

ROBOT'S PUNCH

GAME DESIGN DOCUMENT

PROJET DE SEDLIK GUILLAUME, SARTI BASTIEN, ET CAPITAINE JULIEN



Sommaire:

Game Overview:	2
Règles:	2
Structure:	3
Gameplay:	
Controle:	5
Caméra:	5
HUD:	6
Mécanique:	6
Level Design:	7
L'Avatar:	9
Mr.Robot:	9
Game State:	9
Paramètre atomique:	10
Les Ennemis:	11
Les occupants de l'usine:	11
Game State:	11
Paramètre atomique:	12
Autres Ingrédients :	12
Murs Destructibles:	12
Tapis Roulants:	13
Obstacles:	13
Spawner:	14
Direction Artistique:	14
Intention :	15
Texturing:	18
Intention:	18
Références:	18
Audio:	20
Intention:	20
Référence:	20
SDD:	21
Feedbacks sonores :	21
Musiques : [annulées]	22
Evolution Possible du Projet:	23

Game Overview:

Pitch: *Robot's Punch !* est un jouet se basant sur le comportement, aussi bien celui du joueur que des ennemis, une fluctuation chaotique entre calme, violence et panique. Vous incarnez un petit robots conçu différemment, l'élément perturbateur ayant la capacité de créer le chaos;

USP/KSP : game feel de billard, fluidité du déplacement

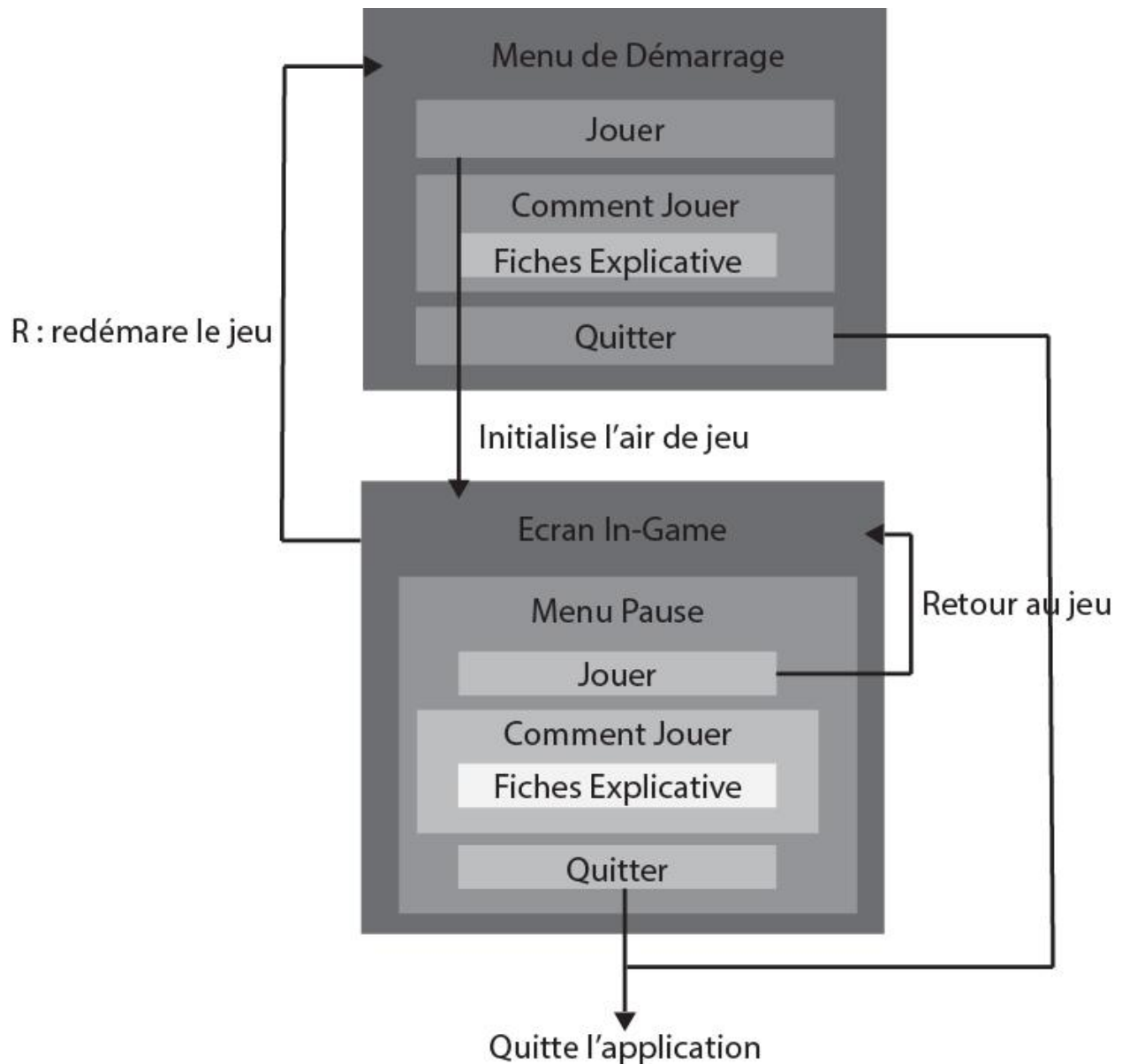
Mise en situation: L'action de *robot's punch !* se passe dans une usine dont aucun employé n'est humain; les robots qui parcourent l'usine ne sont pas hostiles, ils effectuent ce pourquoi ils sont programmés : produire. L'endroit semble par endroit mal entretenu, notamment à cause de la rouille; les robots ne s'occupent ni de l'état du local, ni de leur propre intégrité. Les capacités de l'avatar sont liées au chaos; c'est ce que fera principalement le joueur, que ce soit pour s'amuser à faire du "chamboule-tout" avec les bidons ou les caisses, ou que ce soit pour l'adrénaline de combat à un contre tous.

Règles:

Le jeu prend place dans un espace fermé et obstrué par de nombreux éléments. Des robots effectuent les tâches pour lesquelles ils sont programmés. Ces robots apparaissent de différents endroits jusqu'à atteindre une certaine population. Ils sont programmés pour défendre l'usine en cas d'attaque et pour fuir si une menace en élimine une trop grande partie. Le Joueur est le seul robot qui ne se conforme pas à ces règles et peut donc agir comme il l'entend dans l'usine.

Structure:

Structure de l'application:



Structure d'une partie type:

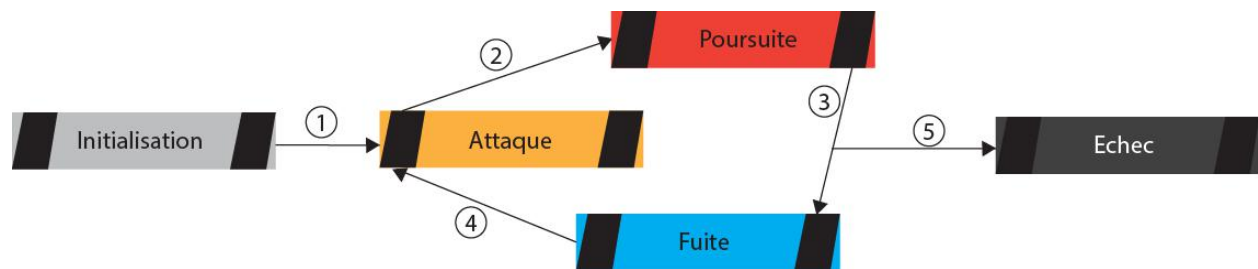
Initialisation: Début de la partie, les ennemis ne sont pas encore apparus. Le joueur peut se déplacer librement dans l'espace de jeu et utiliser son arme. (1) Après 10 secondes, les premiers ennemis apparaissent.

