



Game Design Document

Sebastien BOULET
Yumi COUFFIN
Kyle PRUDHOMME
Thomas VO

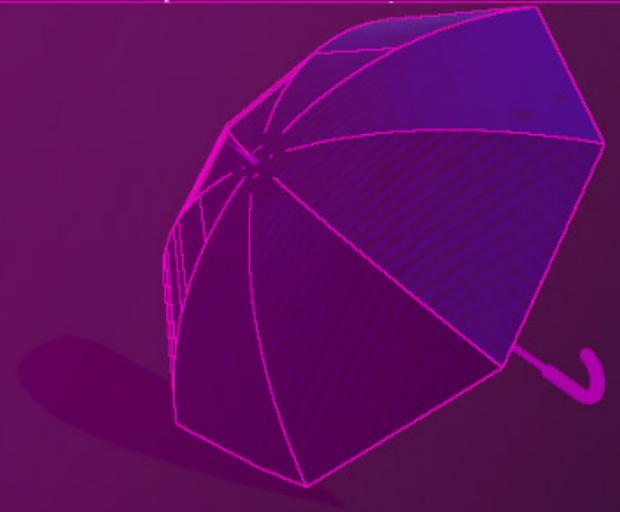


Table des matières

Game Overview Document :

- p5 - Equipe
- p6 - Fiche d'identité
- p9 - 3Cs
- p11 - Mécanique Principale Univers
 - p14 - Ville de Londres
 - p15 - Landmarks
 - p17 - Ingrédients

Level Design :

- p57 - Intentions
- p58 - Ingrédients et blocs LD
- p60 - BlenderOSM
- p64 - Landmarks Naturels
- p66 - Landmark Naturels
- p69 - 30Minutes de Gameplay
- p72 - Progression et Apprentissage

Game Design :

- p20 - Références
- p22 - Tension et Tendence
- p23 - Eventail de possibilités
- p24 - Boucles de Gameplay
- p26 - Boucles de Prédiction
- p28 - 3C
- p37 - Mécanique : Le Flap
- p40 - RGD
- p43 - Diagramme de Ventrice
- p44 - MDA framework
- p48 - Capsule de Gameplay
- p51- Boucles OCR

Programmation

- p76 - Mouvement Parapluie
- p78 - Camera Collision Wall
- p80 - Bonus de flap
- p81 - Vent
- p84 - Tornade
- p85 - Trigger Musique
- p86 - Etoiles

Table des matières

Direction Artistique Visuelle :

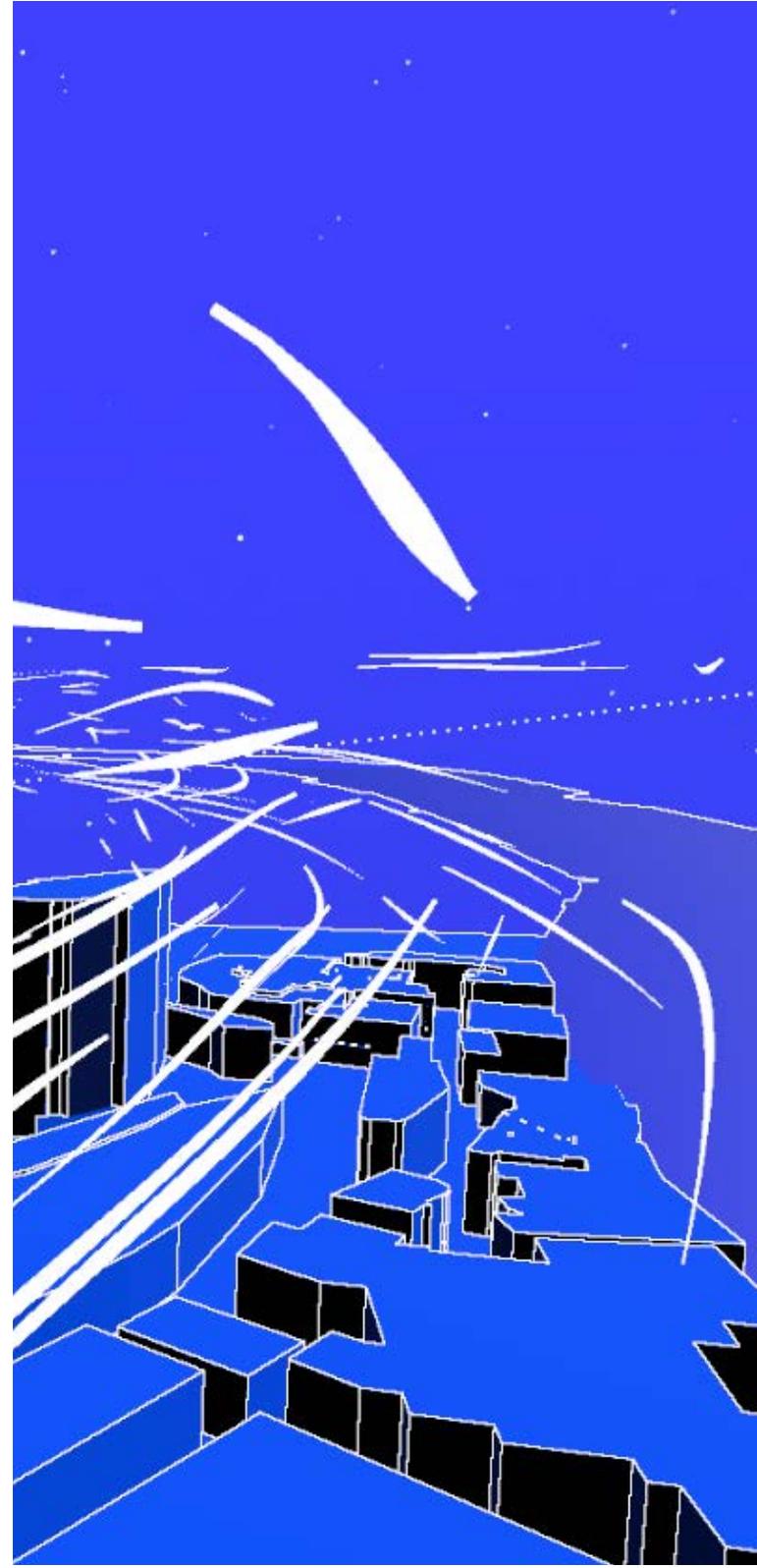
- p88 - Intentions
- p89 - Visuels
- p91 - Signes & Feedbacks
- p92 - Techniques Visuels
- p93 - Références

UI et UX :

- p117 - Intentions
- p118 - Références
- p120 - UI Wireframe
- p121 - UI Feedback et HUD
- p122 - Moodboard
- p123 - Main Menu et éléments HUD
- p124 - Pause Menu et bouton

Direction Artistique Sonore :

- p97 - Intentions
- p98 - Références
- p100 - Overview des sons
- p102 - Production des sons
- p105 - Tracking et Avancement
- p107 - EventList
- p110 - Intégration
- p114 - Répartition Musique





PARAPLUIE

Game Overview Document :

p5 - Equipe

p6 - Fiche d'identité

p9 - 3Cs

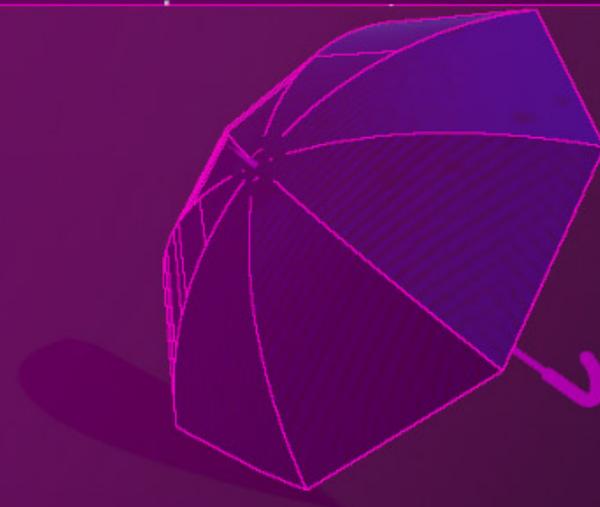
p11 - Mécanique Principale

Univers

p14 - Ville de Londres

p15 - Landmarks

p17 - Ingrédients



Equipe

**Sebastien
BOULET**
Sound Designer

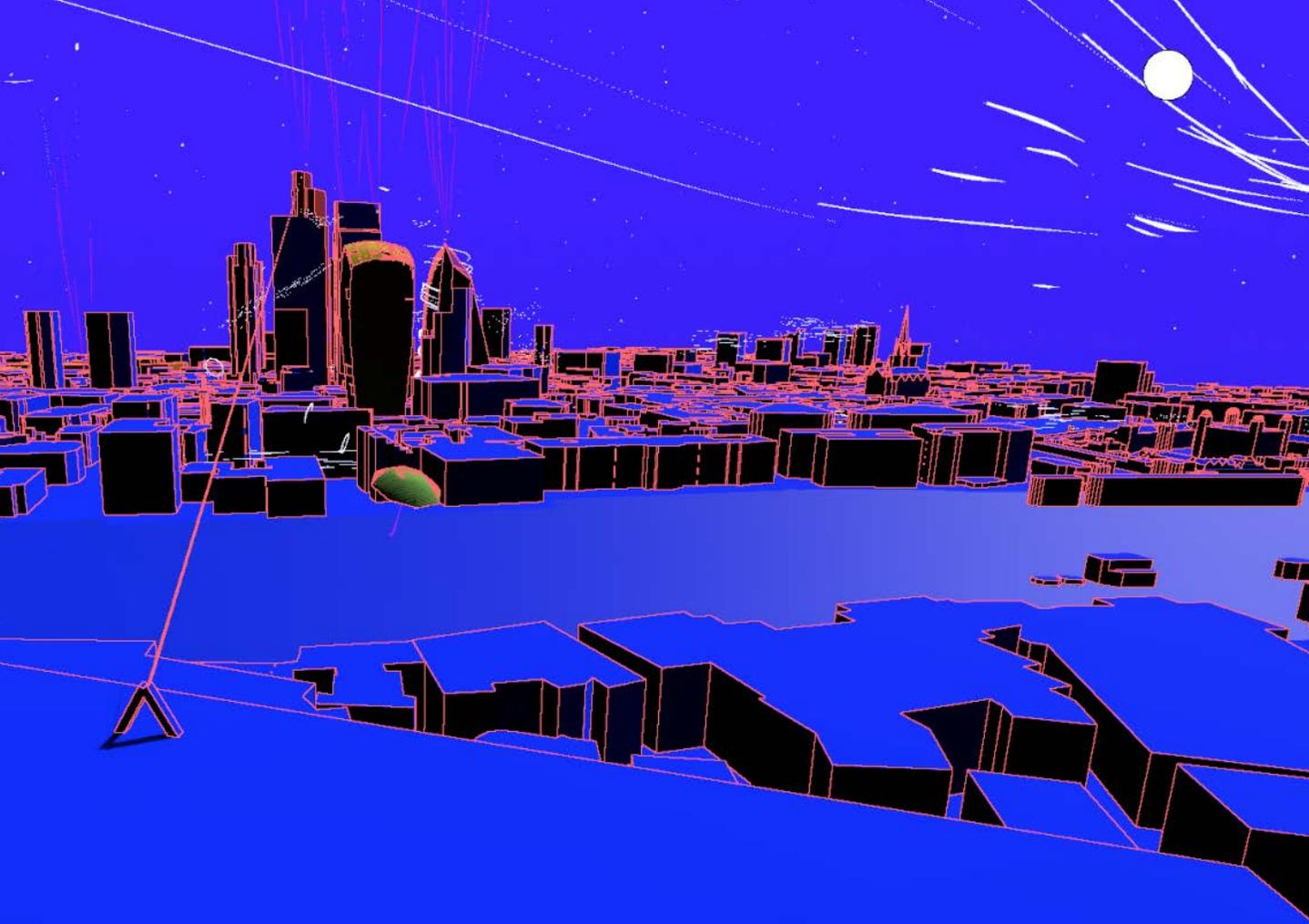
**David COMPAS-
-COLLIN**
Programmer

Yumi COUFFIN
*Lead Programmer
Level Designer
UI/UX Designer
3D Artist*

**Kyle
PRUDHOMME**
Programmer

Thomas VO
*Project Lead
Lead Game Designer
Lead Level Designer
UI/UX Designer*





Fiche d'Identité

Fantasme :

- Incarner un parapluie et voler au gré du vent.
- Un contrôleur atypique et exigeant qui récompense une maîtrise du jeu.
- Explorer la ville de Londres comme jamais vue auparavant

Support :

PC - Unity Engine

Cible :

- Mid et hardcore gamer, joueurs en recherche de challenge et de nouvelles manières de jouer à un jeu vidéo

Pitch :

La ville de Londres, magnifique et synonyme d'élégance et de classe. Et si il existait une autre manière d'explorer cette ville sublime ? Incarnez un parapluie et découvrez ou redécouvrez Londres comme vous ne l'avez jamais vu. Envolez vous et profitez du vent pour vous déplacer et visiter le quartier des affaires de Londres fidèlement reproduit. N'hésitez pas à passer par chacun de ses bâtiments iconiques afin de compléter votre visite !



Fiche d'Identité

Concept :

Parapluie est un contrôler à la 3ème personne qui reprend les caractéristiques du parapluie (planer, se fermer, s'ouvrir). Le joueur a pour objectif de maîtriser les contrôles atypiques de son personnage et de comprendre comment ce dernier interagit avec son environnement afin de coller au mieux au fantasme qu'il se fait d'un parapluie pouvant voler.

Intentions :

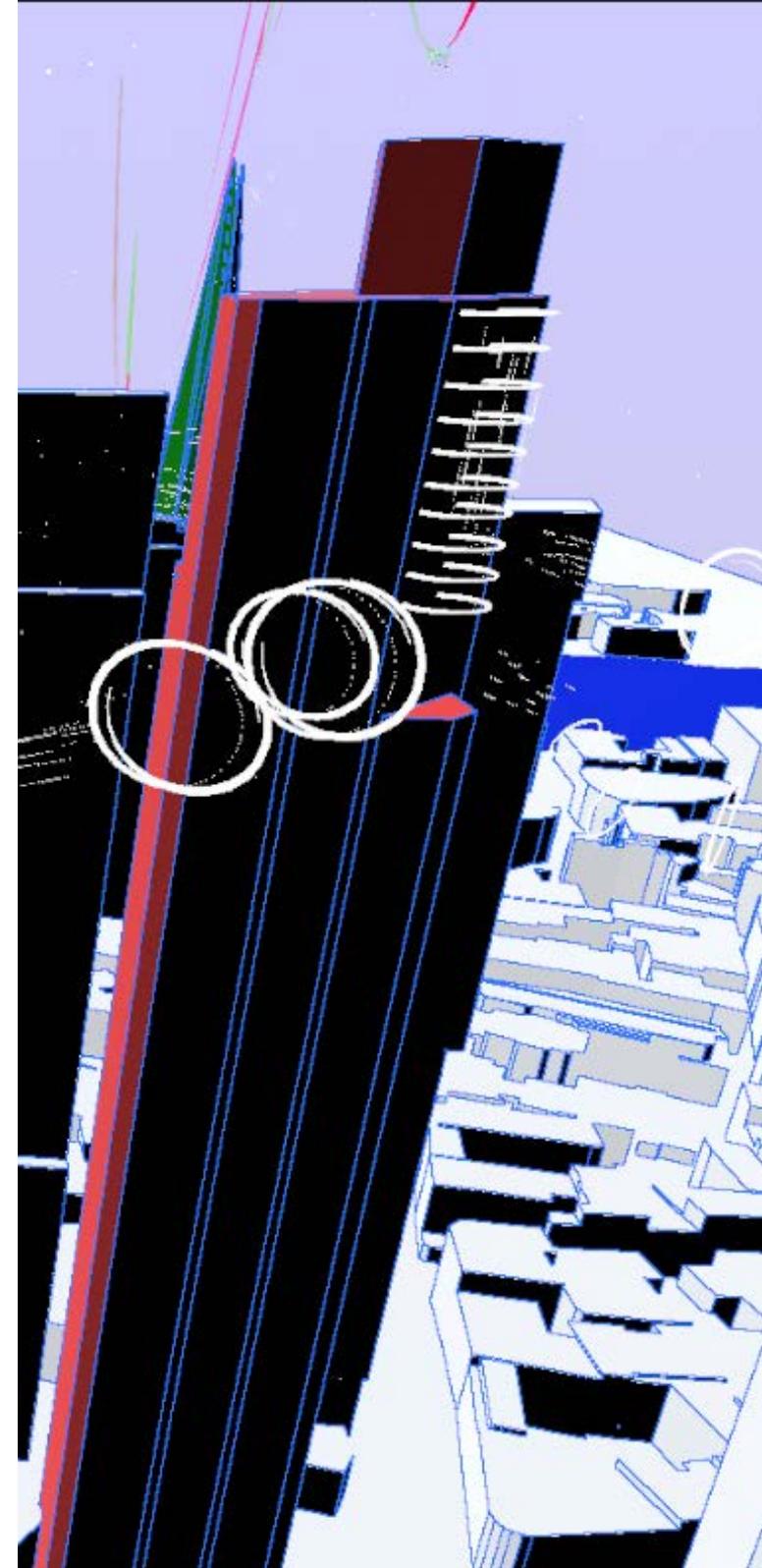
Un contrôler atypique qui valorise la maîtrise du joueur (courbe de progression basée sur la maîtrise du contrôler)

Liberté et légèreté, un contrôler qui, une fois maîtrisé, devient un outils d'exploration et de sensation puissant.

Contemplation : Utiliser le contrôler maîtrisé pour explorer et découvrir, en tirer de la satisfaction et émerveiller le joueur

Mécanique Principale :

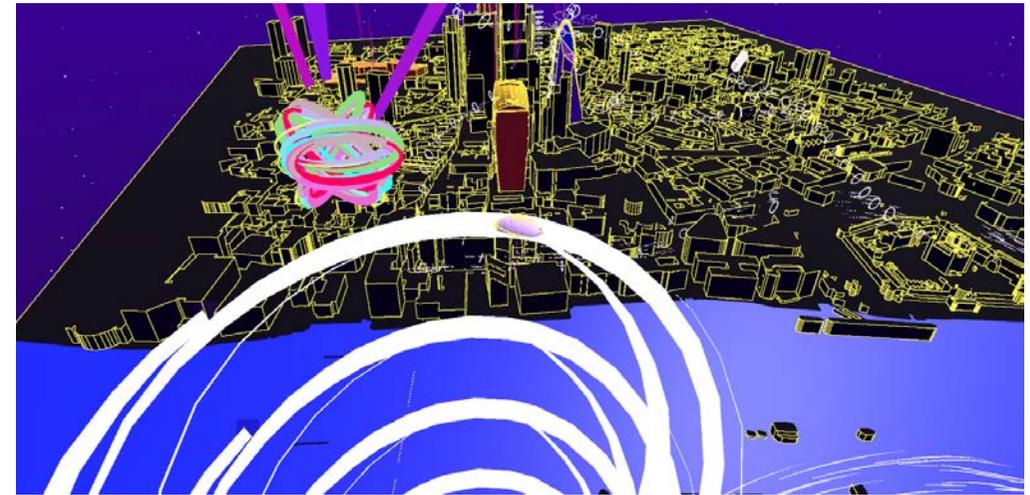
La mécanique principale de Parapluie se trouve dans son contrôler. Ce contrôler atypique reprend les caractéristiques du parapluie : s'ouvrir, se fermer. En position ouverte, le parapluie augmente sa résistance au vent et peut planer sur des distances plus ou moins longues. Le joueur peut orienter le parapluie afin de se diriger dans les airs. En position fermée, le parapluie réduit sa résistance à l'air et chute plus rapidement. Il possède également une hit box plus petite. En se refermant et en se réouvrant, le parapluie peut effectuer un mouvement de «flap flap» pour gagner en altitude ou alors se propulser dans une direction choisie. Effectuer un «flap flap» consomme de l'endurance.



Fiche d'Identité

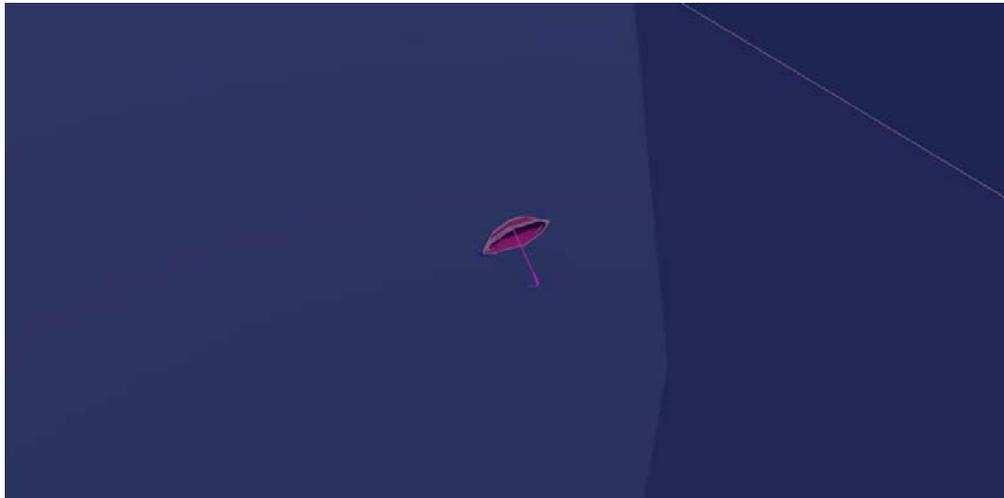
Condition de victoire :

Atteindre le sommet de The Shard, la tour la plus haute du quartier, permet au joueur d'atteindre la fin du niveau. Cependant, pour compléter le niveau à 100 %, le joueur doit visiter chaque bâtiment iconique de Londres pour y récupérer 8 étoiles, preuve de son passage.



