fin de niveau. Ce qui a motivé notre choix est l'expérience résultant d'un combat avec un seul gros ennemi, plutôt que de tirer sur plusieurs petites entités. Cela permettait de créer le rythme soutenu que nous recherchions, c'est-à-dire mettre le joueur en alerte permanente.

Cette direction permettait de répondre à des contraintes techniques que nous imposait notre système (nombre de Blocs limité pour garder un jeu fluide par exemple), tout en proposant un ennemi avec une forme et un comportement aboutis.

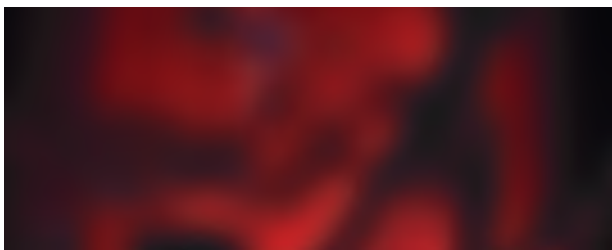
Enfin, notre tension ludique vient de la dissémination et la reconstruction entraînées par l'impact des tirs explosifs, émis par les armes de l'avatar sur les Blocs.





# RÈGLES D'OR

Les règles suivantes ont été des éléments de design considérés comme le cœur du projet et, par conséquent, non modifiables. Elles représentent les éléments que nous avons voulu exploiter et la ligne directrice de leur développement.



BLOCS	<ul style="list-style-type: none"><li>- Ils sont physiques</li><li>- Le joueur peut les déplacer</li><li>- Ils ont une position relative à un Cœur vers lequel ils tendent à revenir constamment</li></ul>
COEURS	<ul style="list-style-type: none"><li>- Ils sont consistants</li><li>- Ils peuvent être détruits</li><li>- Des Blocs peuvent y être rattachés</li><li>- S'ils sont détruits, les Blocs qui y étaient rattachés le sont aussi</li></ul>
TIR	<ul style="list-style-type: none"><li>- Le joueur peut invoquer une mécanique de tir qui crée une explosion repoussant les Blocs visés</li><li>- Le joueur a accès à un tir de projectile qui, au contact des Cœurs, les détruits ou les dégrade jusqu'à destruction</li></ul>
CAMÉRA	<ul style="list-style-type: none"><li>- Le joueur voit l'environnement en 3 dimensions</li><li>- Le joueur peut se déplacer dans les 3 dimensions</li></ul>

## RÉFÉRENCES

---

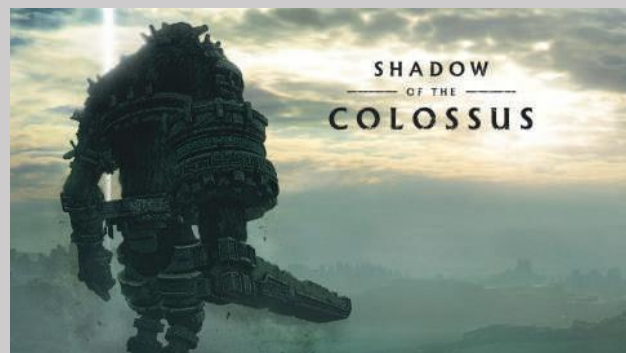


### DEVIL DAGGERS

*Devil Daggers* a été notre référence principale. Plusieurs choses nous ont intéressées dans ce jeu, comme le rythme soutenu, puisque le joueur doit toujours rester en mouvement du fait qu'il soit constamment poursuivi par plusieurs ennemis. Certains d'eux forment d'ailleurs une masse, qui semble être un seul ennemi. Cela nous a inspiré sur la manière de designer le Lodrasq (agencement des Blocs et comportement). De plus, l'une de nos armes, le Mitrapompe, est aussi inspirée de l'arme principale du jeu.

---

---



### SHADOW OF THE COLOSSUS

*Shadow of the Colossus* nous a permis d'analyser et d'exploiter certains codes, comme la sensation de grandeur ou de puissance des boss sur le joueur.

---

---

## FURI

*Furi* nous a inspiré sur la gestion de la difficulté. *Lodrasqe* et *Furi* ont plusieurs gros points communs, comme le combat de boss à phases multiples en arène, se voulant exigeant. Cela nous a apporté une réflexion sur le type de punitivité que nous voulions pour notre jeu. Cependant, nous sommes restés sur un schéma plus proche de *Devil Daggers* que de *Furi*.

---

---

## TEAM FORTRESS 2

*Team Fortress 2*, et plus précisément l'arme de la classe Demoman, a directement inspiré le Lance-grenades, ainsi que la mécanique de rocket jump par l'explosion des munitions.

---

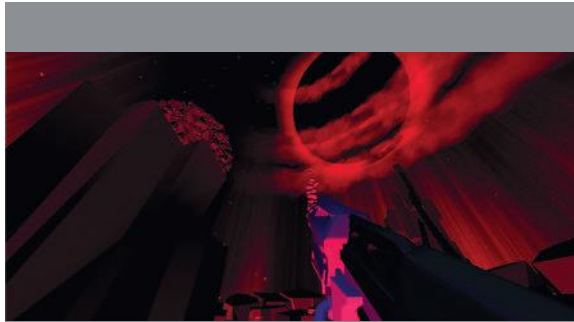


## 3 C s



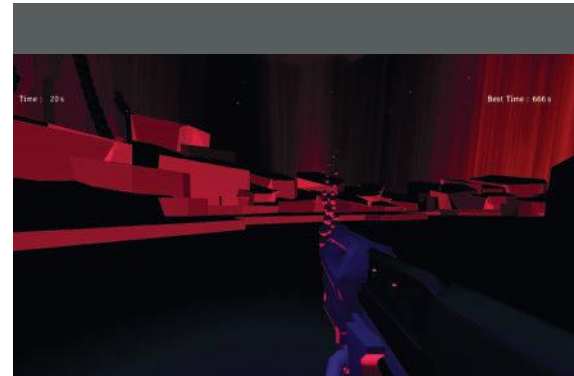
### CAMERA

Vue à la première personne dans un environnement en 3D.

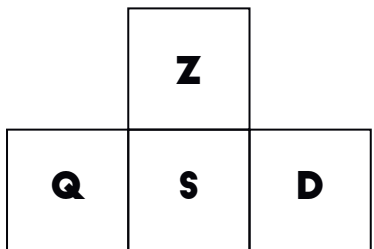


### CHARACTER

L'avatar peut se déplacer et sauter. Il peut tirer des projectiles à l'aide d'armes à feu et se projeter avec ses explosions (rocket jump).



## CONTROLS



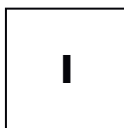
**MAINTENIR ENFONCÉ**  
Déplace l'avatar par rapport à la caméra de manière horizontale



**DÉPLACER**  
Change la direction de la caméra

**BARRE ESPACE**

**MAINTENIR ENFONCER**  
LORSQUE L'AVATAR TOUCHE LE SOL  
Fait sauter l'avatar



**ENFONCER**  
Équipe le Mitrapompe



**ENFONCER**  
Équipe le Lance-grenades



**FAIRE ROULER**  
Change d'arme



**ENFONCER/MAINTENIR ENFONCÉ AVEC MITRAPOMPE ÉQUIPÉ**  
Tire le projectile de l'arme

**ENFONCER AVEC LANCE-GRENADES ÉQUIPÉ**  
Tire le projectile de l'arme

## CONDITIONS DE VICTOIRE/DÉFAITE

---

### DÉFAITE

- Avatar touché par un Cœur du boss ;
- Avatar rentré en contact avec une death zone.

### VICTOIRE

Destruction du dernier Cœur du Lodrasqe.

## MACRO OBJECTIF

---

Le joueur a pour objectif principal de détruire les différentes Formes du Lodrasqe. Il va d'abord tenter de détruire la première forme, puis les suivantes une à une jusqu'à clore le combat.

---

# MÉCANIQUES DE JEU

---

## AVATAR

---

### ORIENTATION DE LA CAMERA

Le joueur peut orienter la caméra de son avatar. Elle peut être tournée sur elle-même par rapport à ce dernier via le déplacement de la souris. La mécanique se comporte comme dans un FPS classique.



---

### DEPLACEMENTS AU SOL

Le joueur peut déplacer son avatar avec les touches Z (avant), Q (gauche), S (derrière) et D (droite). L'avatar accélère dans la direction ciblée jusqu'à obtenir une vitesse maximale. Les déplacements ne peuvent pas faire passer l'avatar au travers d'un objet solide.



---

### GRAVITE

La gravité attire continuellement l'avatar vers le bas, jusqu'à ce qu'il rencontre un objet solide.



---

## **SAUT**

Le joueur peut appuyer sur la barre ESPACE pour obtenir une force de déplacement verticale, plus puissante que la gravité. Le joueur peut utiliser cette mécanique uniquement si l'avatar touche une surface solide sur laquelle il peut tenir immobile.

---

---

## **DEPLACEMENTS EN L'AIR**

Le joueur peut diriger son avatar dans les airs de la même manière qu'au sol. L'avatar accélère dans la direction ciblée jusqu'à obtenir une vitesse maximale. Si l'avatar obtient une vitesse supérieure à ce maximum, donc autrement qu'avec ses inputs, il conserve cette vitesse jusqu'à ce qu'il soit ralenti par un obstacle ou par une redirection brusque. Ces déplacements ne peuvent pas faire passer l'avatar au travers d'un objet solide.

---

---

## **VALEURS DE L'AVATAR**

- Taille : 2,5m\*5m\*2,5 m
  - Vitesse au sol : 50 m/s
  - Vitesse de saut : 25 m/s
  - Accélération : 250 m/s<sup>2</sup>
-



# ARMES

Nous voulions que le joueur puisse choisir sa méthode de confrontation face au boss. Une arme qui répondait à tous les besoins de manière efficace ne nous semblait pas optimal pour un jeu que nous voulions difficile et frénétique. En avoir une plus utilitaire et une autre à utilisation plus agressive, nous a semblé intéressant, car cela oblige le joueur à réfléchir à ce qu'il doit utiliser en fonction de la situation. Le changement d'arme renforce d'ailleurs le côté frénétique du jeu.

## LANCE-GRENADES

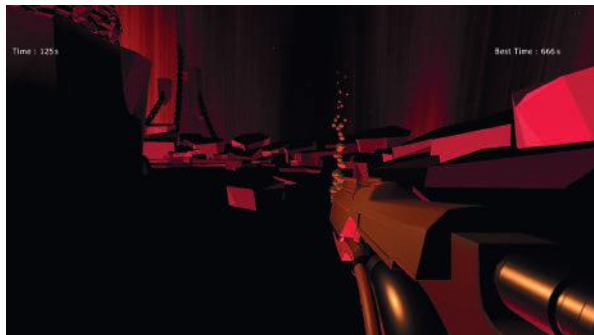
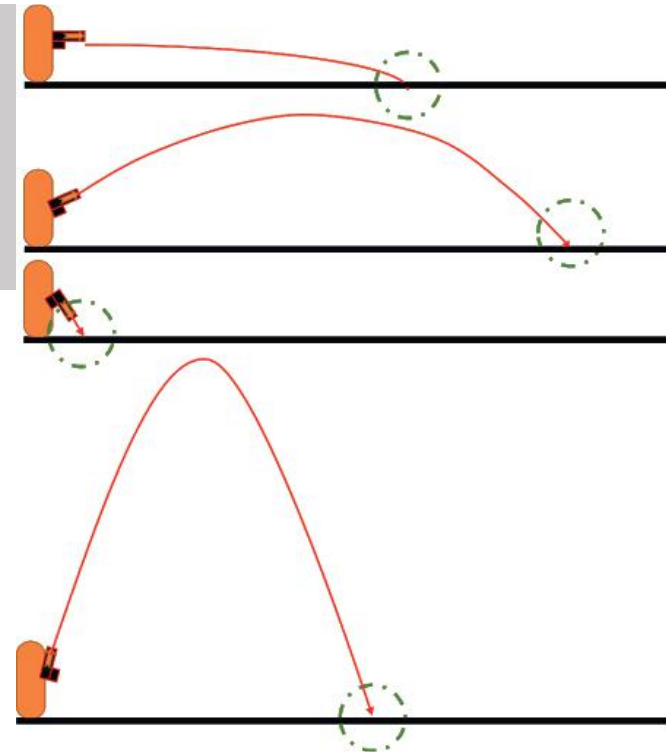


Schéma des différentes trajectoires du projectile selon l'orientation de l'arme au moment du tir.



Le Lance-grenades est une arme utilitaire. Le joueur dirige le départ du projectile en orientant sa caméra. Il crée un projectile en appuyant sur le clic gauche. Il peut invoquer à nouveau cette mécanique un certain temps après la dernière utilisation (cool down), en appuyant à nouveau sur la même touche ou en la maintenant. Le projectile part du centre de l'écran, à l'opposé de l'avatar. Il est affecté par la gravité, donc est attiré vers le bas. Au contact de tout objet solide, le projectile crée une zone. Elle a pour effet de projeter à l'opposé de son centre les objets physiques. Ce projectile inflige aussi des dégâts aux Cœurs de l'entité ennemie.

---


L'intention de création de cette arme provient du fun que crée l'action de repousser des Blocs en masse. La mécanique de gravité rend plus complexe le challenge de précision, mais permet en retour de diversifier les utilisations possibles.

---

L'arme est faite pour utiliser le rocket jump, ainsi que pour dégager la voie au joueur. Nous l'avons rendue moins efficace sur les dommages qu'elle inflige aux Cœurs car c'est la fonction du Mitrapompe (Cf Mitrapompe p. 18). Il offre cependant un meilleur rapport efficacité/distance que ce dernier.

---

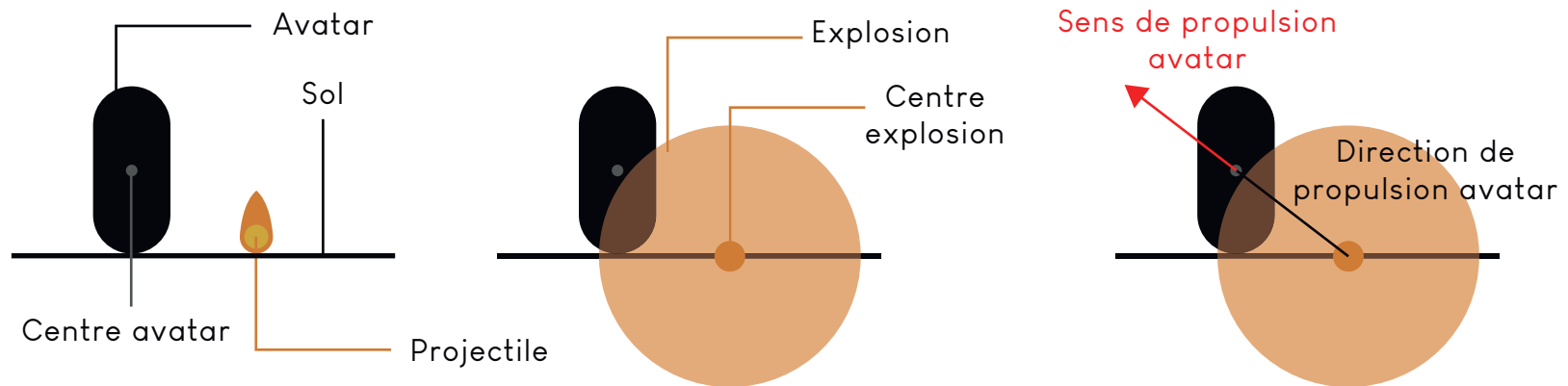
## VALEURS DE L'ARME

- Damage par munition : 20
  - Munitions par tir: 1
  - Cadence de tir : 2-3 tirs par seconde, semi-automatique
  - Intervalle de dispersion : 0 degrés
  - Vitesse de projectile : 250 m/s
  - Gravité :  $0,5\text{m/s}^2$
  - Vitesse de répulsion Blocs : 50 m/s
  - Vitesse répulsive sur l'avatar : 64 m/s
- 
- 

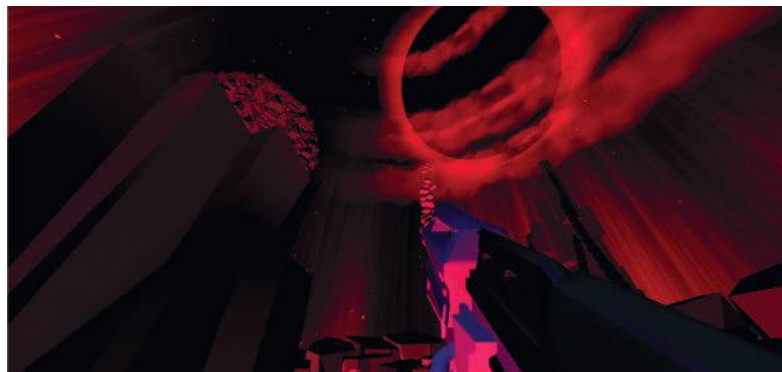
## ROCKET JUMP

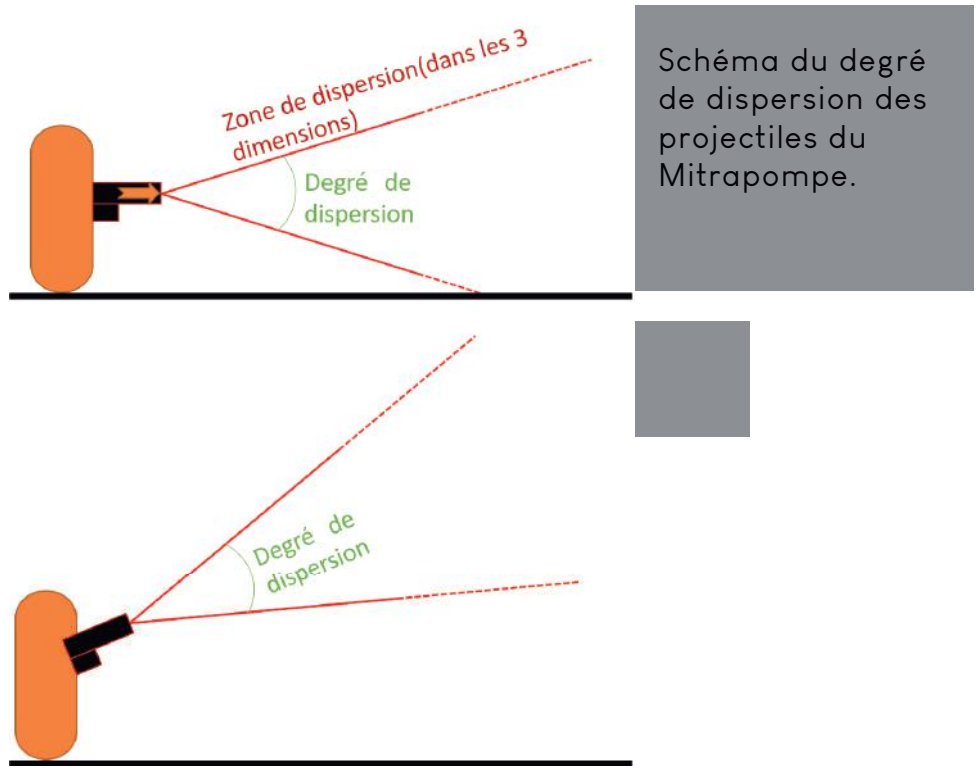
Le rocket jump vient répondre à un besoin de déplacement fun et efficace, tout en étant différent du déplacement de base. Sa maîtrise récompense le joueur par une vitesse accrue, par de nouveaux angles d'attaque et de tir, ainsi que par une accessibilité à de nouveaux lieux.

Le rocket jump est la mécanique qui résulte du contact de la zone d'explosion et de l'avatar. Lorsque l'avatar entre en contact avec cette zone, il est projeté à l'opposé de l'épicentre de l'explosion.



## MITRAPOMPE





Le Mitrapompe est l'arme offensive principale. Le joueur dirige le canon de l'arme en orientant sa caméra. Appuyer sur le clic gauche crée un certain nombre de projectiles, à condition d'avoir équipé l'arme Mitrapompe. Le joueur peut maintenir la commande pour continuer à tirer à un rythme fixe. Les projectiles partent depuis le centre de l'écran, à l'opposé de l'avatar, en suivant un cône de dispersion. Ces projectiles infligent des dégâts aux Cœurs de l'entité.

Le Mitrapompe répond au besoin d'avoir une mécanique rapide, efficace et frénétique pour arriver à détruire rapidement les Cœurs. Plus le joueur est près de sa cible, plus le tir est efficace. Le joueur doit prendre des risques pour s'en servir au maximum de son potentiel. Les dommages qu'il cause sont grandement réduits à moyenne et à longue distance.

---

## VALEURS DE L'ARME

- Dommages par munition : 1,5
  - Munitions par tir: 25
  - Cadence de tir : 2-3 tirs par seconde, automatique
  - Intervalle de dispersion des munitions: cône de 30 degrés
  - Vitesse des projectiles : 500 m/s
  - Gravité : 0 m/s<sup>2</sup>
- 

## CHANGEMENT D'ARME

Le joueur peut changer d'arme instantanément en faisant rouler la roulette de la souris ou en appuyant sur E.

---

## BOSS

---

Le Lodrasqe est une entité que le joueur doit détruire. Ce boss a plusieurs formes. Il ne peut y avoir qu'une seule forme dans l'espace de jeu à la fois.

Nous recherchions d'abord un effet wahoo pour la créature, un fantôme de joueur ou une idée marrante. Ensuite, nous devions concevoir une apparence à cette créature. Finalement, nous voulions trouver un comportement à notre boss, sous forme de patterns. Si une seule de ces étapes ne fonctionnait pas nous reprenions le processus de création à zéro.

---

## COMPOSITION

L'ennemi est constitué de 2 types d'éléments: les Coeurs et les Blocs. Il est composé de plusieurs Blocs espacés les uns des autres. Nous avons choisi de jouer avec les vides et les pleins plutôt que d'avoir un modèle composé de Blocs collés les uns aux autres. Cela sert notre comportement de dissémination.



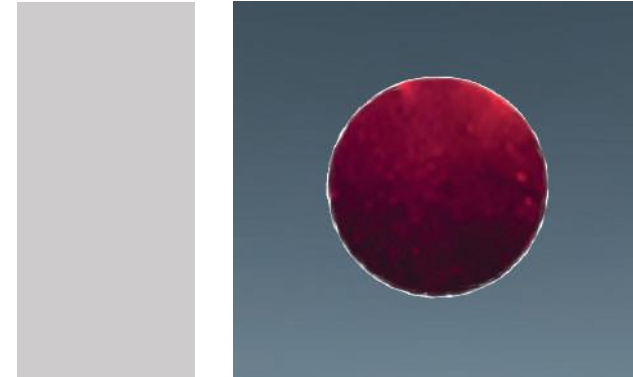
---

## REGLES DE CONCEPTION

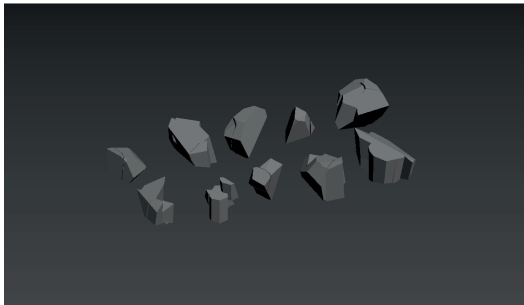
- Le Lodrasqe est divisé en Formes, elles-même divisibles en Phases. Une Forme débute avec un certain nombre de Cœurs. Ce nombre doit être cohérent avec la forme de l'entité
- Le Lodrasqe change de Phase en ayant un certain nombre de Cœurs détruits
- Le Lodrasqe change de Forme en ayant tous les Cœurs d'une Phase détruits
- Chaque Phase et Forme doivent se distinguer par rapport aux précédentes
- Chaque Phase et Forme doivent proposer un challenge de difficulté supérieur au précédent
- Chaque Forme ne peut pas avoir plus de 2 Phases
- Chaque Forme propose un set de patterns de mouvements, qui évolue et/ou varie en changeant de Phase
- Le Boss détruit l'avatar si un de ses Cœurs le touchent

---

## COEURS



- Chaque Cœur est associé à plusieurs Blocs
- Ils ont un nombre de points de vie défini
- Lorsque les points de vie d'un Cœur sont à 0, celui-ci est détruit et les Blocs associés aussi
- Quand un Cœur touche l'avatar, il est détruit



## BLOCS

- Ces objets sont rattachés à un Cœur
  - Ils ont une position de base relative à ce dernier
  - Ils cherchent constamment à regagner leur position initiale
-

---

## MECANIQUE DE REPULSION

Lorsqu'un Bloc est projeté par une explosion, celui-ci arrête de revenir vers sa position. Il est alors dit "désactivé". Un bloc qui se "réactive" va chercher à regagner son point d'origine par rapport au Cœur. Ce comportement se déclenche quand un Bloc a été désactivé 5 secondes ou qu'il est à 80m ou plus de distance.