Attaque: Le joueur finit par toucher un ennemi qui en reculant va agresser un autre robot, et ainsi de suite. En réussissant des attaques puissantes les robots atterrissent de l'autre côté de l'espace de jeu, agressant d'autres robots. (2) A force d'agresser volontairement ou non de nouveau robot la majorité des individus finissent par poursuivre l'avatar.

Poursuite: Le maximum de population est atteint et le joueur doit se déplacer à travers le terrain pour ne pas se faire toucher. Si le joueur veut se débarrasser de la nuée, il doit prendre des risques et charger ses attaques. (3) Avec un coup parfait, le joueur inflige de lourd dégât à tout le peloton ce qui détruit une grande partie des poursuivants.

Fuite: La population de robots est à un niveau très bas, les robots restant fuient le joueur et tente de rejoindre certains points de la carte. Pendant ce temps, la vitesse d'apparition des robots a largement augmenté pour remonter la population à son niveau standard. (4) Une fois ce niveau atteint, les robots recommencent à chasser le joueur si celui-ci les agressent.

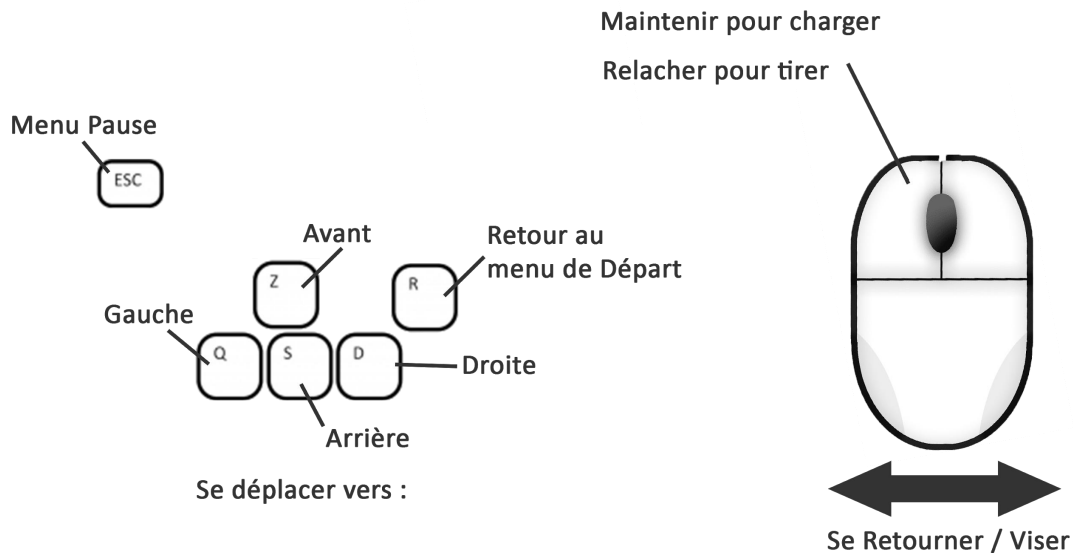
Échec: (5) L'avatar a été touché trop de fois et n'a plus de point de vie, il est détruit et le joueur peut recommencer une nouvelle partie. Cet état arrive le plus souvent lorsque beaucoup d'ennemis sont à la poursuite de l'avatar mais peut survenir à n'importe quel autre moment si le joueur n'est pas vigilant.



Structure d'une partie type

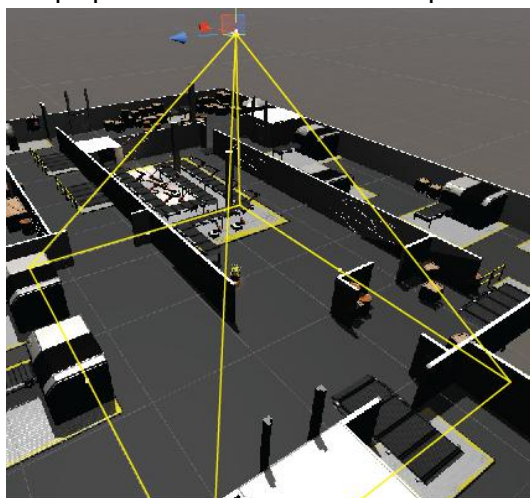
Gameplay:

Controle:

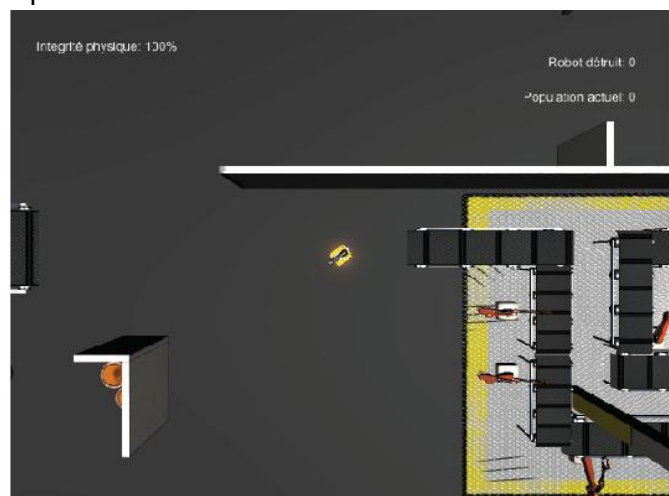


Caméra:

La caméra est en vue du dessus (Top-Down) et suit l'avatar avec un léger délai pour être plus agréable. La caméra ainsi positionner permet la vision d'un large périphérique autour de l'avatar ce qui permet une meilleur anticipation du comportement des robots.



Caméra dans l'éditeur

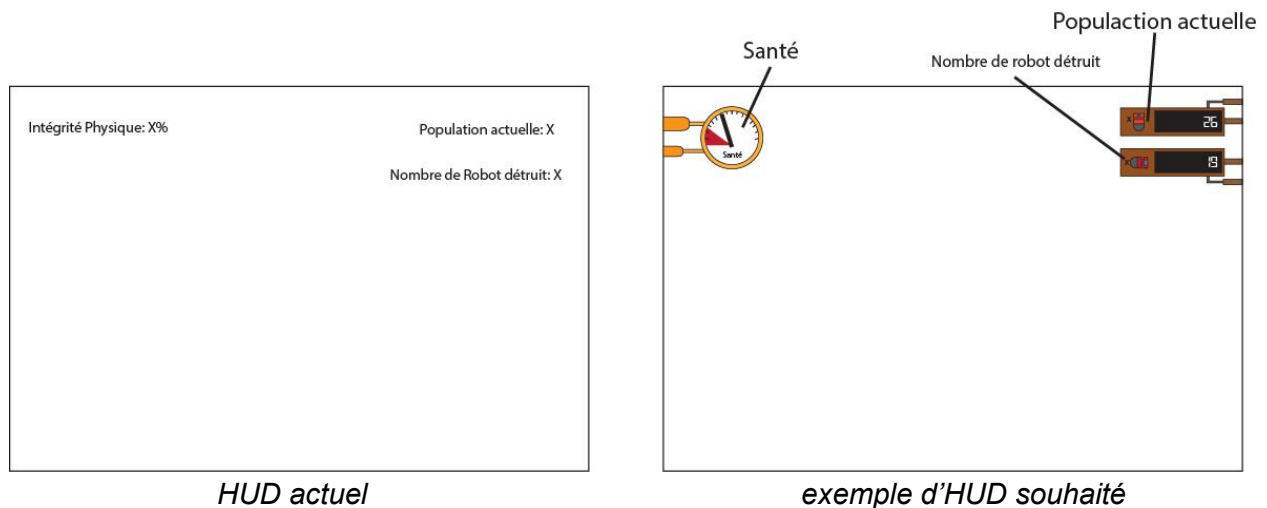


Vue en jeu

HUD:

L' HUD est actuellement composé de 3 lignes de texte qui permettent de connaître l'état de santé de l'avatar, le nombre de robots actuellement sur l'aire de jeu et le nombre de robots détruit jusqu'à là.