Condition de défaite :

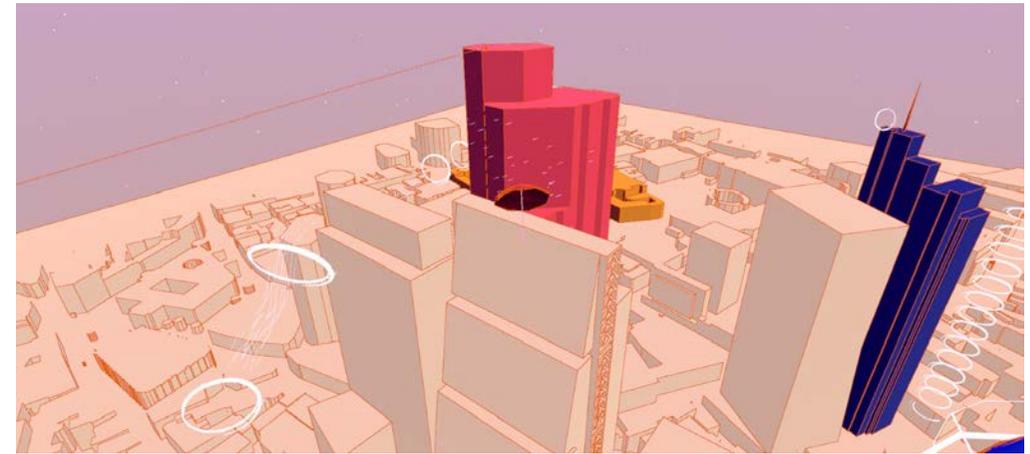
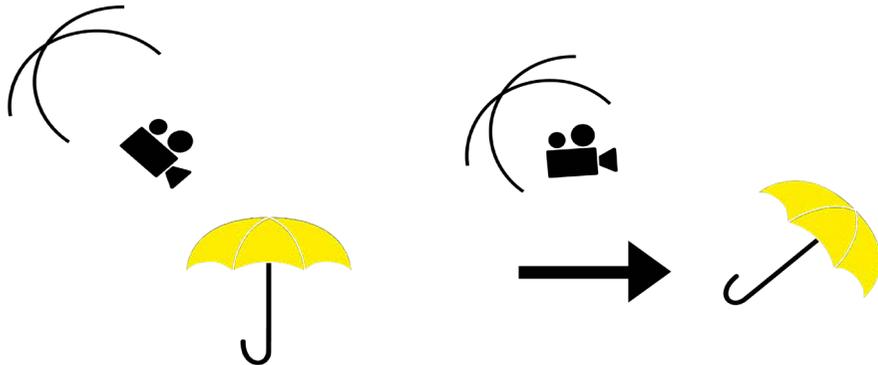
Il n'y a pas de réelle condition de défaite dans Parapluie car quoi qu'il arrive, le joueur va récupérer des flaps. Cependant, il est possible de considérer l'échec du joueur sur un atelier, donc dans ce cas là toucher le sol ou bien rater un bonus serait une situation de défaite. Un des ateliers propose cependant une réelle condition de défaite : tomber dans le trou de cet atelier ne permet pas d'en sortir malgré les 5 flaps.



3Cs

Caméra

La caméra de Parapluie est une caméra à la troisième personne qui suit le joueur. Elle est orientable sur les axes X et Y par le joueur mais peut également s'orienter de manière automatique lorsque le parapluie se déplace. Cette caméra est comme tirée par le parapluie, cela facilite la navigation et le déplacement. Le joueur peut cependant récupérer le contrôle total de la caméra s'il le désire. La caméra permet au joueur de se concentrer sur le parapluie ou bien de choisir d'observer la ville.



Character

Dans Parapluie, le joueur incarne un avatar, un parapluie volant. Cet avatar peut s'ouvrir et se fermer ainsi que se déplacer dans les airs dans un mouvement de battement appelé le flap.

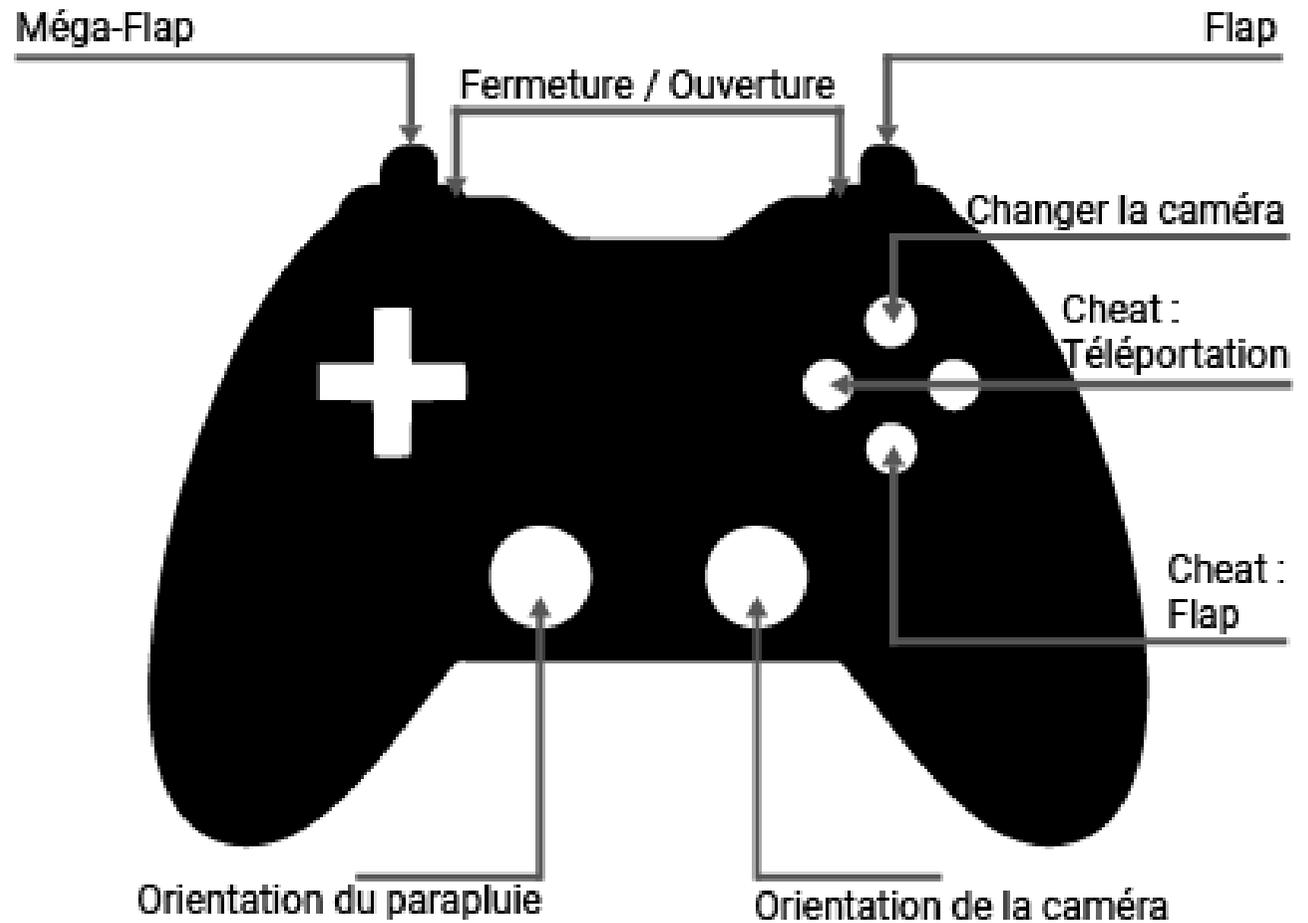
Il peut être influencé par la gravité, lorsqu'il est ouvert il plane et chute lentement alors que si il est fermé il tombe à pic. L'angle d'inclinaison lors du vol affectera également la vitesse de la chute.

Le parapluie est soumis aux changements climatiques, le vent peut le porter et le vent très violent le détruire.

3Cs

Controls :

Parapluie est jouable avec une manette. Orienter le stick gauche permet d'orienter le parapluie tandis qu'appuyer sur A permet de flap. Le parapluie peut s'ouvrir et se fermer si le joueur appuie sur le bouton B



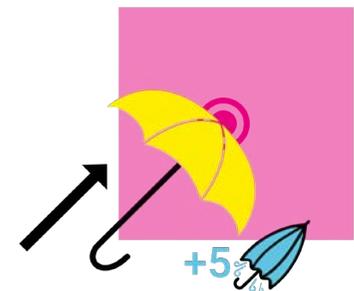
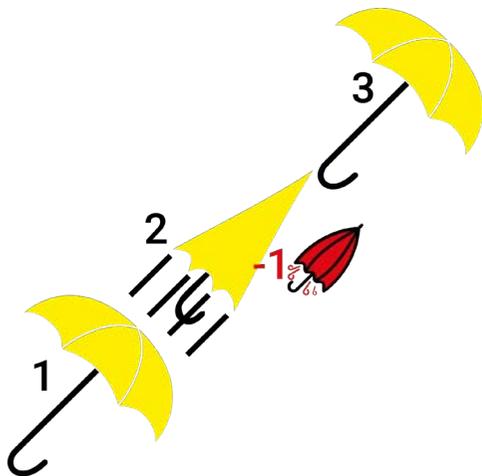
Le Flap

Comment fonctionne le Flap ?

- En appuyant sur un bouton, le parapluie se ferme et s'ouvre de nouveau rapidement et le propulse dans la direction vers laquelle il regarde.
- Pendant l'animation du Flap, le joueur ne peut pas se fermer (freeze input de fermeture)
- Le nombre de flaps est dépendant du niveau.
- Chaque téléportation au début d'un module reset le nombre de flaps du joueur.

Comment récupérer des Flaps ?

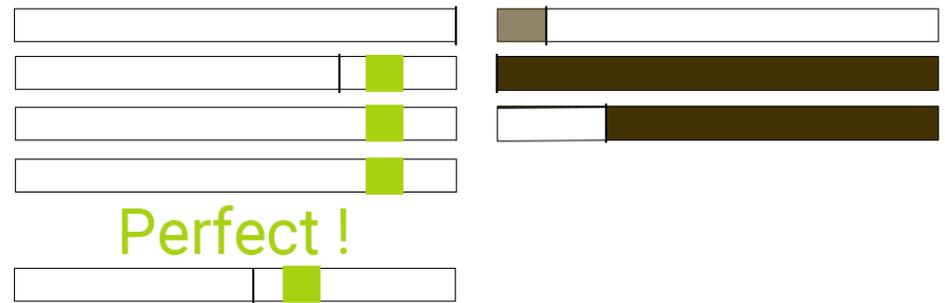
- Deux manières pour récupérer des flaps flaps :
 - Récupérer un bonus de flap (passer à travers remplie instantanément la jauge de flap et fait disparaître temporairement le bonus)
 - Attendre que la jauge de flaps remonte avec le temps. Cela implique de planifier à l'avance sa position.



Signalétique

La jauge de Flap

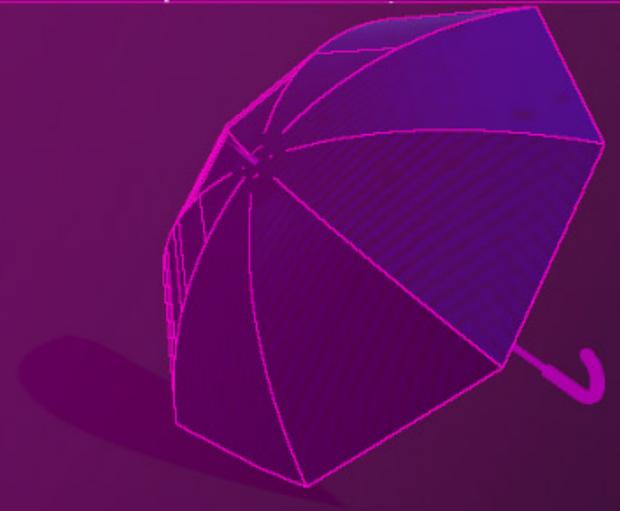
Pour permettre au joueur de suivre sa consommation d'énergie et gérer son flap, l'UI intègre une auge de flap en bas de l'écran. Lorsque le joueur flap, l'énergie consommée fait diminuer la barre tandis que la zone jaune, matérialisant le timing de perfect flap apparait.



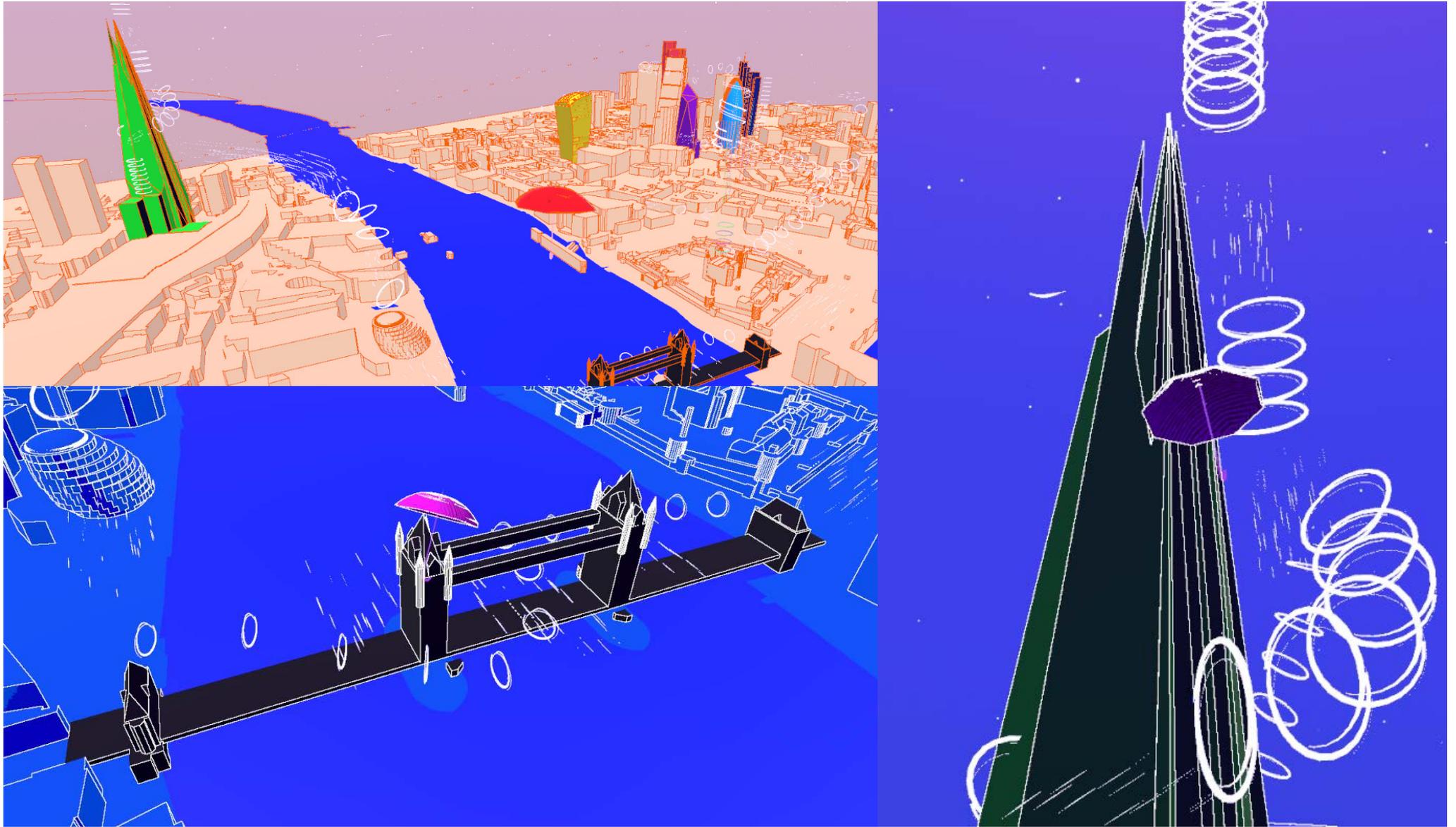


Univers

- La ville de Londres
- Landmarks
- Ingrédients



Ville de Londres



Reproduction à l'échelle du quartier des affaires de Londres. Le joueur peut s'y déplacer librement au gré du vent.

Landmarks

Landmarks

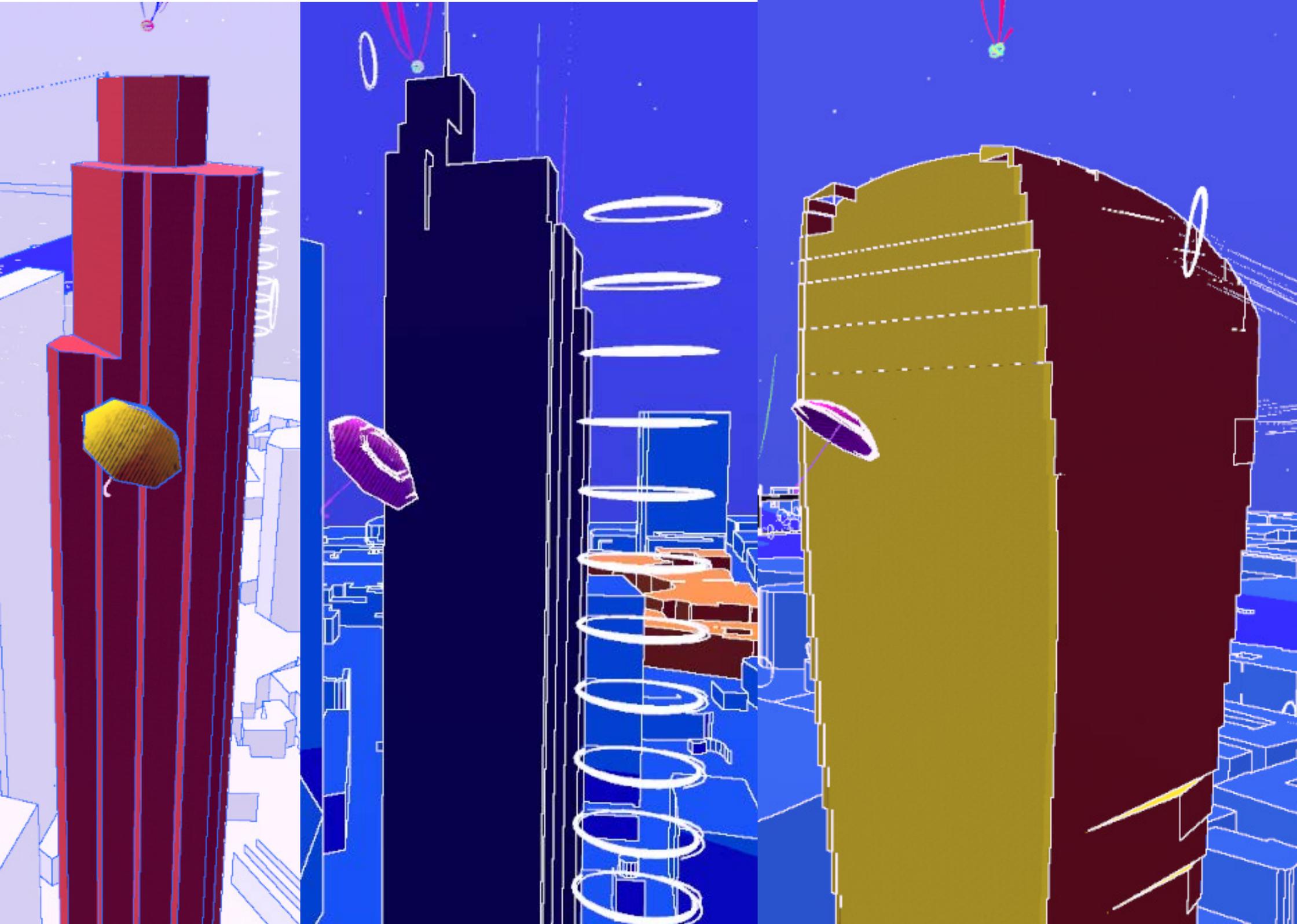
Le niveau est parsemé de Landmarks sur lesquels sont placés des étoiles. Pour compléter le niveau, le joueur doit obtenir toutes les étoiles du niveau, donc visiter tous les Landmarks. Il peut suivre sa progression via une barre colorée. Les Landmarks visités servent aussi de point de téléportation pour les voyages rapides dans la carte.