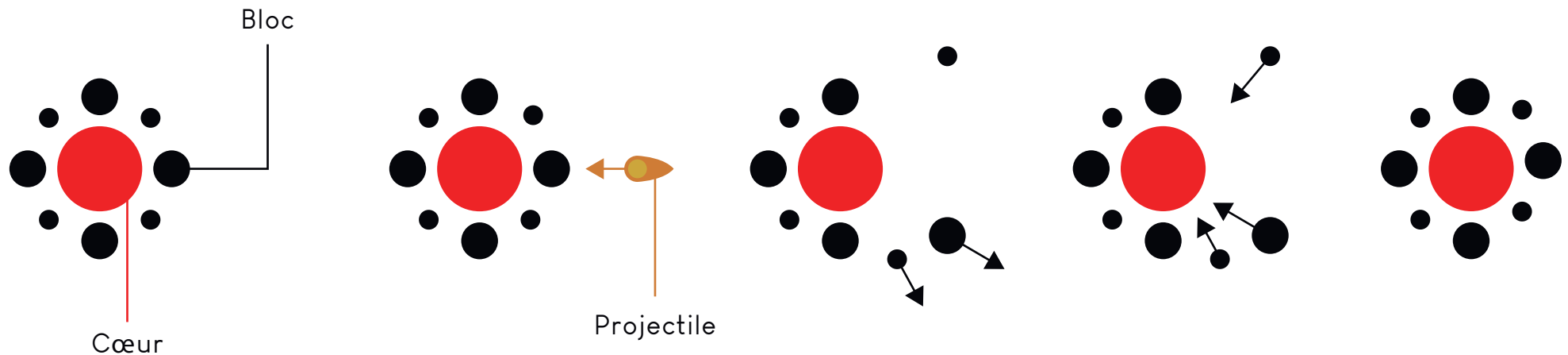
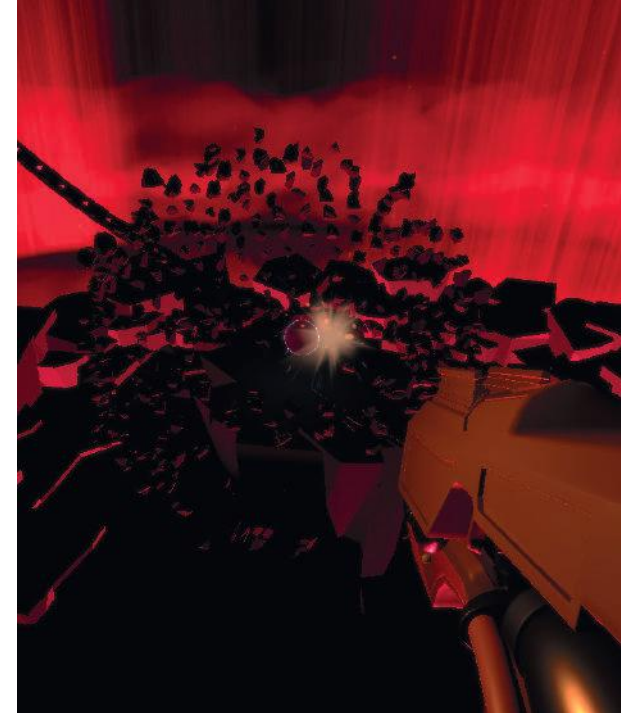

---

---

## MECANIQUE D'ATTRACTION

Les Blocs se dirigent constamment vers un point relatif au Cœur. Plus un Bloc est loin de sa position initiale, plus il va accélérer, et inversement. Arrivé à un certain seuil, il tend à rester à sa position d'origine.

---



# FORMES, PHASES ET COMPORTEMENTS

## STRUCTURE FICHE ENTITÉ

---

### INTENTION GLOBALE

La raison générale du fonctionnement de l'entité et de l'arrivée des formes dans le jeu.

---

---

### INTENTION D'APPRENTISSAGE

Ce que nous voulons que le joueur apprenne à faire ou ce qu'il doit confirmer comme apprentissage avec cette phase.

---

---

### STRUCTURE

Comment la forme est structurée de manière générale et pourquoi.

---

### NOMBRE DE COEURS

Nombre de Cœurs que la phase est censée avoir à son début.

---

---

### POINTS DE VIE DES COEURS

Le nombre de points de vie que chaque Cœur possède de base.

---

---

### DEATH ZONE

Zone qui fait perdre le joueur si l'avatar l'atteint.

---



---

## PHASE X

### **INTENTION**

L'intention de fonctionnement et d'apprentissage de la phase.

### **COMPORTEMENTS**

ACTIVATION : comment le comportement se déclenche.

ACTION : ce que l'entité fait durant ce comportement.

### **CHANGEMENT DE PHASE/ FORME/DESTRUCTION**

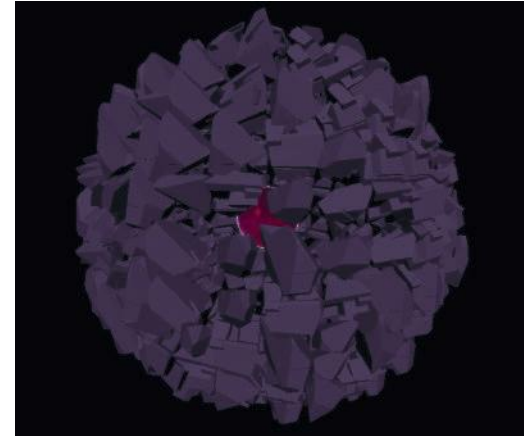
À partir de combien de Cœurs l'entité change de forme/phase ou est détruite.

### **TRANSITION**

Description de l'événement de transition.

---

## FORME 0 : TUTO



---

### **INTENTION GLOBALE**

Avoir une entité statique positionnée en hauteur pour permettre au joueur de comprendre les mécaniques de base du jeu en toute sécurité.

---

### **INTENTION D'APPRENTISSAGE**

Permet de prendre conscience du système de rocket jump, de Cœur, de tir et de répulsion de Blocs. À sa destruction, elle introduit aussi le principe de changement de forme, de phase et de niveau.

---

## **STRUCTURE**

L'entité est composée d'un seul Cœur, entouré de Blocs positionnés en sphère autour.

---

---

## **NOMBRE DE COEURS**

Un seul et au centre.

---

---

## **POINTS DE VIE DU COEUR**

40

---

---

## **DEATH ZONE**

Aucune.

---

---

## **PHASE UNIQUE**

### **INTENTION**

Avoir une entité qui permet d'expérimenter en tout sécurité le jeu sans danger, sans réel challenge.

### **COMPORTEMENT**

ACTIVATION : l'entité vient d'arriver en jeu.

ACTION : ne bouge pas.

### **CHANGEMENT DE FORME**

0 Cœurs restant.

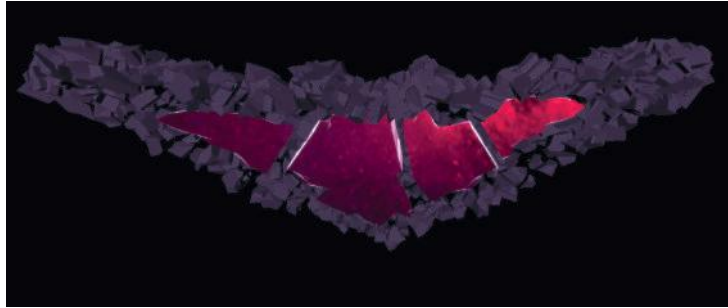
### **TRANSITION**

L'entité se détruit pour laisser apparaître progressivement la forme suivante là où elle se situait. Les chaînes se brisent, le sol tombe, se fissure et l'arène se déforme, se faisant transpercer par les piliers placés en-dessous.

---



## FORME 1 : CHAUVE-SOURAIE



---

### INTENTION GLOBALE

Créer une entité qui donne une impression de combat de toréador : se faire charger, esquiver, frapper.

---

---

### INTENTION D'APPRENTISSAGE

Permet de bien apprendre le rocket jump de manière verticale et s'habituer à un rythme d'esquive, de tir et de repos. Cette entité permet aussi d'introduire le changement forme.

---

### STRUCTURE

L'entité est plus massive au niveau du corps que des ailes, pour permettre une esquive relativement facile. Cela permet de rendre l'entité menaçante et grande sans la rendre trop dure à esquiver. Le dessous de l'entité n'a pas de Blocs de protection, pour avoir un point faible de base.

---

---

### NOMBRE DE COEURS

5 regroupés au centre.

---

---

### POINTS DE VIE DES COEURS

60 chacun.

---

---

### DEATH ZONE

Autour des Cœurs.





---

## PHASE 1

### INTENTION

Cette phase permet d'expérimenter pour la première fois les mécaniques sans qu'il y ait trop de danger. L'entité est facile à affronter par rapport aux suivantes. Le joueur peut apprendre à esquiver de manière verticale avec une précision moyenne.

### COMPORTEMENT A

ACTIVATION :

- depuis le comportement B -> quand l'avatar n'est plus dans le champs de vision du boss.
- depuis le comportement C -> quand l'avatar n'est plus dans le champs de vision du boss.

ACTION : le boss monte dans les airs tout en essayant de se retourner vers le joueur. Sa vitesse ralentit jusqu'à atteindre un minimum.

Décélération :  $16\text{m/s}^2$

Vitesse minimum :  $10\text{m/s}$

Vitesse rotation de dos :  $15 \rightarrow 30 \text{ degrés/s}$

### COMPORTEMENT B

ACTIVATION : depuis le comportement A -> quand le joueur se trouve dans le champs de vision du boss.

ACTION : le boss accélère jusqu'à une vitesse maximum et se dirige vers le joueur.

Accélération :  $80 \text{ m/s}^2$

Vitesse maximum :  $160 \text{ m/s}$

vitesse de rotation de face :  $30 \text{ à } 60 \text{ degrés/s}$

### COMPORTEMENT C

ACTIVATION : depuis le comportement B -> quand la direction du boss est alignée avec le joueur.

ACTION : le boss continue d'accélérer jusqu'à une vitesse maximum. Il arrête de corriger sa trajectoire pour aller vers le joueur.

Accélération :  $80 \text{ m/s}^2$

Vitesse maximum :  $160 \text{ m/s}$

### CHANGEMENT DE PHASE

3 Cœurs restant.

## TRANSITION

L'entité monte pour ensuite adopter le comportement de la phase suivante. Le sol s'affaisse et se déforme plus. Un 5ème pilier traverse la plateforme.



---

### PHASE 2

#### INTENTION

Le boss possède le même comportement, mais il est moins borné pour créer plus d'émergence de patterns et augmenter la difficulté.

#### COMPORTEMENT A

ACTIVATION : depuis le comportement B -> quand l'avatar n'est plus dans le champs de vision du boss.

ACTION : le boss monte dans les airs tout en essayant de se retourner vers le joueur. Il ralentit jusqu'à atteindre une vitesse minimum.

Décélération :  $16 \text{ m/s}^2$   
Vitesse minimum :  $10 \text{ m/s}$   
Vitesse rotation de dos :  $15 \rightarrow 30 \text{ degré/s}$

#### COMPORTEMENT B

ACTIVATION : depuis le comportement A -> quand le joueur se trouve dans le champs de vision du boss.

ACTION : le boss accélère jusqu'à une vitesse maximum et se dirige vers le joueur.

Accélération :  $80 \text{ m/s}^2$   
Vitesse maximum :  $160 \text{ m/s}$   
Vitesse de rotation de face :  $30 \text{ à } 60 \text{ degrés/s}$

#### CHANGEMENT DE FORME

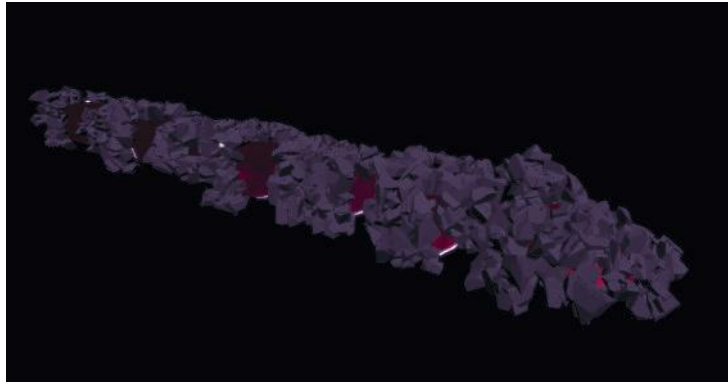
0 Cœurs restant.

#### TRANSITION

L'entité se détruit pour laisser apparaître la forme suivante là où elle se situait. Le sol s'affaisse et se déforme encore plus.

---

## FORME 2 : SERPENT



---

### INTENTION GLOBALE

Faire une entité qui met bien en avant le système de Blocs.

---

---

### INTENTION D'APPRENTISSAGE

Utiliser le rocket jump pour se maintenir à distance, apprendre à se servir du rocket jump de manière horizontale et à rediriger l'avatar.

---

---

### STRUCTURE

Chaque Cœur suit celui qui le précède et reproduit son déplacement. Le premier est le Cœur tête. S'il est détruit, le plus proche de la tête prend le relais et devient la tête à son tour.

Hormis la première tête de l'entité, qui est en partie couverte de Blocs, chaque Cœur est protégé sur 3 faces par des Blocs.

---

### NOMBRE DE COEURS

8

---

---

### POINTS DE VIE DES COEURS

90 chacun.

---



---

## **DEATH ZONE**

Autour des Cœurs.

---



---

### PHASE 1

#### **INTENTION**

Créer une entité qui poursuit le joueur et avec laquelle il est intéressant de ne pas avoir de confrontation directe.

#### **COMPORTEMENT**

ACTIVATION : arrivée en jeu.

ACTION : l'entité se dirige vers l'avatar, s'oriente vers lui, tout en ondulant verticalement.

Décélération :  $16 \text{ m/s}^2$

Vitesse minimum :  $10 \text{ m/s}$

Vitesse de rotation du dos :  $15 \rightarrow 30 \text{ degrés/s}$


---

### CHANGEMENT DE PHASE

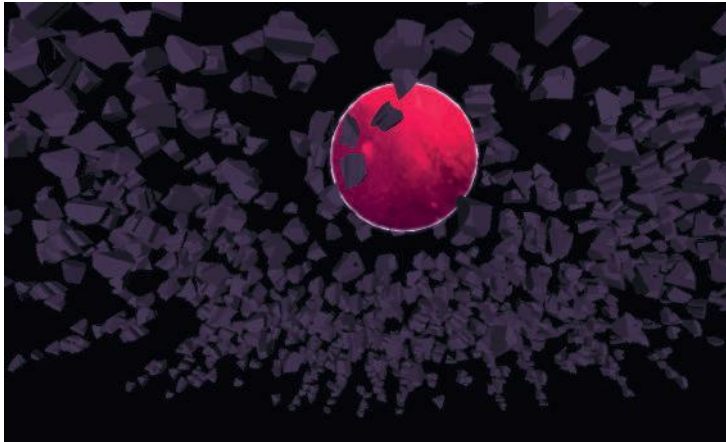
0 Cœurs restant.

#### **TRANSITION**

L'entité se détruit pour laisser apparaître la forme galaxie au-dessus de l'espace de jeu.



## FORME 3 : GALAXIE



---

### INTENTION GLOBALE

Signaler la fin du jeu, l'entité la plus imposante possible en terme de largeur. Un niveau de difficulté élevé, aucun temps mort. Avoir un boss contre la montre, avec l'état de l'espace de jeu qui se dégrade au fur et à mesure que le temps passe.

---

### INTENTION D'APPRENTISSAGE

Cette forme possède une seule phase et correspond au master du jeu. Le joueur doit confirmer une maîtrise du rocket jump et du tir.

---

### STRUCTURE

L'entité est composée d'une myriade de Blocs et d'un Cœur. Les Blocs couvrent la totalité de l'espace de jeu.

---

---

### NOMBRE DE COEURS

1 seul au centre.

---

---

### POINTS DE VIE DU COEUR

400

---

---

### DEATH ZONE

Le Cœur.

---

---

### FIN DU JEU

0 Cœurs restant.

---





---

## PHASE UNIQUE

### **COMPORTEMENT A**

ACTIVATION : l'entité vient d'arriver en jeu.

ACTION : l'entité se déplace continuellement vers le bas du niveau. Elle attire l'avatar si ce dernier est dans la zone d'attraction.

### **COMPORTEMENT B**

ACTIVATION : parallèle au comportement A. Toutes les 15 secondes si les points de vie du Cœur sont supérieurs à la moitié.

ACTION : fait apparaître 2 petites entités.

### **COMPORTEMENT C**

ACTIVATION : parallèle au comportement A. Quand l'entité atteint la moitié de ses points de vie.

ACTION : fait apparaître 8 petites entités.

### **COMPORTEMENT D**

ACTIVATION : parallèle au comportement A. Toutes les 15 secondes si les points de vie du Cœur sont inférieurs à la moitié.

ACTION : fait apparaître 3 petites entités.

### **DESTRUCTION**

0 Cœurs restant.

## FORME : PETITES ENTITÉS



---

### INTENTION GLOBALE

Avoir une entité qui ajoute un challenge de gestion de saturation de l'espace dans la dernière phase de jeu

---

---

### INTENTION D'APPRENTISSAGE

Le joueur doit apprendre à gérer ses actions entre se déplacer, réduire la menace proche et viser l'entité principale.

---

---

### STRUCTURE

Cœurs sans Blocs, se déplaçant en groupe.

---

---

### NOMBRE DE COEURS

1 seul par entité.

---

---

### POINTS DE VIE DES COEURS

20

---

---

### DEATH ZONE

Le Cœur de chaque entité.

---

---

## PHASE UNIQUE

### COMPORTEMENT


ACTIVATION : l'entité vient d'être créée.

ACTION : se dirige continuellement vers l'avatar en accélérant jusqu'à une vitesse maximum.

### DESTRUCTION

0 Cœurs restant.

---

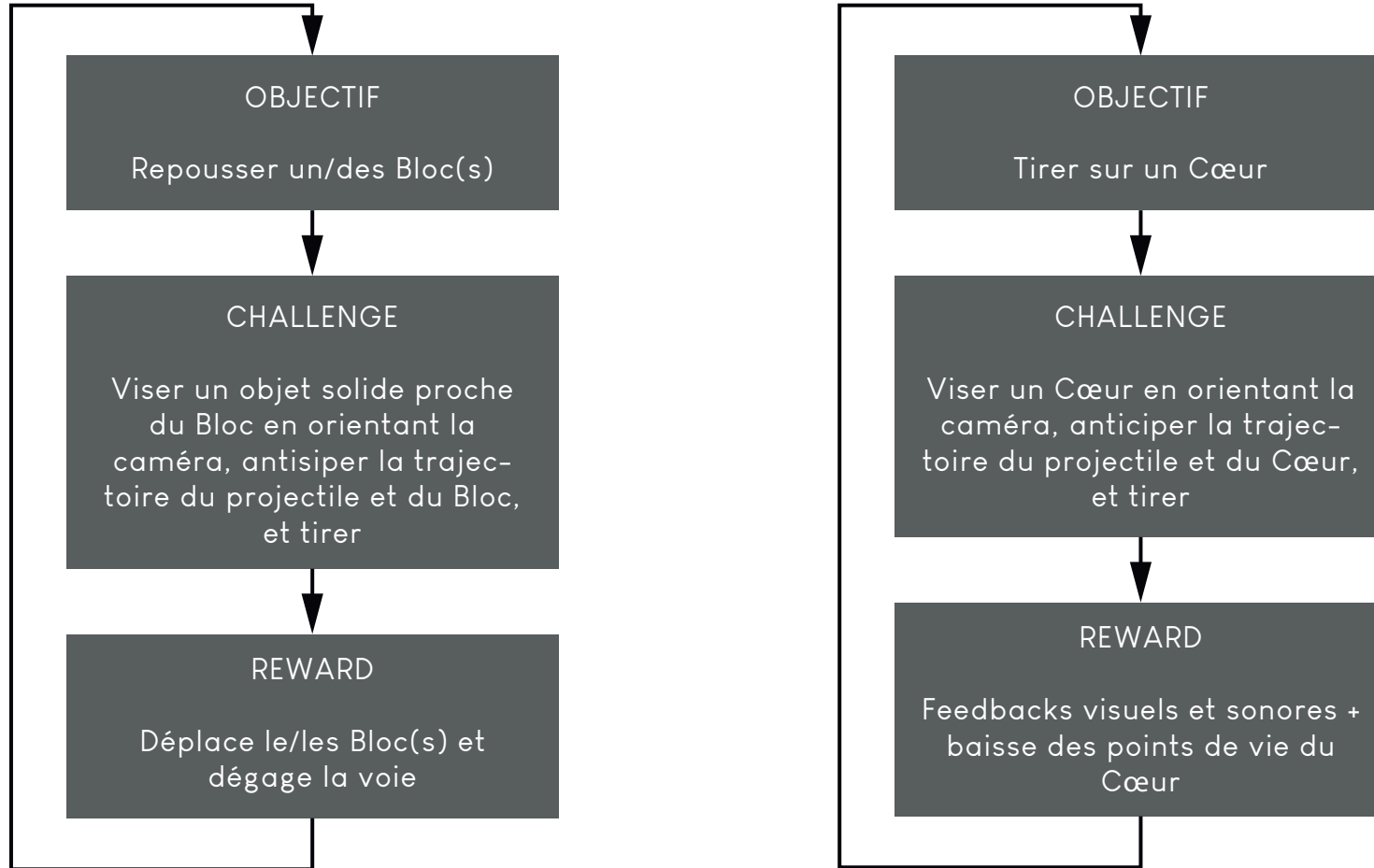


## BOUCLES OCR

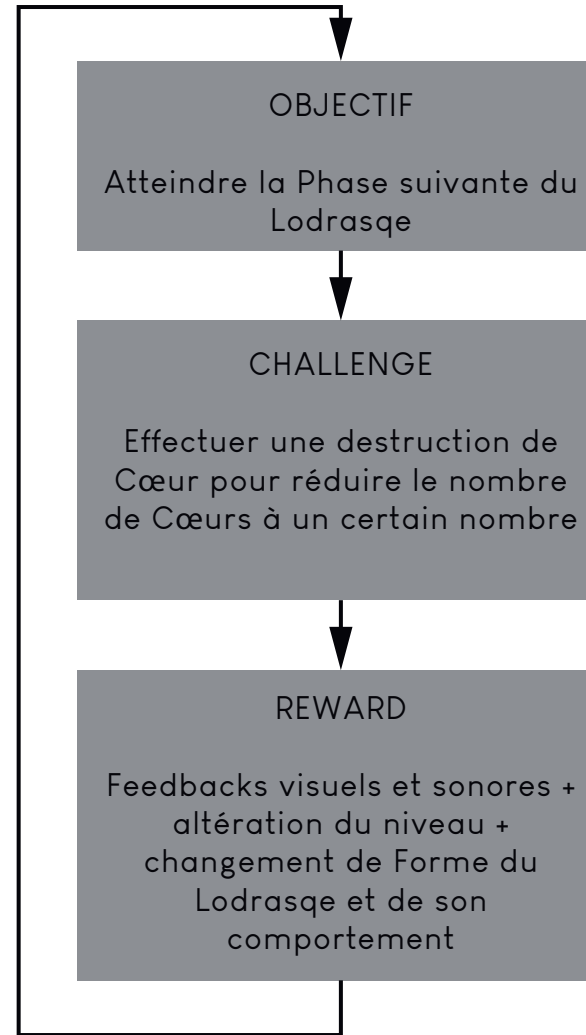
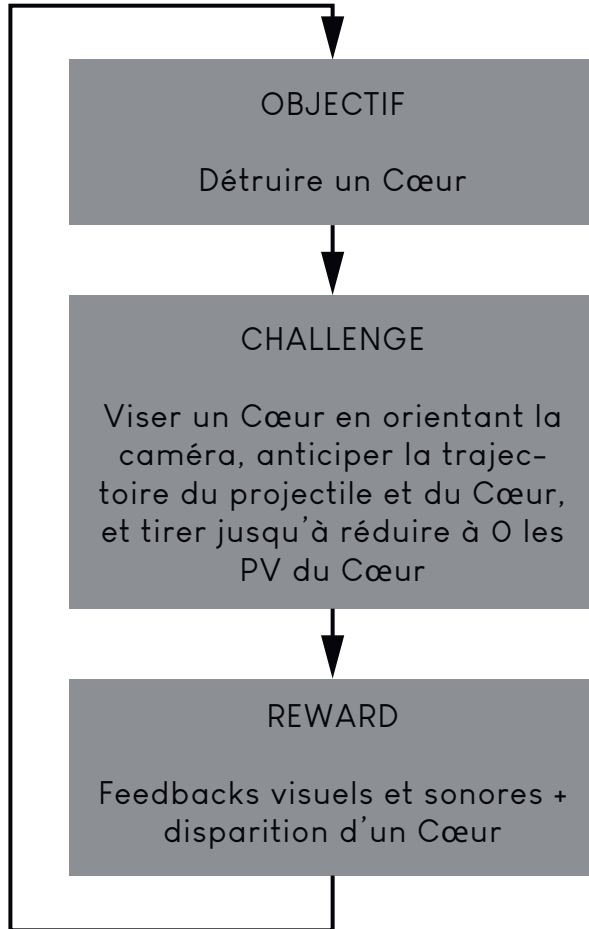
### COURT TERME