Nous aurions voulu implémenter une interface qui irait mieux avec la DA usine/robot que nous avons choisie.



Mécanique:

Mécanique du Joueur:

- Tirer : en maintenant la touche de tir puis en la relâchant (clic gauche), le joueur tire un projectile qui part toujours à la même vitesse mais dont la projection et les dégâts dépendent du temps de charge. Au moment du tir, si la puissance est supérieure à 80% de la puissance maximale, l'ennemi touché passe en "mode projectile". Si le joueur relâche la touche pendant que l'avatar clignote, il effectue un "tir parfait" qui inflige encore plus de dégâts.
- Se Déplacer: touche Z,Q,S,D du clavier, l'avatar se déplace dans la direction désirée.

Mécanique des Ennemis:

- Attaquer: En entrant en collision avec l'avatar, les robots hostiles font perdre des points de vie à ce dernier.
- Se Déplacer: les robots ennemis se déplacent toujours vers un point précis qui dépend de l'état dans lequel ils sont: standard, attaque ou fuite (voir Game State des ennemis).
- L'état de Projectile: lorsqu'ils subissent des chocs violents avec l'arme de l'avatar ou un autre robot déjà dans cet état, les robots ennemis deviennent eux-mêmes des projectiles capables d'infliger des dégâts à leurs congénères.

Mécanique d'objet Neutre:

- Tapis Roulant: Emporte les objets qui se trouvent dessus, n'est prenable que dans 1 sens.

Level Design:

Le Level Design est inspiré de la disposition de chaîne de production d'usine à l'aide de plan qui ont servi de modèle lors de la création de la carte de ce level design.



emière ébauche de level design avec une navigation prise en compte par rapport au prototype:

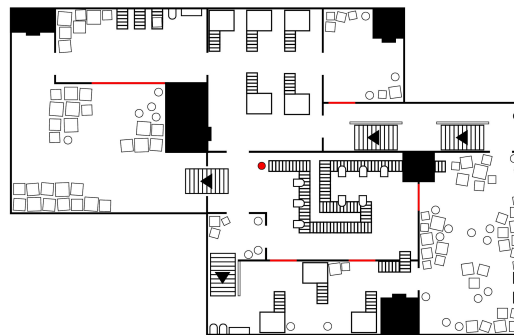
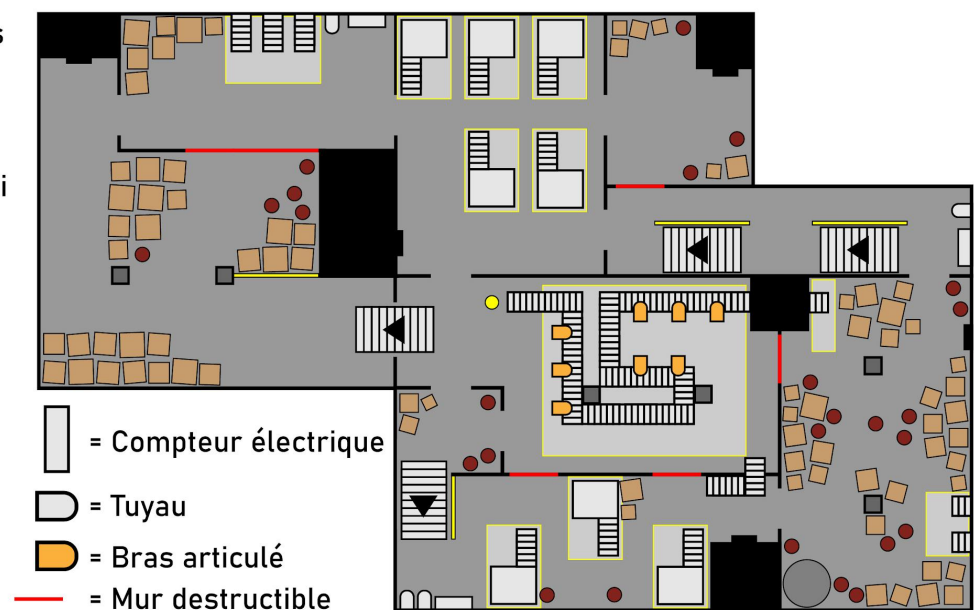


Schéma d'un Level

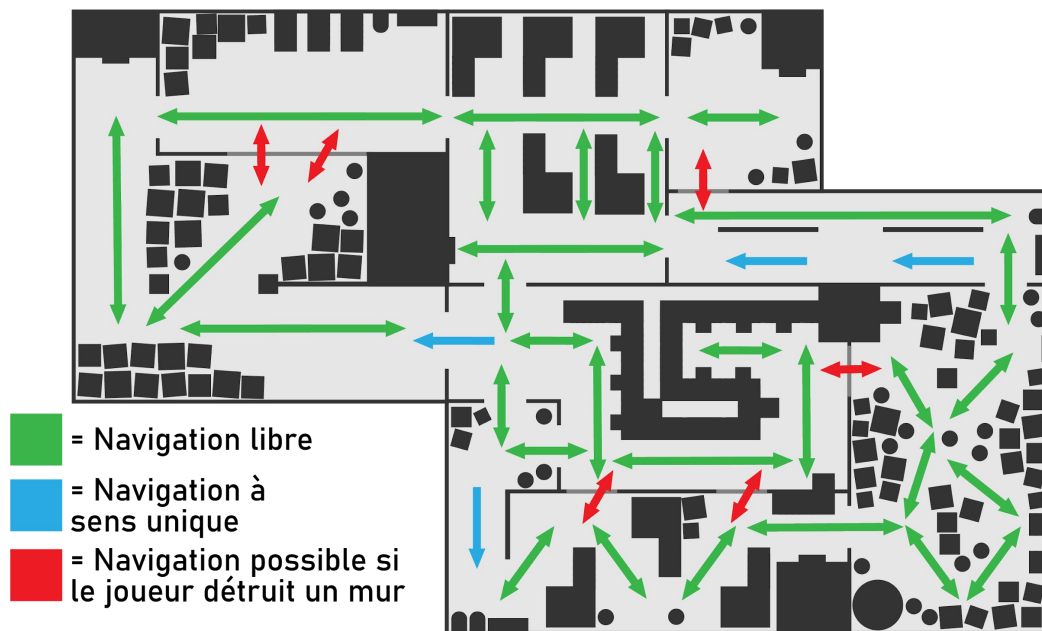
- = Caisse en bois
- = Baril
- = Barrière
- = Spawn ennemi
- = Spawn joueur
- = Tapis roulant
- = Atelier
- = Citerne
- = Poteau
- = Mur

Design définitif et légendé:

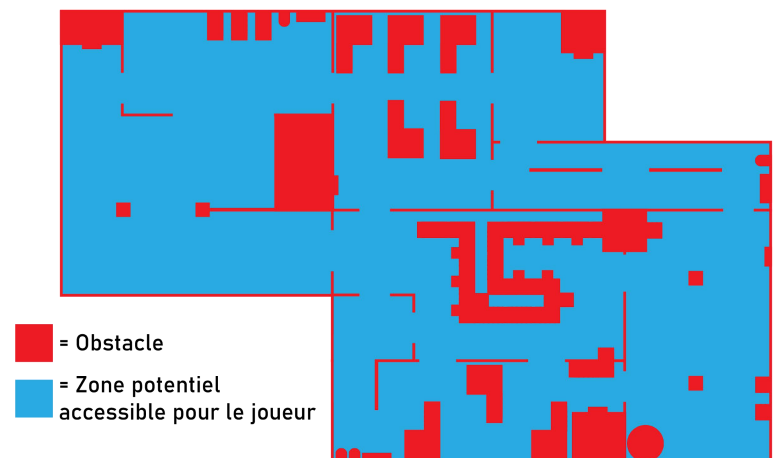
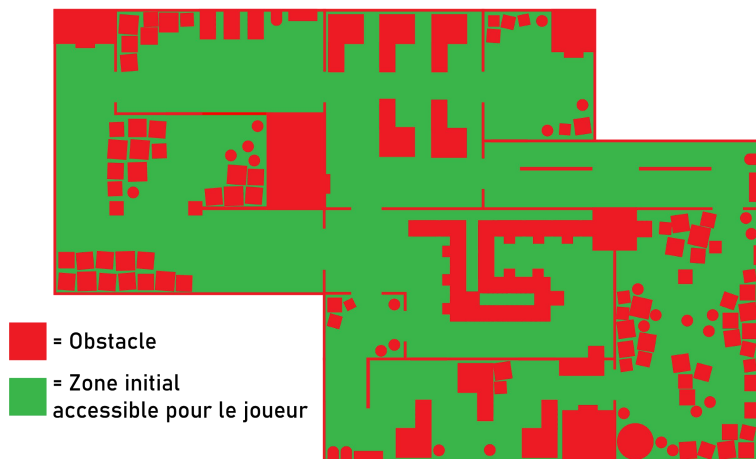


Ce level design est pensé globalement sous 2 angles, créer une navigation fluide au sein d'une usine permettant au joueur de ne pas trop se bloquer et de pouvoir se concentrer sur sa

situation actuel plutôt que d'avoir à réfléchir à son orientation dans cette usine (plusieurs manières de rattrapper un mauvais pathing, comme les boucles créé par la navigation, les tapis ou les murs);



Le second angle pris en compte lors de la création de ce Level Design est celui de la destruction, ou plutôt de l'impact des actions du joueurs sur la carte. Certains ingrédients (comme les barils, les caisses, les murs destructible) permettent donc d'avoir une évolution de l'espace dans lequel le joueur évolue.



L'Avatar:

Mr.Robot:

Créé machinalement par les robots-ouvriers avec des pièces qui n'étaient pas destinées à cette usine, Mr.Robot est resté pendant très longtemps à l'arrêt, ayant été considéré comme défectueux et mis de côté. Alors qu'il allait se faire recycler, il s'est rallumé : il se retrouve donc dans l'usine qui l'a fait différent, le seul aux alentours capable de raison. Enervé d'avoir été laissé de côté, il est maintenant déterminé à gêner le plus possible la production de cette usine.

Game State:

Standard : L'état le plus courant, c'est l'état dans lequel l'avatar est lorsqu'il se déplace ou est immobile. Aucun effet particulier.

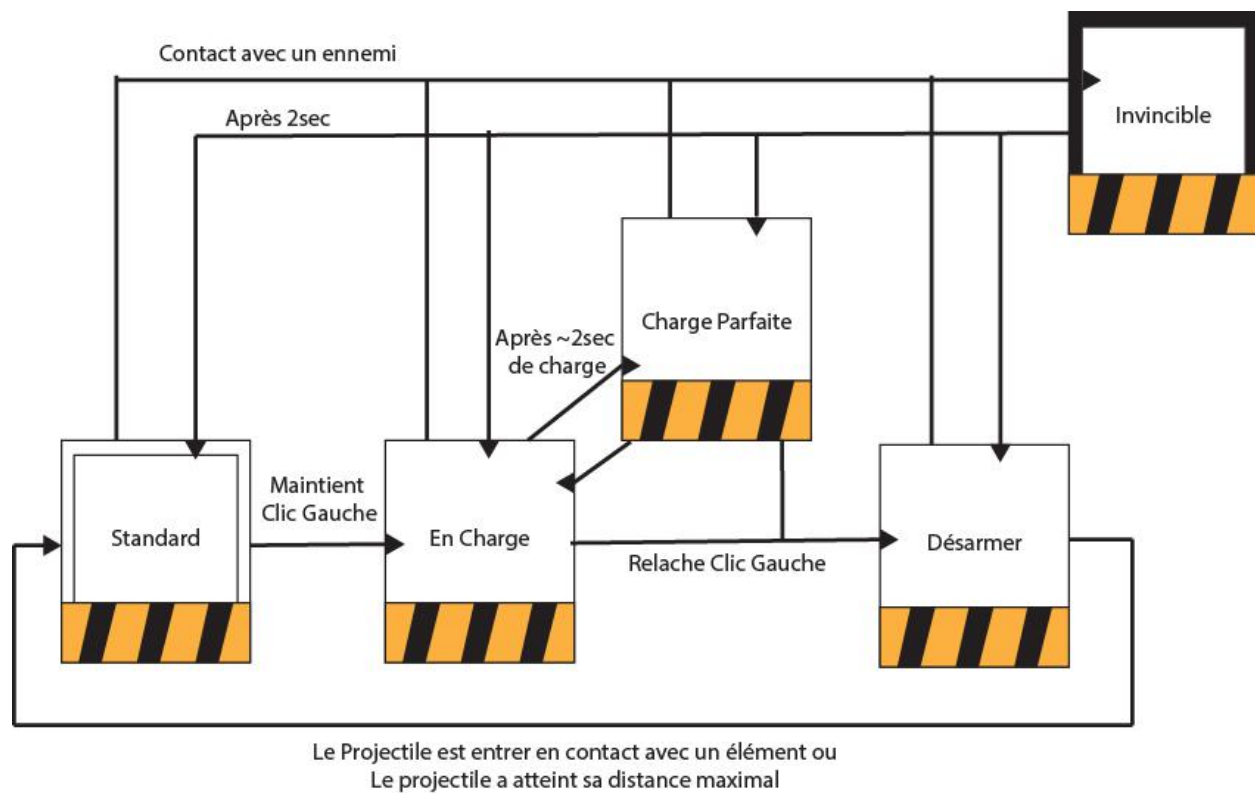
En charge: Actif lorsque le joueur se prépare à attaquer. La vitesse de l'avatar diminue et la puissance de feu augmente. Lorsque la charge atteint la puissance maximale, le joueur dispose de quelques frames pour relâcher le clic gauche et ainsi faire une attaque parfaite. Si le joueur reste appuyé sur le bouton de la souris, la puissance revient à sa valeur maximale standard. Cet état peut être gardé aussi longtemps que souhaité.

Charge parfaite: État éphémère durant lequel l'attaque inflige plus de dégâts et propulse plus loin si le clic gauche est relâché au bon moment.