Le Scalpel



Carte + Frise

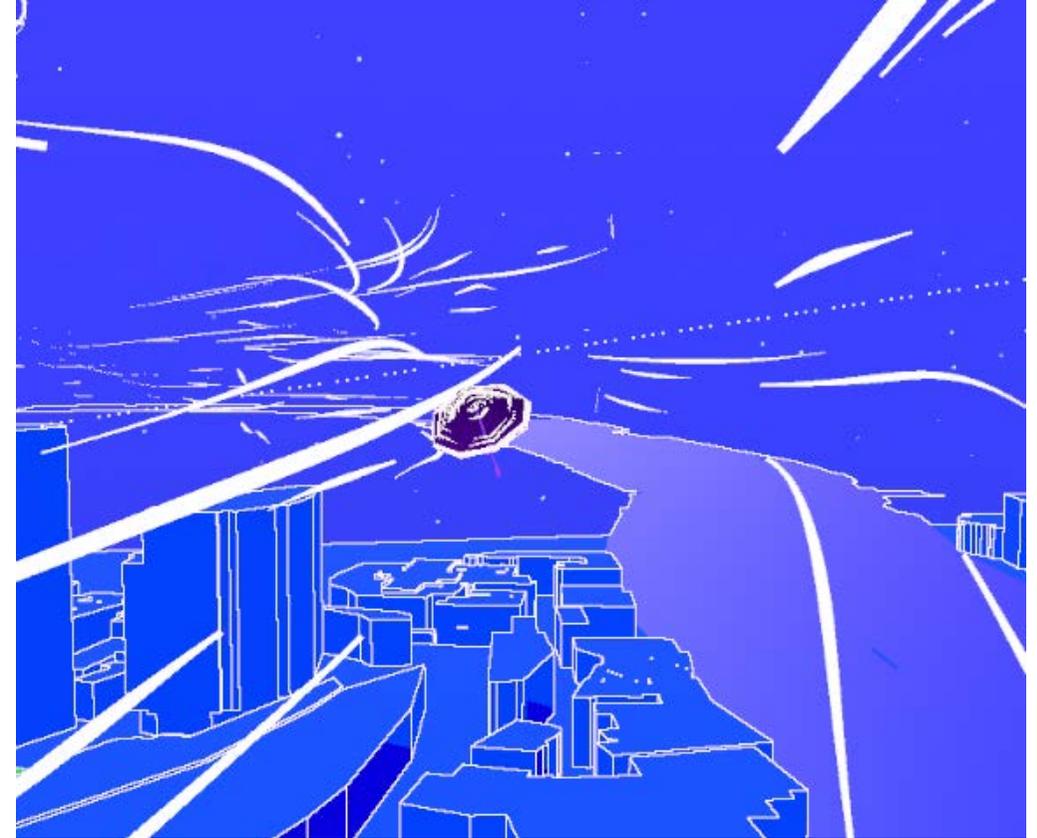


Ingrédients



Bonus de Flap

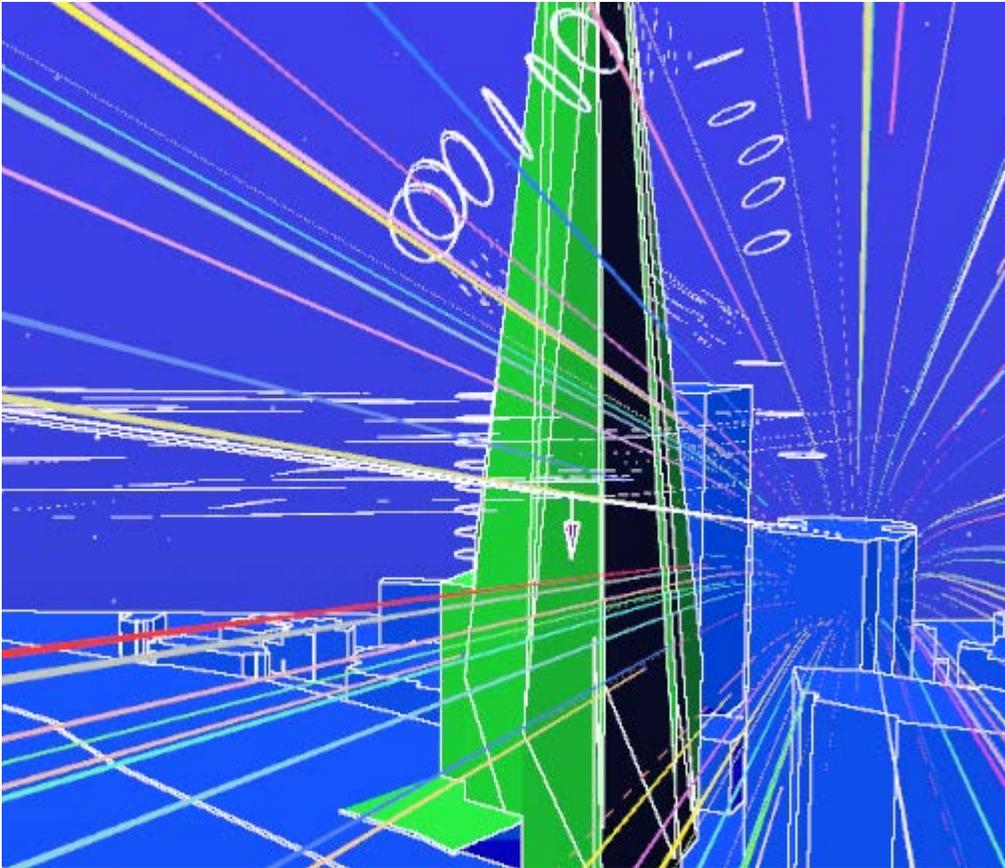
Passer au travers permet de remplir instantanément la jauge de flaps. Agencer des bonus permet de créer des niveaux aériens



Zone de vent

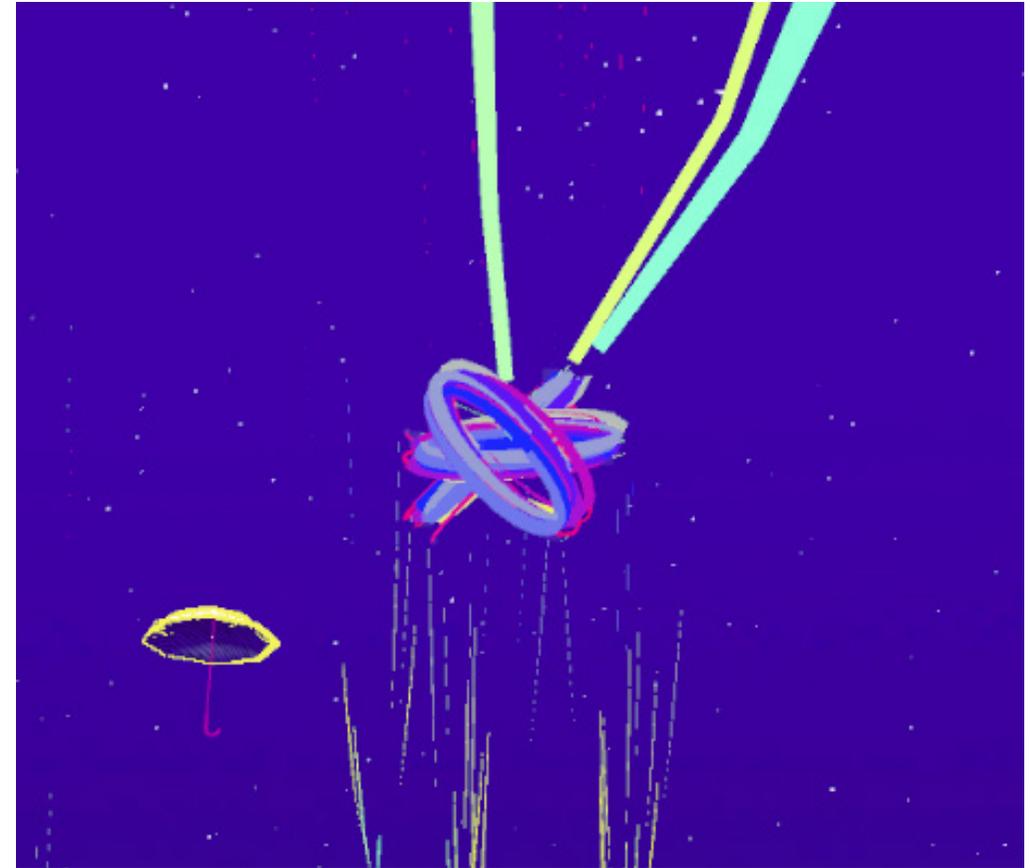
Le parapluie est influencé par le vent. Un vent faible de face ralentira le parapluie tandis qu'un vent arrière puissant le propulsera. Le vent permet de créer des circulations aériennes axées sur la stabilisation.

Ingrédients



Tyrolienne

La tyrolienne permet au joueur de souffler et de profiter de l'environnement tandis qu'il passe d'une zone à une autre. La vitesse et la glisse sont des récompenses que le joueur reçoit après avoir gravi un bâtiment.



Etoiles

Les étoiles sont les collectibles associés aux Landmarks et son partie intégrante de la condition de victoire. Le joueur doit parcourir le niveau et les collecter.



PARAPLUIE

Game Design :

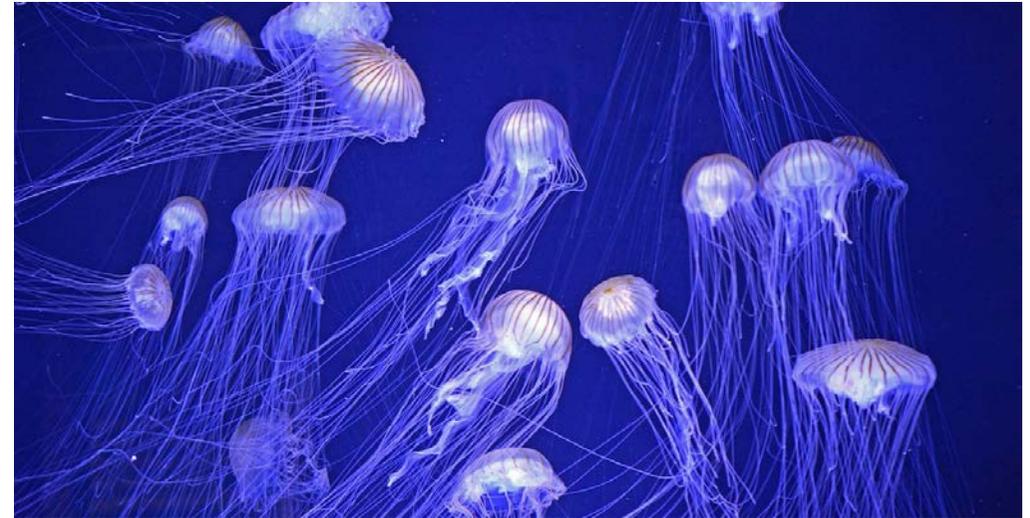
- p20 - Références
- p22 - Tension et Tendence
- p23 - Eventail de possibilités
- p24 - Boucles de Gameplay
- p26 - Boucles de Prédiction
- p28 - 3C
- p37 - Mécanique : Le Flap
- p40 - RGD
- p43 - Diagramme de Ventrice
- p44 - MDA framework
- p48 - Capsule de Gameplay
- p51- Boucles OCR



Références

Controller - Archetype de la méduse

Self Explanatory, la propulsion par la contraction du player. Déplacement par orientation dans une direction suivit d'une contraction qui crée une impulsion. L'archétype de Méduse est un controller intéressant peut référencer dans les jeux vidéos. Il est clair et compréhensible de tous et constitue une référence solide pour Parapluie.



Smash Bros : Kirby

Dans Smash Bros, Kirby est un personnage unique. En effet, contrairement à la majorité des autres personnages qui n'ont que deux grands sauts, Kirby en possède 6 petits et est capable de planer entre chaque saut. Si dans Smash Bros ces caractéristiques sont utilisées dans le contexte d'un jeu de combat, ce type de déplacement est aussi parfaitement adapté à un plateforme 3D dans lequel la gestion du tempo est primordiale afin d'optimiser sa navigation et sa planification tout en offrant une limite dans les déplacements via le nombre maximum de sauts.

Références

I Am Bread

I Am Bread est un exemple type du controller atypique dont le fun et la progression se mesurent via la maîtrise du controller et de toutes les possibilités qu'il offre. Le controller de I Am Bread utilise les gachettes pour se déplacer en se fixant sur certains coins de la tartine pour se propulser ou grimper mais permet aussi de manipuler des objets pour tirer avantage de leurs propriétés. Le fun de Parapluie doit lui aussi venir de la maîtrise du controller et du fait de voir l'éventail de possibilités du joueur lié au controller s'agrandir à mesure qu'il comprend ce controller.



I Am Fish

I Am Fish reprend ce concept de controller atypique et le pousse plus loin en y incorporant la notion de mesure et de dosage. Il demande au joueur de comprendre les propriétés du bocal dans un premier temps (rouler, comment le déplacement du poisson influence le déplacement du bocal) pour ensuite introduire d'autres règles venant appliquer des limites à ces propriétés et demandant donc au joueur une maîtrise de ses limites (une chute d'une trop grande hauteur ou à trop grande vitesse endommage le bocal etc).

Tension et Tendence

Tendance joueur :

Arriver à diriger son character (difficulté du déplacement) --> Maitrise du controller (contrôles atypiques, apprentissage --> maitrise)

Tendance système :

déstabiliser le joueur, le faire chuter

Tension gameplay :

Stabilisation. Le joueur se dirige vers là où il veut aller malgré le système qui veut le déstabiliser et gêner son déplacement. Le joueur doit donc comprendre et maitriser son controller afin de pouvoir se déplacer au mieux et à sa guise.

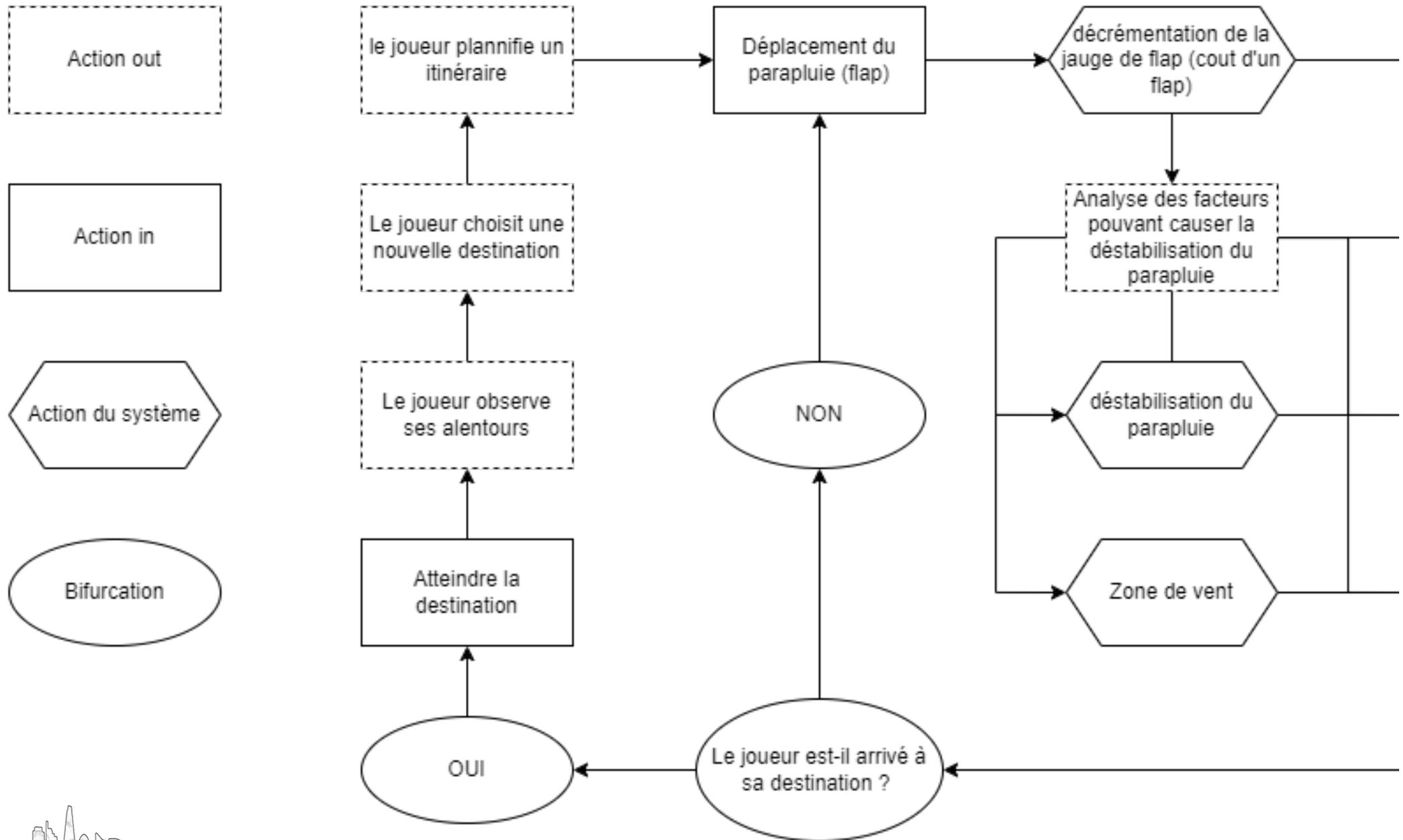


Eventail de possibilités

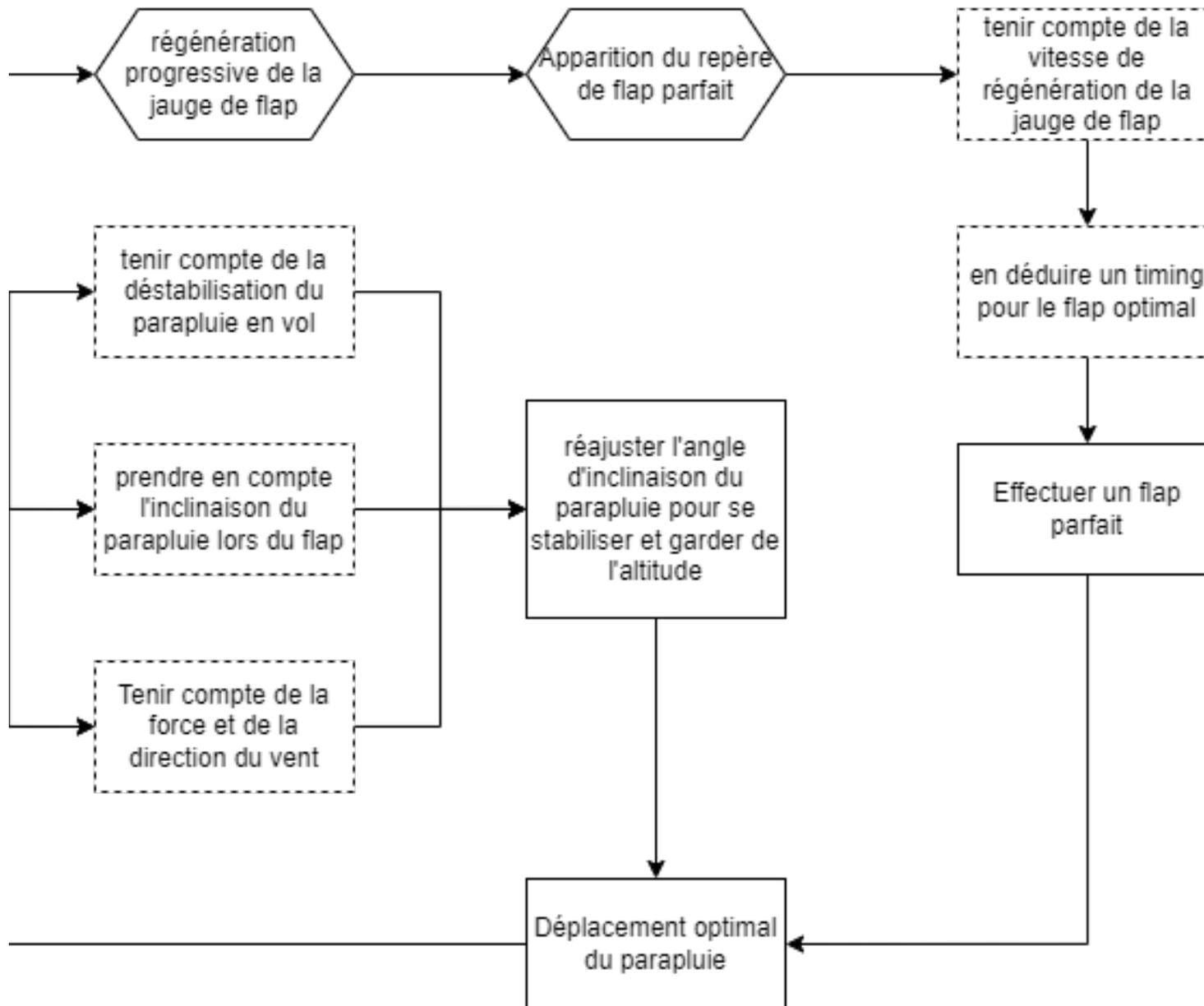
Eventail de possibilités :