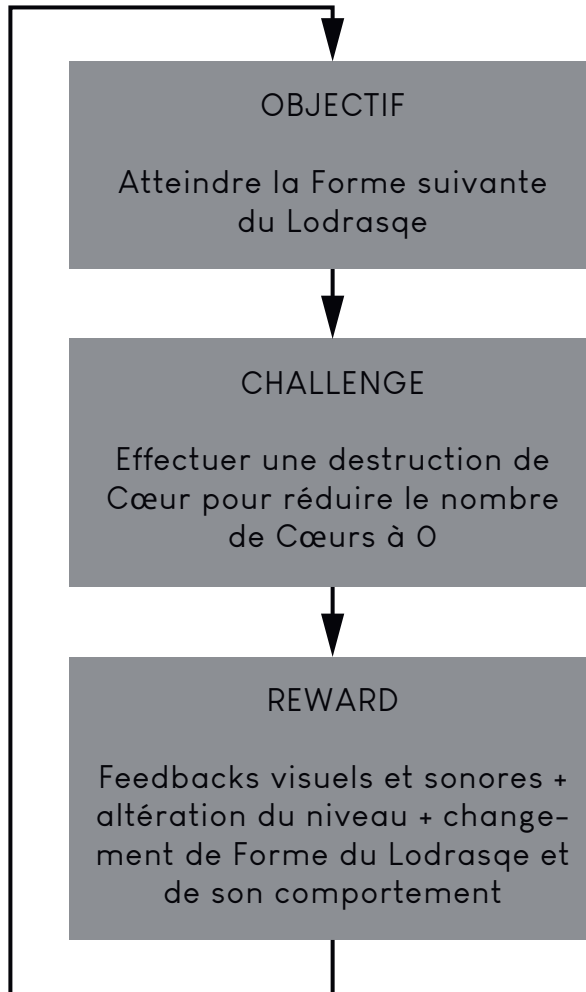
## COURT TERME



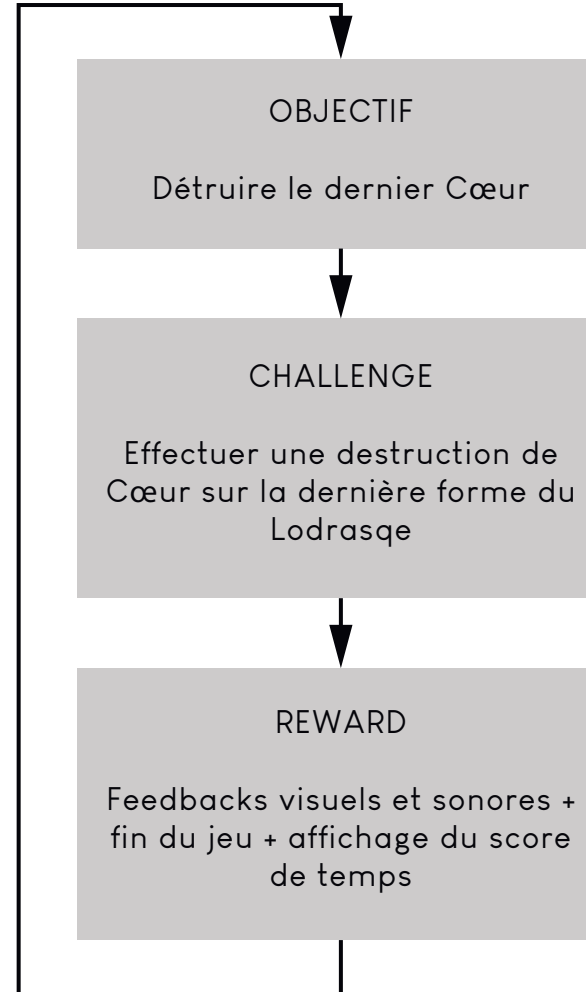
## MOYEN TERME



## MOYEN TERME



## LONG TERME



# ONBOARDING

---

## FICHE PERSONA



Pierre-Louis  
BOURRY

- 
- ÂGE : 23 ans
  - TRAVAIL : étudiant en game design
  - VIE À LA MAISON : habite avec ses parents
  - ACTIVITÉS : musique, jeux vidéo, sortir avec des amis

---

### PROFIL JOUEUR

- TYPE DE JOUEUR : hardcore gamer
  - TEMPS DE JEU : 15 heures par semaine
  - PLATEFORMES : PC
  - HABITUDES DE JEU : sur son PC durant son temps libre ou durant les cours
  - EXEMPLES DE JEUX : Devil Daggers, Dustforce, League of Legends
- 

---

### EXPERIENCES RECHERCHEES

- Expérience de FPS exigeant
  - Recherche le dépassement de soi
  - Recherche l'accomplissement de maîtriser ce que la majorité des gens ne peut pas maîtriser
- 

---

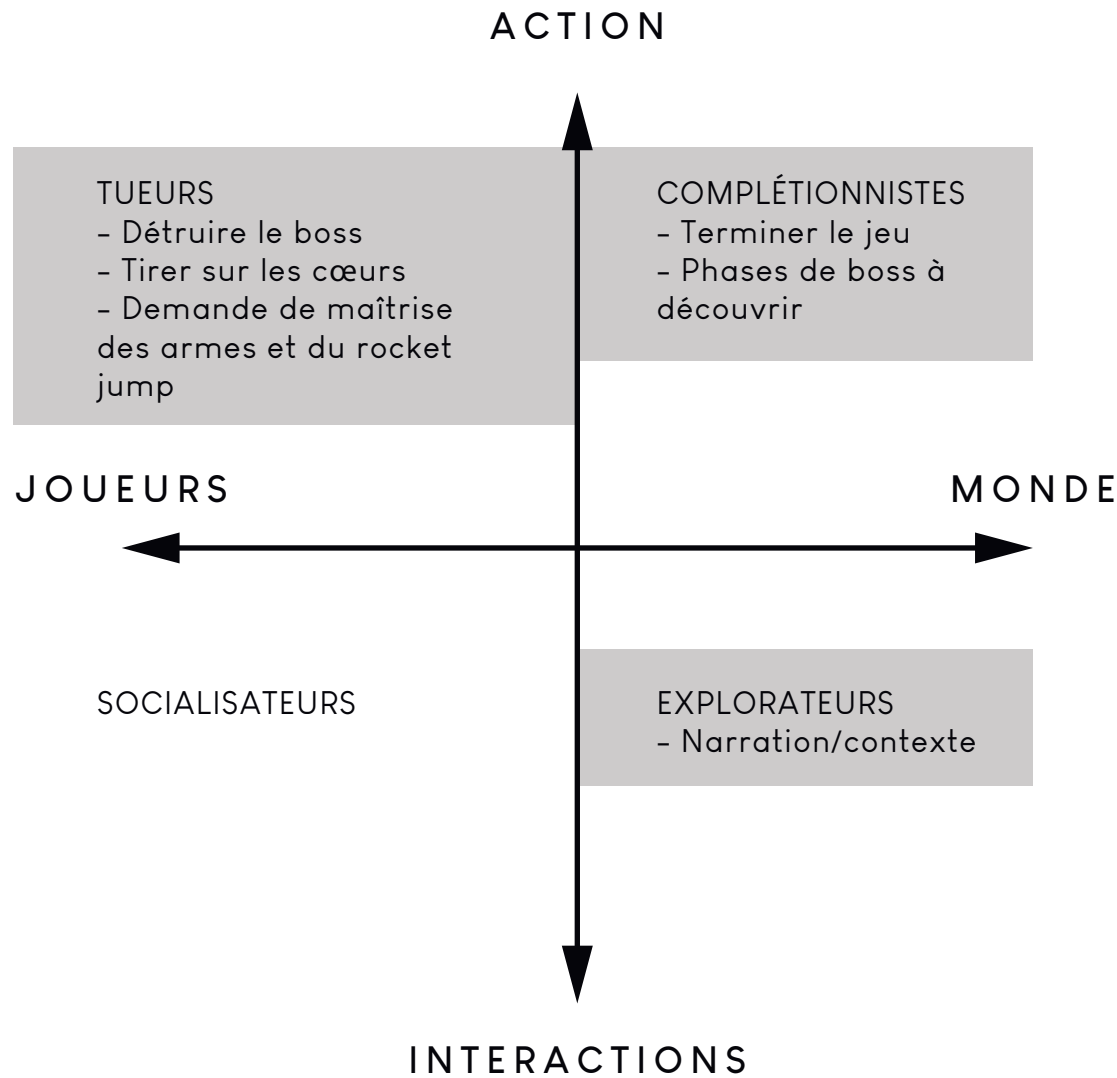
### FRUSTRATIONS

- Subir la narration (barrière au jeu)
  - Manque de granularité
- 

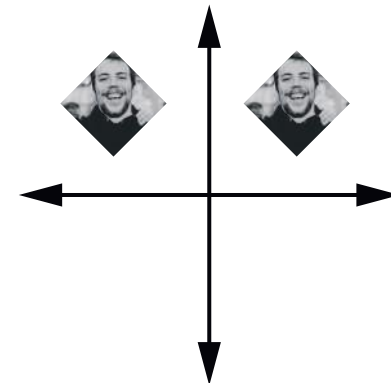
### EN RESUME

Préfère un jeu demandant beaucoup de temps et de pratique pour être maîtrisé, avec très peu de narration (seulement un contexte) pour n'avoir aucune pause involontaire et où il peut exprimer sa propre manière de jouer afin d'affirmer ses skills par lui-même.

# TYPOLOGIE



Pierre-Louis répond au profil de joueur à la fois tueur et complé-  
tionniste, il sera donc orienté  
plutôt vers les jeux d'action.





## PROFIL DU JOUEUR

### GESTION DE LA MOTIVATION

INTRINSÈQUE	EXTRINSÈQUE
<ul style="list-style-type: none"><li>- S'améliorer en continu</li><li>- Maîtriser parfaitement le système</li></ul>	<ul style="list-style-type: none"><li>- Terminer le jeu le plus vite possible</li><li>- Découvrir les phases de boss suivantes</li></ul>

### CONNAISSANCES PREALABLES

- Possède des connaissances de base au niveau des contrôles e jeux FPS (caméra, déplacements, saut, tir...) -> prise en main rapide des contrôles
- Arrive à identifier les éléments à l'écran en faisant le lien avec ceux d'autres jeux joués par le passé

---

### NOUVELLES CONNAISSANCES A ACQUERIR

- Rocket jump
- Utilisation des 2 armes
- Fonctionnement de la mécanique principale

---

### GESTION DE LA CHARGE COGNITIVE

- Le jeu fait part de feedbacks sonores et visuels lors de chaque action du joueur
- Un code couleur peut servir de repère au fil des niveaux

## INTENTIONS ONBOARDING

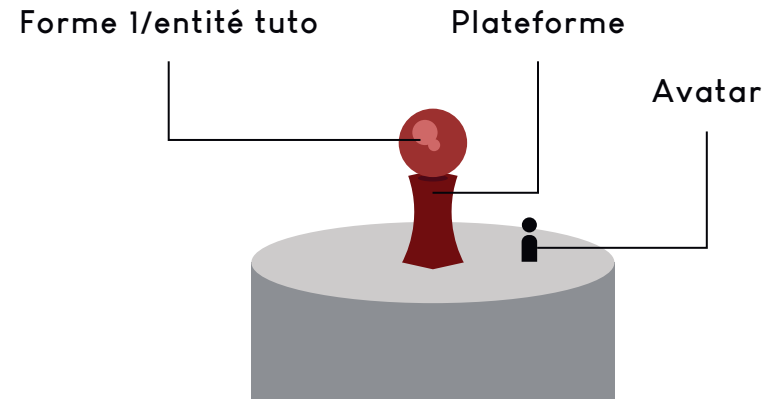
Lodrasqe étant un single-run, le joueur devra toujours repasser par les premières phases de jeu s'il relance une partie.

---

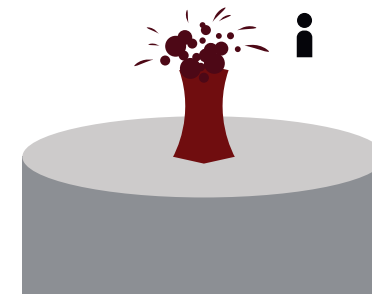
Sachant que notre jeu se doit d'être rejouable très facilement, relancer une partie doit aussi être rapide à faire, afin de répondre aux attentes de notre public cible, à savoir les hardcore gamers souhaitant affirmer leurs compétences.

---

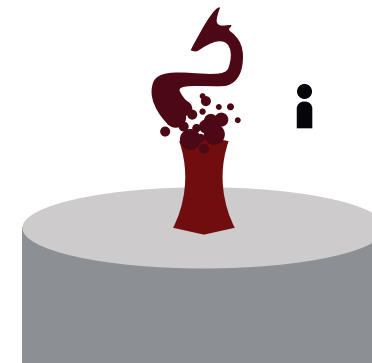
L'onboarding doit permettre au joueur d'apprendre les déplacements de base de l'avatar, le système de tir et le rocket jump.



1. Initiation au rocket jump



2. Destruction de l'entité tuto



3. Transition vers boss 1

---

## LISTE DES CONNAISSANCES A TRANSMETTRE

- Déplacement au rocket jump
- Déplacement de base de l'avatar
- Système d'armes et de tir
- Blocs se replaçant à leur point d'origine sur le mesh
- Destruction des Cœurs du boss
- Destruction des Blocs lorsqu'un Cœur est détruit

Dans *Lodrasqe*, nous voulions que le joueur découvre le jeu par l'expérience. Notre méthode de présentation de la mécanique se fait sans HUD, sans texte.

---

Dans la salle d'introduction, les deux armes présentées permettent au joueur de comprendre qu'il a deux outils différents à sa disposition. Un joueur correspondant au profil de notre public cible saura qu'il peut changer d'arme.

La phase Tuto permet de présenter 4 choses importantes au joueur :

- le rocket jump -> le joueur devra expérimenter ses armes et comprendre qu'il peut se servir de son arme explosive pour se projeter. On confirme que le joueur connaît la mécanique lorsqu'il détruit l'entité, car il est nécessaire de s'en servir pour avancer ;
- la mécanique de Blocs (attraction et répulsion) -> le joueur, pour réussir cette phase, doit forcément tirer avec l'arme explosive pour passer la sphère de Blocs, derrière laquelle se trouve le Coeur ;
- la mécanique de Coeur -> le joueur devra expérimenter en tirant sur les différents éléments de jeu pour comprendre que l'objet Coeur est endommageable. Étant le seul objet destructible par le joueur, qui est en plus protégé et mis au centre de la scène, on attend d'un joueur de FPS qu'il comprenne intuitivement qu'il doit tirer dedans ;
- le concept de Forme/Phase -> en détruisant l'entité tuto et en voyant une nouvelle arriver avec un changement de level design, le joueur apprendra alors les derniers éléments du jeu.

L'onboarding ne s'arrête cependant pas au moment où le joueur détruit la phase tuto. C'est un jeu basé sur l'apprentissage de patterns et la découverte. La première session de jeu d'un joueur a de fortes chances de s'arrêter à la Forme Chauve-souraie. Il devra recommencer un certain nombre de fois le jeu, afin d'apprendre les bases du système et déréussir les deux premières Phases.

---

L'onboarding s'arrête une fois que le joueur a compris le principe dans sa globalité. L'accès à la Forme Serpent est une bonne délimitation, car cette phase est une augmentation réelle du challenge, et le joueur devra simplement apprendre à maîtriser les mécaniques.

## DESIGN DE LA DIFFICULTÉ

---

—

*Lodrasge* a une difficulté volontairement élevée. Le jeu reprend des codes de combat de boss : apprentissage de comportements d'entité, mécaniques simples à large degré d'apprentissage, variations de level design...

## DIFFICULTÉ LIÉE AUX MÉCANIQUES JOUEUR

Les mécaniques joueur ont été designées pour avoir une grande marge de maîtrise, tout en étant fun et intuitives. Ce sont des mécaniques légèrement plus complexes que celles d'un FPS classique.

Ce qui est ici considéré comme un FPS classique est un jeu de tir à la troisième personne, dans lequel les déplacements sont classiques (ZQSD/joystick) et où le challenge principal est de viser. Les joueurs ciblés sont des hardcore gamers ou des joueurs ayant une bonne expérience des FPS classiques. Exemple : Série *Call of Duty*, *Medal Of Honor*.

Nos inspirations proviennent d'une difficulté de maîtrise proche de *Team Fortress 2* et de *Devil Daggers*. Les mécaniques sont "Easy to learn, hard to master". C'est une volonté de notre part d'avoir un écart énorme entre un novice et un bon joueur, et entre un bon joueur et un excellent joueur. Nous avons fait en sorte que le joueur doive atteindre un certain niveau de maîtrise des mécaniques pour terminer une Phase.

---

## DIFFICULTÉ LIÉE AUX PHASES ET À L'ENVIRONNEMENT

Les entités, dont le comportement est structuré en Phases, sont ce qui régule réellement la difficulté de jeu. Nous avons designé chaque Phase de sorte que le joueur acquiert un certain niveau de maîtrise d'une, ou de plusieurs mécaniques. En effet, chaque Phase a un niveau de difficulté qui lui est propre, et la passer confirme donc une certaine maîtrise des mécaniques mises en jeu.

L'environnement est la seconde variable importante dans la gestion de la difficulté. Les Phases et les Formes du niveau associé ont été désignées dans deux optiques : apprendre au joueur à se servir d'une mécanique à un certain niveau et fonctionner ensemble. Par exemple, la Forme Chauve-souris est, du fait de sa forme et de son comportement, plus intéressante à affronter au sol, avec une faible vitesse de déplacement, pour pouvoir entraîner le joueur efficacement. La Forme Serpent en revanche, avec son comportement plus agressif et sa Forme qui nécessite de le contourner pour lui tirer dessus plus efficacement, est très intéressante pour mettre le joueur en difficulté avec une vitesse élevée et sans le sol.

---

## CHAQUE PHASE A SON INTÉRÊT ET SA DIFFICULTÉ

---

### **FORME 0 - TUTO**

Confirmer que le joueur sait utiliser de manière basique le rocket jump et qu'il sait tirer sur une cible immobile proche. Elle est sans difficulté, même pour un joueur novice, exceptée, peut-être, la compréhension du rocket jump, car il n'y a pas d'explication.



---

### **FORME I - PHASE I**

Confirmer que le joueur sait utiliser le rocket jump pour esquiver de manière verticale, ainsi que pour tirer sur une entité mobile. Cette Phase représente, par rapport au reste du jeu, une difficulté très basse, mais normale par rapport aux expériences classiques de FPS. Cela permet tout de même au joueur d'apprendre tout en ayant une expérience de jeu intéressante et fun.



---

## **FORME 1 - PHASE 2**

Confirmer que le joueur sait utiliser le rocket jump pour esquiver de manière verticale et horizontale, et qu'il sait tirer sur une entité mobile plus rapide. Confirmer que le joueur sait évoluer dans un environnement avec des obstacles hauts et enchaîner rocket jumps-tirs. Cette Phase représente une difficulté moyenne par rapport au reste du jeu et une expérience avancée pour un FPS classique. Il y a déjà un bond de difficulté important par rapport à la Phase précédente, pour montrer au joueur que le jeu devient vraiment plus dur à chaque changement de phase

---

---

## **FORME 2 - PHASE 2**

Confirmer que le joueur sait évoluer dans un environnement sans sol via l'utilisation du rocket jump contre des parois verticales pour rester en l'air, et qu'il sait tirer sur une entité. Cette Phase est la dernière phase d'apprentissage réelle du joueur. L'intensité augmente beaucoup car le sol disparaît, le joueur n'a plus de surface de repos facilement exploitable, et l'entité est très agressive. La montée en difficulté est volontairement importante pour mettre le joueur en situation délicate, afin de le préparer à la Forme finale.

---

---

## **FORME 2 - PHASE 1**

Confirmer que le joueur sait se déplacer avec le rocket jump pour esquiver une entité et pour naviguer entre des obstacles. Confirmer que le joueur sait tirer sur une entité qui le pourchasse constamment, et qu'il sait enchaîner rapidement rocket jumps-tirs. Cette Phase devient bien plus dure et représentera un cas de difficulté haute pour un FPS classique. Elle est ici pour bien montrer au joueur que l'intensité du jeu grimpe de plus en plus.

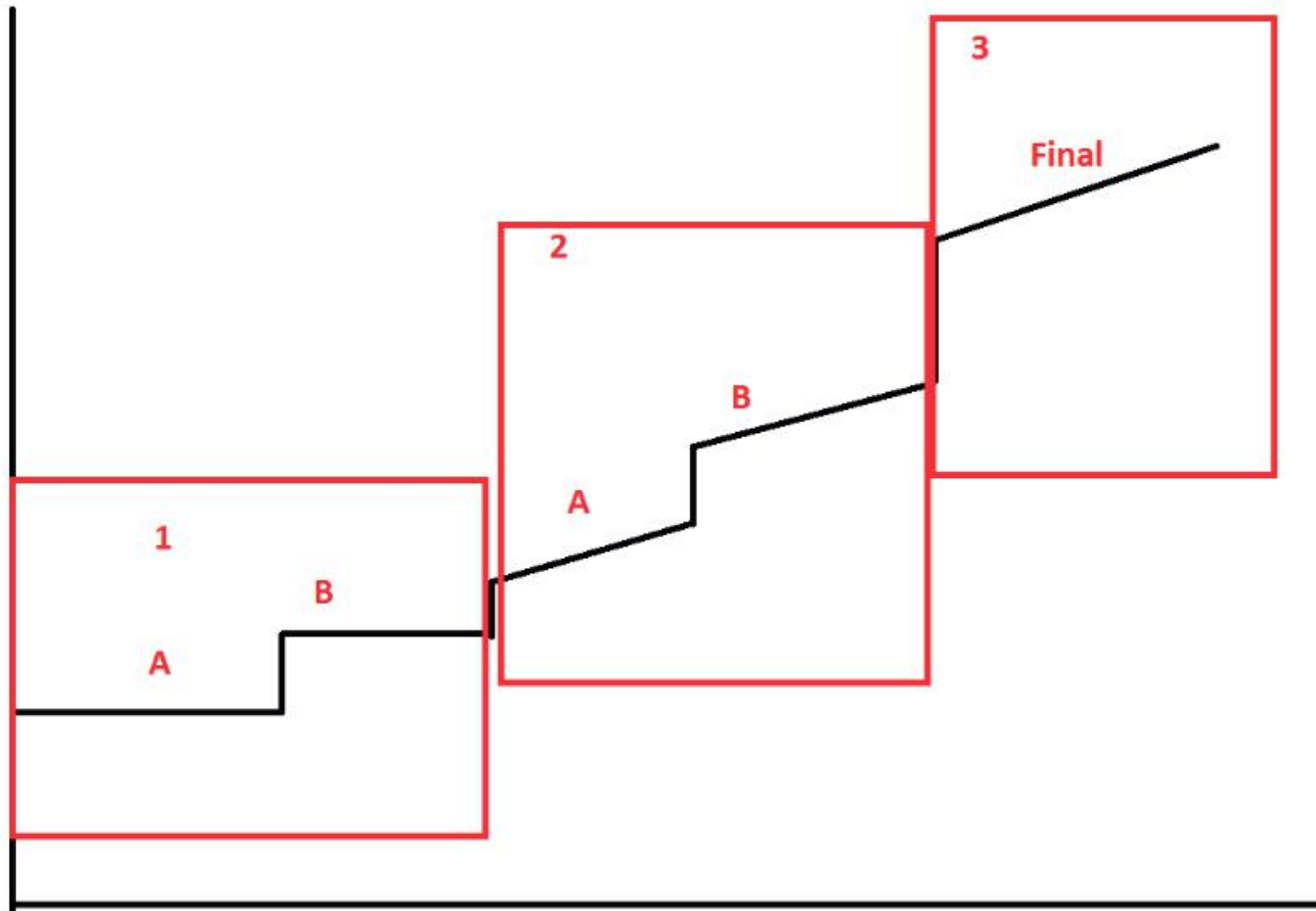
---

---

## **FORME 3**

Cette Phase étant la Phase Master du jeu, elle confirme que le joueur a un haut niveau de maîtrise de toutes les mécaniques. Cette Phase demande au joueur de tirer sur une entité, qui est constamment au-dessus de lui, tout en éliminant ses projectiles autoguidés. Il doit faire face à une situation très différente des autres, tout en étant chronométré du fait que l'entité se rapproche de plus en plus de la zone de mort du joueur.