Désarmer: Lorsque le projectile de l'attaque est lancé, le joueur ne peut pas préparer de coup jusqu'à ce que le projectile revienne de lui-même.

Invincible: Après être rentré en contact avec un ennemi, l'avatar devient invincible un court instant mais reste libre de ses mouvements et peut encore attaquer.





Paramètre atomique:

Vie	Vitesse de déplacement	Portée du projectile	Vitesse du projectile	Dégat du projectile (dégât max)
100	20	10	50	16-73(180-310)

Les Ennemis:

Les occupants de l'usine:

Ce sont les robots qui travaillent dans l'usine, ils sont programmés pour faire certaines tâches dont celle d'éliminer les agents hostiles au bon fonctionnement de l'usine. Ils peuvent être assez résistants, et sont très remplaçables : Ils sont créés à la chaîne et se font remplacer indéfiniment.

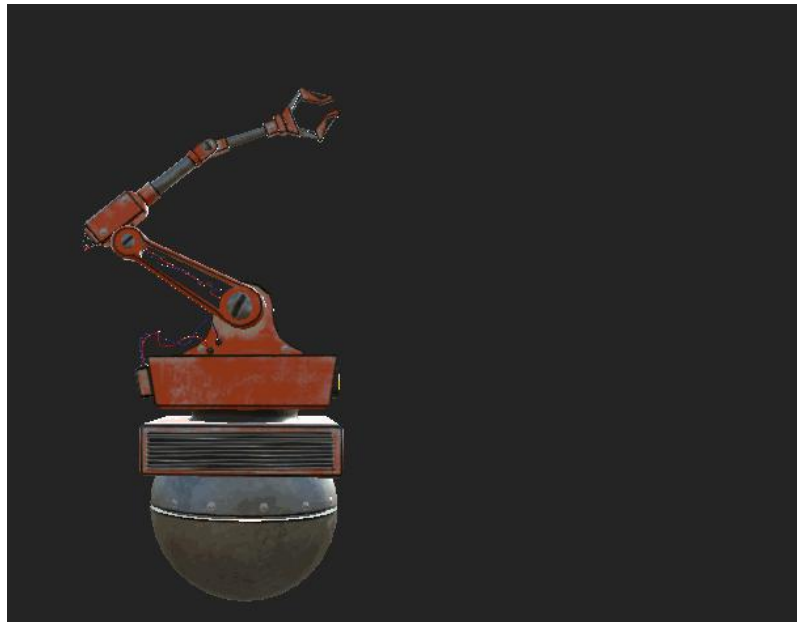
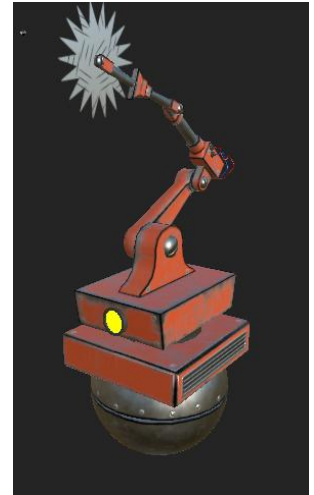
Game State:

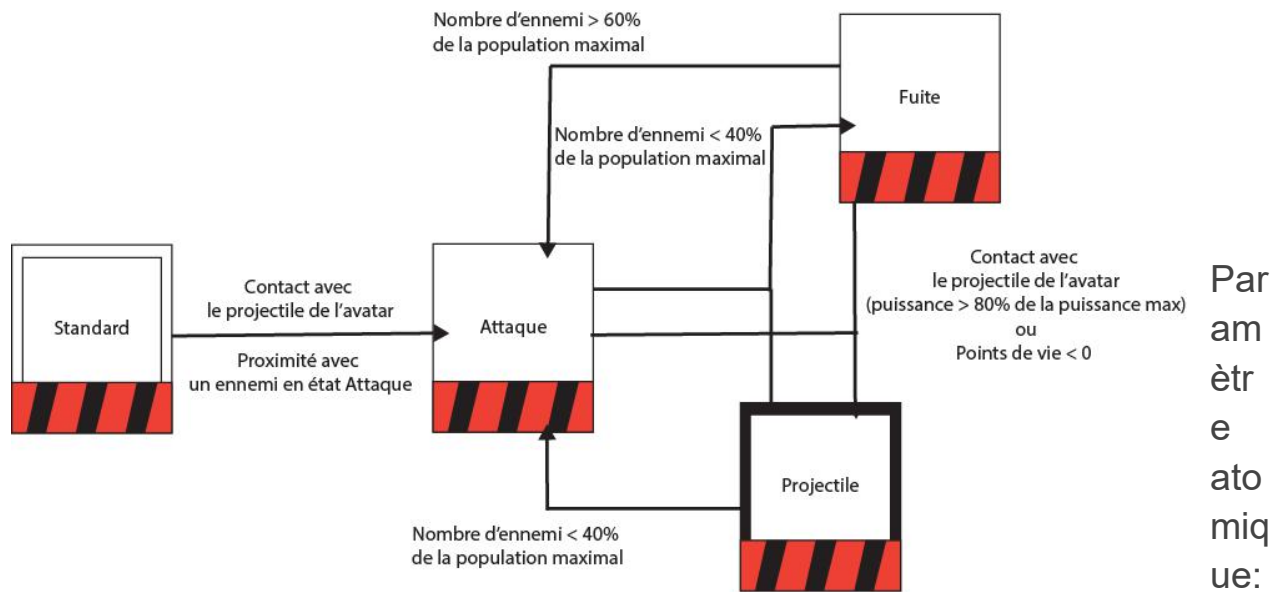
Standard: Etat dans lequel est l'agent lorsqu'il apparaît dans l'espace de jeu. Il se rend à certain endroit de l'usine pour effectuer les tâches auxquelles il est affecté. Dans cet état, il n'attaque pas le joueur.

Attaque: Après être entré en contact avec l'arme de l'avatar, l'agent prend l'avatar en chasse. Cet état peut également être atteint si un robot qui pourchasse déjà l'avatar passe trop près d'un robot qui ne l'attaque pas.

Fuite: Lorsque l'usine voit sa population chuter, les robots restants essaient d'atteindre des endroits où ils pourraient être en sécurité au lieu d'attaquer le joueur. Cet état dure jusqu'à ce que le nombre de robots dépasse un seuil.

Projectile: Un robot entre dans cet état après un contact suffisamment violent avec l'arme de l'avatar ou avec un autre robot dans l'état projectile. Dans cet état, le robot n'est plus capable de se diriger lui-même et part comme une balle de manière rectiligne. Il rebondit contre les murs de l'usine et peut infliger des dégâts aux autres robots avec lesquels il entre en contact. Si un robot n'a plus de point de vie, il passe également dans cet état avant de disparaître.





Points de vie	Vitesse de déplacement	Dégat	Hitstun (Hitstun max)
250	15	17	1sec (10sec)

Autres Ingrédients :

Murs Destructibles:

Murs qui disparaissent si un ennemi en état de projectile le percute.



Tapis Roulants:

Surface qui emporte les ingrédients qui entrent en contact avec lui. Pendant qu'ils sont déplacés par le tapis, les ingrédients ne peuvent pas se déplacer autrement.

Les tapis ne peuvent être empruntés que dans un sens.

Vitesse de déplacement des objets: 40.