- Se laisser porter par le vent, mais un vent trop puissant peut être problématique
- « Flap flap » en s'ouvrant puis en se refermant (remonter un peu en l'air)
- S'ouvrir (et planer, créer du drag, de la résistance au vent)
- Se fermer (moins de résistance au vent, chute libre)
- Se retourner (si vent trop puissant)
- Se casser (si impact avec un élément)
- Protéger quelque-chose de la pluie, du vent en faisant barrage
- Inclinaison du parapluie (profiter du vent ou protéger)

Boucles de Gameplay



Boucles de Gameplay



Boucles de Prédiction

Boucles de prédiction :

Prédiction	Décision	Action	Régulation	Apprentissage
Je veux atteindre un objet en hauteur	Je décide de l'objet à atteindre	Je prends de la hauteur	Je modifie ma trajectoire afin d'atteindre mon objectif	Je sais comment atteindre des objets en hauteur

Prédiction	Décision	Action	Régulation	Apprentissage
Je veux atteindre un objet en contrebas	Je décide de l'objet à atteindre	Je chute de mon point de départ	J'ouvre et ferme le parapluie afin de contrôler ma vitesse de chute en plus de contrôler ma trajectoire	Je sais contrôler ma vitesse de chute et atteindre des objets en contrebas

Prédiction	Décision	Action	Régulation	Apprentissage
Je veux savoir l'utilité des courants	Je décide de prendre un courant	Je vais dans le courant		Je sais que les courants me portent dans la direction de ce dernier

Boucles de Prédiction

Boucles de prédiction :

Prédiction	Décision	Action	Régulation	Apprentissage
Je veux faire la différence entre vent "doux" et vents "violents"	Je décide de prendre les 2 types de courants	Je prends un vent "doux" et un vent "violent"		Je comprends que les vents "violents" peuvent retourner le parapluie, et donc provoquer un Game Over, et non les "doux"
Prédiction	Décision	Action	Régulation	Apprentissage
Je veux passer entre deux colonnes	Je décide de passer entre les deux colonnes	Je me déplace vers les colonnes	Je modifie ma trajectoire afin de passer entre les colonnes	Je maîtrise mieux le déplacement du parapluie
Prédiction	Décision	Action	Régulation	Apprentissage
Je veux protéger un objet de la pluie	Je décide de protéger l'objet	J'ouvre le parapluie près de l'objet afin de protéger ce dernier	J'incline le parapluie afin de créer une plus grande surface de protection	Je sais comment protéger des objets de la pluie

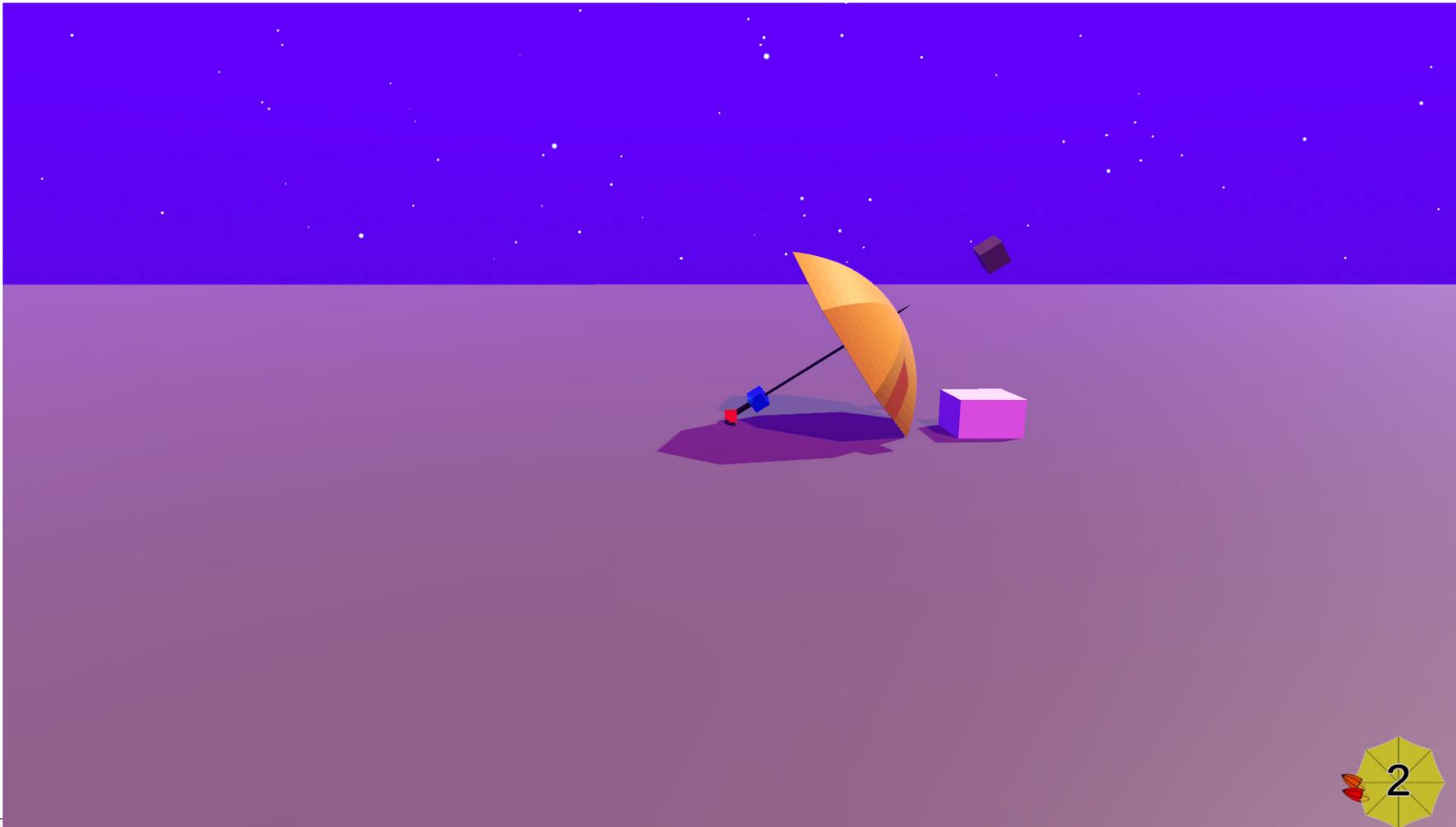
3C : caméra

Intentions de la caméra :

Nous avons un jouet qui se repose sur la difficulté de manipulation, la caméra se doit donc d'aider au maximum le joueur à s'orienter selon ses actions automatiquement.

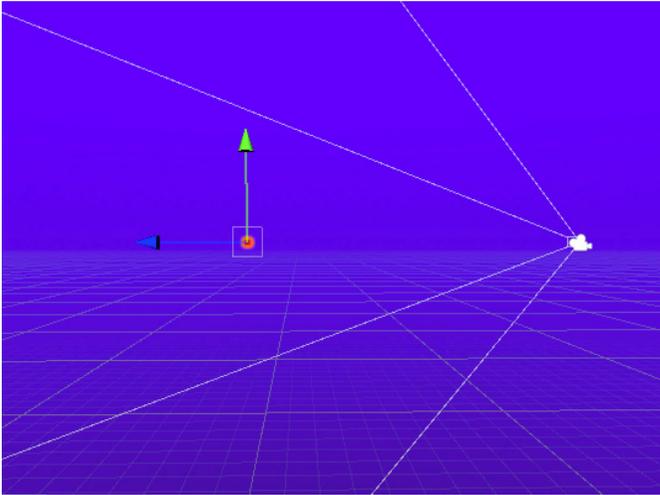
La manipulation de la caméra ne doit pas être un fardeau, elle doit être utile si nécessité d'observation, et se déplacer automatiquement au bon endroit au bon angle.

Parapluie au sol :



3C : caméra

Centre de la caméra :



Le cube rouge (cube R) est le centre (parent) de la caméra qui est reculé en Z par rapport à celui-ci. La souris ou joystick droit fait rotationner ce cube R, ce qui fera bouger la caméra (enfant) autour de ce point (avec offset vertical).

Cube R suit et se déplace de manière constante vers la position du bas du parapluie de manière smooth.

La position et la rotation de cube R par rapport au parapluie a un impact sur les mouvements automatiques de celui (et donc au mouvements de la caméra).

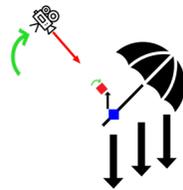
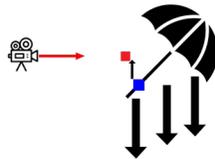
Parapluie en chute verticale :



Cube Bleu (Cube B), il est attaché au parapluie, il permet de comparer la hauteur de la cam par rapport au parapluie.

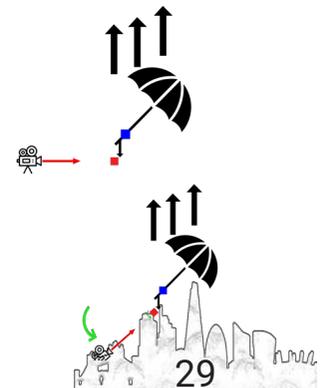
Rotation verticale automatique :

- si cube R est au dessus de cube B, alors le parapluie est en chute, cube R rotationne de manière à monter la caméra vers le haut en l'inclinant vers le bas



inversement :

- si cube R est en beaucoup en dessous de cube B, alors le parapluie est en train de s'élever, cube R rotationne de manière à descendre la caméra vers le haut en l'inclinant vers le bas



3C : caméra

Rotation horizontale :



Cube attaché au parapluie indiquant son orientation en X et Z.

Cube gris (cube G) reprenant les coordonnées X et Z de celui attaché au parapluie, plus la hauteur (Y) de cube R.

Conditions spéciales :

- si le joueur rotationne le cube R avec l'input, alors les rotations automatiques verticales ne fonctionnent pas.

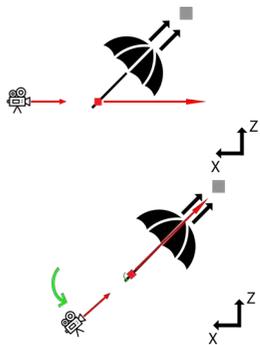
- si le joueur rotationne le cube R avec l'input, alors les rotations automatiques horizontales ne fonctionnent pas pendant 2 secondes.

- les vitesses de rotations automatiques du cube R dépendent de la différence entre les positions/rotations cible et les positions/rotations actuelles.

- les rotations verticales automatiques sont limitées, elles s'arrêtent de manière à ne pas trop monter et descendre la caméra selon la distance entre cube R et cube B (si le parapluie chute ou s'élève lentement)

- la caméra ne rotationne pas automatiquement jusqu'aux offsets limites.

Rotation horizontale automatique :

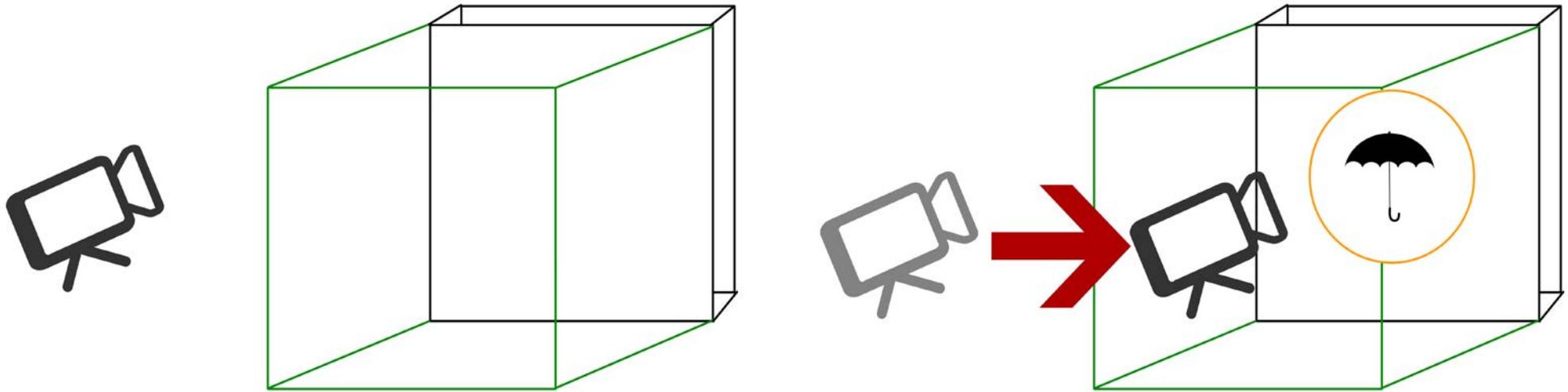


- si le joueur rotationne le cube R ne regarde pas cube G, alors cube R rotationne afin de regarder cube G.

3C : caméra

Vision du Parapluie à travers les murs :

Pour l'effet qui permet de voir le parapluie à travers les murs nous avons utilisé un shader qui met un cercle transparent au centre de la camera sur le material si le collider de la camera entre a l'interieur du trigger collé de l'autre coté du mur. Si la camera sort de se trigger l'effet s'estompe. Si le mur est plus large que la distance Caméra/shader, avant même l'entrée dans le trigger, le mur sera juste invisible.



3C : character

Le Parapluie Volant :

Parapluie est un controller reprenant les caractéristiques du fantasme du parapluie volant.

Le joueur incarne donc un Parapluie volant à la recherche de son propriétaire à travers divers décors et environnements. Il est capable de s'ouvrir et de se fermer et ainsi, il peut se propulser dans les airs dans un mouvement de battement, le flap. Le flap est limité et ne se régénère qu'au contact d'un certain objet ou d'une certaine matière dans les niveaux.

Influencé par la gravité, il chutera lentement en planant lorsqu'il est ouvert et tombera à pic s'il est fermé. Lorsqu'il plane, son angle d'inclinaison va également affecter sa vitesse de chute. Moins il est droit par rapport au sol, plus il tombera vite.

Enfin, le parapluie est soumis aux conditions météorologiques comme le vent pouvant le porter ou le faire s'envoler voir le détruire en cas de vent très violents.

3C : controles

Mapping et Choix (Manette)

Orientation du parapluie :

Le joueur oriente son avatar (parapluie) pour se stabiliser dans les airs : effectuer des rotations pour corriger son angle et sa position vis à vis des objets de la scène et de son objectif. Manette stick gauche, orienter le stick pour modifier l'angle du parapluie.

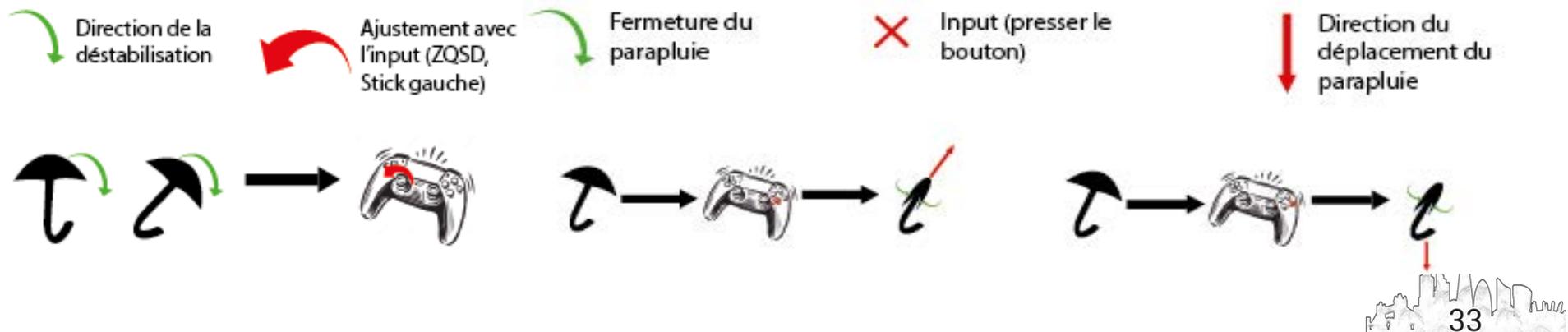
Les conventions veulent que le déplacement se fasse via le stick gauche de la manette, le déplacement de Parapluie se fait via des impulsions actionnées par un bouton. Modifier l'angle et la direction du parapluie influence directement vers où le parapluie va se propulser. Les boutons étant le plus souvent actionnés par la main droite, la main gauche est souvent libre. Le stick gauche comme stick directionnel (au sens propre du terme) nous semble donc être une solution optimale.

S'ouvrir/se fermer et Flap :

Le flap propulse le joueur selon l'angle et l'inclinaison du parapluie dans la direction vers laquelle il fait face. Manette bouton A

Le parapluie fermé possède une résistance à l'air inférieure au parapluie ouvert, permettant au joueur de perdre plus vite de l'altitude. Manette bouton B

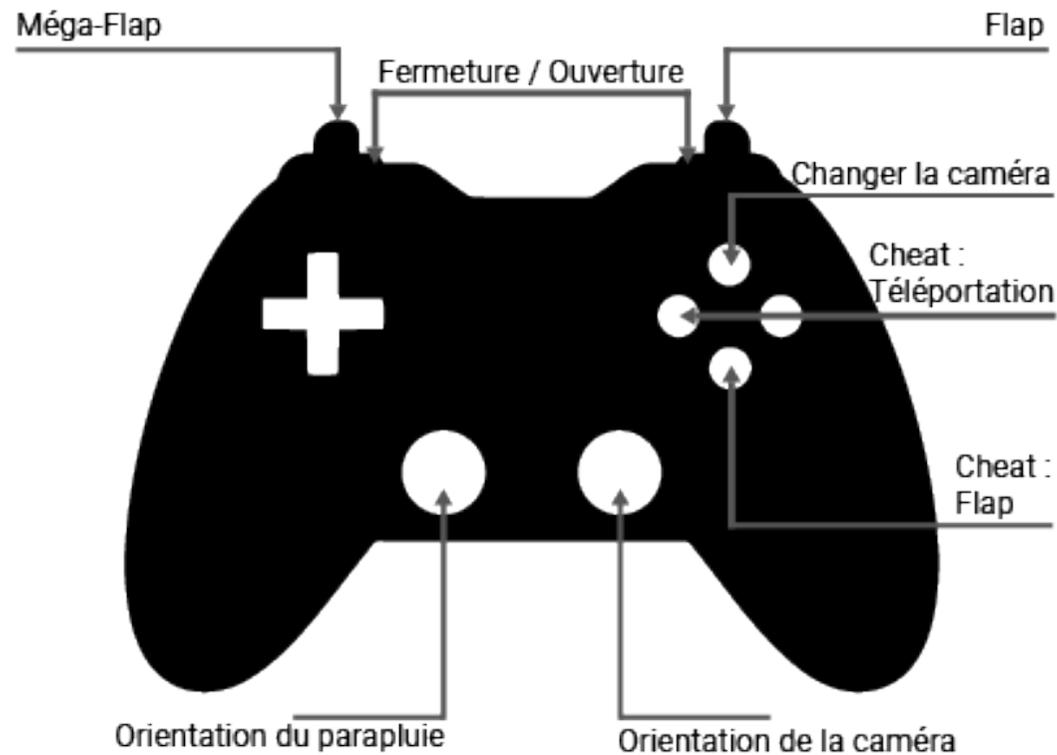
Pareillement, par convention, la main droite actionne les boutons. Il est plus simple d'utiliser le flap et de se fermer à sa guise en utilisant A et B, boutons faciles d'accès par le pouce. Le jeu demande de flap plus souvent que de s'ouvrir/se fermer, le flap prend la priorité pour le bouton A



3C : controles

Orientation de la caméra :

La majorité des déplacements de la caméra sont automatiques et suivent les déplacements du joueur sur les axes verticaux et horizontaux. Cependant, lorsque le joueur est au sol ou dans les airs, il peut orienter la caméra afin de se diriger vers là où il le souhaite. Lorsque le joueur est dans une optique de planification de son déplacement, il est très rarement en train de se déplacer, sa main droite est donc inactive et libre. L'orientation de la caméra peut donc être mappée sur le stick droit. Cela correspond également aux conventions ce qui permet de shift l'attention du joueur vers la maîtrise du parapluie plutôt que l'apprentissage des boutons



3C : controles

Mapping et Choix (Clavier + Souris)

Orientation du parapluie :

Le joueur oriente son avatar (parapluie) pour se stabiliser dans les airs : effectuer des rotations pour corriger son angle et sa position vis à vis des objets de la scène et de son objectif. ZQSD (ou WASD) pour modifier l'angle du parapluie.

Les conventions veulent que le déplacement soit géré par la main gauche dans le cas où il est conçu pour le clavier. Le déplacement de Parapluie se fait via des impulsions actionnées par un bouton. Modifier l'angle et la direction du parapluie influence directement vers où le parapluie va se propulser. Les boutons étant le plus souvent actionnés par la main droite, la main gauche est souvent libre. Le clavier comme controller directionnel (au sens propre du terme) nous semble donc être une solution optimale.