---



COURBE DE DIFFICULTÉ



---

## EXPÉRIENCE ET DIFFICULTÉ

Toute l'expérience du jeu, et sa progression, sont là pour créer un game feel final très fort. La difficulté et son augmentation exponentielle, permettent d'accentuer l'intensité de jeu et de surprendre le joueur.

Nous avons calibré la difficulté par rapport au déplacement du joueur. L'expérience est faite pour ne pas être injuste. Nous avons fait en sorte que chaque Phase ne semble pas impossible pour le joueur, et que la seule raison d'échec soit une erreur de placement/déplacement.

Nous avons délibérément fait en sorte que le niveau de compétence demandé pour le déplacement augmente beaucoup, d'une part car c'est une des mécaniques du joueur les plus intéressantes et les plus fun à utiliser, mais aussi car c'est celle qui a le plus de degrés de maîtrise possible.

---

## PUNITIVITÉ

Nous avons choisi une punitivité maximale, qui est de faire recommencer à la Forme 1 - Phase 1 le joueur si l'avatar se fait toucher par un Cœur, pour 4 raisons :

- proposer une expérience plus intense : le fait de devoir enchaîner toutes les Phases à la suite sans fautes pour réussir accentue la difficulté ;
- l'apprentissage : le joueur doit échouer pour apprendre. Un joueur risque de perdre rapidement en découvrant une nouvelle Phase. Comme le jeu demande une grande maîtrise des mécaniques, il faut aussi qu'il pratique, qu'il découvre et qu'il expérimente pour avancer. Le jeu est

designé pour que le joueur acquiert un haut niveau de compétence pour réussir ;

- la découverte et la satisfaction : le jeu est designé pour que chaque Phase soit éprouvante et que le joueur voit le passage d'une Phase à une autre comme une étape importante. Pour accentuer cela, rendre difficile d'accès les Phases avec la condition de ne pas échouer permet au joueur de sentir non seulement une évolution dans ses capacités, mais aussi une évolution dans le jeu. La tension entre la punitivité et la difficulté permet d'accentuer cet effet.

Comme le joueur doit enchaîner toutes les Phases, cela lui demande une plus longue concentration. Ce facteur a été pris en compte pour la gestion de la difficulté, car une concentration plus longue demande plus d'effort.

Le fait de faire recommencer le joueur lui permet de refaire les Phases. Cela lui permet de se rendre compte de sa propre progression. Refaire un challenge qui, précédemment, était éprouvant, est devenu facile grâce à l'expérience acquise et permet de donner une sensation de progression au joueur.

---

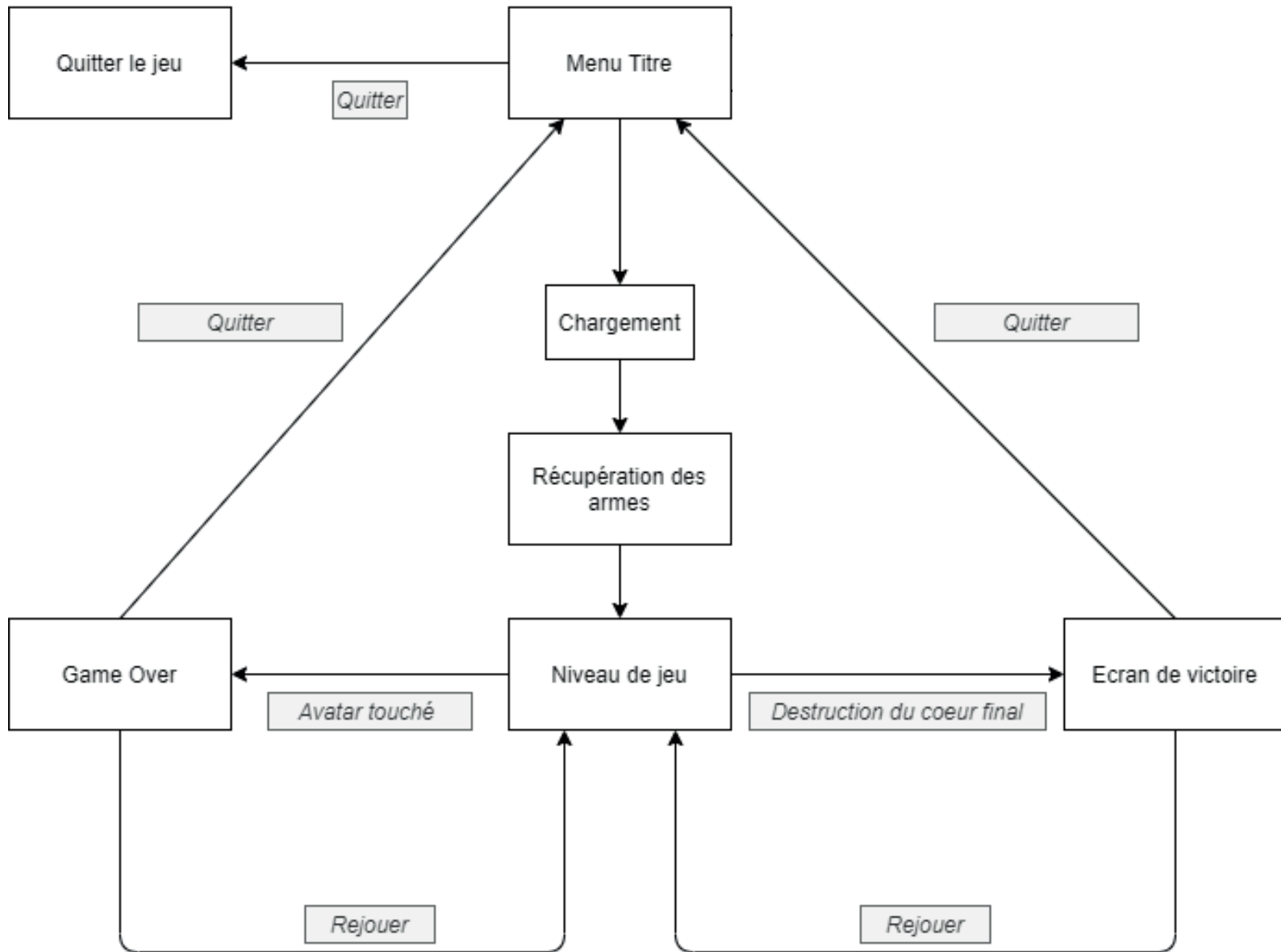
## STRUCTURE DE PROGRESSION

Le jeu est un single run. Le concept fait qu'il n'y a qu'un seul niveau fixe avec une difficulté élevée. Le joueur doit alors recommencer jusqu'à réussir. Chaque nouvelle Forme/Phase atteinte par le joueur peut être considérée comme une étape/avancée, mais n'est pas enregistrée par le système.

La progression au niveau des règles de jeu est binaire : le

joueur réussi ou non. La progression au niveau du joueur se fait par une accumulation d'expérience personnelle lui permettant d'aller de plus en plus loin dans le jeu, jusqu'à le finir.

La seconde partie de la progression se fait au niveau de sa performance, par exemple, finir le jeu le plus vite possible ou le plus de fois à la suite sans perdre.



FLOWCHART DE FONCTIONNEMENT DU MENU

# LEVEL DESIGN

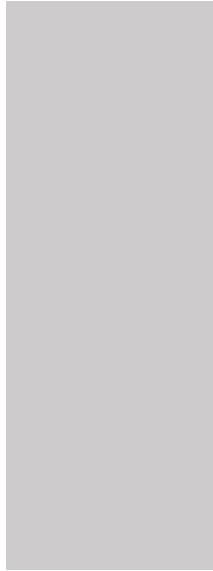
## INTENTIONS GÉNÉRALES

—

Durant le deuxième semestre, nous voulions renouveler notre level design en proposant quelque chose d'original, tout en restant en cohérence avec notre univers. C'est ainsi que nous avons eu l'idée de créer un espace de jeu dont les niveaux s'adapteraient à la forme du Lodrasq. Le défis étant de travailler avec les artistes sonores et visuels pour retranscrire une ambiance logique, tout en répondant à notre intention initiale d'environnement sombre.

# DÉROULÉS DES NIVEAUX

---



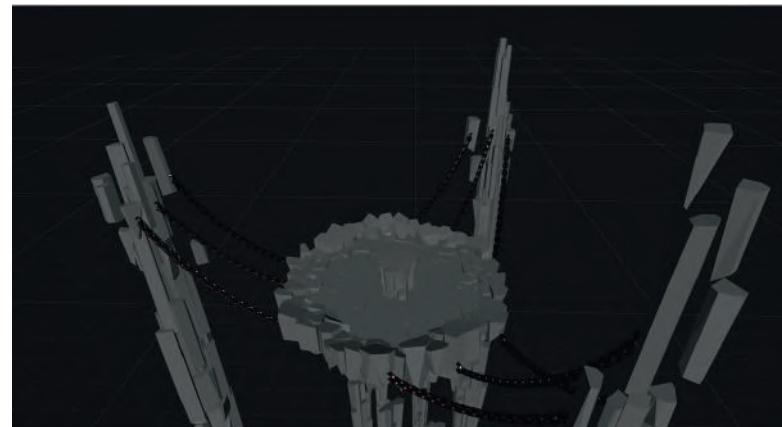
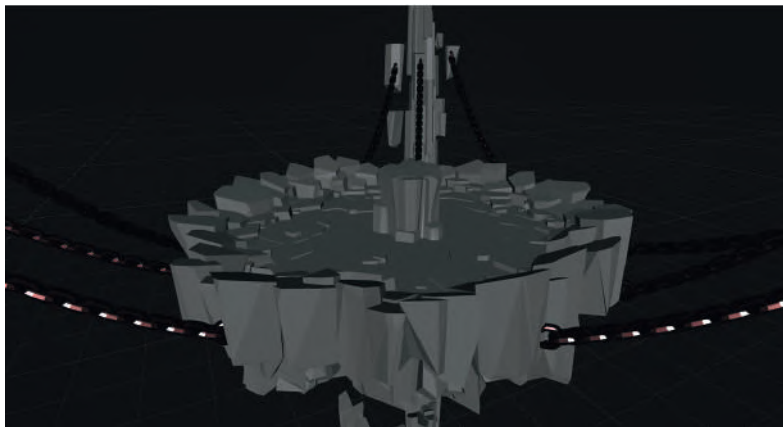
---

## PHASE 0 : TUTO

INTENTION : créer un espace ouvert, mettant en hauteur l'entité par rapport au joueur, l'objectif premier de cette phase étant que le joueur valide une maîtrise minimale du rocket jump.

DESCRIPTION : terrain relativement plat entouré de vide, avec un pilier central surélevant l'entité cible, qui est immobile. Trois piliers sont disposés autour de l'arène, de manière équidistante, et sont reliés à celle-ci par des chaînes. L'avatar apparaît sur la plateforme, devant le pilier central.

---



---

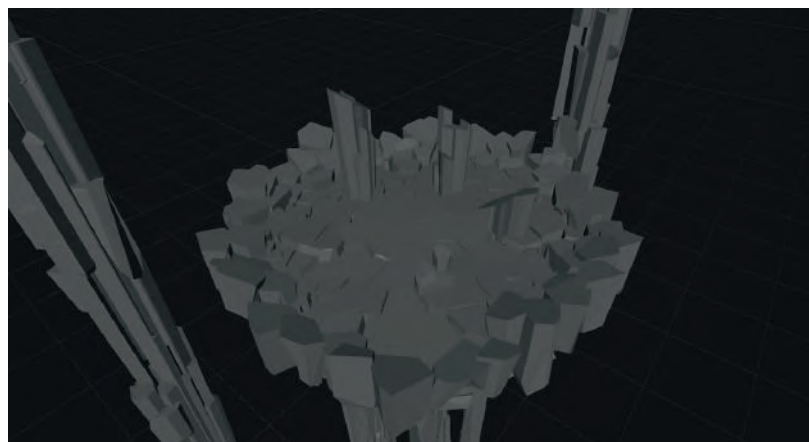
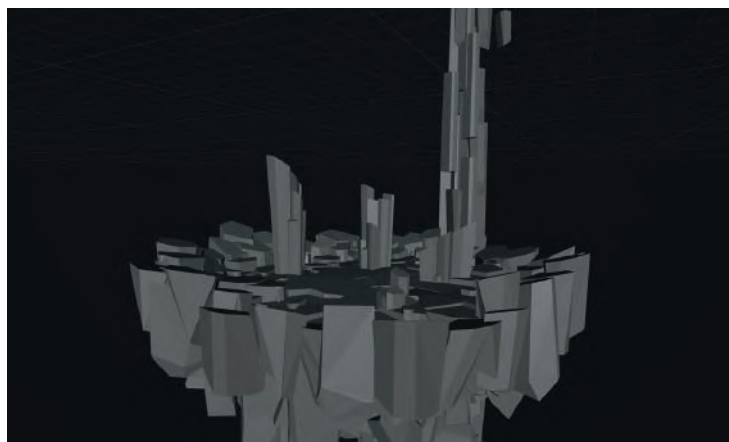
## PHASE I : CHAUVE-SOURAIE

INTENTION : avoir un espace de jeu avec le sol et les piliers servant de points d'appui au déplacement, via le rocket jump, tout en gênant la vue et la circulation du joueur.

TRANSITION : le sol tombe, se fissure et déforme l'arène, se faisant transpercer par les piliers placés en-dessous.

DESCRIPTION : le sol est un plan craquelé et déformé. Le pilier central disparaît et quatre autres de tailles différentes surgissent de la plateforme.

---





---

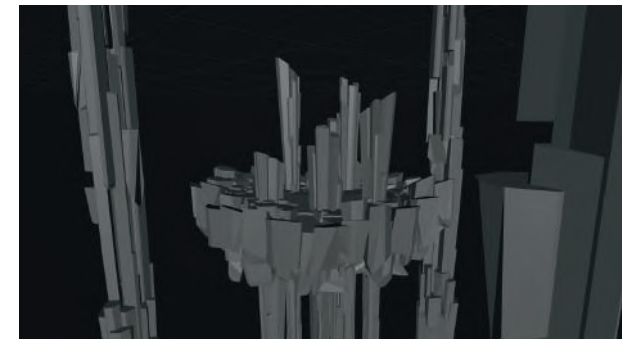
## PHASE 2 : CHAUVÉ-SOURAIE

INTENTION : continuer dans la lancée du 2ème niveau, en accentuant la difficulté. Nécessite de l'adaptation de la part du joueur à la suite du changement de l'espace de jeu.

TRANSITION : le sol s'affaisse et se déforme encore plus. Les piliers semblent devenir plus grands du fait de la tombée de l'arène. Un 5ème pilier surgit et vient compléter une disposition en cercle avec les autres.

DESCRIPTION : même structure que la phase 1 avec l'arène encore plus basse et un pilier supplémentaire venant la transpercer. Ces cinq piliers sont placés de manière équidistante entre eux et par rapport au centre de l'arène.

---



---

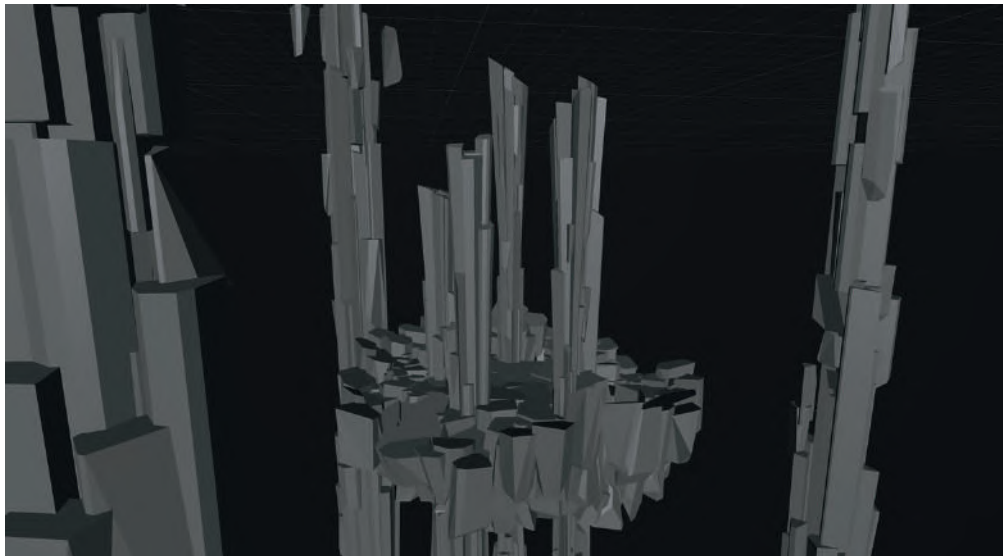
### **PHASE 3 : SERPENT**

INTENTION : continuer dans la lancée du 3ème niveau, en accentuant la difficulté. Nécessite de l'adaptation de la part du joueur à la suite du changement.

TRANSITION : le sol tombe encore et se déforme plus.

DESCRIPTION : même structure que la phase précédente avec l'arène plus basse.

---







---

## PHASE 4 : SERPENT ET GALAXIE

INTENTION : faire disparaître les surfaces sur lesquelles l'avatar peut tenir facilement, en retirant le sol et obliger ainsi le joueur à se servir encore plus souvent du rocket jump pour se déplacer. Afin de garder un climax logique, le niveau ne change pas de l'avant dernière phase à la dernière.

TRANSITION : le sol se brise et disparaît.

DESCRIPTION : il ne reste plus que les piliers.



# DEVELOPPEMENT

## OUTILS DE PRODUCTION

—

Nos entités sont composées de deux parties : les Cœurs et les Blocs. Ceux-ci sont modélisés par le pôle artistique. La forme de l'entité, en revanche, est gérée par un script en jeu, faisant que l'artiste de notre groupe ne pouvait pas modifier directement l'entité et la formation d'une nouvelle Forme, au risque de devoir passer par un processus beaucoup trop fastidieux. Nous avons donc mis en place un système permettant à notre artiste d'avoir le contrôle sur les entités, accélérant ainsi grandement la production.

L'outil développé utilise un mesh (un ensemble de données permettant à un logiciel 3D d'afficher un volume), modélisé sur un logiciel tiers afin de placer les Blocs autour des Cœurs.

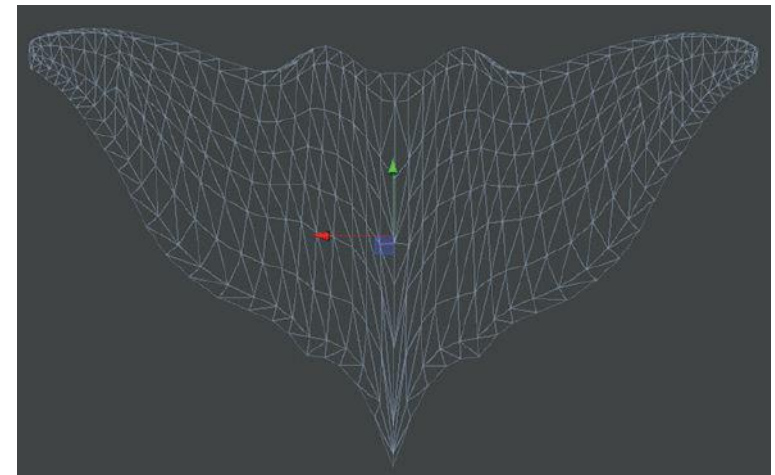
Un mesh est séparé en 3 parties :

- les vertices (ou 'sommets' en géométrie), qui servent à indiquer les points dans l'espace où seront chaque coin de la forme ;
- les triangles, servant à indiquer quel trio de vertices vont ensemble pour former des faces ;
- les coordonnées UV, servant à indiquer comment une texture s'affiche sur les triangles d'un mesh. C'est le même principe que lorsque que l'on doit emballer un volume quelconque dans du papier cadeau par exemple.



## ASSETS

Notre outil utilise seulement les vertices d'un mesh, ce qui change complètement la manière dont l'artiste 3D doit penser la forme de son modèle. Dans notre cas elle devait penser son mesh comme un quadrillage régulier.



**MESH CHAUVÉ-SOURAIE**

L'outil va donc utiliser toutes les intersections dans cette forme pour placer les Blocs aléatoirement parmi une liste finie de vertices. Enfin, on peut choisir la taille de la forme et la taille de chaque Bloc individuellement depuis l'outil.

# PROBLÈMES TECHNIQUES RENCONTRÉS

---

En prototypant les premières versions du jouet, nous nous sommes vite rendus compte que notre mécanique prenait énormément de puissance de calcul. La première version était limitée à environ 200 Blocs simultanés. Nous avons donc d'abord cherché un moyen d'optimiser sur la physique de *Unity* pour augmenter le nombre de Blocs, tout en limitant le nombre d'itérations dans les calculs de collisions.

Quand un objet physique entre en contact avec une boîte de collision, le moteur physique utilisé par *Unity* va

répéter le même procédé un certain nombre de fois pour calculer dans quelle direction l'objet physique doit se déplacer pour ne plus être en collision, tout en simulant les lois mécaniques. Augmenter le nombre d'itérations de calculs augmente la précision de la simulation, et baisse les performances lors d'une collision. Dans notre jeu, il arrivait souvent que des centaines de Blocs restent en collision pendant de grands laps de temps. Il était donc nécessaire de baisser le plus possible le nombre d'itérations.

En continuant de changer les paramètres physiques, nous avons réussi à augmenter le nombre de Blocs simultanés jusqu'à environ 300 (toujours sur le même PC).

Nous avons cherché du côté des compute shaders. Malheureusement, pour que cela soit vraiment efficace, il était nécessaire de faire en sorte que la détection de collisions et la résolution de collisions se fassent en shader. La résolution était relativement simple à réaliser, sachant que nous avons réduit la forme des boîtes de collisions des Blocs à seulement une sphère. Ainsi, nous n'avons plus eu besoin de calculer les forces de rotation.

Le problème venait de la détection de collisions. Le seul moyen que nous voyions à ce moment était de faire une liste complète de toutes les boîtes de collisions. En limitant toutes les boîtes à des sphères, il était nécessaire de faire un algorithme qui place des sphères de collisions sur toutes les formes du terrain, puis de faire en sorte que seules les sphères nécessaires soient prises en compte. Tout cela demandait encore plus de recherches et de travail que notre problème initial. Nous avons donc décidé d'abandonner la piste du compute shader pour le système ECS de *Unity 2019*.

Même problème qu'avec le compute shader, tout était à refaire pour la physique en ECS. Cependant, nous avons trouvé des scripts en ligne permettant de faire une simulation physique suffisante pour nos besoins. Mais les performances étaient beaucoup trop hasardeuses et pendant les rares moments durant lesquels la simulation était stable, le gain de performance ne dépassait pas les 20 %, sachant que nous n'avions pas encore intégré le système de déplacement des Blocs, risquant de nous faire redescendre en-dessous de la barre des 300 Blocs simultanés. Nous sommes donc revenus sur notre version de base en changeant grandement le système de calcul de déplacement des Blocs.

Cet ancien système appelait un calcul de déplacement extrêmement simple. Malgré la simplicité de ce calcul, le jeu tournait lentement lorsque l'on montait trop haut le nombre de Blocs. Le problème ne venait pas du calcul, mais de l'appel en lui-même, autrement-dit de l'Update. L'Update est un évènement qui permet de stocker un nombre d'instructions écrites par le développeur d'un projet, qui vont ensuite être exécutées à chaque frame. Il y a un Update d'appelé pour chaque script présent dans la scène qui en possède un.



Quand on appelle l'évènement Update, Unity va faire un nombre conséquent de tâches en arrière-plan pour permettre le bon fonctionnement du moteur de jeu. Nous n'avions malheureusement aucun contrôle sur ces tâches. De plus, Unity possède énormément de fonctionnalités secondaires, l'obligeant à faire un travail conséquent rien que pour lancer l'Update, quelque soit le nombre d'instructions ordonnées par le développeur.

Dans notre ancien système, chaque Bloc possédait un Update. Baisser le nombre d'appels d'Update était donc vital pour augmenter le nombre de Blocs présents simultanément. Nous avons donc appliqué la logique ECS : un script sur chaque objet servant simplement à garder en mémoire des paramètres, et un script parent qui va exécuter la même instruction sur chaque objet

possédant un script donné, en utilisant les paramètres propres de chaque objet.

Cela limitait le nombre d'Updates à seulement un par script parent. Nos entités possédant au maximum de 8 Cœurs différents, nous pouvions nous permettre d'avoir jusqu'à 8 Updates simultanées.

Le calcul dans chaque Update était beaucoup plus long qu'avant, mais cela nous a permis d'augmenter drastiquement notre nombre de Blocs présents simultanément dans la scène. Notre entité contenant le plus de Blocs étant la Galaxie avec ses 2500 Blocs, le jeu peut tout de même tourner à 120 fps en moyenne dans l'éditeur, et tout cela sans contraindre notre processus de création d'entités.