Obstacles:

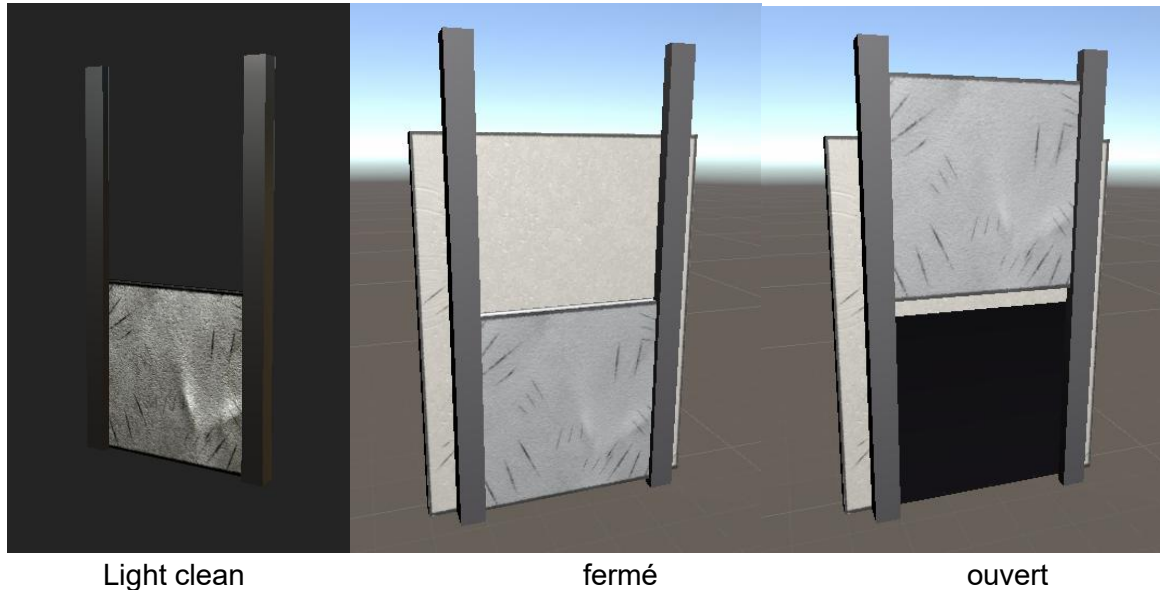
Des éléments qui ne sont pas fixes. Ils peuvent avoir des masses différentes ce qui les rend plus ou moins faciles à déplacer.

Masse allant de: 1-5.



Spawner:

Les Spawners sont situés à des endroits fixes sur la carte, la fréquence à laquelle ils font apparaître des robots varie selon la phase de jeu dans laquelle on se trouve.



La fréquence d'apparition commence à 10 sec pour ne pas submerger le joueur immédiatement sous un flot incessant d'ennemis.

Une fois que le nombre de robot a atteint une fois son maximum puis à chaque fois que la population est supérieure à 80% de la population maximale, la fréquence d'apparition augmente à 30 sec (le nombre de robot est satisfaisant pour faire tourner l'usine correctement)
Si le nombre d'ennemis est inférieur à 60% de la population maximale, l'usine crée des robots en urgence pour combler le manque. La fréquence d'apparition passe à 4 sec jusqu'à ce que la population repasse le seuil des 80%.



Fréquence d'apparition en fonction du nombre de robot dans l'espace de jeu

Direction Artistique:

Intention :

La Direction Artistique de ce projet avait pour objectif de contextualiser une situation cohérente avec le Gameplay développé dans le prototype, ce qui plus précisément signifiait au niveau de nos intentions:

1- Développer un chara-design cohérent avec l'avatar que le joueur contrôle, permettant d'expliquer les comportements induit par la physique du projet et la mécanique principal du jeu, à savoir le poing qui décolle de l'avatar (le choix d'un poing comme projectile est une de nos

étapes du chara-design

Inspiration pour l'avatar (et ennemi)

BB-8 de star wars



inspiration pour le mode de déplacement

WALL-E de WALL-E



inspiration pour l'aspect

Bras articulé (jouet)