S'ouvrir/se fermer et Flap :

Le flap propulse le joueur selon l'angle et l'inclinaison du parapluie dans la direction vers laquelle il fait face. Souris click droit

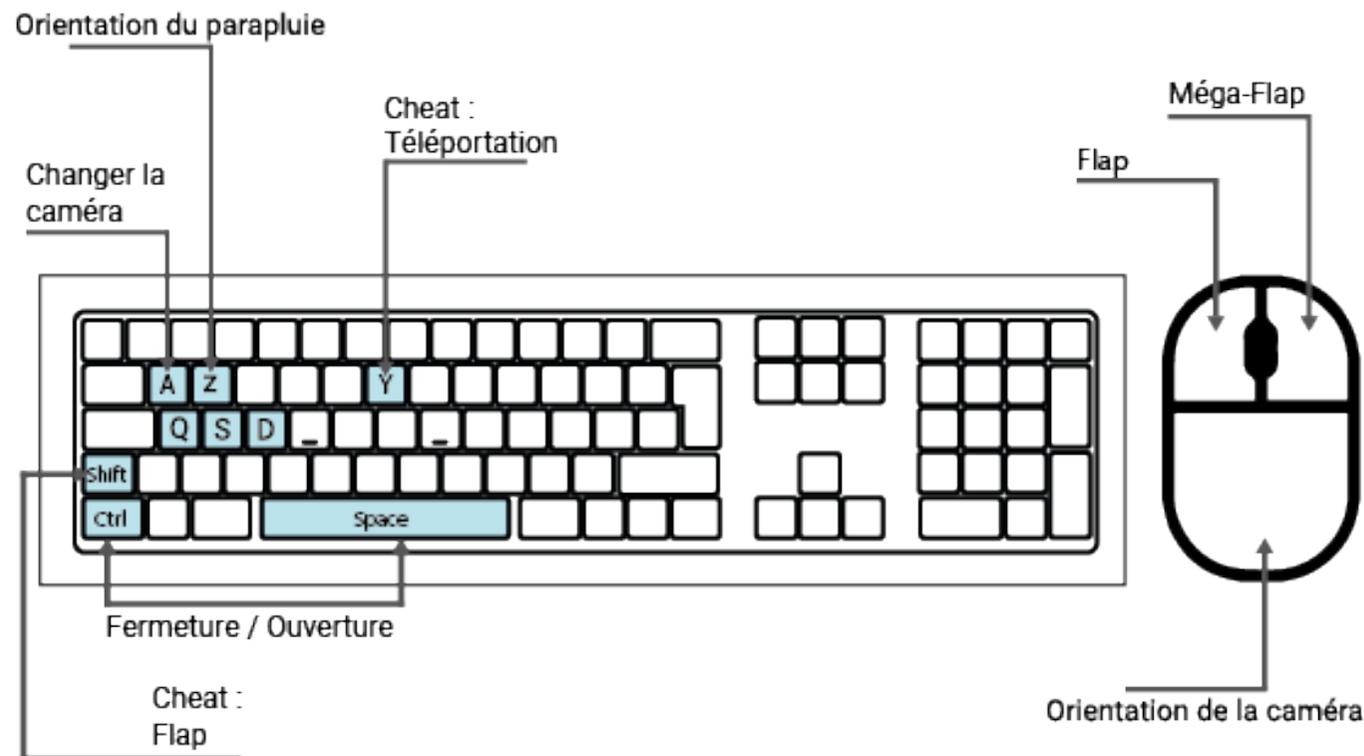
Le parapluie fermé possède une résistance à l'air inférieure au parapluie ouvert, permettant au joueur de perdre plus vite de l'altitude. Souris click gauche

Pareillement, par convention, la main droite utilise les boutons de la souris. Parapluie ne possède que deux input nécessitant des boutons, parfait pour une souris.

3C : controles

Orientation de la caméra :

La majorité des déplacements de la caméra sont automatiques et suivent les déplacements du joueur sur les axes verticaux et horizontaux. Cependant, lorsque le joueur est au sol ou dans les airs, il peut orienter la caméra afin de se diriger vers la où il le souhaite. Lorsque le joueur est dans une optique de planification de son déplacement, il est très rarement en train de se déplacer, sa main droite est donc inactive et libre. L'orientation de la caméra peut donc être mappée sur le stick droit. Cela correspond également aux conventions ce qui permet de shift l'attention du joueur vers la maîtrise du parapluie plutôt que l'apprentissage des boutons. Ce problème se pose beaucoup moins si le contrôle de la caméra est lui aussi sur la souris (en plus des deux boutons). Il n'y à plus aucun conflit entre les phases de recherches de destinations et de déplacement.



Mécanique - Le Flap

Comment fonctionne le Flap ?

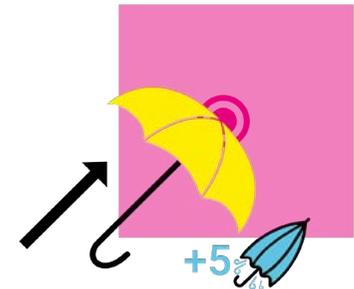
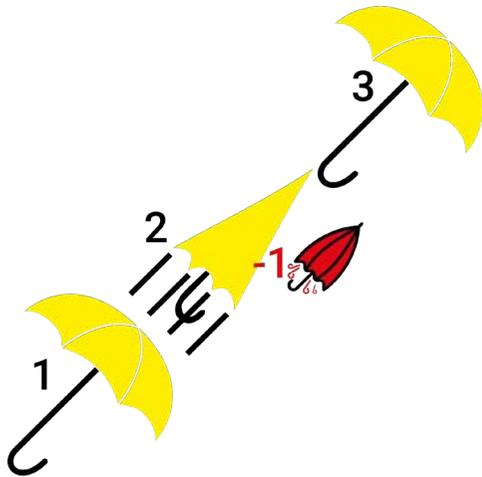
- En appuyant sur un bouton, le parapluie se ferme et s'ouvre de nouveau rapidement et se propulse par impulsion dans la direction vers laquelle il regarde.
- Pendant l'animation du Flap, le joueur ne peut pas se fermer (freeze input de fermeture)
- un flap consomme de l'énergie, l'énergie max du joueur est à 100 et un flap coûte 15 d'énergie.
- si le joueur flap dans le bon timing après avoir effectué un flap, il effectue un perfect flap plus puissant et qui augmente la régénération d'énergie.
- Le joueur peut effectuer 7 flaps sur une barre s'il effectue uniquement des perfect flap

Comment récupérer de l'énergie ?

Lorsque le parapluie se pose au sol, le joueur récupère l'intégralité de son énergie

Lorsque le parapluie traverse un bonus de flap, le joueur récupère l'intégralité de son énergie

L'énergie est régénérée lorsque le parapluie est dans les airs de manière automatique petit à petit.

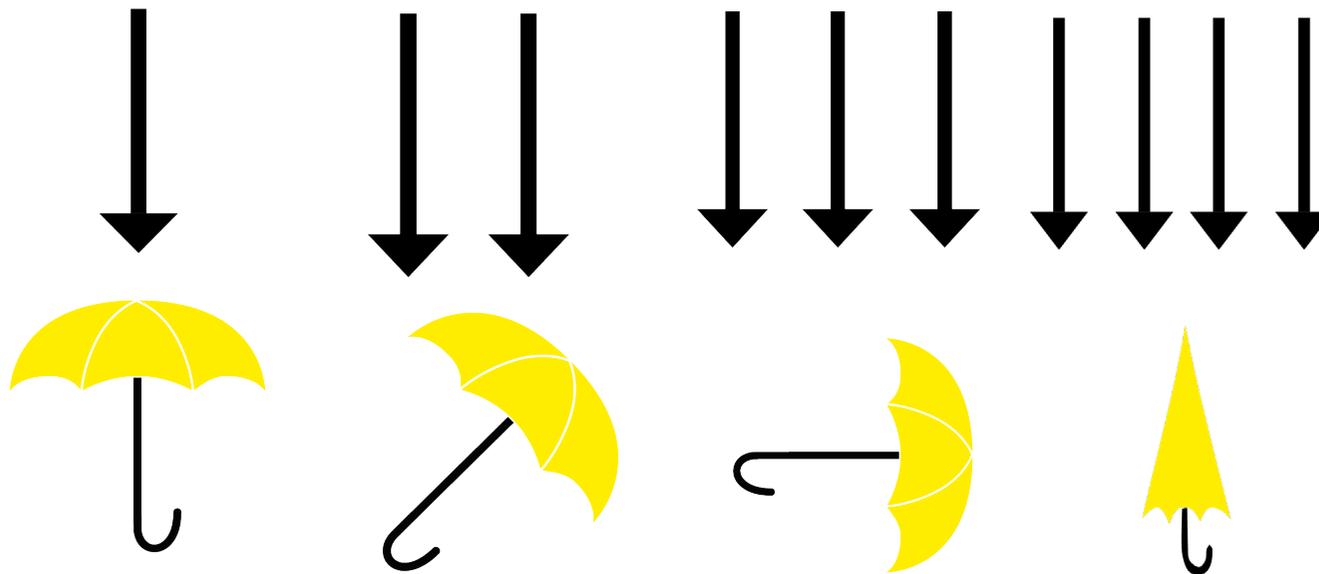


Mécanique - Comportement dans les Airs

Comportement dans les Airs :

En fonction de son état, le parapluie possède différents comportements dans les airs :

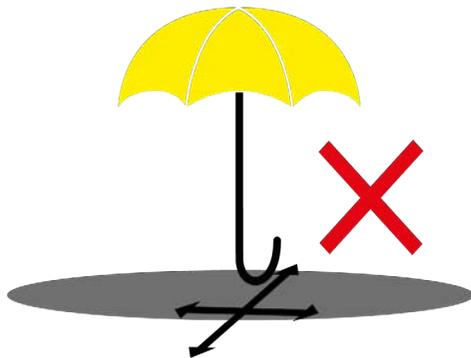
- S'il est fermé, il tombera à pic (aucun drag) jusqu'au sol ou jusqu'à ouverture de la part du joueur.
- S'il est ouvert, il va lentement chuter en planant en pouvant s'orienter dans les 3 axes.
- Lorsqu'il est ouvert et qu'il plane, son angle et son orientation vont affecter sa vitesse de chute.



Mécanique - Comportement au Sol

Comportement au Sol :

- Lorsque le parapluie est au sol, il ne peut pas se déplacer. Il ne peut que s'orienter sur les 3 axes afin d'offrir au joueur la possibilité de repartir
- S'il possède encore des flaps, il peut repartir dans les airs
 - S'il est sur une Plateforme, il récupère des flaps jusqu'à en avoir 5 s'il en a moins de 5, sinon son nombre de flap est inchangé
 - S'il n'est pas sur une plateforme et qu'il n'a plus de flap, c'est un «Game Over»



RGD - Vent

Type	Paramètre Atomic			
	Pas de challenge	Facile	Normale	Difficile
Taille vent	Omniprésent	Grand	Normal	Petit
Hitbox parapluie	Très Grand	Grand	Normal	Petit
Fréquence d'indication	Omniprésent	Régulièrement	Normal	Rarement
Taille de l'objet d'indication	Très grand	Grand	Normal	Petit
Position de l'indication	Omniprésent	Visible	Normal	Caché
Indication direction de déplacement du vent (indiqué par un objet)	Omniprésent	Visible	Normal	Caché
Indication vitesse vent	Omniprésent	Visible	Normal	Caché
Puissance du vent	Très faible	Faible	Normal	Fort
Nombre obstacle	Aucun	Peu	Normal	Beaucoup

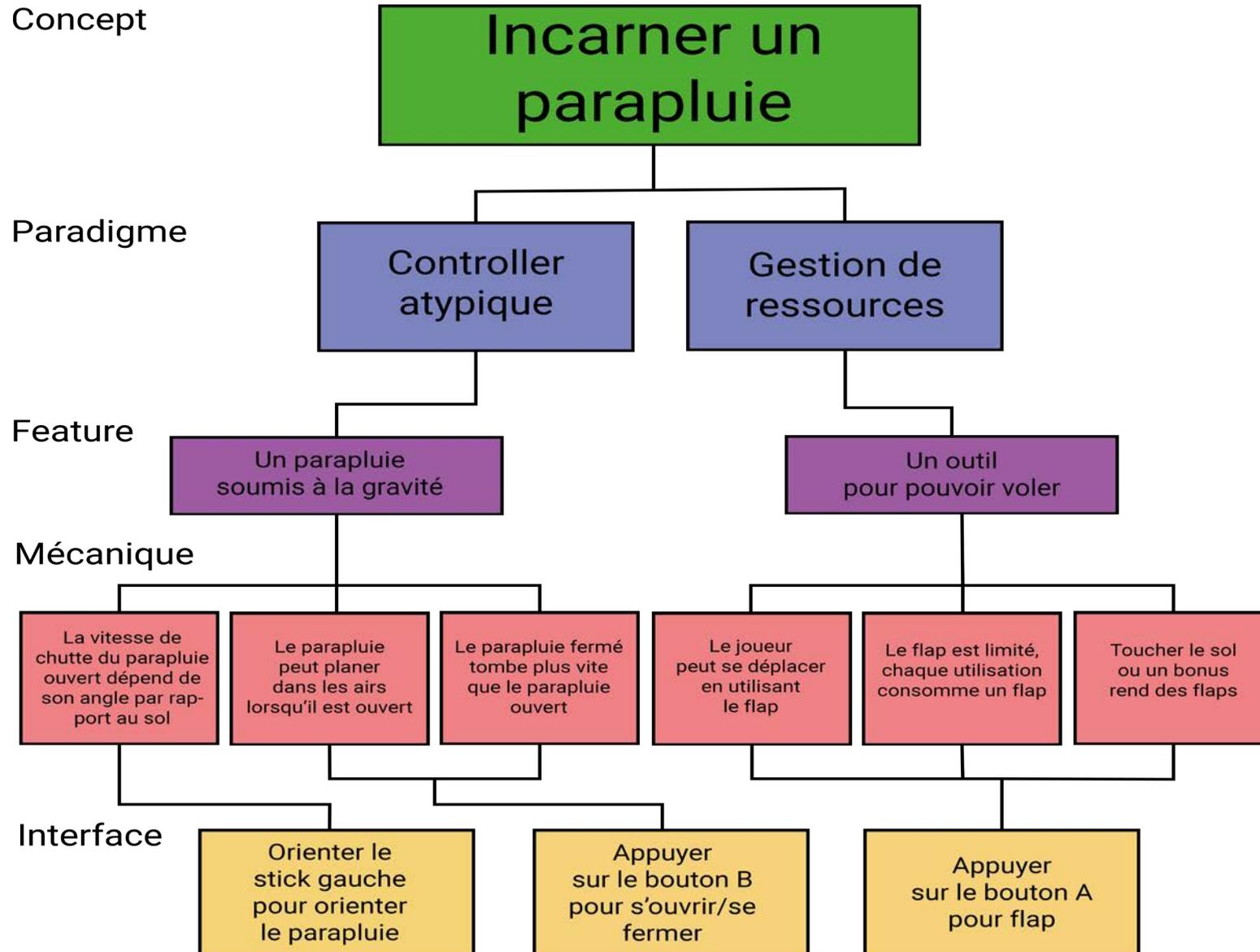
RGD - Obstacle

Paramètre Atomic					
Type	Pas de challenge	Facile	Normale	Difficile	
amplitude d'inclinaison du parapluie	Très faible	Faible	Normal	Fort	
vitesse d'inclinaison	Très lent	Lent	Normal	Rapide	
vitesse de chute	Très lent	Lent	Normal	Rapide	
taille obstacle	Aucun	Petit	Normal	Grand	
hitbox parapluie	Très petit	Petit	Normal	Grand	

RGD - Déplacement

Paramètre Atomic				
Type	Pas de challenge	Facile	Normale	Difficile
amplitude d'inclinaison du parapluie	Très faible	Faible	Normal	Fort
vitesse d'inclinaison	Très lent	Lent	Normal	Rapide
vitesse de chute	Très lent	Lent	Normal	Rapide
Nbr flap	Infini	Beaucoup	Normal	Peu
puissance flap	Très grand	Grand	Normal	Petit

Diagramme de Ventrice



MDA Framework

Déclinaison de la sensation	Dynamics de la sensation
<ul style="list-style-type: none">□ Joystick, Vitesse d'orientation jaugeable et en fonction de l'angle de la caméra□ Caméra suivant le parapluie avec délais + s'oriente automatiquement en fonction de la direction du parapluie□ Gâchettes sur les flaps, -> appuie à moitié pour activation.□ Décroissance + croissance de l'énergie□ Perfect du flap, feedback + flap du parapluie□ Flottaison dans le vent, gravité presque malléable□ Orientation en accord avec le vent, accélération et décélération: vents, fermeture, bonus.	<p>Controllers manettes fluides, et stable, peu besoin de spam ou de changer la disposition des doigts.</p> <p>Parapluie qui suit le vent, visiblement ou qui va en contre sens. Feedback sonore & visuelle de la récupération d'un bonus ou d'un objectif</p> <p>Stabilité au sol, et difficulté à bien jauger les orientations et les distances en l'air.</p> <p>Animations du parapluie, les actions ont des effets visibles.</p> <p>La complétion du niveau se traduit à l'écran.</p>

MDA Framework

Déclinaison de la Fantasy	Dynamics de la Fantasy
<ul style="list-style-type: none">❑ Incarner un parapluie volant : être libre dans ses déplacements, dans la ville, la nature ou au-dessus de l'eau. Aucun terrain n'est inaccessible pour le joueur❑ Entrer dans des courant de vent et se propulser : fluidité et poésie, se laisser porter par le vent, oublier la difficulté des contrôles pour profiter du paysage, de l'architecture et contempler❑ Grands espaces, plaisir d'explorer la ville de Londres par les airs : atteindre et voir des lieux inaccessible en tant qu'humain (lieux interdit au public, toit de gratte-ciels, monuments)❑ Landmark forts : Le joueur visite Londres depuis les airs avec son parapluie, la complétion du niveau vient en visitant tous les bâtiments iconiques indiqués pour terminer la carte. Le joueur peut terminer le niveau sans le compléter, mais il n'aura pas terminé le niveau à 100 %	<p>Vol : utiliser les propriétés du parapluie et du vent afin de se déplacer dans les airs librement. une fois le système de déplacement et le comportement du parapluie dans le vent maîtrisé, je joueur peut laisser de côté l'aspect challenge et exigence du système pour se concentrer sur un aspect plus contemplatif et expérimental.</p> <p>Contemplation : le joueur peu lors de son exploration lâcher l'aspect challenge et complétion du jeu et se concentrer sur la contemplation. Chercher les changements de couleurs et d'architectures plutôt que les différents challenges et le gameplay. Poser la manette et regarder, se remémorer les expériences vécues, nouvelle dimension poétique. Difficulté des contrôles → maîtrise des contrôles → accès à la dimension poétique du jeu (être en harmonie avec le système une fois la bataille avec le système terminée).</p> <p>Exploration - Liberté et choix : le joueur observe le niveau et détermine un itinéraire vers la destination choisie et depuis cette destination, il choisit une autre destination. L'itinéraire optimal n'est qu'une suggestion et le joueur se crée son propre chemin à travers Londres selon ses désirs.</p> <p>Exploration - Landmark fort : Dans le niveau, le joueur peut trouver des bâtiments et monuments iconiques de Londres pour les visiter. Chaque bâtiment offre un POV intéressant pour le joueur lui donnant des informations sur les itinéraires à prendre pour se rendre aux autres bâtiments et contribuent à la complétion de sa visite</p> <p>Apprentissage : Expérimentation, theory, practice, trial and error, mastery</p>

MDA Framework

Déclinaison du challenge	Dynamics du challenge
<ul style="list-style-type: none">❑ Le challenge se trouve dans la demande de la maîtrise de la navigation, du controller pour atteindre un lieu donné.❑ Les trajectoires visibles demandés sont marqués par les ingrédients comme le vent ou le bonus de flap, et demande parfois une optimisation ou un choix à faire pour atteindre l'objectif.❑ Le perfect flap, méga flap et fermeture peuvent être indispensable pour des challenges exigeants.	<p>La limite d'énergie qui oblige une optimisation des flaps, le perfect flap permet d'étendre la limite en utilisant le même taux d'énergie pour gagner de la distance avec le flap.</p> <p>La vitesse de rotation du parapluie et la puissance des flaps qui sont pas modifiable : Modifier la distance d'un flap avec la rotation du parapluie est très utile.</p> <p>Utilisation du vent, s'aligner avec ou aller en contre sens modifie la vitesse et donc la trajectoire, une bonne utilisation du vent réduit considérablement le temps passé sur un challenge.</p> <p>Le bonus de flap, il est optimal de vider son énergie avant de le prendre, comme utiliser un méga flap juste avant pour avoir 100% d'énergie en étant loin après le bonus de flap.</p>