# CHARTE GRAPHIQUE

## INTENTIONS

---

Pour l'aspect graphique de *Lodrasqe*, notre fil conducteur était de proposer un boss démesurément grand dans un environnement lui étant adapté, le tout en respectant notre mécanique principale de 'nuée de Blocs' revenant à leur point d'origine. Nous voulions également faire émaner une aura pesante et mystérieuse sur le joueur, donnée à la fois par l'entité et par son univers.

Notre seconde intention était de proposer une arène ouverte et libre d'actions en guise d'espace de jeu. Celle-ci donnant un maximum une sensation de liberté, le joueur peut y utiliser tous les outils qui lui sont fournis afin de s'y déplacer comme bon lui semble.

Enfin, un autre de nos objectifs était de faire comprendre au joueur, de la manière la plus intuitive possible, la répercussion de ses actions par des effets visuels.

# PROCESSUS DE RECHERCHE



Nos premières inspirations pour le jeu étaient dirigées vers les représentations de la Rome Antique, avec ses arènes fermées, ses gladiateurs, ses armes... L'espace de jeu était pensé comme fermé et les ennemis, de forme humanoïde, devaient arriver par vagues pour laisser place à une entité plus grande à la fin.

## EXEMPLES DE RÉFÉRENCES

### ARENES



### ARMURES

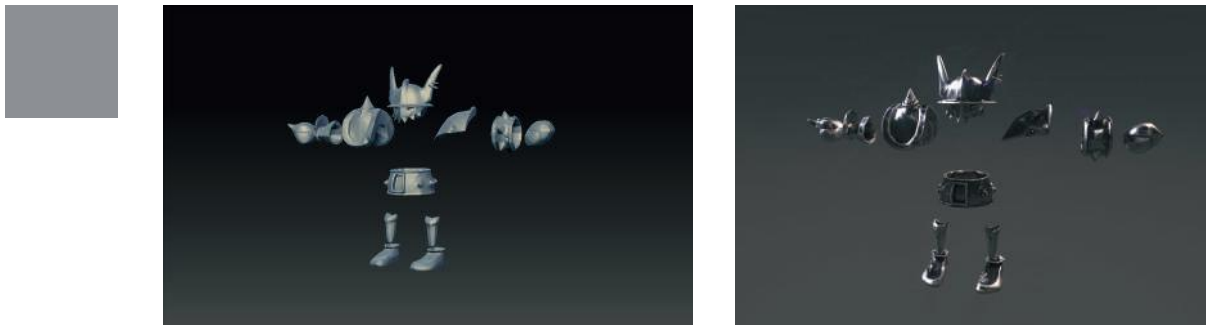
### ARMURES & ARMES



## QUELQUES RÉALISATIONS



**CONCEPTS ART** Armure et armes des ennemis



**ASSET 3D** Ennemi de niveau 1

## PROCESSUS DE RECHERCHE ET DE CRÉATION

### ETAPE 1

Recherche de références et réalisation de moodboards.

### ETAPE 2

Réalisation de concepts art adaptés aux contraintes techniques.

### ETAPE 3

Création d'assets game ready d'après les concepts art.



L'idée d'une arène fermée ne convenait finalement pas, car cela entraînait en contradiction avec notre volonté d'un boss que nous voulions gigantesque. L'environnement devait être mieux adapté à sa taille.

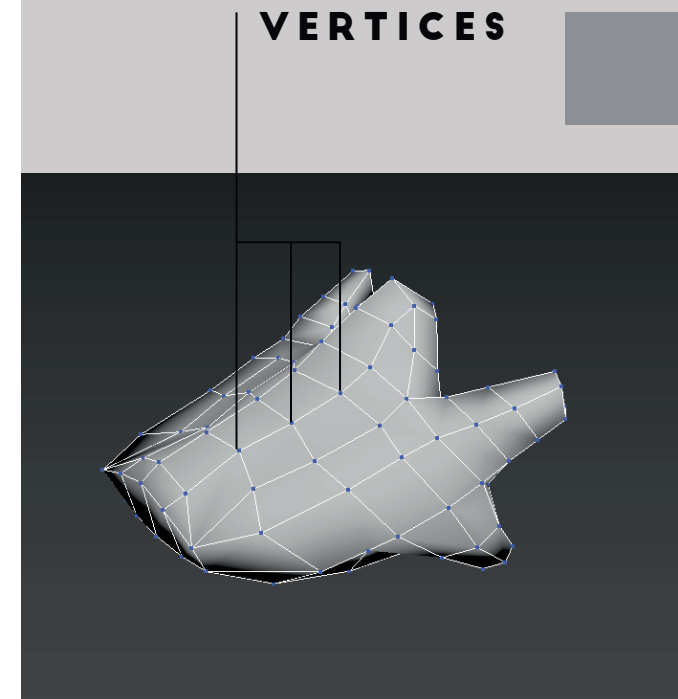
De plus, notre mécanique appliquée sur des morceaux d'armures ne rendait pas un bon game feel. Nous avons alors repensé tout le design graphique et abandonné les recherches sur la Rome Antique.

## MOYENS TECHNIQUES

Avant de nous lancer dans la production d'assets, nous devions prendre en compte une contrainte technique principale : la mécanique de 'nuée de Blocs' qui limitait les performances du jeu. Les assets devaient donc être réfléchis d'une manière particulière quant à leur conception.

Nous avons donc cherché des moyens efficaces de concevoir des entités composées de plusieurs centaines de Blocs. Après plusieurs tests techniques, nous avons décidé de raccorder les blocs nécessaires à la mécanique par script.

Le principe consiste à convertir les vertices d'un modèle 3D en Blocs, eux aussi modélisés en amont. Lors de la modélisation, il était important d'avoir une distance égale entre les vertices pour donner un résultat homogène lors de la conversion.



# ÉVOLUTION

Avant d'arriver au script permettant de faire nos 'nuées de Blocs', nous sommes passés par d'autres procédés.

---

## TECHNIQUE 1

À l'aide de l'outil ProCutter de *3ds Max*, il était possible de placer des planes sur une modélisation de sorte que ceux-ci la coupe. Cependant, quand nous en placions trop, les planes fusionnaient avec certaines parties, rendant impossible l'exploitation de cette technique pour nos besoins.

---

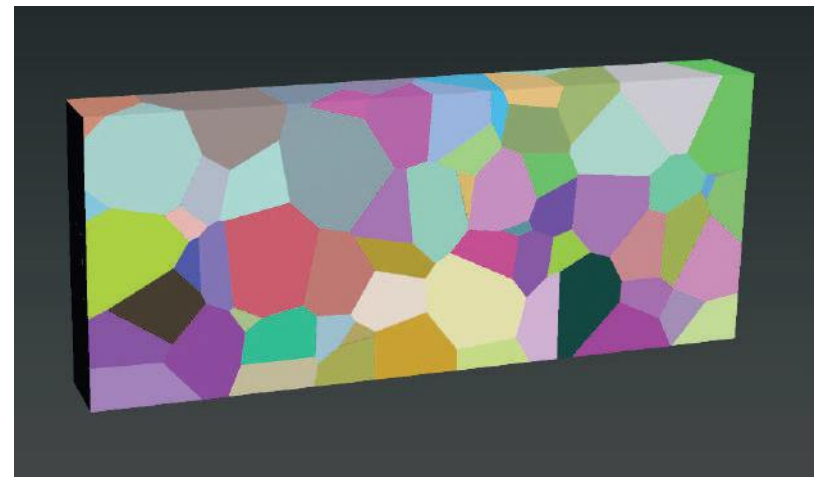


---

## TECHNIQUE 2

Grâce à un script applicable à *3ds Max*, nous arrivions à découper un modèle 3D aléatoirement, avec la possibilité de choisir le nombre de morceaux. Malheureusement, l'effet obtenu en jeu permettait au joueur de prédire l'emplacement de retour des Blocs expulsés, limitant ainsi le game feel.

---



## ANIMATIONS

---

Afin de donner plus de vivant à notre boss, nous nous sommes renseignés sur différentes manières d'animer un modèle 3D.

Finalement, nous avons abandonné l'idée de faire des animations en amont pour les intégrer, car cela ne rendait pas le résultat voulu. Le mouvement de l'ennemi est désormais exclusivement le résultat de son comportement de déplacement, réalisé par script.

## ARMURES

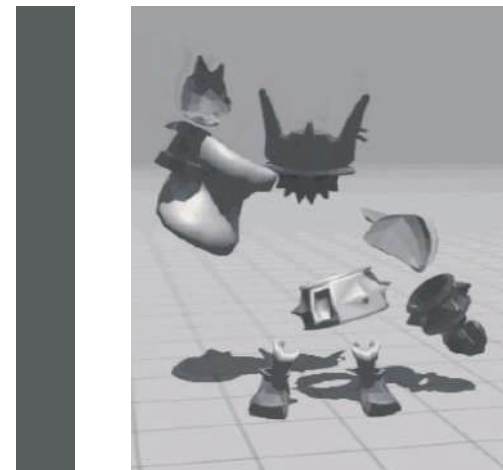
---

Durant notre phase de recherche sur les méthodes d'animation, nous sommes passés par le rigging des modèles d'armures, sur *3ds Max*, afin de les tester. Les animations de tests ont été faites sur *Mixamo*.

---

Il y avait une incohérence visuelle en jeu, car les Blocs avaient des difficultés à se replacer avec l'animation. De plus, il n'était plus possible de reconnaître la forme humanoïde lorsque les Blocs bougeaient.

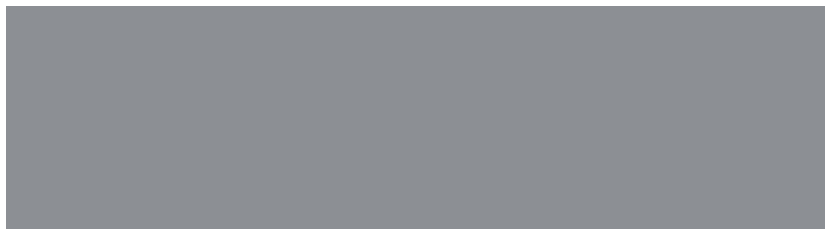
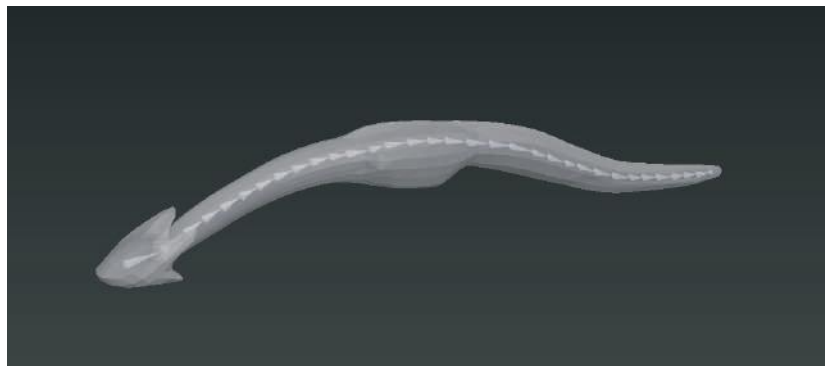
## ÉVOLUTION



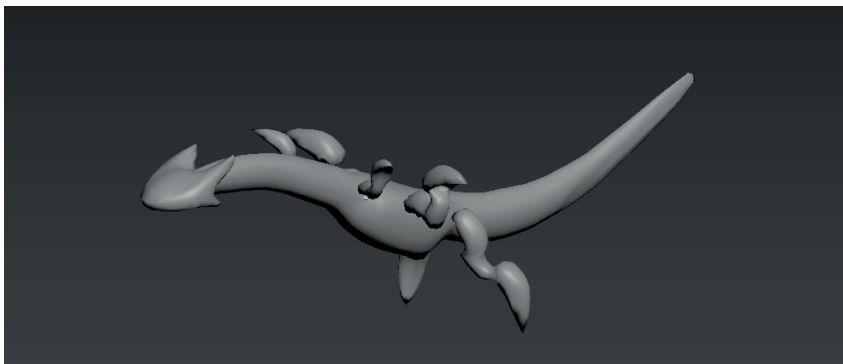
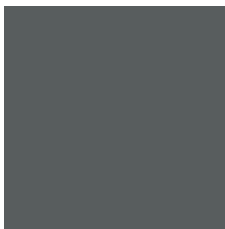
## CREATURE

---

Nous avons également appliqué des rigs sur les premiers essais de boss pour ensuite les animer sur *3ds Max*.



Par manque de temps et de résultats concluant, nous sommes rabattus sur l'animation par script, dépendant ainsi du comportement du boss, le rendant plus vivant. De plus, il était possible de le contrôler plus facilement et à notre guise, directement en jeu.



# BOSS



## BOSS SERPENT

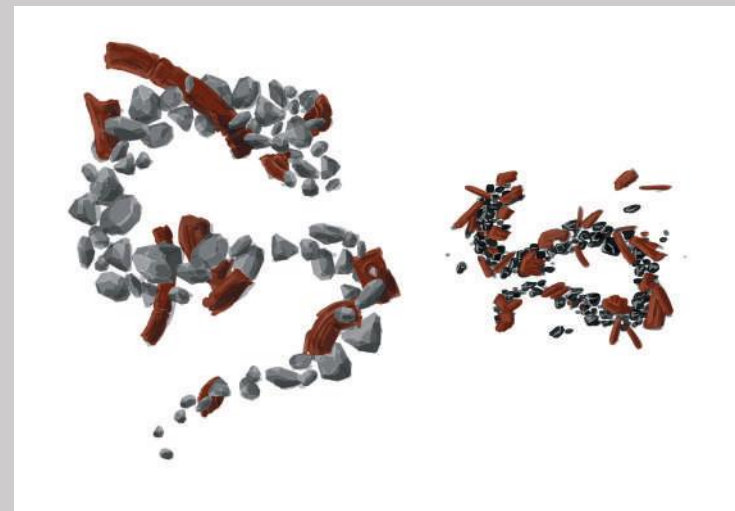
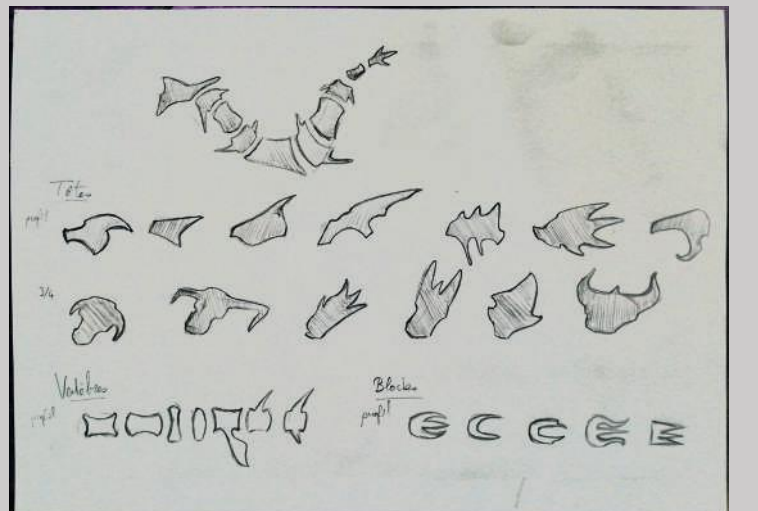
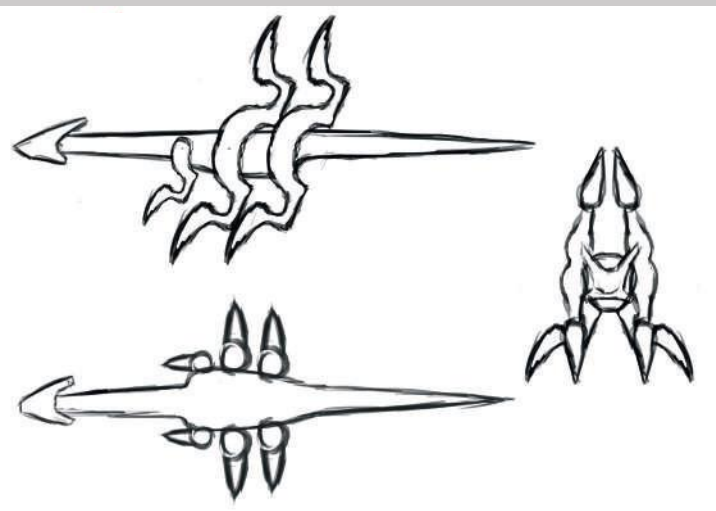


### RÉFÉRENCES

Au premier semestre, nous nous sommes concentrés sur une seule apparence pour notre boss. Nous voulions qu'il soit en longueur, tout en étant imposant.



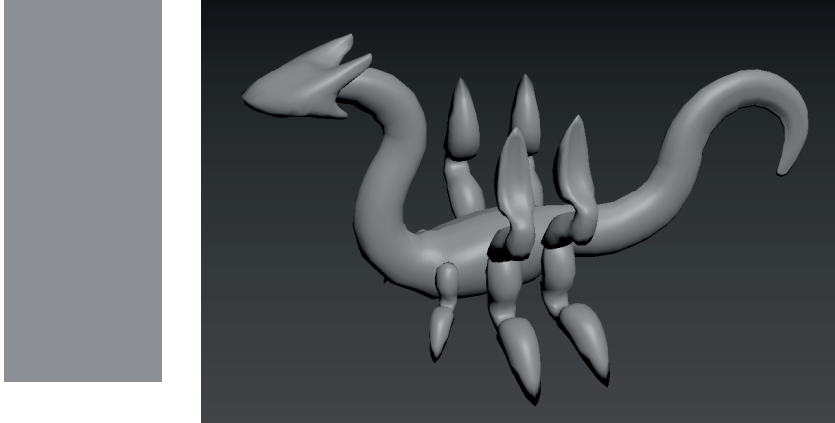
Au départ, ce boss se déplaçait en faisant des bonds depuis le sol, pour y retomber et s'y enfoncer de nouveau (déplacement en vagues). Désormais, il ondule dans les airs, tel un dragon chinois.



CONCEPTS ART

Quelques recherches de formes pour le boss serpent.

## ASSETS ET RENDU IN GAME

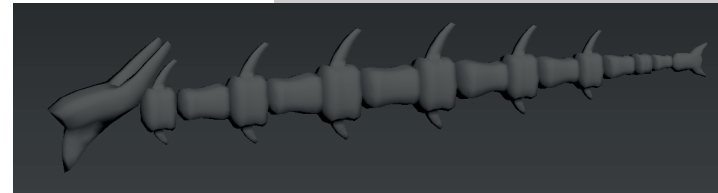


### ASSET 1

Premier test avec la modélisation d'une forme complète à animer.

### ASSET 2

Autres tests avec chaque partie séparée, animée par script.



## RENDU IN GAME

Une fois l'asset 2 terminé, il est importé dans le projet et chacune de ses vertices est convertie en Bloc.

Lors du deuxième semestre, nous avons eu en projet de donner plusieurs apparences à notre boss. L'idée étant qu'il ait des pouvoirs de changement de forme, et donc de capacités, proposant ainsi des challenges différents.

Cependant, il ne nous était pas possible de prendre autant de temps que pour le boss serpent au premier semestre. C'est pourquoi certaines étapes du processus de recherche et de création ont été sautées pour les conceptions des nouvelles entités.

## BOSS CHAUVE-SOURAIE



## RÉFÉRENCES

Le boss Chauve-souraie précède le serpent et se déplace dans les airs en battant des ailes. Nous avons donc cherché des références en rapport avec cette intention.

---



## ASSETS ET RENDU IN GAME



---

### ASSETS

Modélisation du boss aérien (2 variantes).

---

### RENDU IN GAME

La modélisation est importée dans le projet et voit toutes ses vertices converties en Blocs.

---

## BOSS GALAXIE

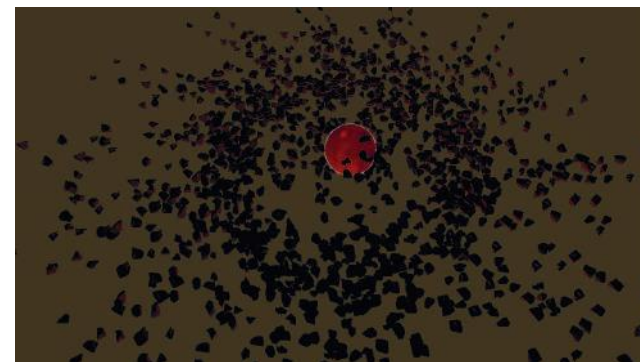
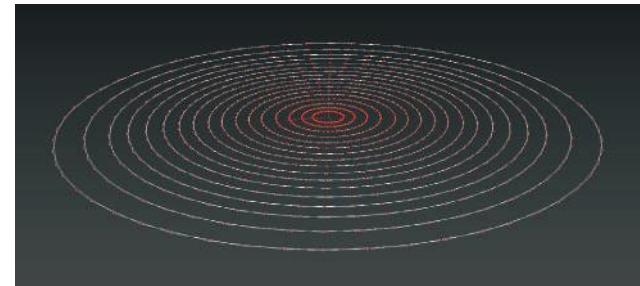


## RÉFÉRENCES

Notre référence principale durant tout le développement de ce boss aura été la forme des galaxies. Nous voulions une entité finale prenant tout l'espace de jeu pour le climax. Notre intention étant que le joueur se sente oppressé et dominé.

## RENDU IN GAME

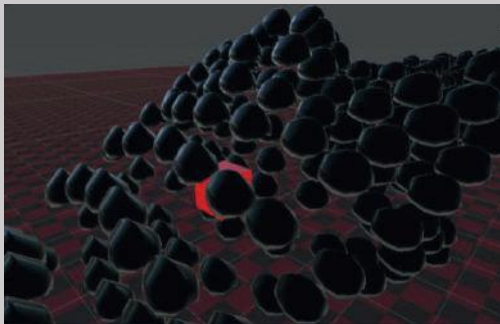
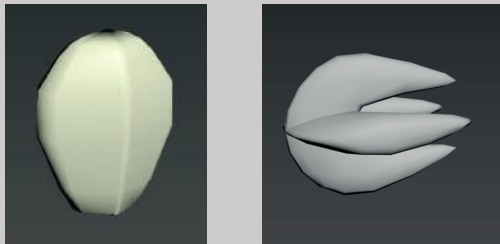
L'asset 3D de ce boss est seulement constitué de plusieurs planes circulaires superposés. Les vertices se touchant, amènent les Blocs en lesquels elles ont été converties, à rentrer en collision, provoquant une énorme nuée d'apparence organique (sensation de respiration) une fois en jeu.



## BLOCS ET COEURS

### BLOCS

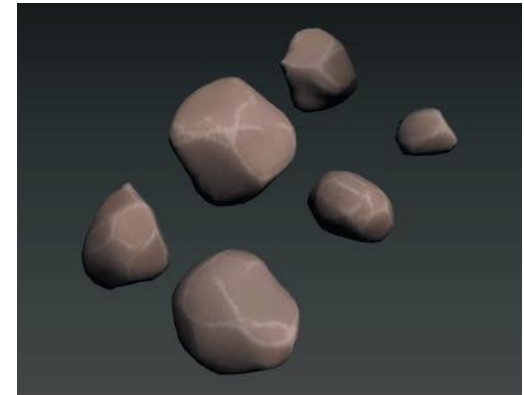
La conception des Blocs devait respecter une technicité particulière, puisque, en jeu, leur orientation dépendait de la physique qu'ils rencontraient. Nous devions donc trouver un design dont l'orientation pouvait être aléatoire, tout en gardant une certaine esthétique.



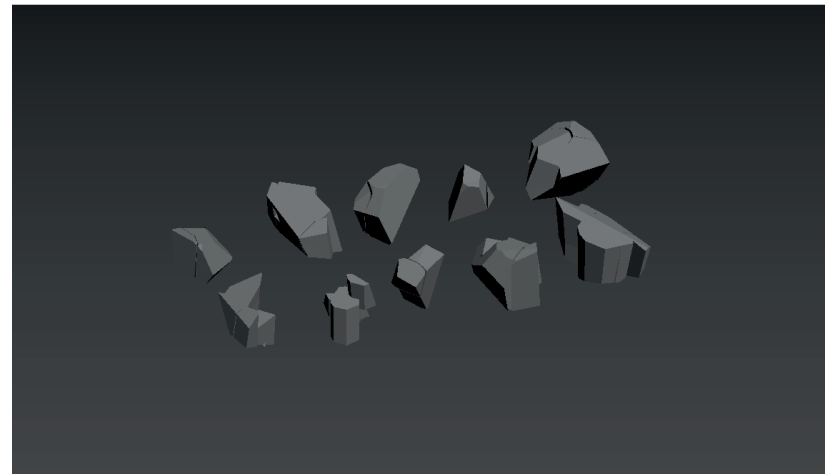
---

Au premier semestre, nous nous sommes arrêtés sur une esthétique de pierre, en cohérence avec notre univers de départ.