inspiration pour les bras articulés de l'ennemi

premières ébauches au dessin

Diverses ébauche pour l'avatar

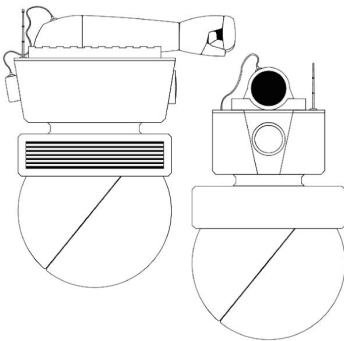


ébauche d'un bras

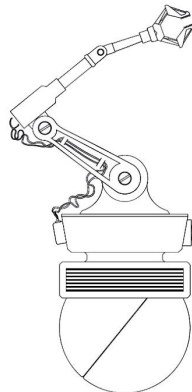


Concept Art pour servir de modèle

Turn around de l'avatar



référence pour l'ennemi



Rendu final

L'avatar en 3D

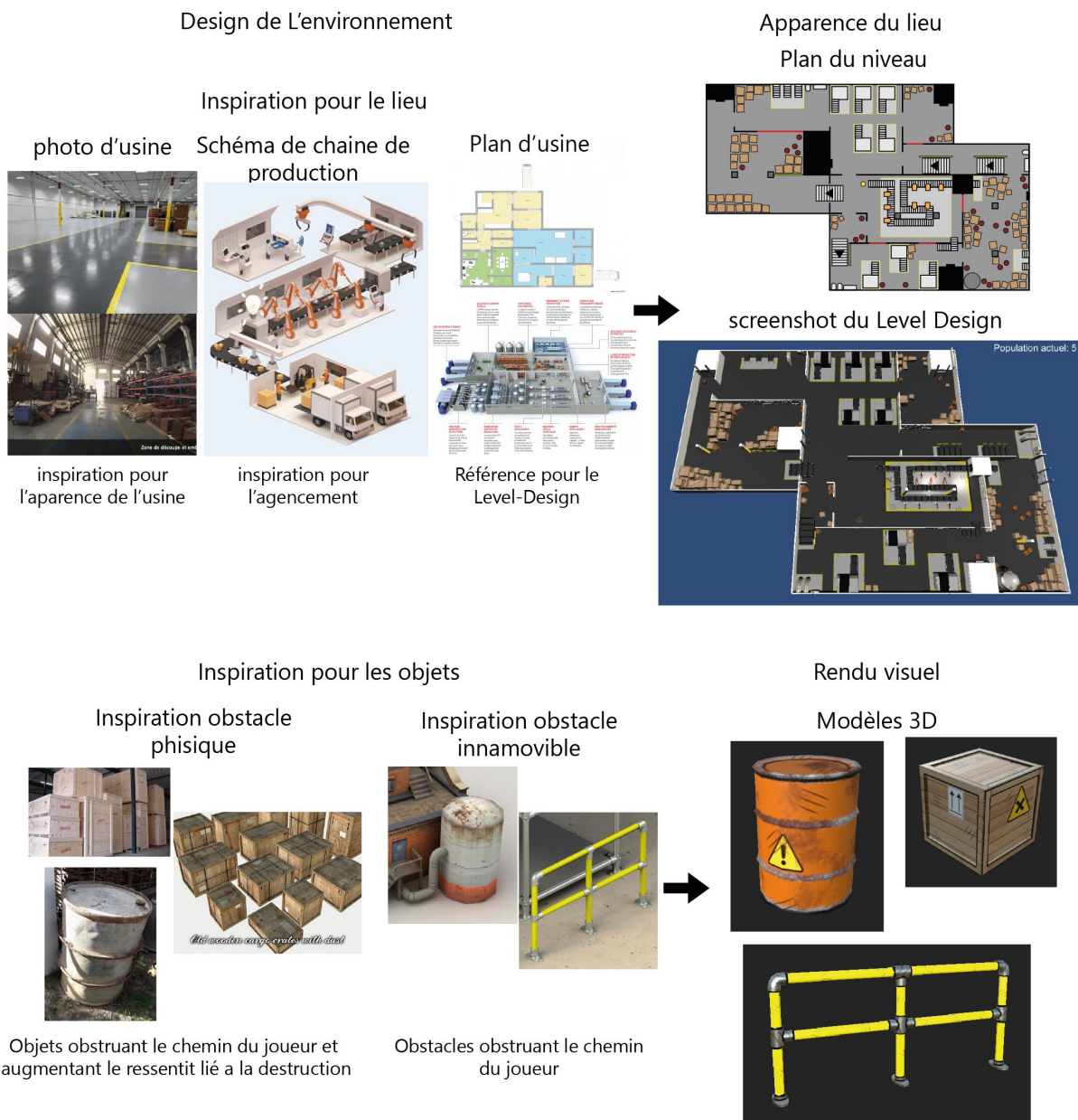


Ennemi en 3D



intentions fortes, notamment pour jouer sur l'iconographie lié à la violence et à la destruction); Le choix du robot comme représentation pour l'avatar nous a permis de rendre cohérent le déplacement obtenu dans le proto (brusque changement de direction sans réorientation du corps), le fait que le haut du corps soit indépendant du bas (notre prototype a les contrôles d'un

twin stick shooter, une main dirige les déplacements de l'avatar et l'autre la rotation du haut du corps), la physique des ennemis, et la mécanique du poing détachable.



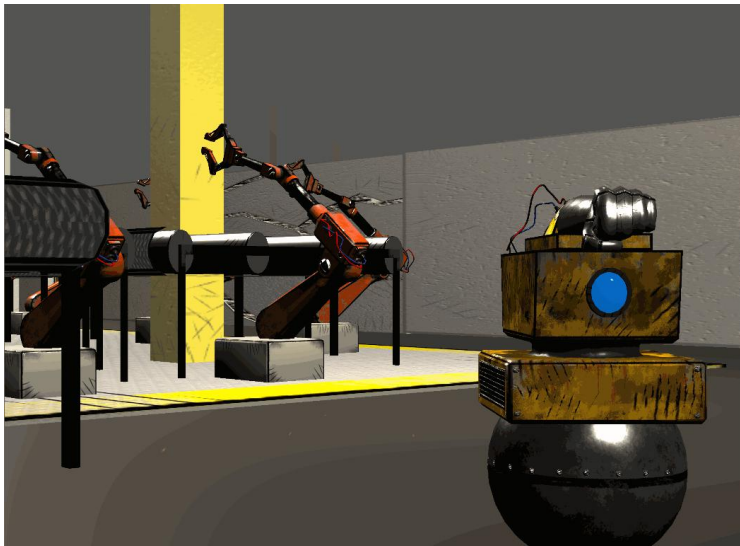
2- Trouver un lieu permettant de créer une arène dans laquelle le joueur peut exploiter les décors, et dont l'agencement met en avant le comportement des ennemis (suite de grandes pièces plutôt qu'un seul grand espace avec peu d'obstacle ou d'un labyrinthe), avec un grand nombre d'objet présent dans le décors et la possibilité de créer facilement de nouveaux ingrédients de Level Design;

L'usine a été choisie car cohérente avec le pitch et la présence de nombreux robots dans son espace, collait avec les intentions d'architecture de l'arène, permettait la présence de nombreux objets permettant d'augmenter l'impact des projections provoqué par le poing,

Et permet d'être assez souple dans l'optique de développer divers ingrédients (comme les tapis roulant) qui interagissent avec le gameplay.

3- Développer des sign et feedback cohérent avec le reste de la DA, c'est a dire trouver des idées de réaction visuel ou sonore permettant d'indiquer au joueur des information crucial lié à l'état du système tout en restant cohérent avec le reste de la DA établie;

Genre de feedback réalisé:



Lorsque le joueur est blessé

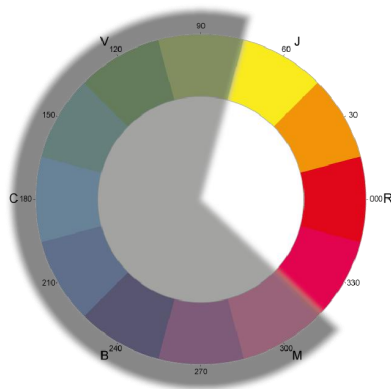


Quand le joueur tire le poing

Pour le choix des signs et feedbacks visuel, on s'est basé sur un effet de projection d'étincelle lorsque le joueur ou l'ennemi est blessé, une autre différente lorsque celui ci tire le poing, et un flash lumineux au niveau du sol lui indique le meilleur timing pour relâché le chargement du poing.

Pour plus de détails sur les signs et feedbacks sonore, il faut se référer à la partie dédiée à l'audio et au SDD dans le dossier de DA.

4- Choisir un style cohérent avec le pitch et le genre d'objet modélisé;



Pour le style général, on a choisis un style brut pour les modèle avec des texture un peu sale sur les objet, afin de rester cohérent avec le pitch et le reste de la DA (une usine autonome qui fonctionne grâce à des robots qu'elle construit elle même) pour plus de détails, voir la partie Texturing du dossier de DA.

Pour la colorimétrie, on a choisis de basé notre choix de couleur sur les teintes chaudes du spectre lumineux, tant pour l'ambiance voulu que par rapport a la colorimétrie affordante des divers objets présents dans une usine.

Texturing:

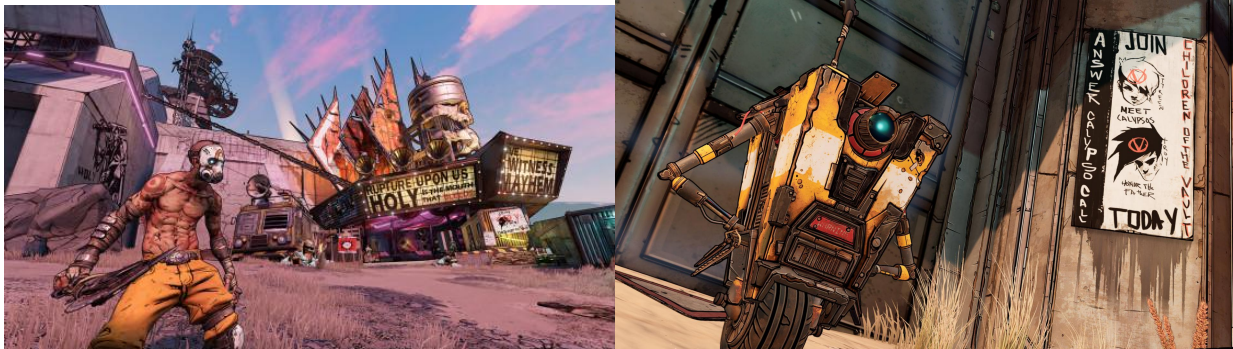
Intention:

à partir du thème de l'usine et des robots, créer une ambiance propre au jeu, comme un endroit à l'origine aseptisé, puis endommagé.