MDA Framework

Déclinaisons de la Discovery	Dynamics de la Discovery
<ul style="list-style-type: none">❑ Grands espaces, plaisir d'explorer la ville de Londres par les airs : atteindre et voir des lieux inaccessible en tant qu'humain (lieux interdit au public, toit de gratte-ciels, monuments)❑ Landmark forts : Le joueur visite Londres depuis les airs avec son parapluie, la complétion du niveau vient en visitant tous les bâtiments iconiques indiqués pour terminer la carte. Le joueur peut terminer le niveau sans le compléter, mais il n'aura pas terminé le niveau à 100 %❑ Plaisir de découvrir plusieurs architectures caractéristique d'un quartier/d'une époque. Découvrir les couleurs liées à chaque quartier/monument iconique et leur effet sur le parapluie❑ Découvrir les interactions du parapluie avec son environnement et tirer profit de ces interactions pour découvrir le niveau❑ Découvrir de nouvelles circulatoires que je joueur n'avait pas vu ou visualisées lors de son premier passage suite à sa maitrise des éléments de déplacement du jeu (flap, vent et gestion de l'angle du parapluie)	<p>Exploration - Liberté et choix : le joueur observe le niveau et détermine un itinéraire vers la destination choisie et depuis cette destination, il choisit une autre destination. L'itinéraire optimal n'est qu'une suggestion et le joueur se crée son propre chemin à travers Londres selon ses désirs.</p> <p>Exploration - Landmark fort : Dans le niveau, le joueur peut trouver des bâtiments et monuments iconiques de Londres pour les visiter. Chaque bâtiment offre un POV intéressant pour le joueur lui donnant des informations sur les itinéraires à prendre pour se rendre aux autres bâtiments et contribuent à la complétion de sa visite</p> <p>Ref : genshin impact/ zelda BOTW : les Landmark et les viewpoint qui permettent de se définir des objectifs et des itinéraires vers d'autre Landmark.</p> <p>Apprentissage : le joueur expérimente avec son controller et apprend par la pratique comment son parapluie interagit avec les éléments du monde. Ces connaissances sont ensuite applicables lorsque le joueur explore. L'exploration amène le joueur à rencontrer d'autres éléments/situation métant à profit ses connaissances</p> <p>Un second niveau de maitrise avec le vent et son influence sur les déplacement et la déstabilisation du parapluie. Comprendre comment le vent influence le parapluie en focntion de sa nature, sa direction et sa puissance pour pouvoir s'en servir comme moyen de déplacement ou alors éviter les effets négatifs du vent.</p>

Capsule Type de Gameplay

Navigation

Observation

- Observer la map et trouver comment naviger dedans.
- Trouver un objectif / lieu de repère
- Trouver comment atteindre l'objectif

Collection

- Apprentissage pattern LD
- Complétion du niveau / Incrementation du score
- Connaissance du monde

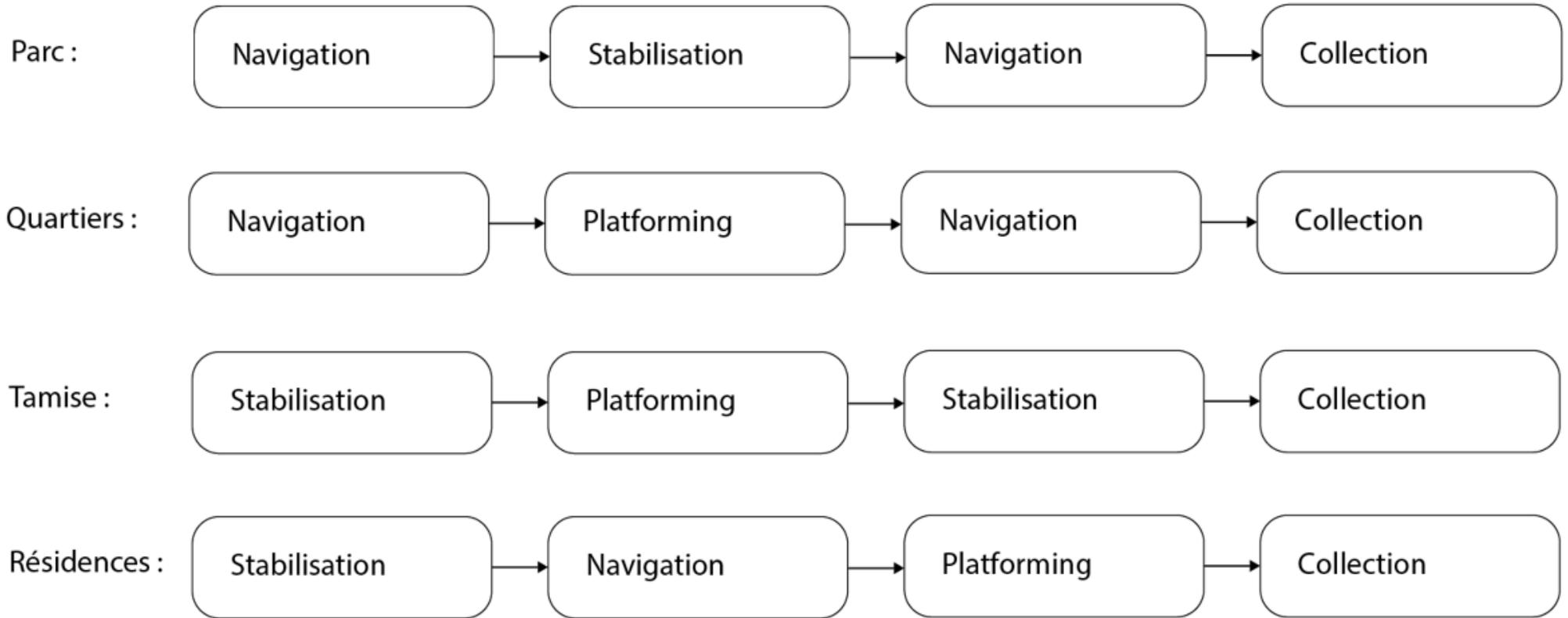
Challenges

- Orientation + flap.
- Optimisation de la trajectoire
- Optimisation de l'énergie