---



Au deuxième semestre, l'idée de pierre est restée, mais leur apparence a changée. Ce sont finalement des parties de l'environnement morcelées (-> assets piliers. Cf. Environnement p. 25) pour accentuer l'appartenance de l'entité à son milieu.



## COEURS

---

Les Cœurs devaient eux aussi respecter certaines conditions. Ils devaient pouvoir être contenus par les Blocs, afin que ceux-ci les protègent, et devaient aussi être plus petits que la surface convertie en Blocs à laquelle ils sont associés.

---

Pour les Cœurs, nous nous sommes donc concentrés à la fois sur une apparence osseuse et sur un rendu cristalin.



---

## IDÉE CONCEPT

Réalisation d'un concept de Coeur en cristal sur *Substance Painter*.

---



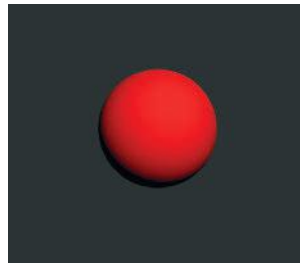
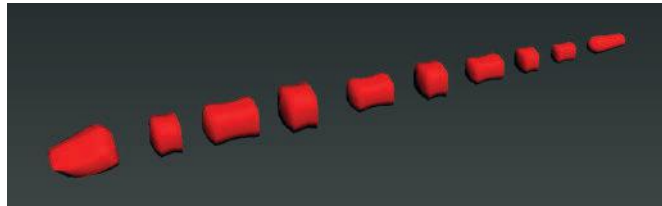
---

## RECHERCHES

Exemples de recherches de shaders sur *Unity* et de textures cristallines.

---





## ASSETS

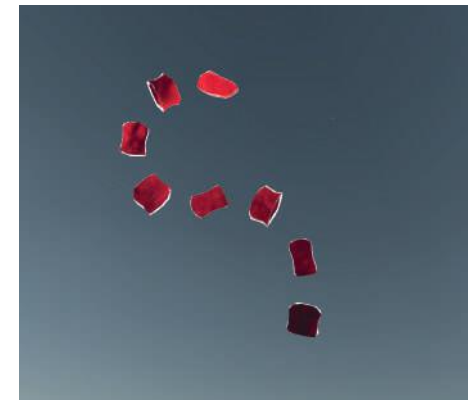
Finalement, la forme des coeurs est restée sur une apparence osseuse, reprenant la en surface des boss, sauf pour l'entité galaxie qui est une simple sphère, comme un noyau.



## RENDU IN GAME

Une fois les assets intégrés en jeu, nous avons appliqué un shader, créé à l'aide de Cubemaps dans *Unity*, sur les assets, afin de rendre claires les cibles (ce sont les seuls éléments du jeu avec ce shader). Cela nous a permis de rappeler l'effet cristalin que nous recherchions, à travers la texture.

Un autre shader, cette fois animé, est présent également sur les Cœurs lorsqu'une nouvelle forme du boss va apparaître.



## ENVIRONNEMENT

---

Lors du premier semestre, nous nous sommes concentrés sur un univers désertique. L'idée d'un désert convenait parfaitement, puisque c'est un environnement à l'apparence infinie, donc adaptée pour la grandeur de notre premier boss, et pouvant accueillir presque n'importe quel thème.

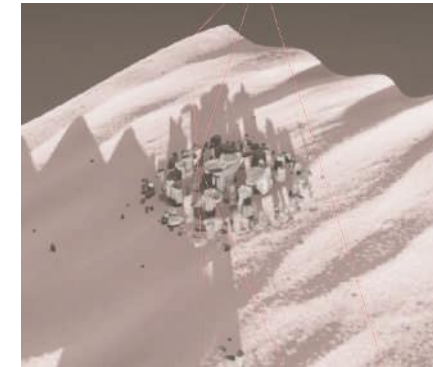
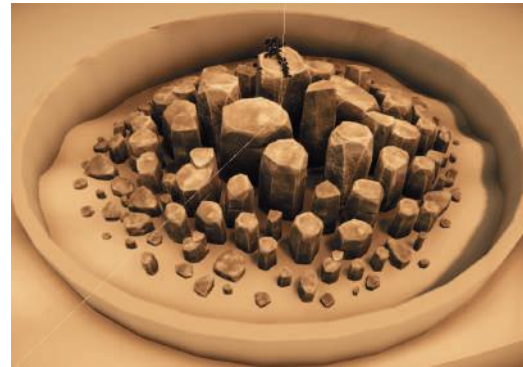
Une première version consistait à placer des rochers hauts, car nous voulions garder l'effet des Blocs passant entre des obstacles.

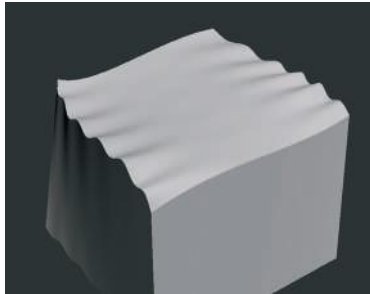
---

## RENDU IN GAME

Premiers tests d'environnement avec des assets de rochers, le Post Process de *Unity* et *Aura 2*.

---





---

## RENDU IN GAME

---

Second test d'environnement avec un sol fait sur 3ds Max (dunes), une texture de sable et des assets de piliers romains.

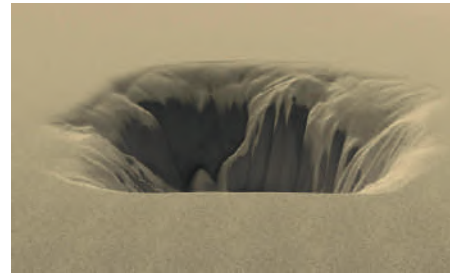
Une deuxième version de notre environnement a été faite : elle était toujours dans le désert, mais cette fois, avec des piliers en ruines, jonchant le sol, rappelant les traces d'une civilisation passée.



Durant le deuxième semestre, le Level Design a été changé pour mieux s'adapter aux différentes formes de boss, et ainsi proposer de nouveaux challenges. Le design de l'environnement a donc été revu.

---

Durant la conception d'une troisième version, nous nous sommes concentrés sur un espace de jeu central bien délimité, le tout dans un environnement toujours désertique et immense.



## RÉFÉRENCES

La troisième version tient ses inspirations notamment de cascades de sable et de représentations de planète plate. Quant aux éléments de décor, nous nous sommes d'abord inspirés de différents concepts de ruines, et même d'immeubles, puis, nous nous sommes dirigés vers des formes et dispositions de piliers sur certaines installations artistiques.

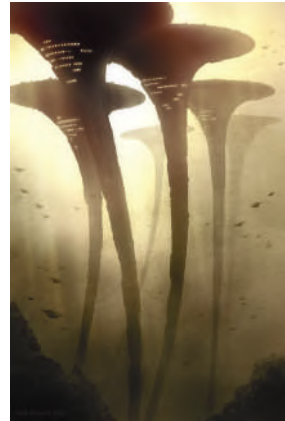




## REFERENCES RUINES



## REFERENCES PILIERS





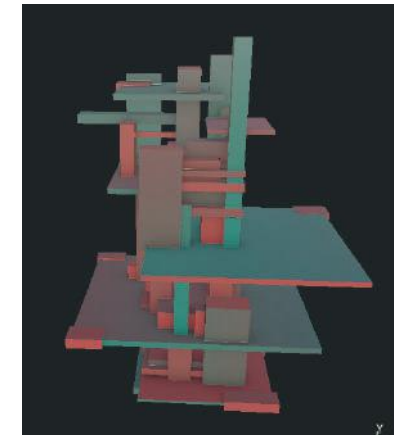
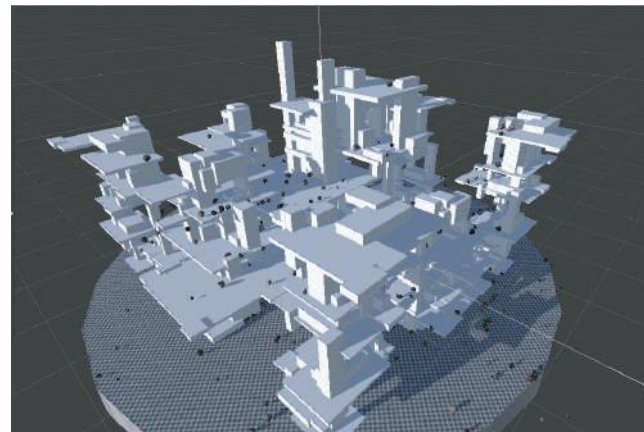
## CONCEPT ART

D'après les quelques références trouvées, nous avons conceptualisé un nouvel environnement. Il se compose d'un espace de jeu central, dont les extrémités terminent en cascades de sable, jonché de ruines pour permettre au joueur de se déplacer en prenant appui dessus, et d'un décor externe, inaccessible, présent uniquement pour mettre

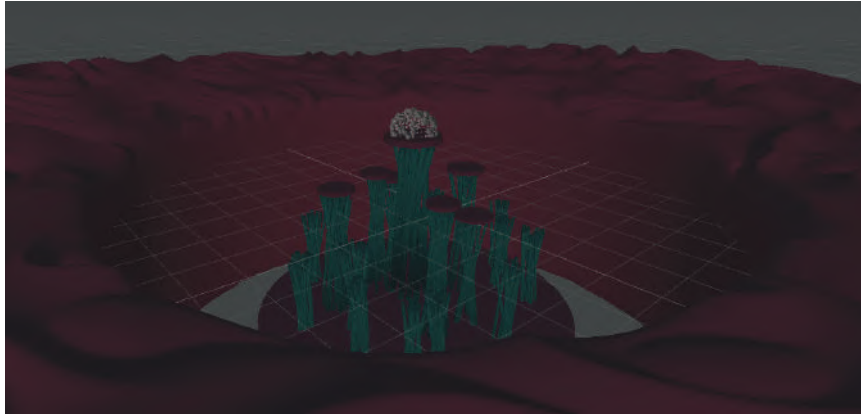
en avant l'ambiance d'environnement géant et désolé. Enfin, pour accentuer l'inaccessibilité de cette zone, un énorme fossé sans fin a été pensé, conduisant l'avatar à la mort s'il tombe.

## ASSETS ET RENDUS IN GAME

L'objectif des assets de ce troisième test, d'après les références de ruine, a été de les travailler en hauteur et de donner des points d'appui afin que l'avatar puisse faire de grands sauts. Une texture a aussi été travaillée. L'idée étant qu'elle soit un dégradé de couleurs servant de repère au joueur, quant à la hauteur de son avatar dans l'environnement.



En effet, s'il est trop bas, il risque d'avoir un game over car il tomberait. Ce moyen trouvé permettait de servir le gameplay, demandant de bouger constamment, et ne permettant pas de s'attarder sur des détails.



Les assets précédents ne convenant finalement pas, car le joueur ne pouvait pas anticiper facilement la disposition des obstacles du fait de leur complexité, nous avons repris nos références de piliers afin de proposer un nouveau concept d'obstacles.

Un quatrième test a été fait. Il se compose uniquement de cylindres, auxquels la texture de

dégradé a été appliquée, disposés de sorte que le joueur puisse prendre appui dessus sous n'importe quel angle. Leur disposition a été élaborée à l'aide d'un script, permettant de gérer plus facilement leur nombre, leur angle par rapport au sol et leur taille. Une plateforme est, de plus, posée sur chacun des îlots de cylindres.

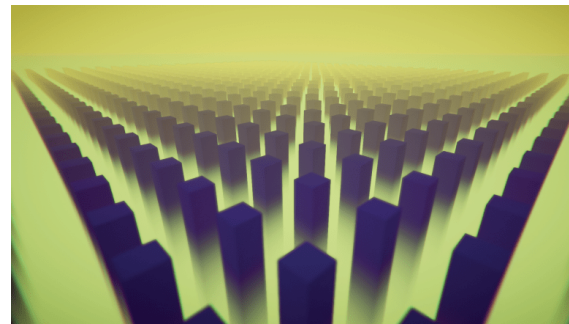
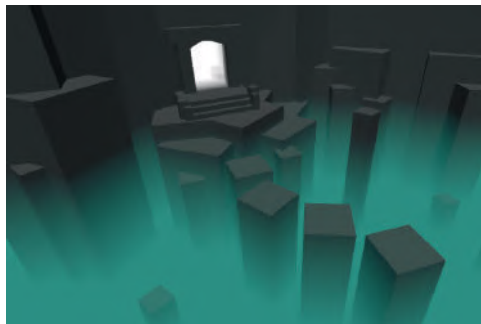
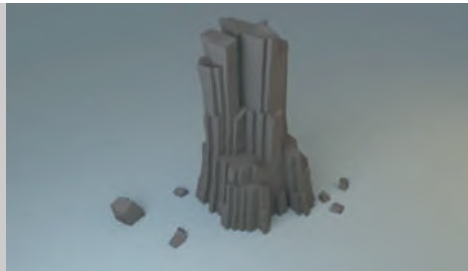


L'association de tous les éléments du quatrième test n'étant pas convaincante, nous avons décidé de faire un reboot de tout le design de l'environnement.

---



## REFERENCES ROCHERS



## REFERENCES BROUILLARD

## CONCEPT ART

Ce cinquième test aura été le dernier changement fait. Nous avons repris des références de rochers et de paysages low poly du plug-in Unity : *Polyquest*. À la suite de ça, nous avons gardé la disposition globale de l'espace de jeu central du quatrième test pour donner un nouveau concept d'environnement de jeu.



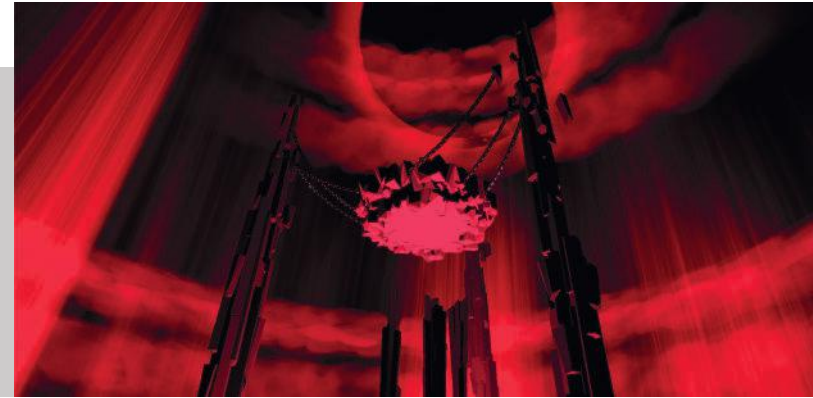
## ASSETS

Les assets ont tous été réalisés à l'aide de 3ds Max. L'arène a été morcelés à l'aide du script de découpe (Cf. Moyens techniques p. 5) afin de s'adapter au level design.



## RENDU IN GAME

Les assets ont été agencés en amont directement sur 3ds Max afin d'avoir directement leurs bonnes dispositions dans Unity. D'autres éléments ont été rajoutés plus tard pour servir le décor, comme les effets de brouillard, les god rays, les lumières ou encore des particles systems.



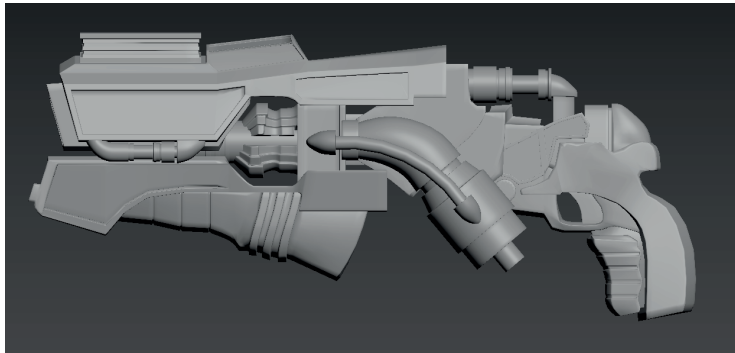
## ARMES



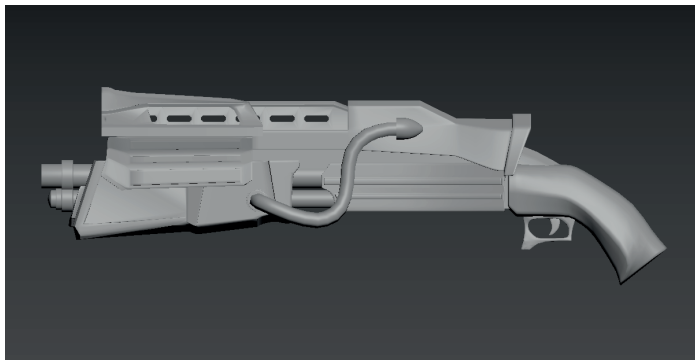
Les designs des armes ont été pensés pour retranscrire leur utilité, ainsi que leur fonctionnement et leur puissance. Des codes couleurs ont été élaborés afin que le joueur puisse se repérer lorsqu'il change d'arme.



## ASSETS ET RENDUS IN GAME



**LANCE-  
GRENADES**

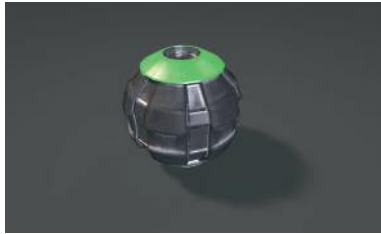


**MITRAPOMPE**

Les armes ne sont pas texturées afin de rester en cohérence avec notre environnement low poly sans textures. Nous voulions un assemblage d'assets qui soit à la fois épuré et esthétique, sans aller vers le réalisme.

## ÉVOLUTION

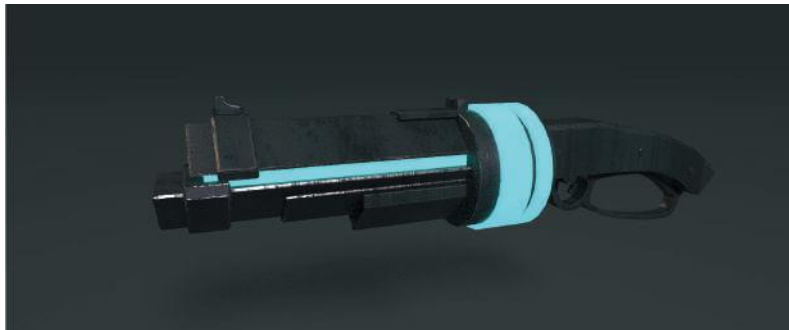
Avant d'arriver aux deux armes actuelles, nous sommes d'abord restés sur trois. D'autres designs ont été faits avec d'autres codes couleurs.



### STICKY



### LOB



### MITRAPOMPE

---

Les codes couleurs utilisés pour ces armes nous ont plus servis de repères lors du développement qu'au gameplay. En effet, notre jeu était au départ en noir et blanc (Cf. Couleurs p. 35) et les couleurs n'étaient donc pas visibles. Cela n'empêchait cependant pas le joueur de distinguer les différentes armes.

---



# VFX



Nous avons utilisé des VFX afin de montrer les feedbacks visuels des actions du joueur et celles de l'entité. D'autres servent seulement le décor. Ils ont été réalisés à l'aide du particles system de *Unity*.

## 1er SEMESTRE

---



### TIR

**Description :** création d'un effet de fumée, qui se répand dans toutes les directions, à la sortie du canon de l'arme.

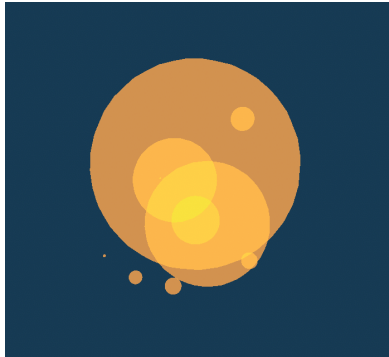
**Condition d'apparition :** le joueur actionne l'input de tir.

### IMPACTS PROJECTILES MITRAPOMPE

**Description :** des projectiles sont propulsés avec une certaine vitesse vers le haut et retombent sous l'effet de la gravité.

**Condition d'apparition :** les projectiles issus du Mitrapompe entrent en contact avec un obstacle.





## **EXPLOSION PROJECTILES LOB/STICKY**

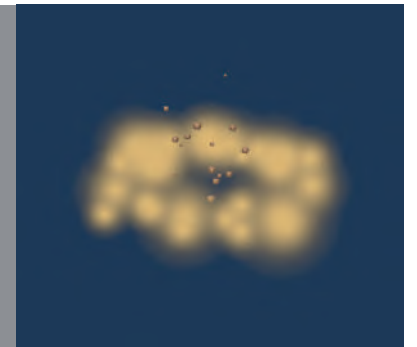
Description : sphères en 3D dont le volume augmente puis diminue rapidement.

Condition d'apparition : le projectile du Lob ou du Sticky explose/se détruit.

## **IMPACT COEUR**

Description : effet de fumée et particules qui vont vers le haut puis retombent avec la gravité.

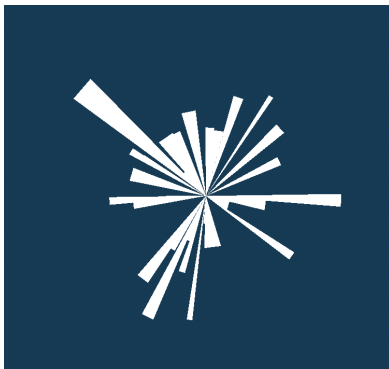
Condition d'apparition : un Cœur entre en contact avec un élément solide de la scène.



## **DESTRUCTION COEUR**

Description : rayons de lumière avec une emissive partant du coeur et allant dans toutes les autres directions

Condition d'apparition : un Cœur a ses points de vie à 0.



## DOMMAGES COEUR

Description : clignotement rapide du Cœur qui passe du noir au blanc pendant moins d'une seconde

Condition d'apparition : un projectile du Lance-grenades ou du Mitrapompe entre en contact avec un Cœur



## EXPLOSION COEUR

Description : sphères en 3D dont le volume augmente puis diminue rapidement avec une emissive.

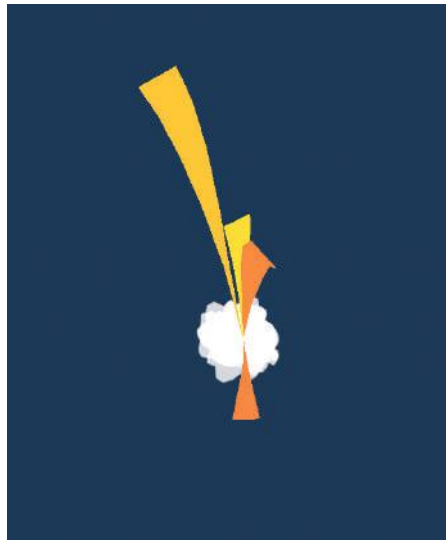
Condition d'apparition : un Cœur est détruit par le système.



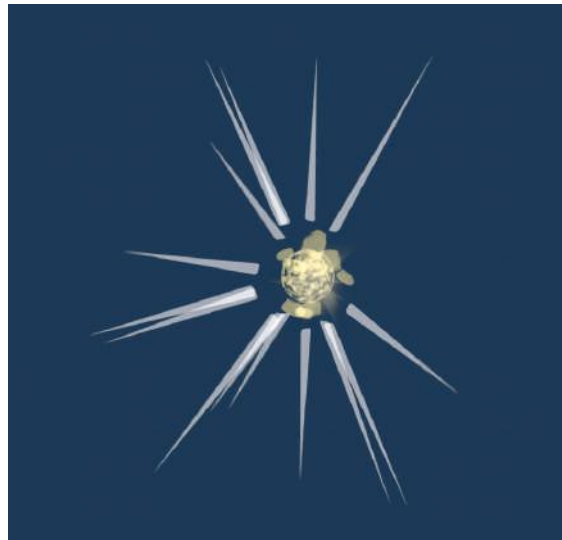
## 2ème SEMESTRE

---

Certains effets visuels ont été changés, d'autres non et certains sont nouveaux par rapport au premier semestre.