Références:

- **Borderland** :

Jeu très connu, borderland a une esthétique unique, à base de cell-shading et de textures aux aspérités renforcées par une ligne noire : C'est ce dernier point qui a influencé le texturing, avec l'exagération aux trait noir des angles et des différentes parties abimées des modèles. La colorimétrie du personnage est également influencée par le personnage de Claptrap (ci-dessous à droite).



- **Crossroad park** (projet en beta) :

Ce projet encore en cours (et donc non commercialisé) a principalement eu une influence pour l'outline marquée et les modèles comprenant au final peu de



polygones.



- Wall-E :

Ce film d'animation dépeint une terre ravagée, complètement recouverte de déchets. C'est cette ambiance de rouille et de saleté qui a été une influence.



Audio:

Pour le sound design, l'emphasis a été mise sur les différentes interactions entre les différents éléments, notamment les chocs.

Intention:

L'intention de base était de mélanger une ambiance sonore crissante et rouillée d'usine mal entretenue, avec des sonorités "robotiques", métalliques et rappelant les bruits de tôles.

Référence:

Halo reach : lors de la recherche de références initiales, certaines machines (notamment des véhicules) pouvaient produire certaines sonorités "d'alarmes". Les Bips des ennemis en sont inspirés, que ce soit pour le combat ou pour la fuite.

Daxter (version psp) : le niveau de l'usine à été une inspiration principale dans la création des bruitages des caisses et des barils.

Crosscode : les sonorités robotiques des ennemis mécaniques et des chocs lors des combats avec ceux-ci ont grandement influencé les chocs, à la fois dans leur manière de résonner et dans leur pitch légèrement grave.

SDD:

Feedbacks sonores :

→ robots ennemis :

1- robots ennemis qui s'entrechoquent entre eux (déplacement classique) (son 2d) **|devenu “s’entrechoquent entre eux ou avec l’avatar”|**

2- robots ennemis qui s'entrechoquent entre eux (propulsé par le poing) (son 2d) **|annulé|**

3- robots ennemis qui heurte un mur (son 3d)

4- robots ennemis qui attaquent pince (son 3d)

5- robots ennemis qui attaquent scie (son 3d)

6- robots ennemis qui s’énervent entrée en situation combat – bips, blips, etc (son 3d)

7- robots ennemis qui paniquent entrée en situation critique (son 3d)

→ avatar :

1- robots ennemis qui s'entrechoque avec l'avatar (son 2d) **|annulé, identique à celui avec les ennemis|**

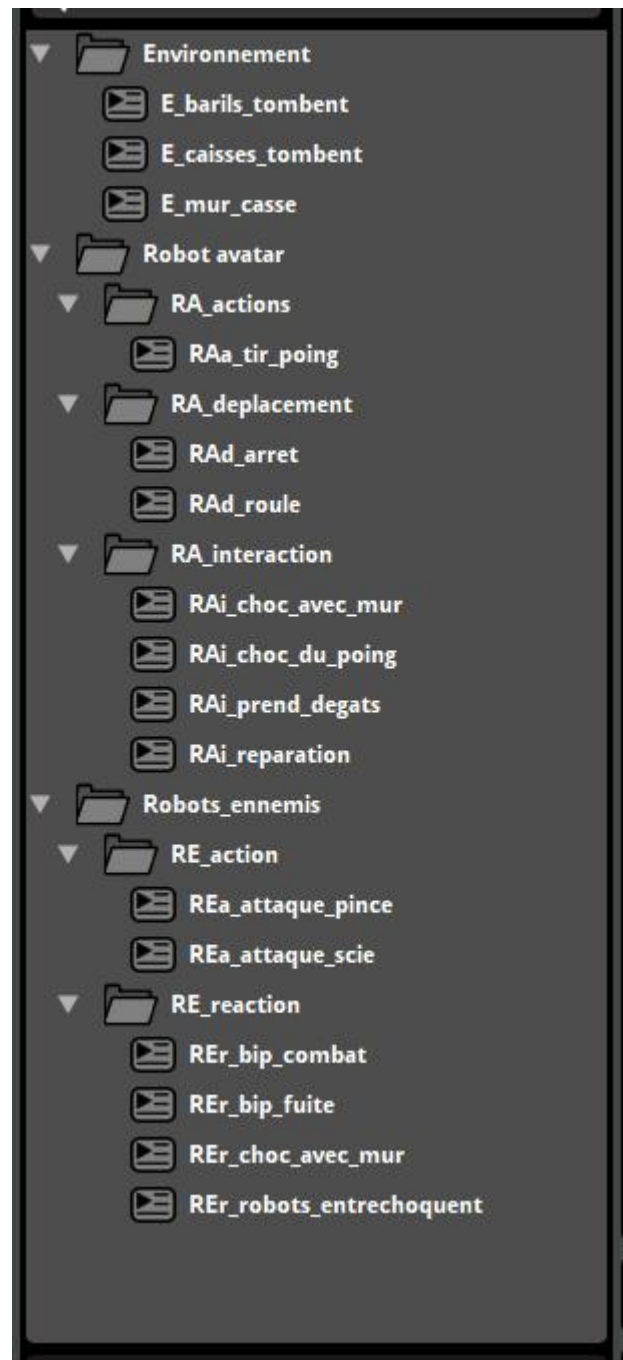
2- avatar qui heurte un mur (son 3d)

3- tir du poing (son 3d)

4- poing qui heurte un mur
|et|

5- poing qui heurte un ennemi (son 3d)
|4 et 5 : les deux sont identiques|

6- l’avatar se répare (plus il crée de chaos, plus il se répare) (son 3d) **|mécanique**



annulée|

7- l'avatar prend des dégâts (son 3d)

→ déplacement avatar :

1- bruit de moteur calme : avatar à l'arrêt (son 2d, boucle)

2- bruits de moteur en marche + boule de métal qui roule (son 2d, boucle)

→ environnement :

1- bruit de tapis roulant (son 3d, boucle) **|annulé|**

2- fracas caisses (sons 3d, multi-instrument)

3- fracas barils (sons 3d, multi-instrument)

4- mur qui casse (son 3d)

→ annonces systémiques : **|tous annulés, certains n'étaient pas utiles, d'autres n'étaient pas à leur place|**

1- feedback perte de vie

2- feedback regain de vie

3- feedback vie basse

4- situation critique atteinte

Musiques : |annulées|

1- musique situation calme

2- musique situation combat

3- musique situation critique (fuite des ennemis)

4- jingle de mort **|fait mais pas intégré|**

Evolution Possible du Projet:

Liste de potentielles améliorations :

- feedbacks de vie faible : glitches sur l'écran
- amélioration du navmesh des ennemis
- ennemis différents qui ont des interactions spéciales (rebond, knockback, etc...)
- fenêtres destructibles (envoyer un ennemi à l'extérieur = ennemi éliminé)
- LD plus grand et aboutis (et plus de murs destructible)
- plus d'objets et de décors + décors mouvants (pièces détachées sur les tapis roulants, etc...)
- plus d'objets qui sont interactifs, plus d'effets de combos entre les différents éléments ennemis ou objets (rebonds, dégâts supplémentaires, etc)
- feedback de quantité de dégâts infligés aux ennemis
- plus de sons intégrés
- un système de lights intégré à la DA (Lampes/spots/sources lumineuses présentes en usine)
- le sol, que nous aurions dû faire plus tôt
- des feedback visuels supplémentaires (ex: effet de ralenti lors de la charge parfaite, murs cassants qui ont été prévus pour que les morceaux s'éparpillent, etc...)