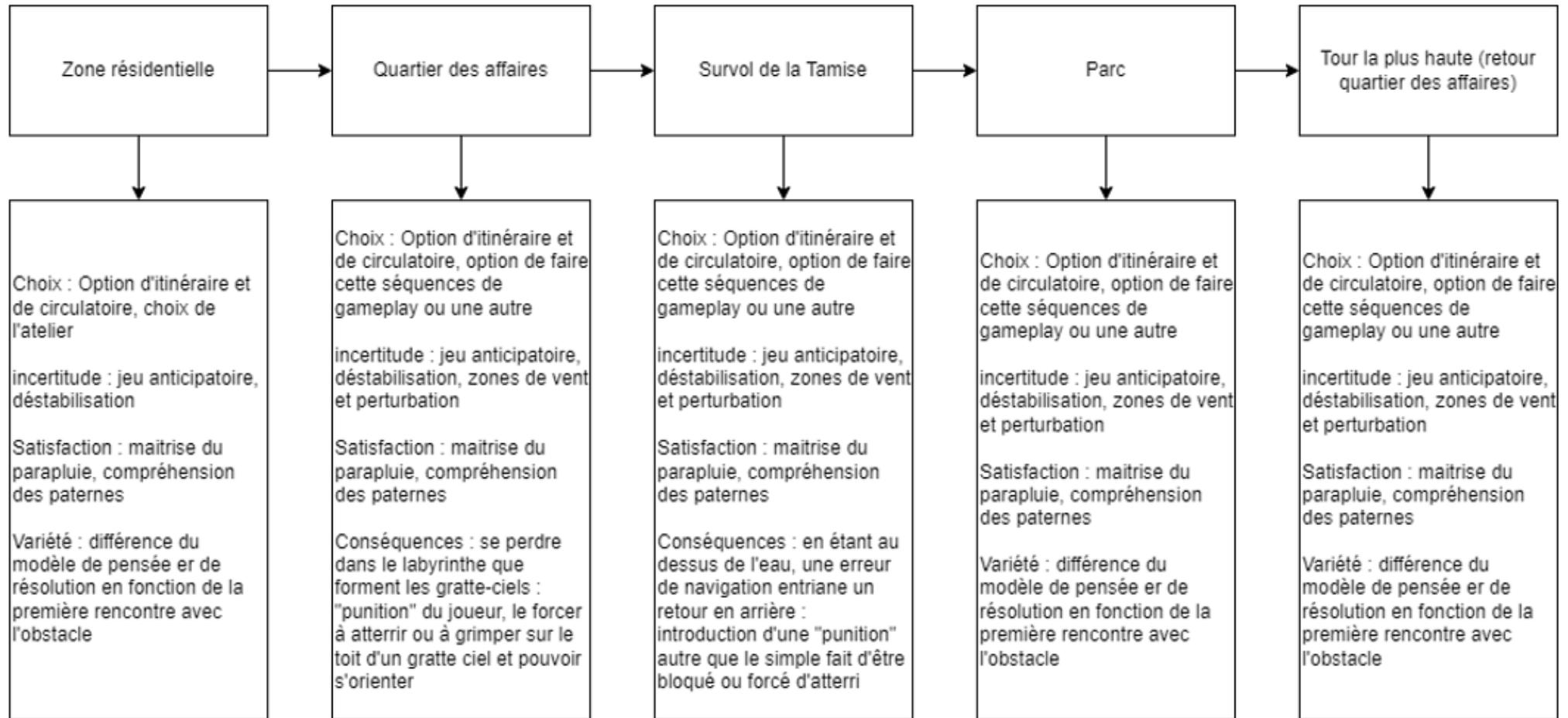


- 1 map
- 11 objectifs (batiments iconique)
- x zones de couleurs différentes

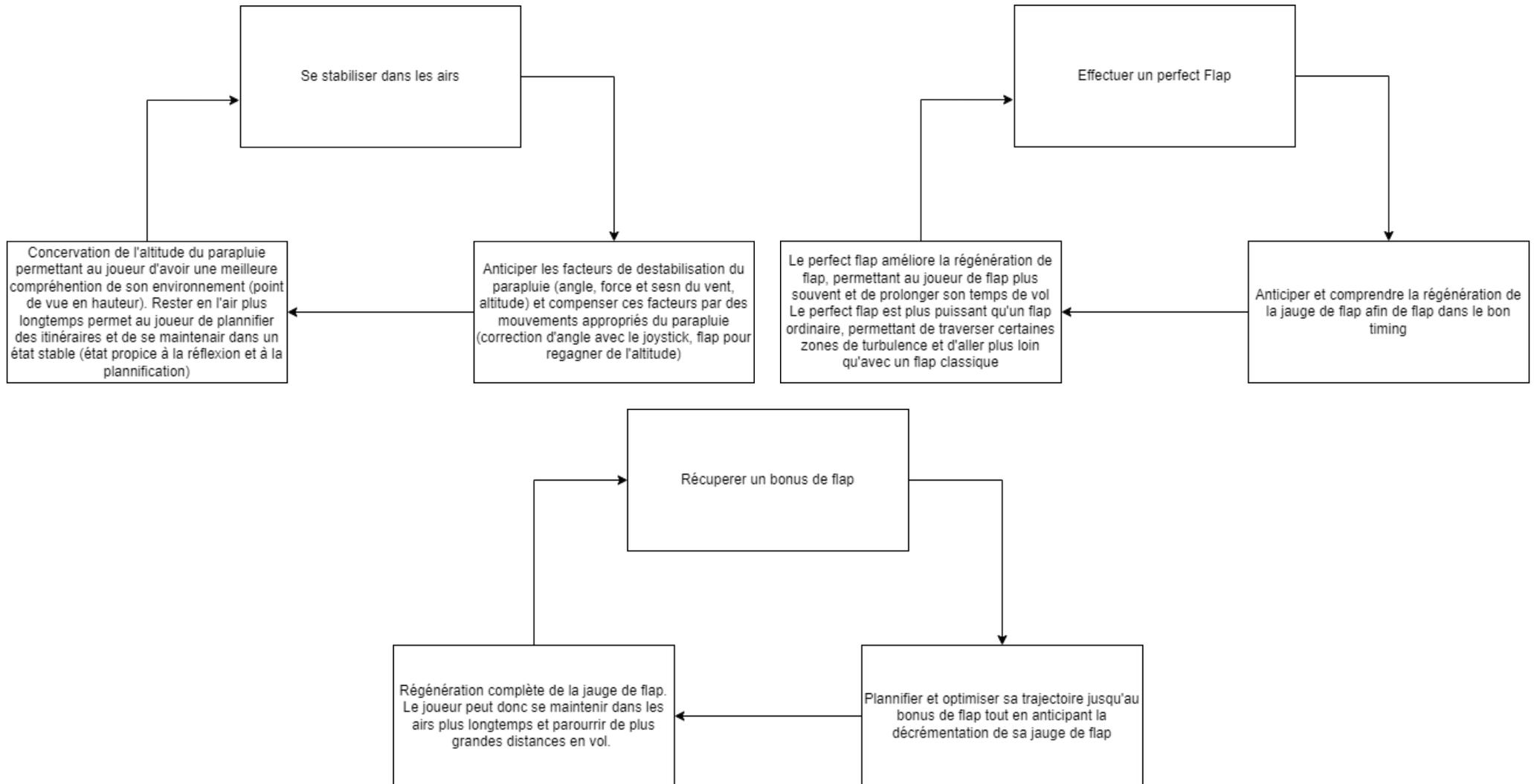
Capsule Type de Gameplay



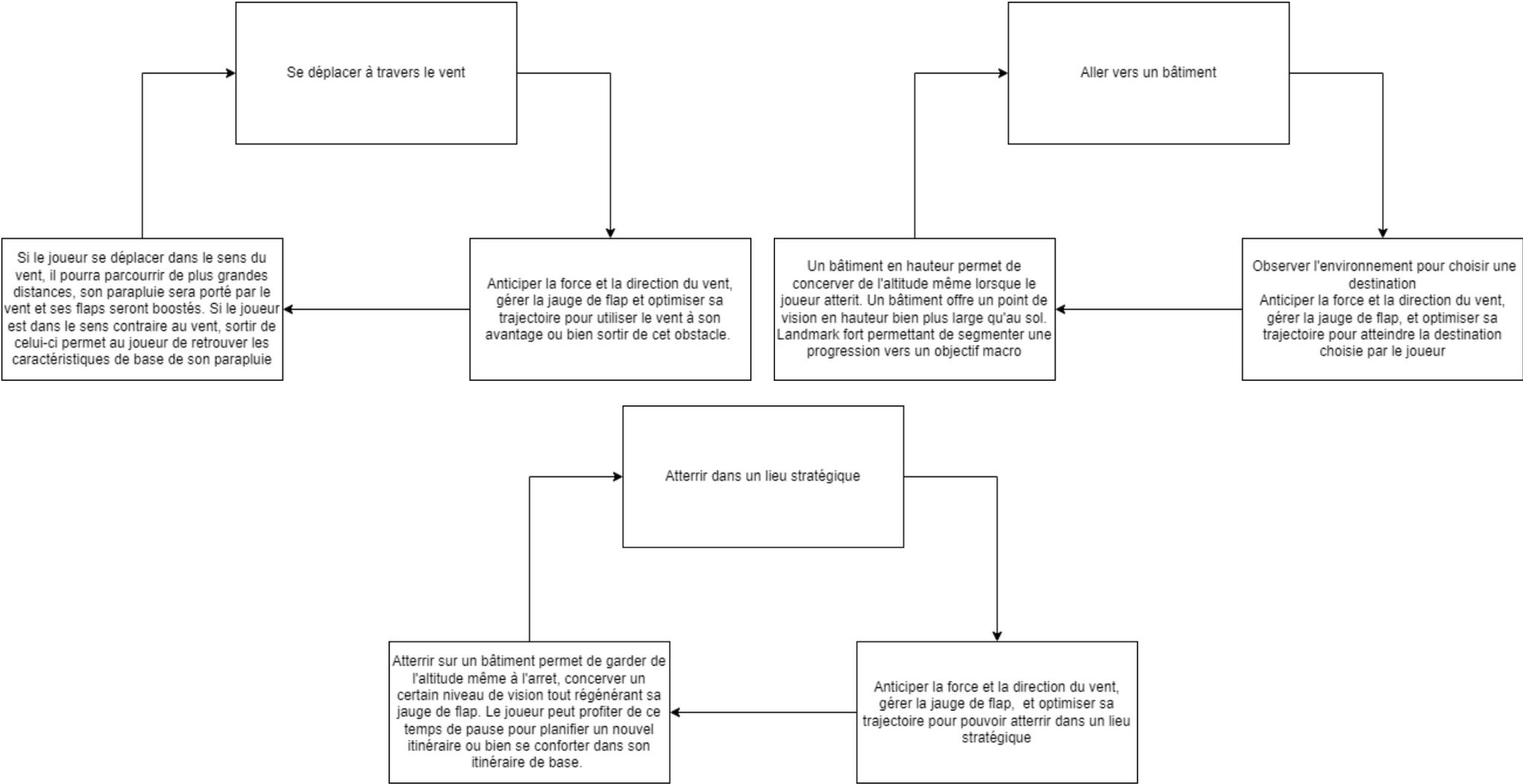
Capsule Type de Gameplay



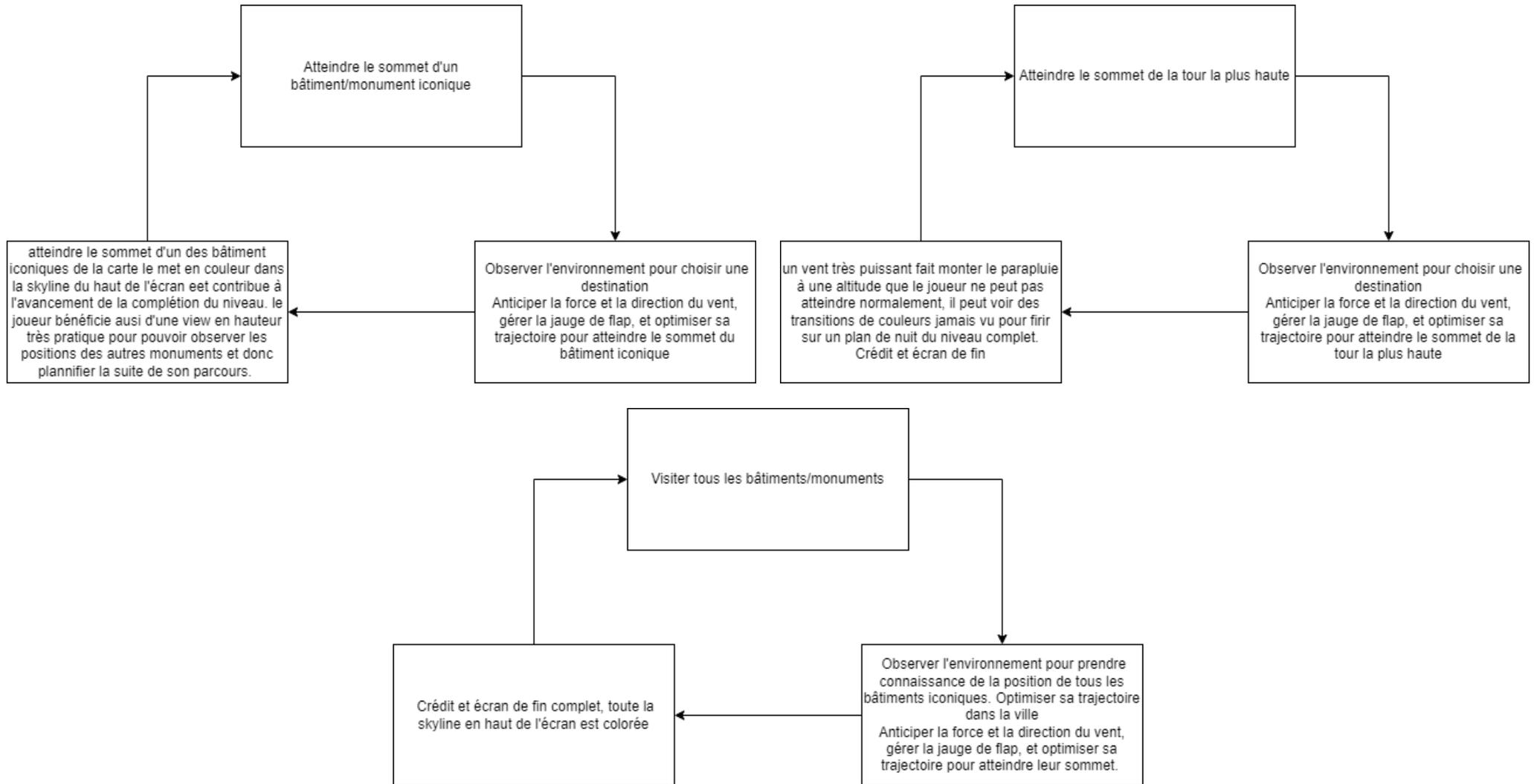
OCR court terme

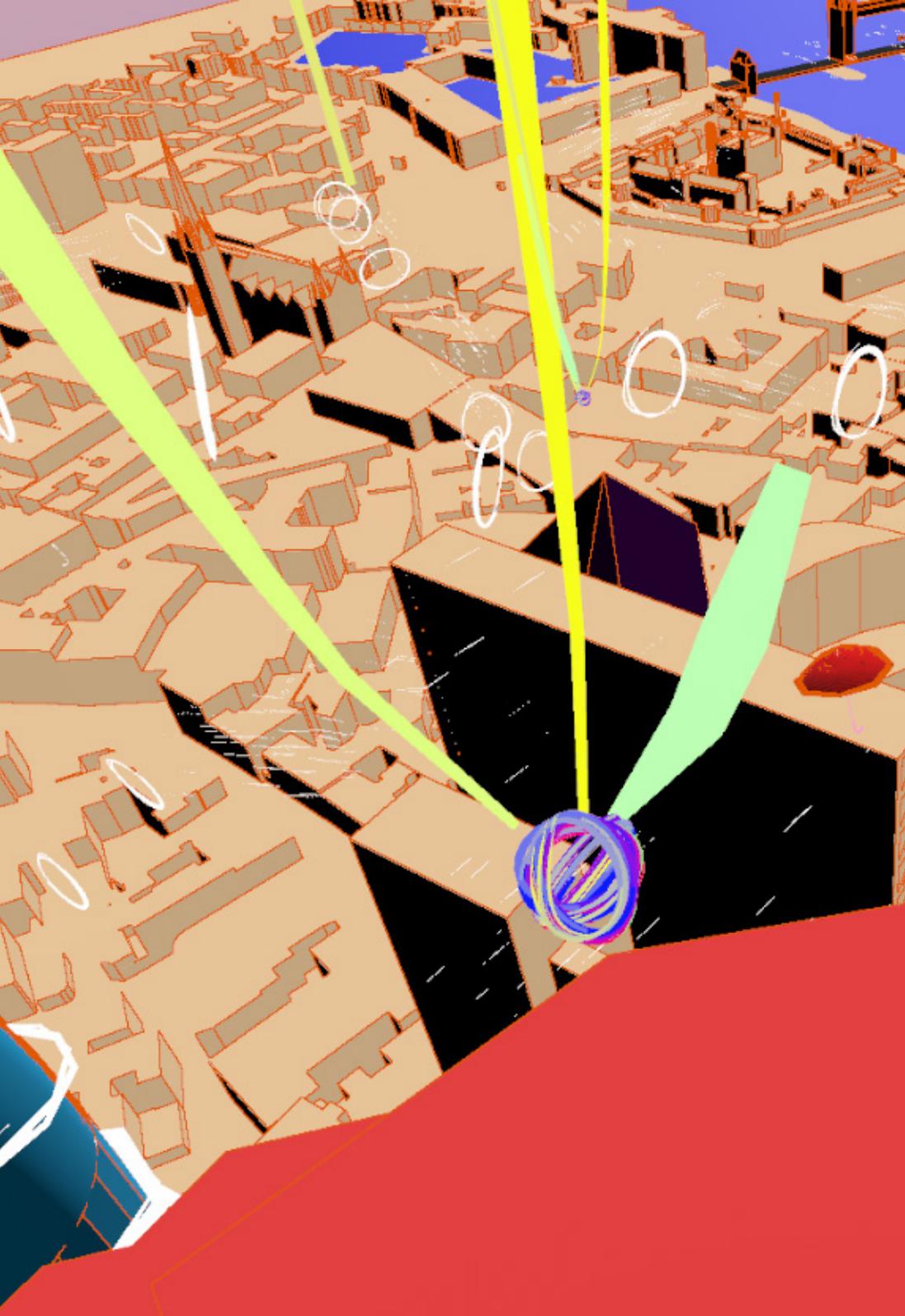


OCR moyen terme



OCR long terme





Choix de Design

Limitation du flap - Energie ou flat number :

la limitation du flap était obligatoire pour pouvoir garder un aspect fun du controller et apporter de l'enjeu dans les mouvements et déplacements dans les airs, mettre l'accent sur la notion de timing.

Deux options possibles :

- un nombre flat de flap qui reset lorsqu'on touche le sol ou lorsqu'on récupère un bonus.
- une jauge d'énergie qui se régénère au fil du temps, un flap coutant de l'énergie.

l'énergie nous permet de créer une variation du déplacement en fonction du skill, un joueur expérimenté pourra parcourir plus de distance s'il utilise le flap dans le bon timing.

Récupération d'énergie :

la régénération d'énergie se fait de deux manières : récupérer un bonus de flap ou bien attendre que la barre d'énergie se remplisse au fur et à mesure. L'énergie ici challenge le joueur sur deux aspect : sa capacité à naviguer et à maîtriser le parapluie pour atteindre un bonus de flap tandis que la vitesse de régénération dépend de si le joueur a effectué un flap parfait ou non, le challengeant sur son timing et sa maîtrise du timing lié au flap.

Choix de Design

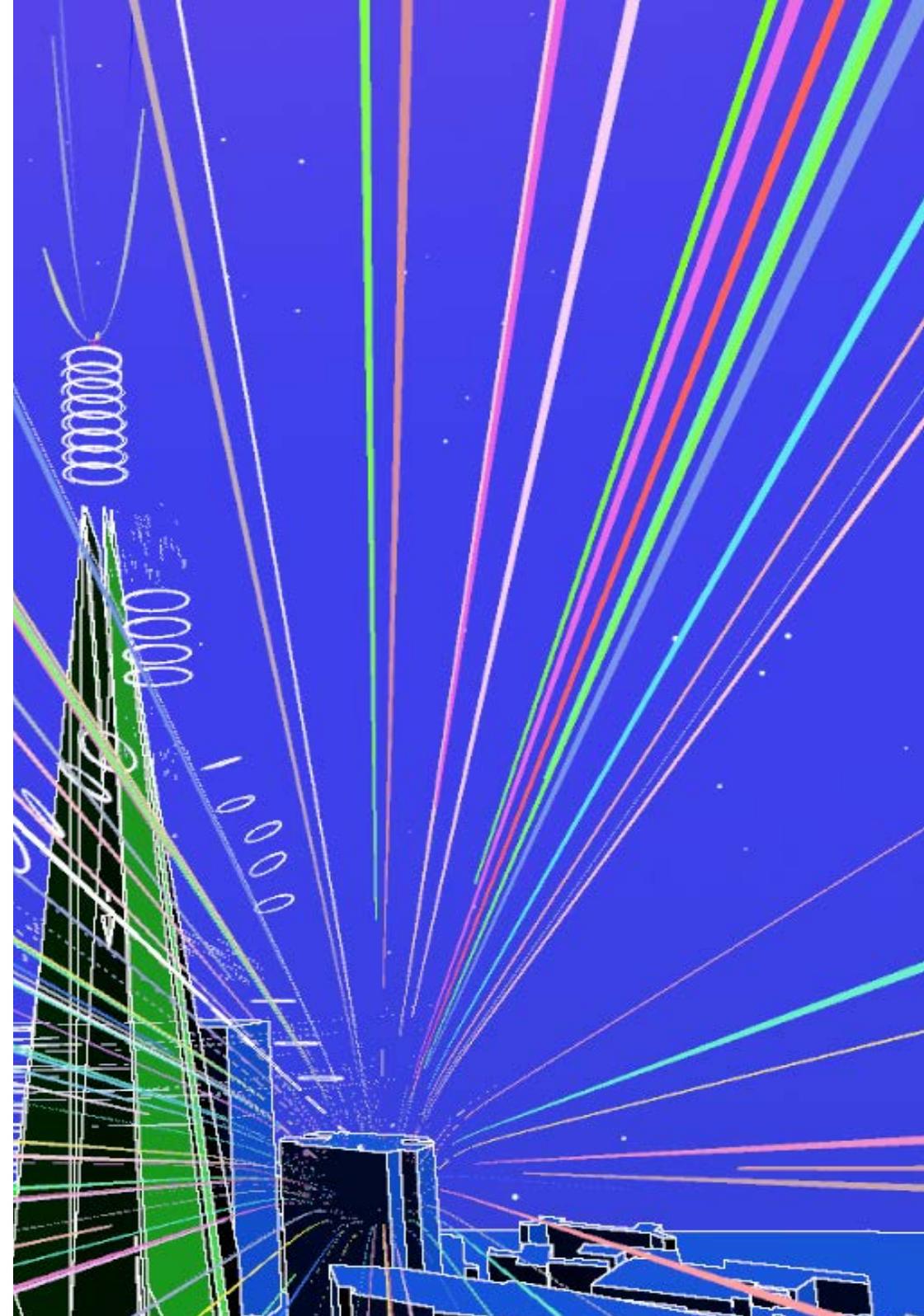
Un manque de Punitivité qui nuit au Fun :

Dans les premières phases du jeu, le parapluie était capable de se déplacer lorsqu'il était au sol, il était alors difficile d'apporter de la valeur au déplacement en l'air, bien plus complexe. Il était bien moins prise de tête de se déplacer au sol et de ne s'envoler que pour passer certains obstacles. Cette mécanique allait complètement à l'encontre de nos intentions de design (valoriser la maîtrise du système) et pire, allait à l'encontre de notre tension système souhaitée. De plus, se déplacer au sol après avoir perdu tous ses flaps jusqu'à une plateforme ou un bonus ne faisait aucun sens et retirait tout enjeu et toute punitivité au jeu.

Apporter de l'enjeu et pousser le joueur à rester dans les Airs :

-En limitant les mouvements au sol tout en gardant la liberté offerte par les mouvements dans les airs, le joueur est poussé à rester dans les airs le plus souvent possible. Rester dans les airs permet aussi au joueur de disposer de plus d'options de déplacement et d'un plus grand champ de vision lui permettant de plus facilement planifier ses déplacements.

Ne plus pouvoir se déplacer lorsque le parapluie est au sol, couplé au fait que le flap soit limité (le flap étant le seul moyen de quitter le sol autre que de changer d'atelier), créé une véritable condition de défaite. Elle est claire et menace le joueur à tout instant, apportant une tension permanente (renforcement de la tension système de la déstabilisation) et un véritable enjeu. Limiter les mouvements au sol apporte du Fun dans Parapluie.

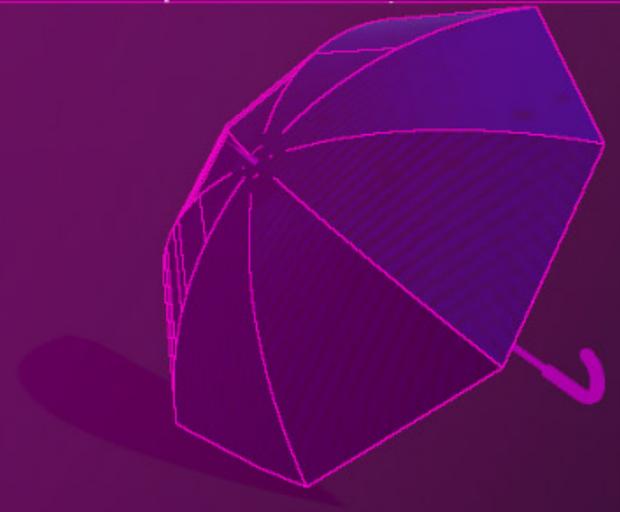




PARAPLUIE

Level Design :

- p57 - Intentions
- p58 - Ingrédients et blocs LD
- p60 - BlenderOSM
- p64 - Landmarks Naturels
- p66 - Landmark Naturels
- p69 - 30Minutes de Gameplay
- p72 - Progression et Apprentissage



Level Design - Intentions

3 Intentions de Level Design :

- Controller Aerien ; tirer profit du controller et créer des niveaux qui apportent verticalité et vertiges, changer d'altitude et proposer des expériences nouvelles avec le vent
- Exploration et découverte : Reproduire un quartier de Londres que le joueur peut visiter et découvrir ou redécouvrir d'une manière totalement nouvelle grâce au controller aérien
- Contemplation : Offrir au joueur des vistas uniques via les architectures et les couleurs pour le récompenser d'avoir maîtrisé le parapluie (controller qui devient un outils d'exploration maîtrisé)

Références LD

I Am Fish

Dans les Niveaux de I Am Fish, le joueur utilise le déplacement du poisson et les propriétés du bocal afin de rejoindre l'arrivé. La maîtrise et les connaissances les limites du bocal (hauteur de chute max, vitesse d'impacte max) sont nécessaires pour compléter les niveaux. Le joueur est challengé sur sa capacité à doser ses outils de déplacement. Dans parapluie, c'est un peu la même chose qui est recherché : Doser l'orientation du parapluie et gérer son nombre de flap pour pouvoir terminer les niveaux.



Ville de Londres

Une reproduction du quartier des affaires de Londres. Découvrir ses icones architecturales d'une manière inédite. L'utilisation de la tech BlenderOSM nous permet de pouvoir concerver les proportions et les échelles du quartier.

Références LD

Mario Odyssey

un grand niveau ouvert que le joueur peut explorer à sa guise. Il peut compléter le niveau par un chemin critique sans le finir à 100 %. Il peut aussi explorer le niveau de fond en comble pour trouver toutes les lunes (étoiles) et avoir tous les secrets, le 100 %. Chaque lune est un reward d'une situation spécifique de gameplay.

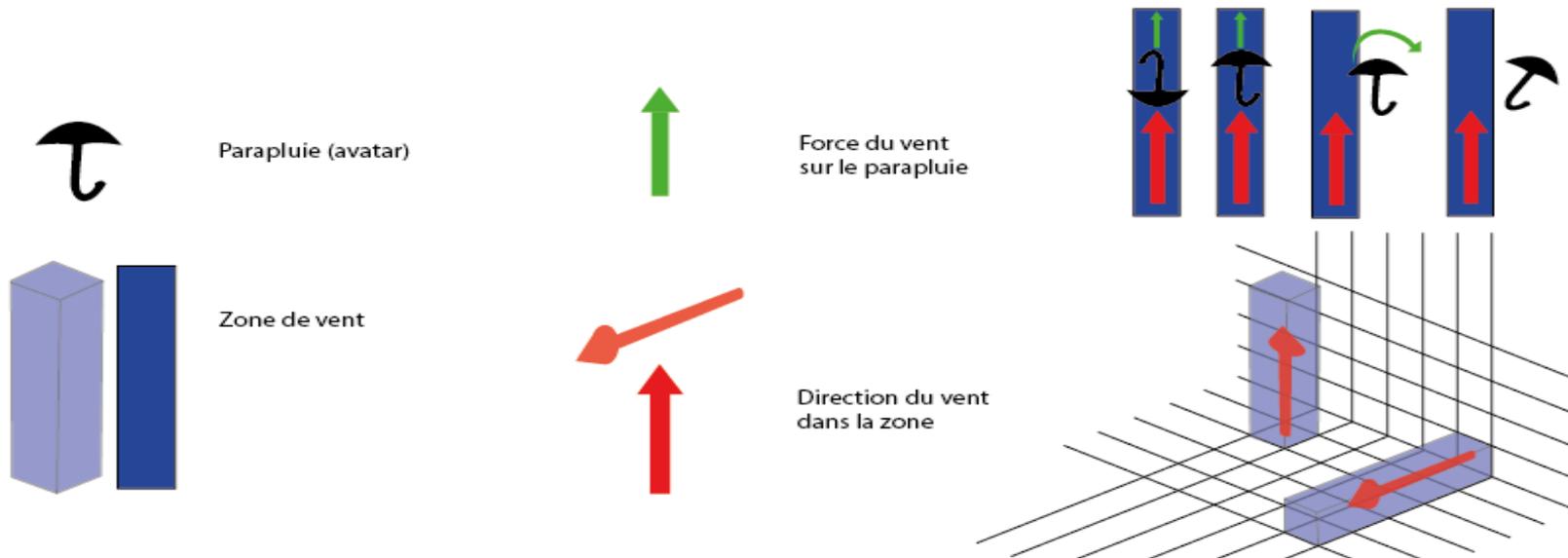


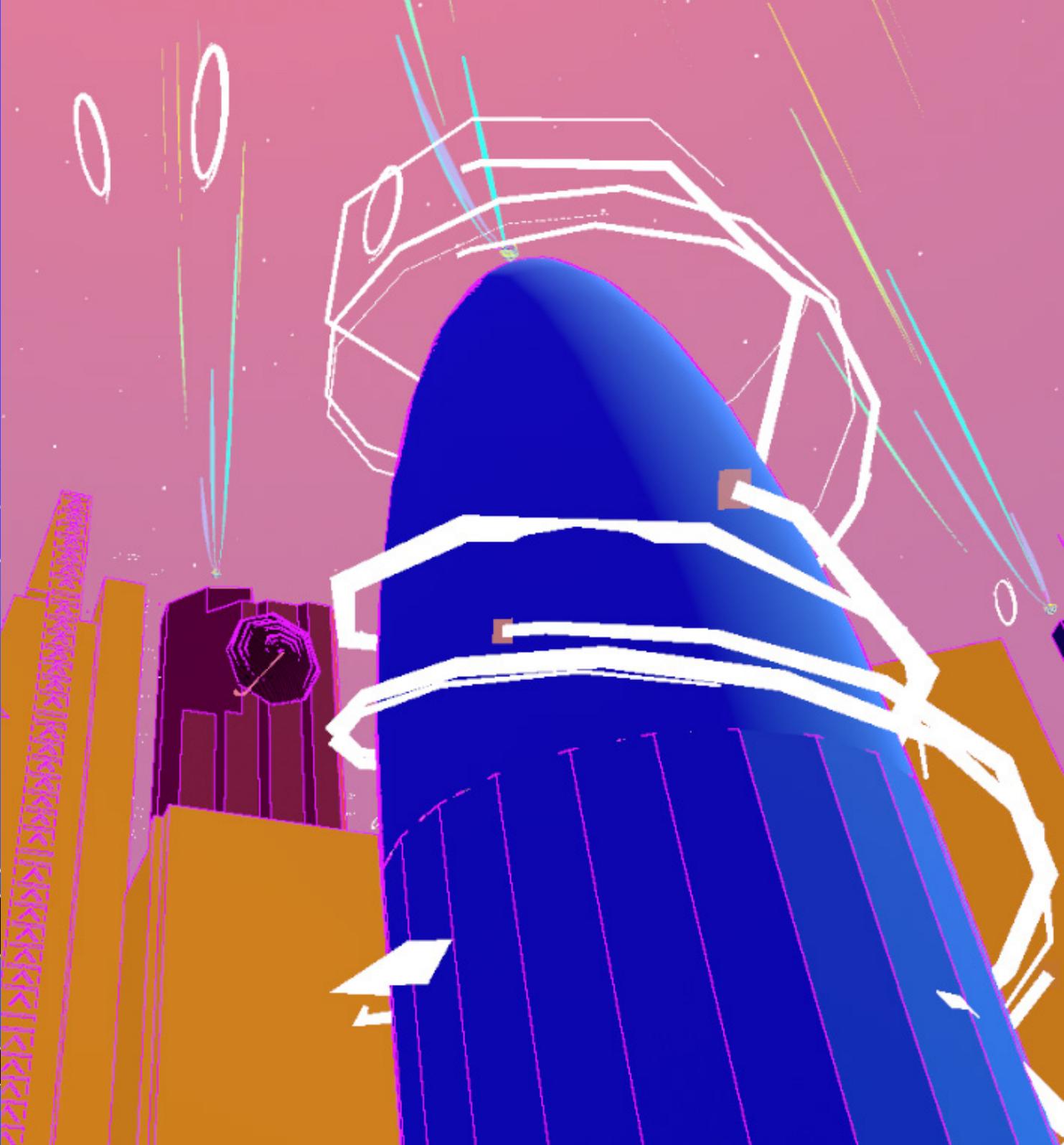
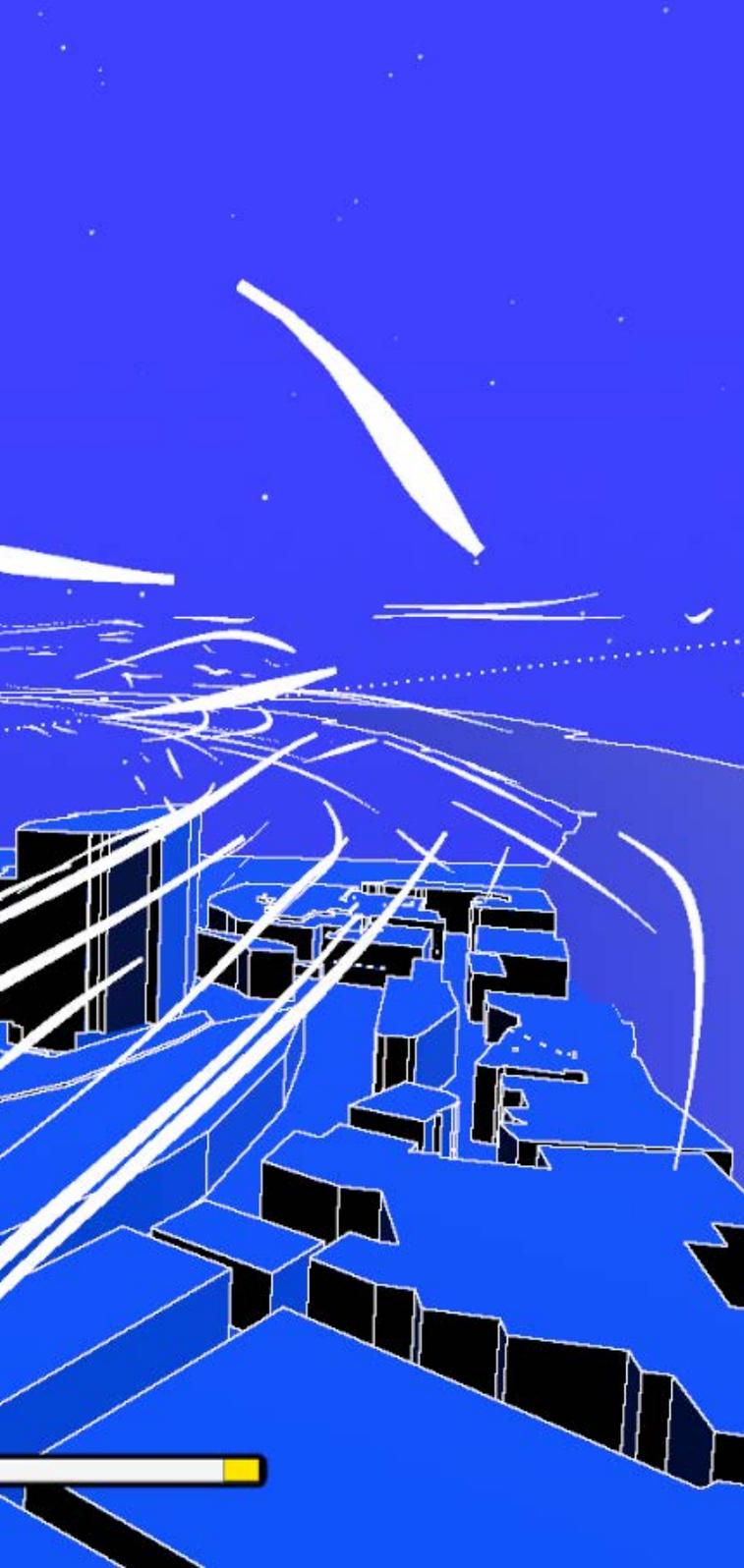
Ingrédient - le Vent

Comment fonctionne le vent ?