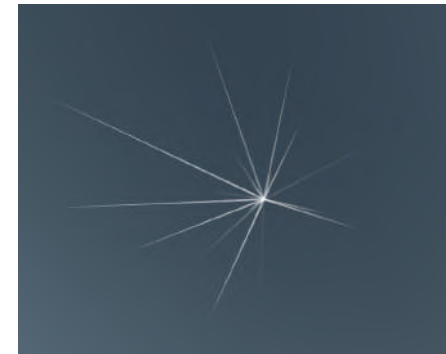


**IMPACTS  
PROJECTILES  
MITRAPOMPE**



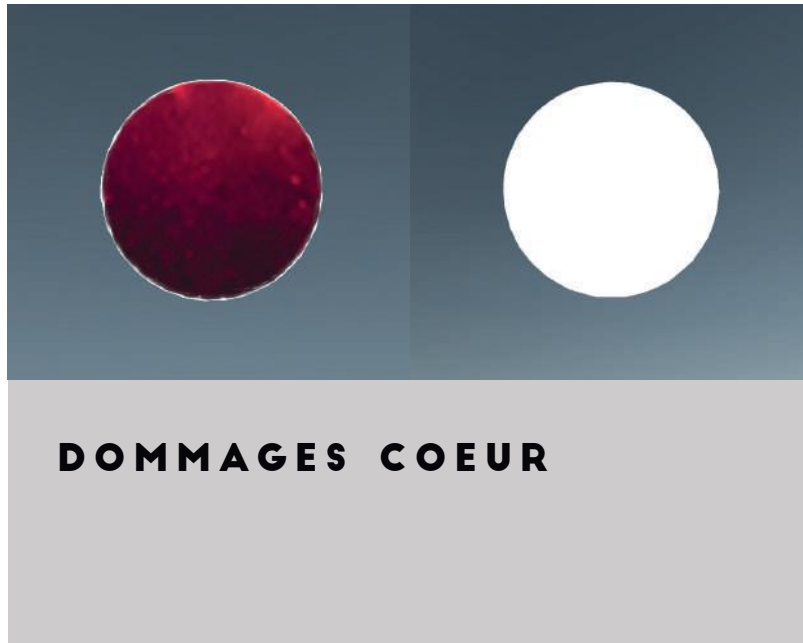
**EXPLOSION  
PROJECTILES  
LANCE-GRENADES**

(D'après 'Explosion projectiles  
Lob/Sticky')

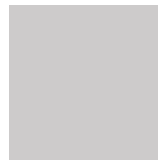


**FISSURE  
COEUR &  
SPAWN**

(D'après 'Destruction  
Cœur')



**DOMMAGES COEUR**

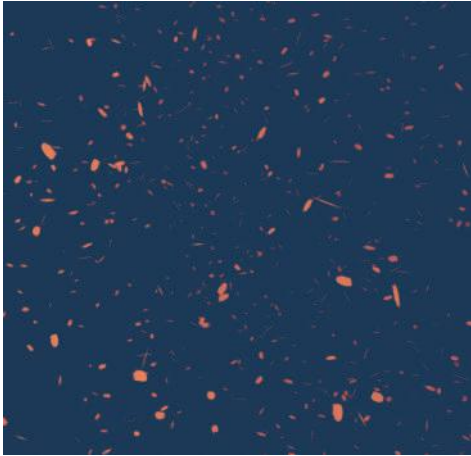


## **FUMEE PILIERS**

Description : effet de fumée et particules créés sur les côtés d'un cercle.

Condition d'apparition : les piliers entrent en collision avec l'arène tombante.





## **CENDRES**

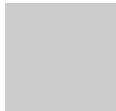
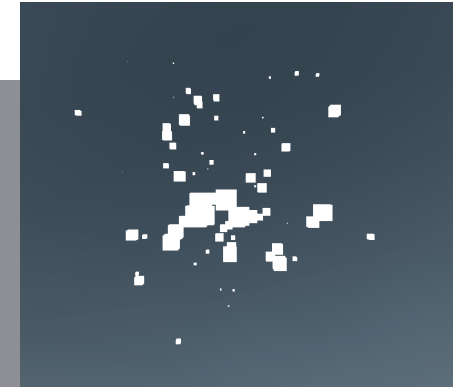
Description : effet de bouts de cendres se faisant porter par le vent.

Condition d'apparition : tout le temps.

## **DESTRUCTION CHAINES**

Description : explosion des maillons des chaînes accrochées à l'arène.

Condition d'apparition : lors de la destruction des maillons.



## COULEURS

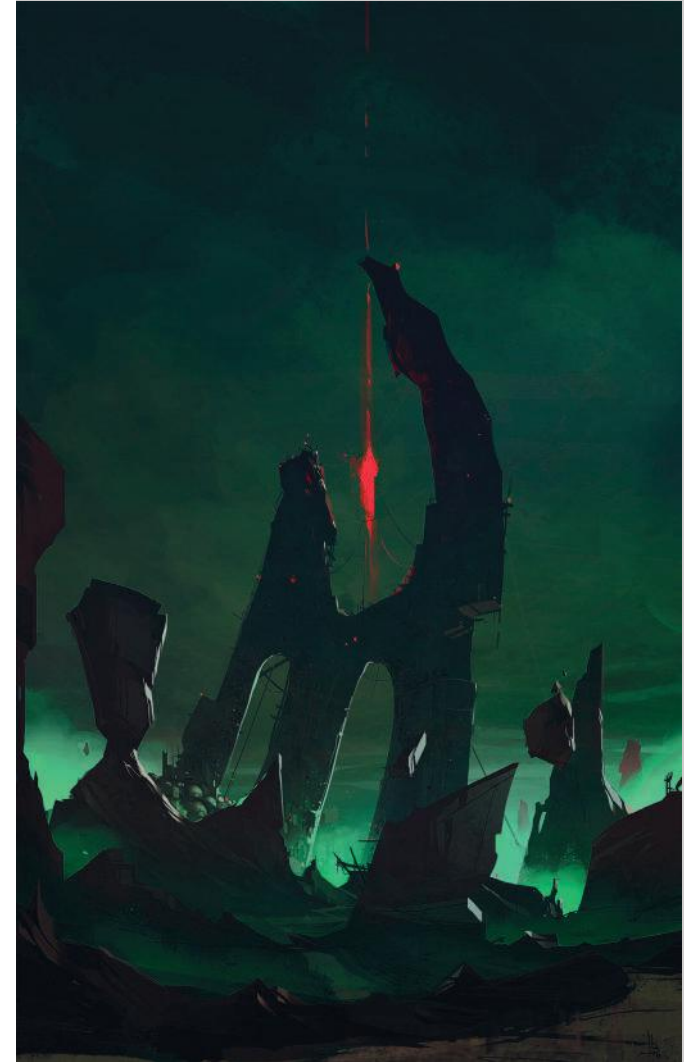
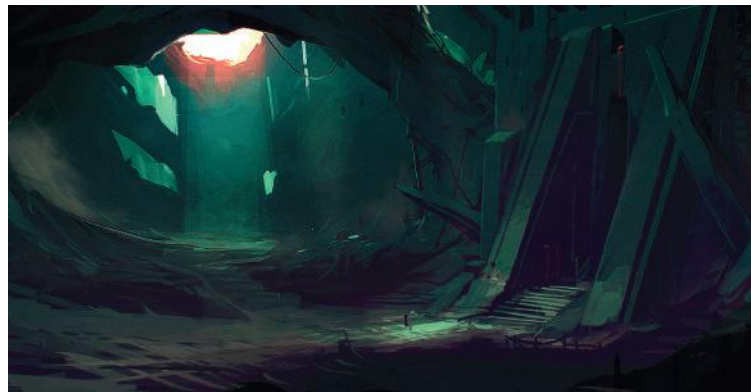
---

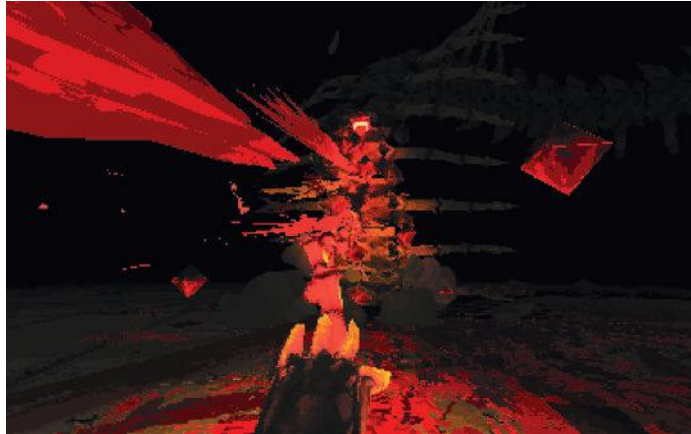
Durant le premier semestre, le projet était volontairement mis en noir et blanc afin de nous concentrer sur la disposition globale des éléments. C'est une technique également utilisée en art, pour avoir d'abord une vue d'ensemble sur une production et ensuite venir rajouter les couleurs progressivement, de manière cohérente et esthétique. Les couleurs ont donc été travaillées à partir du deuxième semestre.



## RÉFÉRENCES

Nous trouvons l'agencement du rouge et du vert intéressants. Cependant, étant des couleurs complémentaires, il nous a été trop difficile de les gérer dans le projet.





Finalement, ce sera les couleurs de notre référence principale *Devil Daggers* qui aura été l'inspiration pour les couleurs finales. La dominance de rouge, avec des zones sombres, correspondait parfaitement à l'ambiance dont nous voulions faire part.



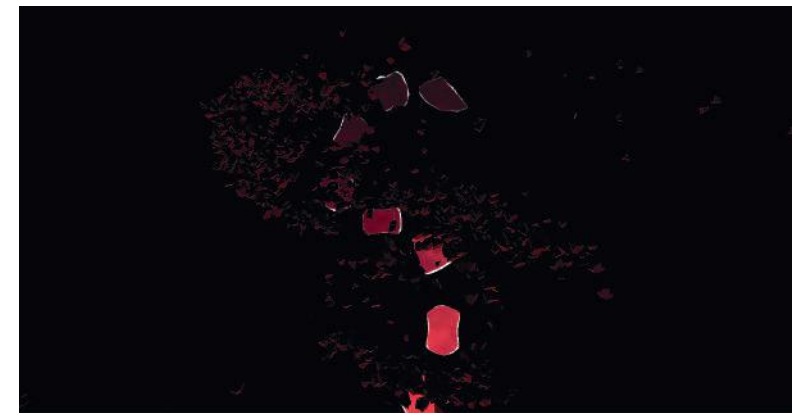
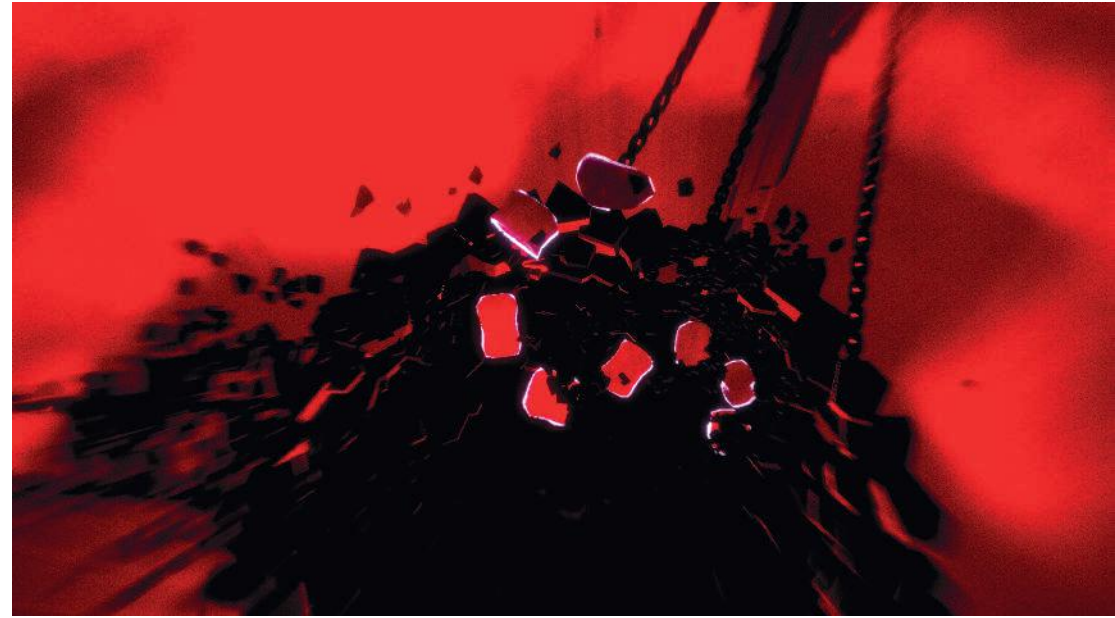
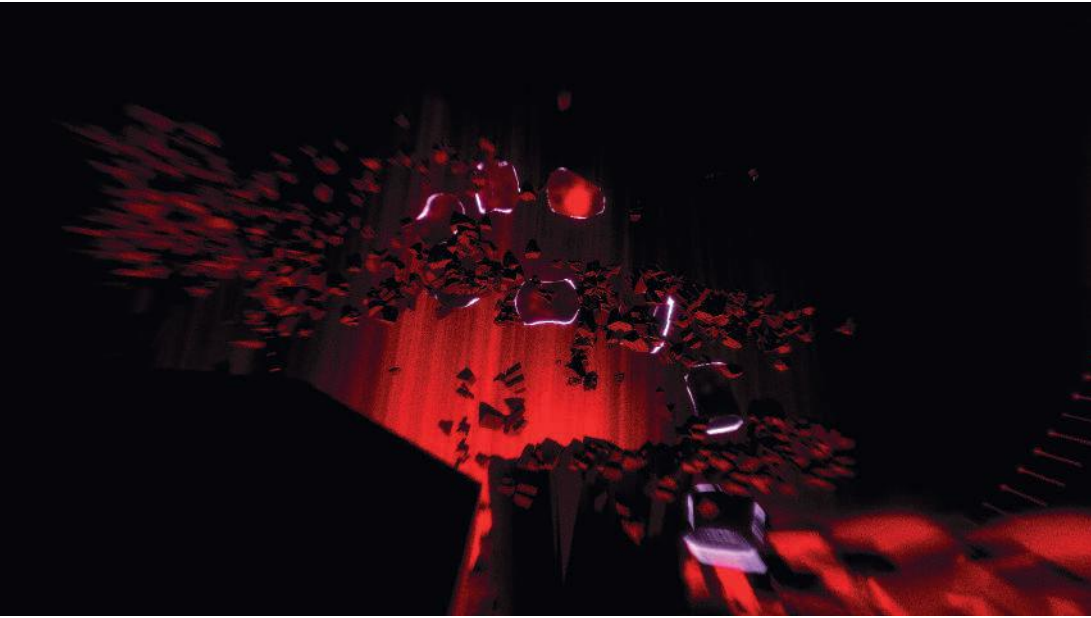
**1ER  
SEMESTRE**

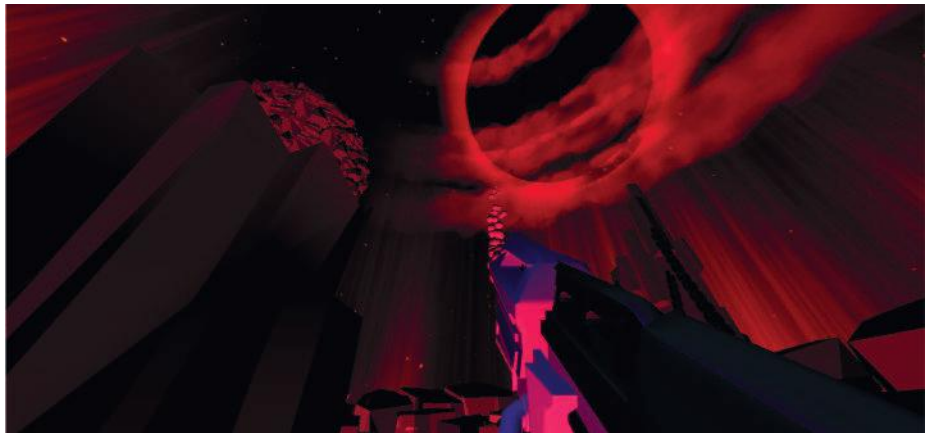
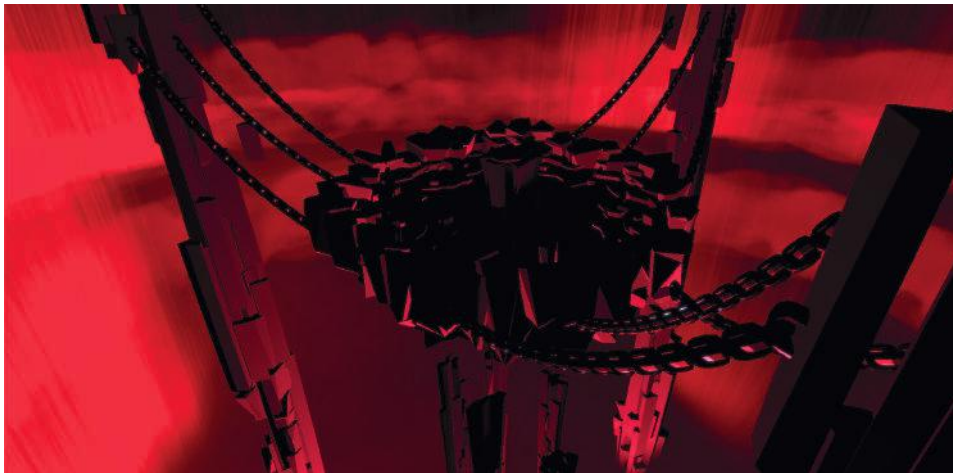
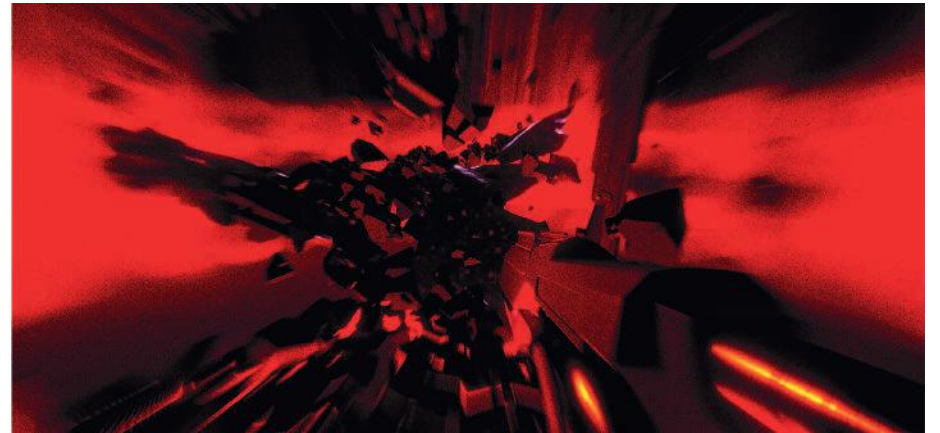
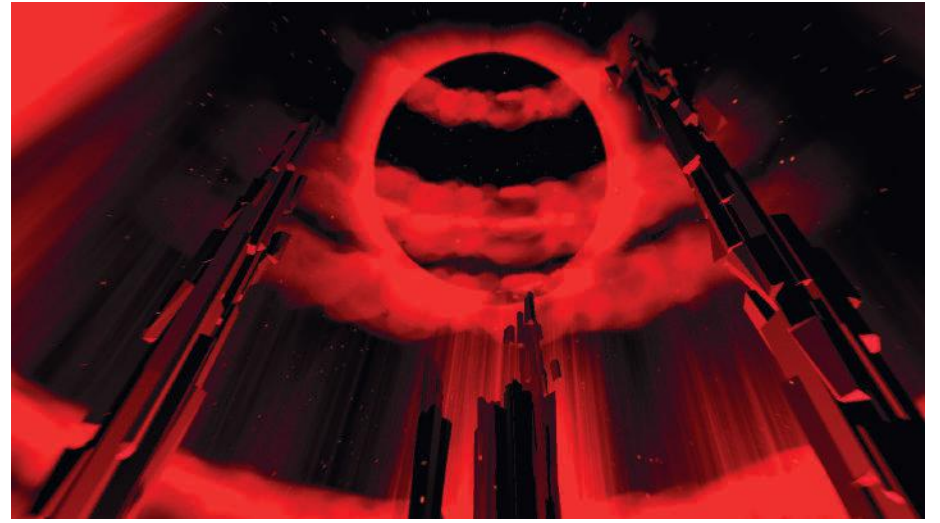
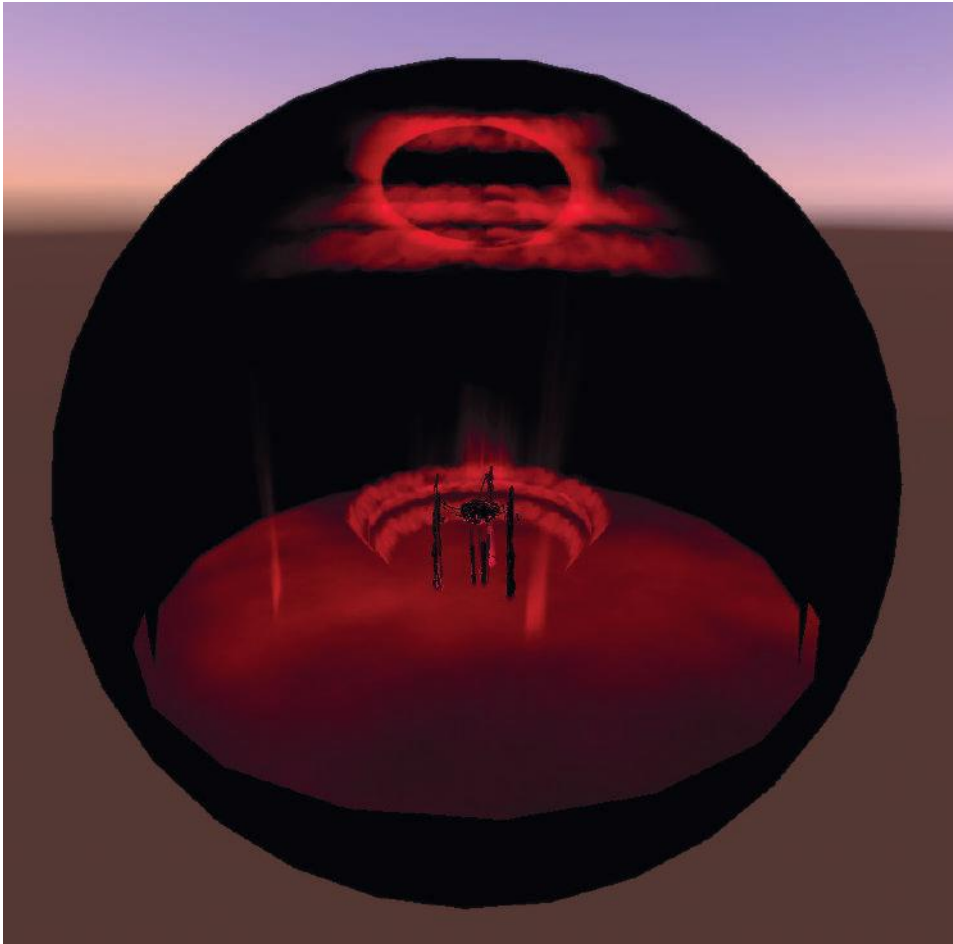


**2EME  
SEMESTRE**



# VISUELS DU JEU





# CHARTRE SONORE

## INTENTIONS

—

Pour *Lodrasqe*, nous voulions proposer une ambiance sonore travaillant conjointement avec celle visuelle, cela afin de renforcer cette atmosphère angoissante présente dans notre jouet du premier semestre. Nous avons donc, dans un premier lieu, cherché les points rendant notre univers angoissant et quels seraient les éléments sonores pouvant appuyer cela.

À la suite de diverses recherches vidéo-ludiques, nous avons trouvé les éléments suivants particulièrement importants : la spatialisation du son, les effets sonores rendant nos sons organiques proches du synthétique, et la puissance sonore de certains feedbacks. Ces points ont été nos principaux axes de travail afin de créer une ambiance sonore qui nous convenait.

## MÉCANIQUE SONORE

---

La plus grande partie de nos events sonores sont spatialisés. Le layer sonore du boss possède des sons avec une portée très faible. Cela permet de rajouter des sons très imposants sans créer une cacophonie constante.

Tout cela nous a permis de rendre le Lodrasqe angoissant. Les sons du layer d'environnement ont une portée très importante afin que le joueur puisse ressentir la présence du Lodrasqe de loin, tout en le rendant encore plus massif.

## RÉFÉRENCES ET CRÉATIONS SONORES

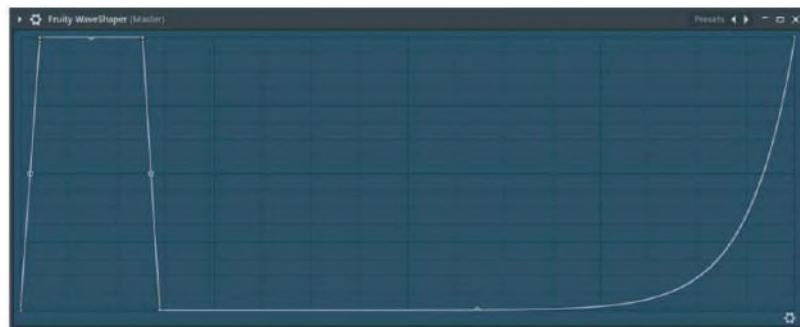
---

### **BOSS**

Les effets sonores présents dans *Devil Daggers*, et plus précisément le Centipède, ont été pour nous une de nos sources d'inspiration principale pour les sons émis directement par notre créature. En effet, le côté organique et angoissant des effets sonores liés à ce Centipède étaient vraiment intéressants pour notre boss, par le côté pesant que nous voulions créer dans le jeu.

Pour le retranscrire, nous avons enregistré des bruits d'inspirations « difficiles », en appuyant sur notre gorge pour simuler l'effet « râle d'agonie » que nous avons stretch en utilisant l'algorithme « speech » dans *FL Studio* pour l'allonger sans changer le pitch. Nous avons ensuite appliqué un effet de distorsion particulier appelé le WaveShaper.

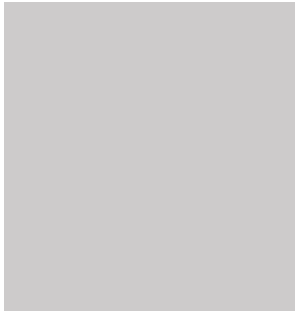
## CENTIPEDE



## WAVESHAPER

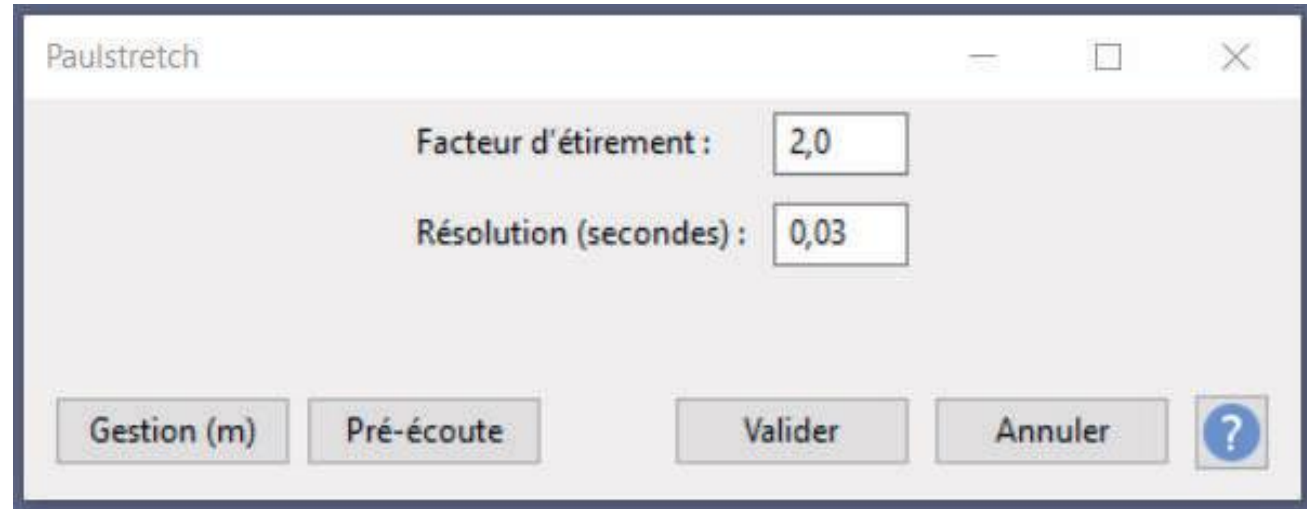
Le WaveShaper va changer le volume de sortie selon une courbe. L'abscisse correspond au volume du son d'entrée et l'ordonnée correspond au volume du son de sortie. Plus le volume d'entrée est à gauche, plus il est faible, et inversement à droite. Pour celui de sortie, plus il est en hauteur, plus il sera fort, et inversement s'il est en bas.

Cet effet nous permet donc de faire ressortir les sonorités faibles d'un son et/ou d'enlever les sonorités fortes. C'est un moyen très utile pour isoler des sons différents qui ont la même fréquence. Mais dans notre cas, il nous a permis de faire une « bouillie » sonore avec tous les bruits d'imperfection de notre micro (micro intégré à un PC portable de basse qualité par exemple), mais aussi avec des sons faibles que le micro captait : les bruits de gorge, de bouche en tout genre, les respirations... Cet outil nous a permis de beaucoup expérimenter et de créer plusieurs sonorités différentes en partant d'un seul sample. Ce qui a accéléré notre production puisque nous n'avons pas à constamment chercher de nouveaux samples.



Un autre logiciel que nous avons utilisé pour créer les sons du Lodrasq est Audacity, et principalement l'effet de paulstretch.

Le facteur d'étirement aura pour fonction de définir la durée du son après traitement selon sa durée de base. Par exemple, un enregistrement de 1 minute avec un facteur d'étirement de 10 fera 10 minutes après modification.



La résolution consiste en la taille de chaque sample que le paulstretch va utiliser pour étirer le son. Par exemple, si on choisit une valeur de 0.5 secondes, il va prendre des morceaux de cette durée-là et les répéter. Plus la valeur est basse, plus la résolution du son sera haute.

Dans notre cas, nous avons utilisé le paulstretch de deux manière différentes :

- effet « Démon » : résolution très courte (0,01 seconde) ;
- effet « Fantôme » : résolution longue (+ 1 seconde) et facteur d'étirement élevé ( 5+).

L'effet Démon donne un aspect mécanique au bruit du Lodrasq (bruit de déplacement, bruit de respiration, ect...). La résolution très courte va donner un effet « haché » aux sonorités.

L'effet Fantôme lui va servir pour les voix du Lodrasq. La résolution longue et le facteur d'étirement élevés vont étendre le son sans déformer sa nature, mais les sonorités seront toutes mélangées.

## ARMES

Les armes sont composées de trois sons : un enregistrement d'impact mécanique, un son d'impact synthétique, avec une légère distorsion pour l'accentuer, et un bruit mécanique aigu joué juste après, pour créer une sorte d'impression de « mouvement » mécanique dans l'arme, comme si le tir déclenchait une sorte d'articulation.

Pour comprendre le fonctionnement des armes, et donc pouvoir proposer un son cohérent, nous nous sommes intéressés à des schémas de coupes d'armes afin de comprendre les éléments les composant et leur fonctionnement.



## SOUND LIST



LAYER	ASSET	CONDITIONS DE MODIFICATION	CONDITIONS D'APPARITION	3D
				102

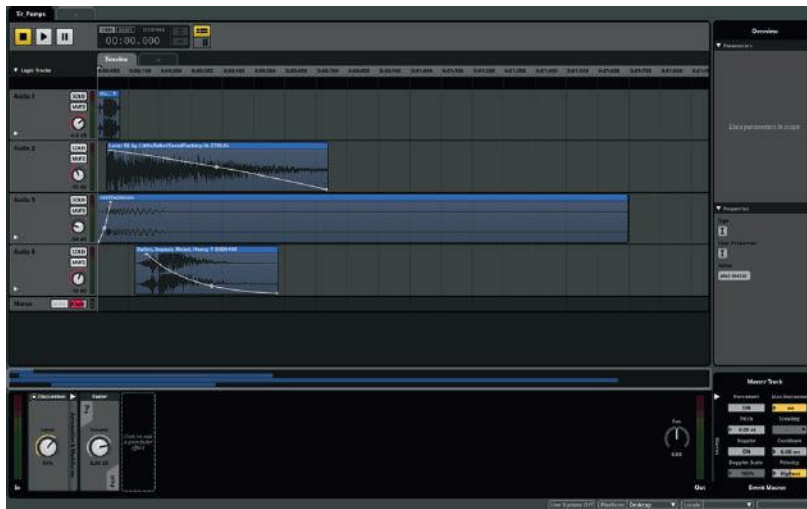
Avatar	Avatar_GameOver		Se déclenche à la mort de l'avatar	
Avatar	Avatar_RocketJump		L'avatar n'est plus en contact avec le sol suite à un tir de Lance-grenades	
Avatar	LanceGrenade_Lancement		Le joueur appuie sur l'input de tir tout en ayant sélectionné le lance grenades	
Avatar	LanceGrenade_Explosion	Réverbération quand l'avatar est éloigné	Le projectile du Lance-grenades entre en contact avec un élément solide, peut aussi se déclencher si le projectile n'a rencontré aucun obstacle durant 3 secondes	■
Avatar	Mitrapompe_Tir		Le joueur appuie sur l'input de tir tout en ayant sélectionné le Mitrapompe	
Lodrasqe	Chauve-souraie_BoulderImpact		Se déclenche quand les Blocs de la Chauve-souraie entrent en contact avec un élément solide de la scène	■
Lodrasqe	Chauve-souraie _Damage		Se déclenche quand un des Cœurs de la chauve-souraie est atteint par un projectile d'une arme de l'avatar	■
Lodrasqe	Chauve-souraie _Death		Se déclenche quand le dernier Cœur de la Chauve-souraie est détruit	■
Lodrasqe	Chauve-souraie _Screech	Réverbération quand l'avatar est éloigné	Se déclenche quand l'entité apparaît	■
Lodrasqe	Chauve-souraie _Spawn		Se déclenche à la fin de la phase précédente	■



Lodrasqe	Chauve-souraie_Transition		Se déclenche quand deux Cœurs de la Chauve-souraie ont été détruit	■
Lodrasqe	GalaxieMob_Screech	Réverbération quand l'avatar est éloigné	Se déclenche quand l'entité apparaît	■
Lodrasqe	GalaxieMob_Death		Se déclenche quand un monstre invoqué par la galaxie meurt	■
Lodrasqe	GalaxieMob_Spawn		Se déclenche à la fin de la phase précédente	■
Lodrasqe	Galaxie_Death		Se déclenche quand le Cœur du boss Galaxie est détruit	■
Lodrasqe	Galaxie_Screech		Se déclenche quand l'entité apparaît	■
Lodrasqe	Serpent_BoulderImpact		Se déclenche quand les Blocs du boss Serpent entrent en contact avec un élément solide de la scène	■
Lodrasqe	Serpent_Damage		Se déclenche quand un des Cœurs du boss Serpent est atteint par un projectile d'une arme de l'avatar	■
Lodrasqe	Serpent_Death		Se déclenche quand le dernier Cœur du boss Serpent est détruit	■

Lodrasqe	Serpent_Screech	Réverbération quand l'avatar est éloigné	Se déclenche quand l'entité apparaît	■
Lodrasqe	Serpent_Spawn		Se déclenche à la fin de la phase précédente	■
Lodrasqe	Tuto_Death		Se déclenche quand le Cœur de la phase tuto est détruit	■
Lodrasqe	Tuto_Lévitacion		Boucle sonore permanente, ne s'arrête que suite à la destruction du Cœur	■
Environnement	Rumble		Se déclenche quand l'environnement change de forme	■
Environnement	ChainBreak		Se déclenche au moment de la première phase de transition, lorsque les chaînes se détruisent	■
Environnement	Impact		Se déclenche quand l'arène s'empale sur les piliers et qu'elle se détruit	■
Introduction	Récupération Lance-grenades		Se déclenche quand l'avatar entre en contact avec le Lance-grenades	
Introduction	Récupération Mitrapompe		Se déclenche quand l'avatar entre en contact avec le Mitrapompe	

Introduction	Son émis par les armes		Se déclenche à l'apparition des armes	■
UI	Menu principal		Loop sonore durant toute la période où le joueur est dans les menu	
UI	Clic		Se déclenche quand le joueur fait un clic gauche sur un des boutons du menu	



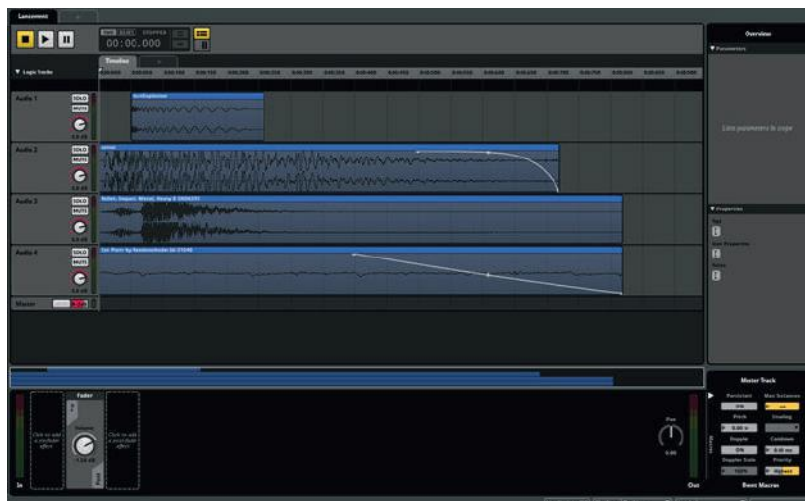
## EXPLICATIONS DES EVENTS SONORES FMOD

### TIR MITRAPOMPE

One shot 2D composé de 4 sons :

- son de tir d'arme à feu réaliste ;
- son de réverbération du son de tir avec des aigus diminués ;
- son de kick synthétique saturé ;
- son d'impact de métal.

Le son du Mitrapompe doit être satisfaisant, tout en donnant l'impression qu'on tient une vraie arme dans les mains. Le son est en 2D puisque l'arme est toujours dans la même position par rapport aux oreilles du joueur. Chaque son est pitché aléatoirement de manière légère pour éviter une répétitivité.

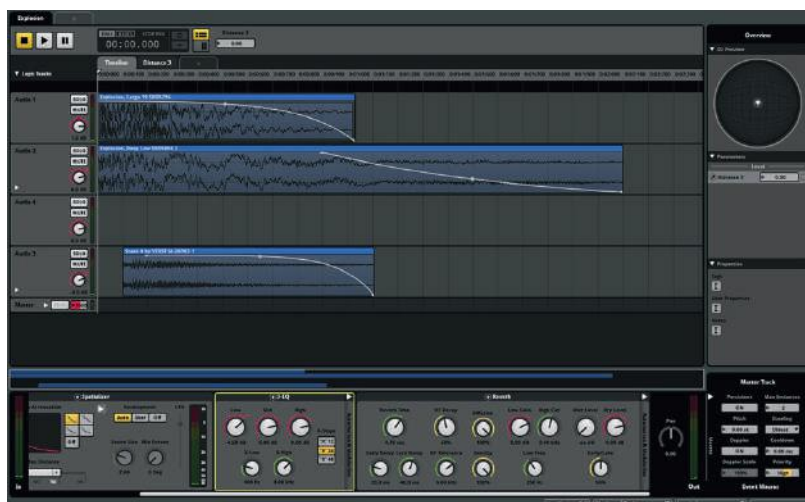


## TIR LANCE-GRENADES

One shot 2D composé de 4 sons :

- son de kick synthétique ;
- son d'explosion pitché pour le rendre plus grave ;
- son d'impact de métal aigu ;
- son de ronronnement de chat distordu et étiré.

Le bruit du tir devait être synonyme de puissance et de lourdeur, avec les mêmes contraintes que le bruit du Mitrapompe. Nous avons donc opté pour la même logique que le Mitrapompe mais en plus grave et en plus « ample » pour accompagner la puissance brute du tir du Lance-grenades. Chaque son est pitché aléatoirement de manière légère pour éviter une répétitivité.



## EXPLOSION PROJECTILE LANCE-GRENADES

One shot 3D composé de 3 sons :

- son d'explosion réaliste avec un filtre passe-bas ;
- son d'explosion réaliste plus grave et saturé ;
- son de caisse claire.

Le son d'explosion n'est pas lié à un objectif primaire (faire des dégâts aux Cœurs), mais plutôt à un objectif secondaire qui est d'éloigner les Blocs des Cœurs. Nous avons intentionnellement diminué l'impact de l'explosion sur l'environnement sonore. Le volume de l'explosion doit être important à courte portée mais avait besoin de pouvoir être entendu de très loin. La courbe

d'atténuation du son a été choisie pour qu'il soit audible au même volume partout sur le niveau. Il y aussi un effet de reverb qui diminue au fur et à mesure que la distance entre l'explosion et le joueur diminue.

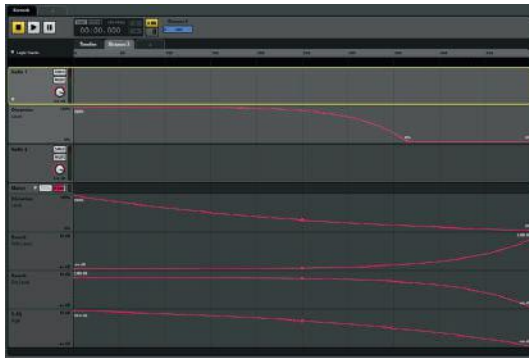


## SCREECH CHAUVE-SOURAIE

Boucle 3D composée de 2 sons :

- bruit d'inspiration saturée faible ;
- cri aigu saturé.

Ce son sert à angoisser le joueur et en même temps paraître « naturel », comme si le Lodrasqe respirait en faisant ce bruit. Sa spatialisation a une portée faible pour qu'on puisse l'entendre quand le joueur est en danger. Le début de lecture de la piste est décidé aléatoirement, il est aussi pitché aléatoirement de manière légère. Chaque Cœur va produire ce son de manière indépendante, il faut donc qu'ils puissent être différenciés. Une reverb est appliquée sur tout le son, permettant de donner une présence à l'entité même si elle est à distance, et de donner également une impression que l'entité prend de la place. Le volume de la reverb du screech du boss est proportionnel à la distance entre l'avatar et l'entité. À l'inverse, la distorsion devient plus forte lorsque l'avatar en est proche.



## DOMMAGES SUBIS PAR LE LODRASQE

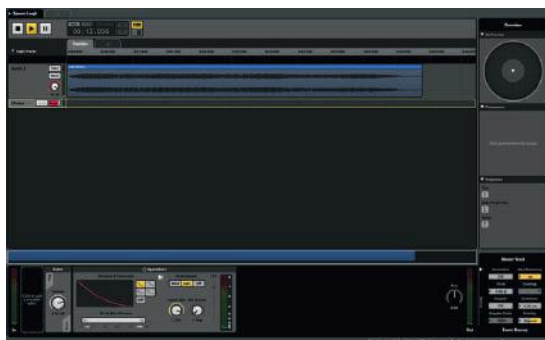
One shot 3D composé de 2 multi-instruments :

- petit cri strident saturé ;
- petit bruit d'impact synthétique style 8-bit saturé.

Le son sert de feedback de dégât au Cœur, il devait donc être satisfaisant et compréhensible tout en gardant l'identité du Lodrasqe. Il possède une portée d'entente importante puisqu'il sert de récompense au joueur ayant touché un Cœur, il doit pouvoir l'entendre de partout. Chaque son est pitché aléatoirement de manière légère pour éviter une répétitivité.

Un son est utilisé pour signifier que l'entité perd des point de vie, un autre pour marquer que la créature souffre/est affectée, et un pour donner du contexte.

Le son du cri de douleur a la même portée que les autres sons de la créature, plus faibles que la distance d'impact. Cela pour rendre cohérent l'univers et la spatialisation des sons.



## APPARITION DU BOSS GALAXIE

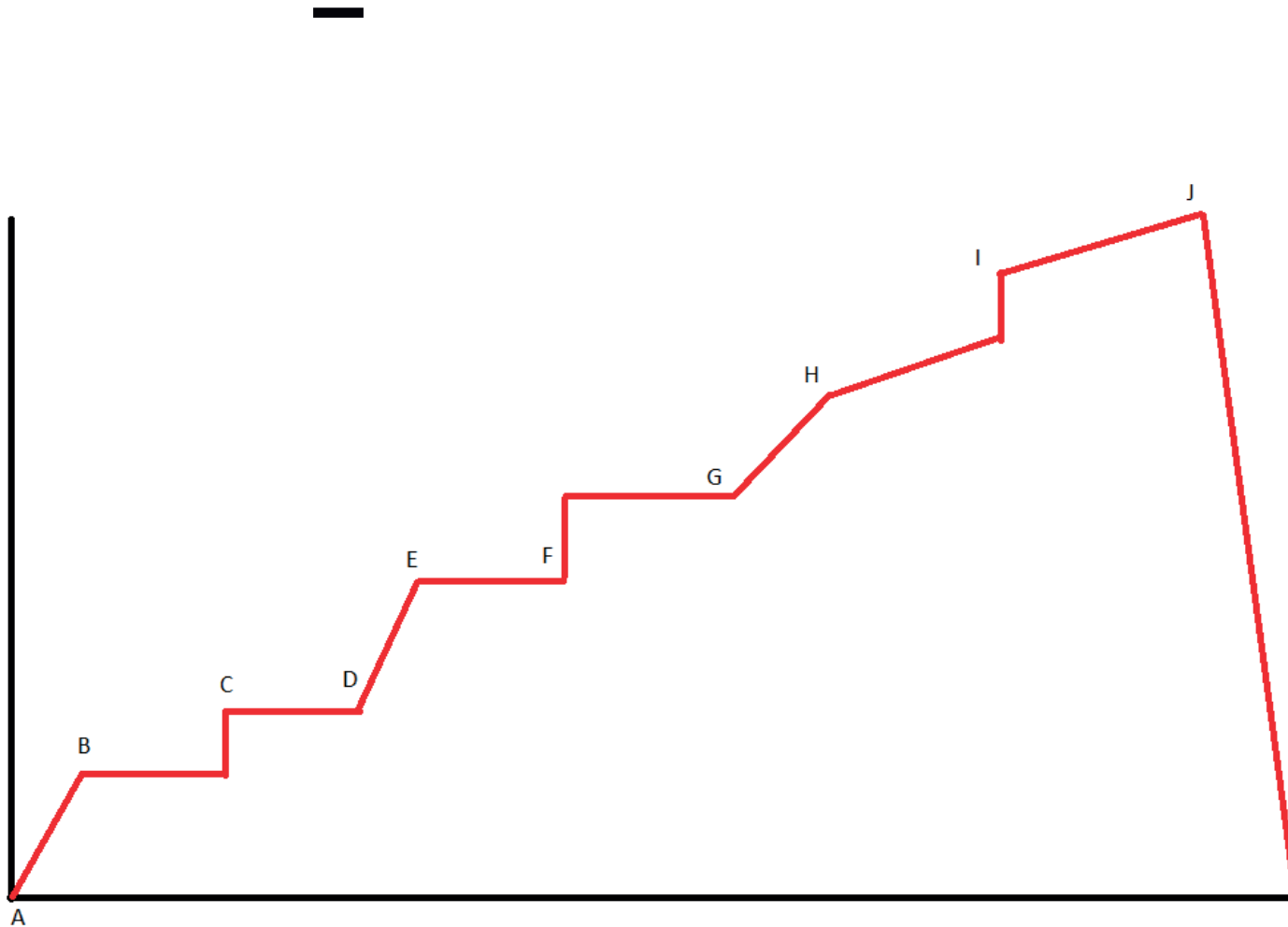
One shot 3D composé d'un seul son de rire démoniaque, créé à l'aide d'un site de Text to Speech (<https://ttsdemo.com/>). Le son de rire lui-même correspond à un rire humain, rendu plus grave. Il a été étendu avec un paulstreich à résolution courte, mais avec un étirement important. Le son étiré a ensuite été modifié avec le WaveShaper de sorte à créer un « kick » au début de la lecture. Le rire a été tellement distordu que le clipping obtenu nous sert de kick. Le son a ensuite été lourdement compressé (en volume) pour qu'il soit plus audible, et une forte reverb a été ajoutée.



## SITE DE TEXT TO SPEECH UTILISE



# TIMELINE DE GAMEPLAY



A : Spawn  
Chauve-souraie

B : Apparition  
Chauve-souraie à l'écran

C : Passage phase 2  
Chauve-souraie

D : Mort Chauve-souraie

E : Spawn + apparition à  
l'écran du Serpent

F : Passage Phase 2  
Serpent

G : Mort Serpent

H : Spawn + apparition à  
l'écran Galaxie

I : apparition de la vague  
supplémentaire de mini  
entités

J : Mort de la Galaxie

# FUTUR DU PROJET

L'aventure *Lodrasqe* s'est déroulée sur une année scolaire. Durant cette période, nous avons pu réaliser à quel point produire sur un semestre et sur une année était différent. Nous avons fait face à de nombreux soucis techniques et humains mais avons su rester soudés au sein de l'équipe.

Après cette année de production, nous sommes fiers du jeu que nous avons réalisé, fiers du fait qu'aujourd'hui, plus qu'un jeu, nous avons une méthode de production qui nous aurait permis, si nous l'avions souhaité, de produire d'autres boss. Nous avons le système, les règles et toute la méthode de production de créés. Aujourd'hui, si une nouvelle équipe décidait de reprendre le projet, nos règles de design pourraient permettre, sans soucis,

de produire du nouveau contenu. De cela, nous en sommes fiers, car c'est une preuve que nos années d'études sont le fruit d'une expérience riche, nous ayant donné les connaissances nécessaires au design de jeux.

Nous ne continuerons pas *Lodrasqe*, car à présent, chacun a ses ambitions et il est temps pour l'équipe de se séparer. Cependant, grâce à ces efforts communs, nous avons tous un projet magnifique à montrer dans notre portfolio.

Merci d'avoir pris le temps de lire ce document. Toute l'équipe de *Lodrasqe* vous souhaite une bonne expérience de jeu !

**TEAM TEKATE** —