- Une zone qui fonctionne avec un Trigger. Dans cette Zone, une force est appliquée sur le Parapluie dans une direction avec une puissance donnée
- Le vent pousse le Parapluie et agit différemment en fonction de la partie du parapluie qui lui fait face :
 - si le vent vient de dos, le parapluie va accélérer et l'amplitude des flaps vers l'avant est amplifiée
 - si le vent vient de face, il ralentit le parapluie et l'amplitude des flaps vers l'avant est réduite
 - si le vent vient d'en haut, il pousse le parapluie vers le bas et l'amplitude des flaps vers le haut est réduite
 - si le vent vient d'en bas, il pousse le parapluie vers le haut et l'amplitude des flaps vers le haut est amplifiée

Variables du vent : taille de la zone, sens du vent, puissance du vent, angle du vent





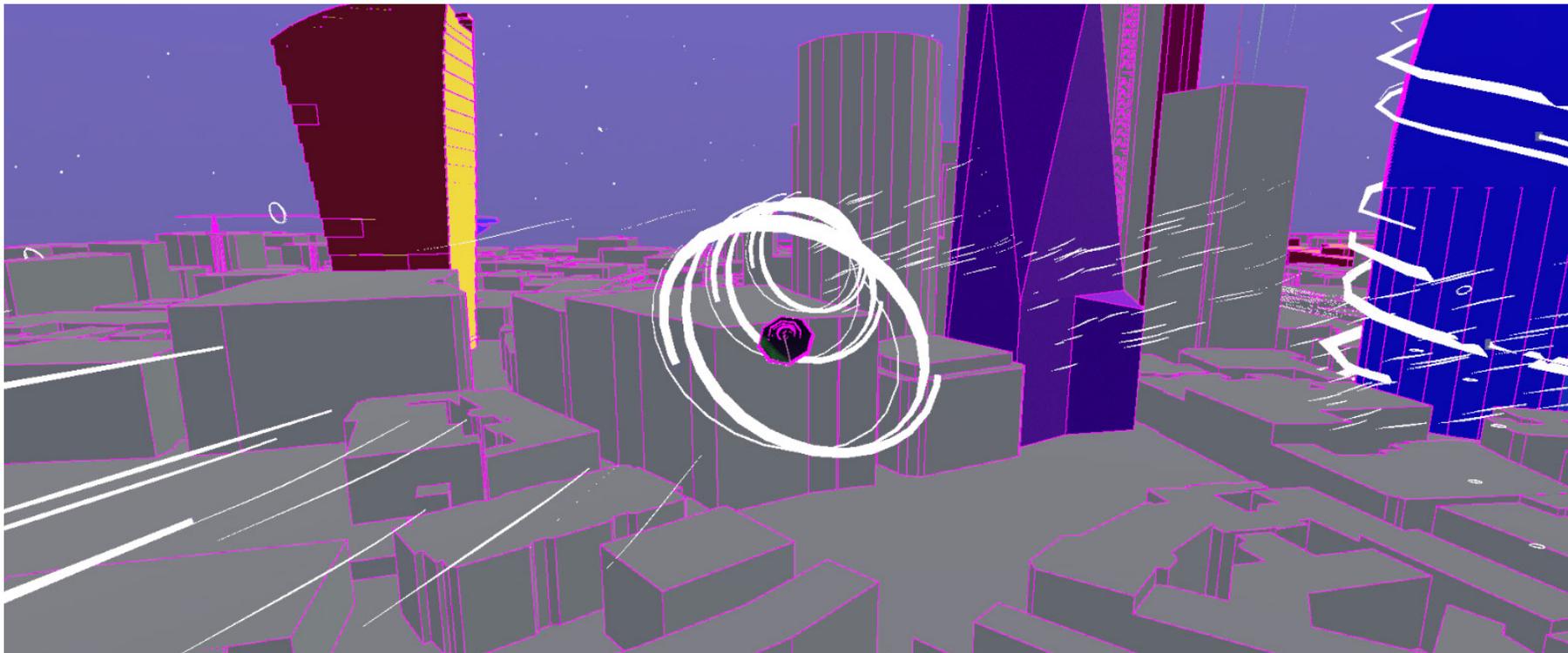
Ingrédient - le Vent

Différents types de vents

- Vent classique (une direction) : il pousse le parapluie dans la direction vers laquelle il souffle
- Recovery : un vent qui sert d'assurance au joueur lorsqu'il s'attaque à un atelier
- Barrier : un vent puissant qui repousse le joueur, permet de bloquer un chemin
- Death zone (vent violent) : dès que le parapluie y entre, il déclenche un game over. Le joueur est naturellement repoussé par les death zones mais il peut quand même forcer pour y aller.
- Bourrasque (une direction) : un vent qui souffle de manière alternée souffle --> pause --> souffle --> pause
- Tornade : un vent qui souffle de manière circulaire

Ingrédient - Bonus de flap

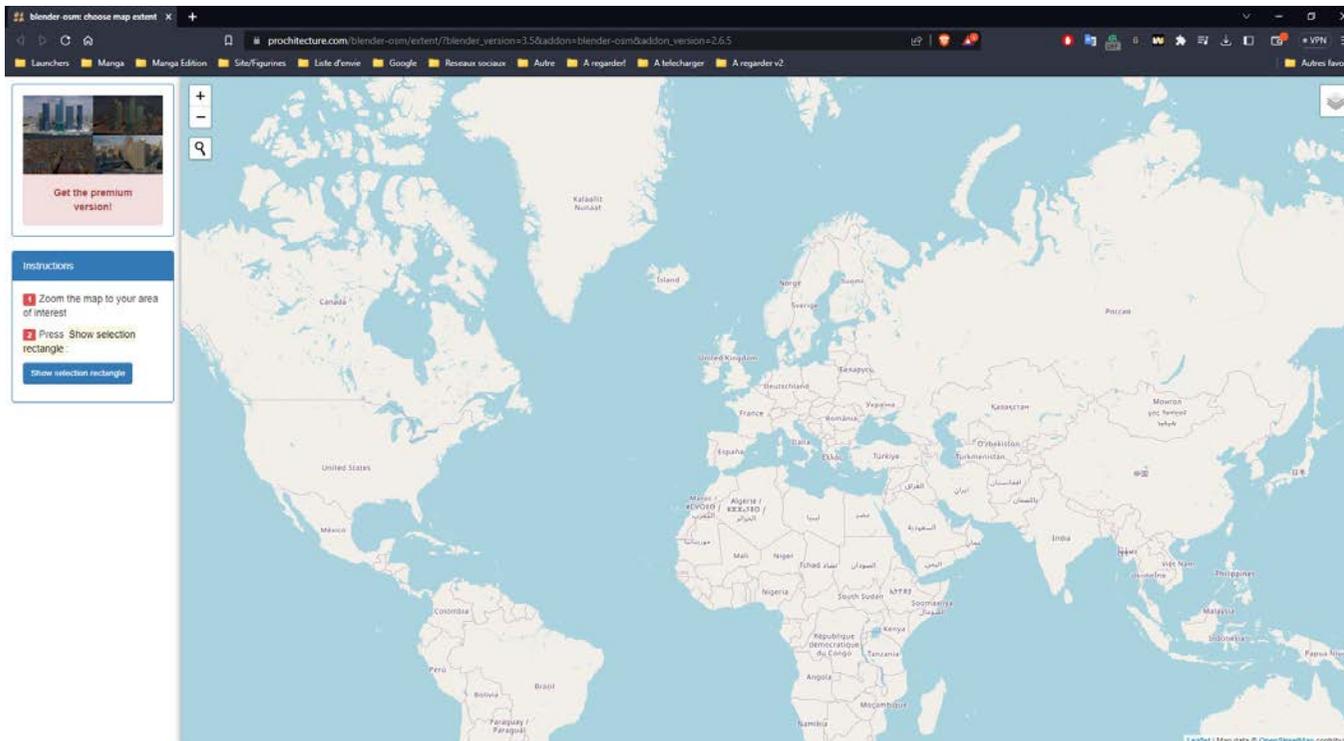
2. Des obstacles mouvants: des formes (sphères, cubes, pavés,etc....) qui se déplacent automatiquement de manière continue (avec ou sans bounce) entre plusieurs points, qui parfois bloquent la sortie / l'accès à la suite du niveau. Pour le déplacement, sachant que l'orientation et la stabilisation sont nécessaires à celui-ci, des obstacles mouvants peuvent être pertinents pour combiner la mesure du flap avec du timing et de l'anticipation, en installant de la pression en réduisant les fenêtres d'opportunités par moment et en punissant une mauvaise maîtrise de ces micro-challenges (mesure, timing, précision, anticipation).



Blender OSM

Gain et temps et fidélité avec la réalité

Nous avons pour intention de permettre au joueur d'explorer et de découvrir le quartier des affaires de Londres. Ce quartier étant gigantesque, il était impossible de concerver les proportions et les échelles des rues et des batiments en modélisant et plaçant ces batiments un par un nous même. Nous avons donc utilisé la technologie BlenderOSM. Cette technologie nous permet de fournir à blender des données OSM (Open Street Map) pour qu'il nous génère la topologie du lieu. Cela nous permet d'avoir rapidement une map modélisée avec un niveau de détail suffisant. Nous retravaillons ensuite la map pour y implémenter certains bâtiments que nous faisons nous même pour les landmarks par exemple.



Blender OSM

The image illustrates the Blender OSM workflow in three stages:

- Map Selection:** A 2D map of London with a selection rectangle. A sidebar on the left provides instructions: "Zoom the map to your area of interest", "Press Show selection rectangle", "Drag the rectangle to select your area of interest", and "Press Copy to copy the string with coordinates to the clipboard". A "Copy" button is visible.
- Coordinate Extraction:** A dark overlay window titled "blender-osm" displays the "Extent:" coordinates: 51.5151, -0.1323, -0.0796, and 51.4766. It includes buttons for "select", "paste", "from a...", "OpenStre...", and "import".
- 3D Model Rendering:** A 3D perspective view of the city model. The Blender interface shows the "blender-osm" add-on settings, including "Import buildings", "Import water objects", "Import forests", "Import other vegetation", "Import roads and paths", and "Import railways". The "Scene Collection" panel on the right shows the loaded OSM data.

Landmarks Naturels

Un fit parfait pour le gameplay de Parapluie

Le quartier des affaires de Londres est composé de multiples tours très élevées aux formes uniques entourées de bâtiments plus austère et plus bas. Cela dégage naturellement des Landmarks visible de très loin pour le joueur et nous permet de très facilement manipuler sa progression dans le niveau.



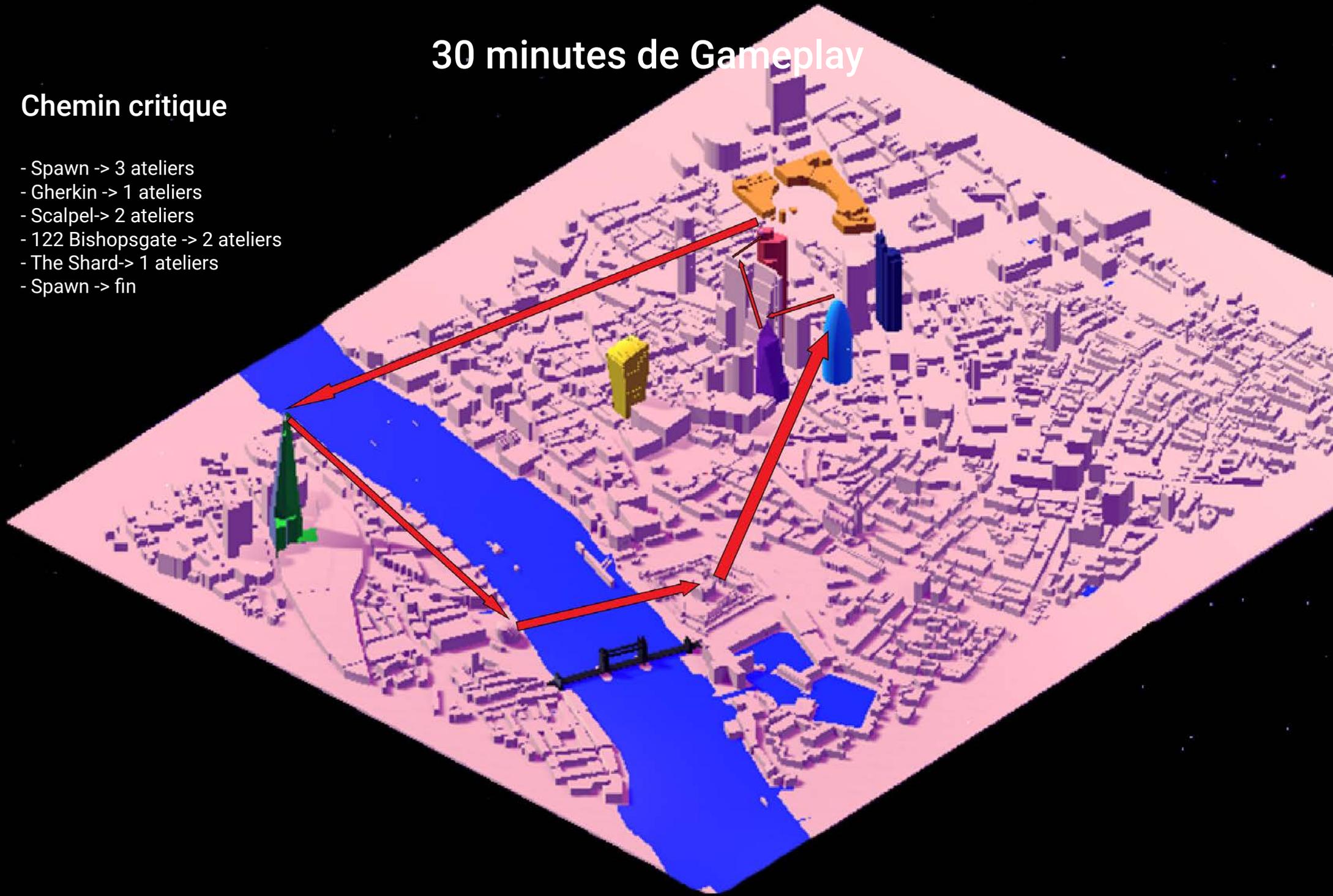




30 minutes de Gameplay

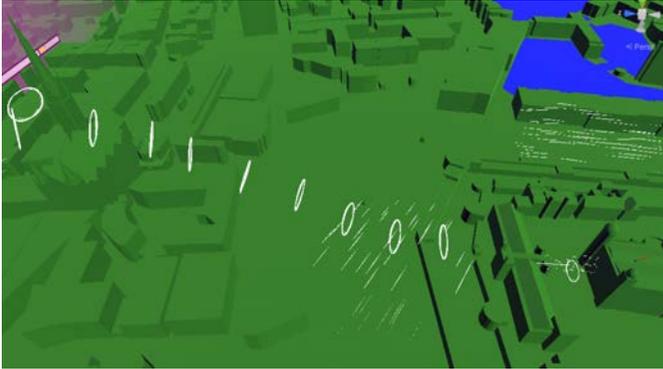
Chemin critique

- Spawn -> 3 ateliers
- Gherkin -> 1 ateliers
- Scalpel -> 2 ateliers
- 122 Bishopsgate -> 2 ateliers
- The Shard -> 1 ateliers
- Spawn -> fin

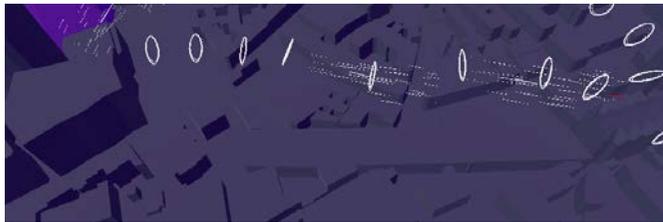


30 minutes de Gameplay

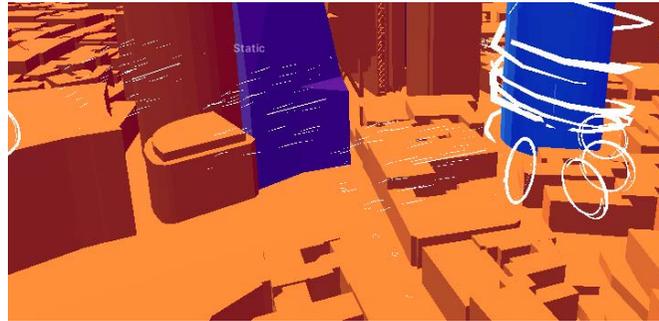
1 : Spawn



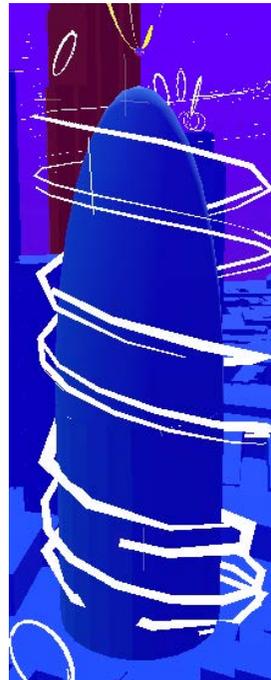
début composé de vent et de bonus, chemin plutôt facile, et échouer nous ramène dans un chemin autre du chemin critique.



Suite en hauteur, tomber force à revenir en hauteur pour suivre le chemin, le prochain endroit de pause est visible et proche.

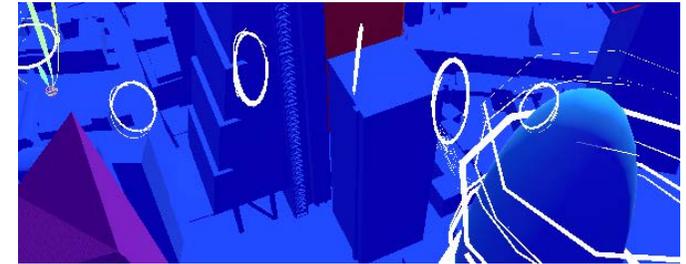


Escalier de vent, proche du sol et du gherkin : premier landmark. il est difficile d'échouer, l'objectif est d'atteindre ce premier landmark facilement car il rejoint tous les autres.



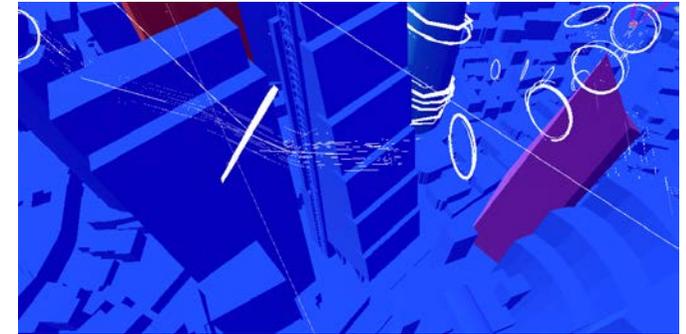
2 : Gherkin

Premier vent en tornade, il suffit juste de se laisser s'élever. En tournant autour du bâtiment, on perçoit tout ce qu'il y a autour, et en arrivant en haut, on atteint la première étoile.



Chemin de bonus de flaps pour faire l'aller-retour avec le Scalpel.

3 : Scalpel

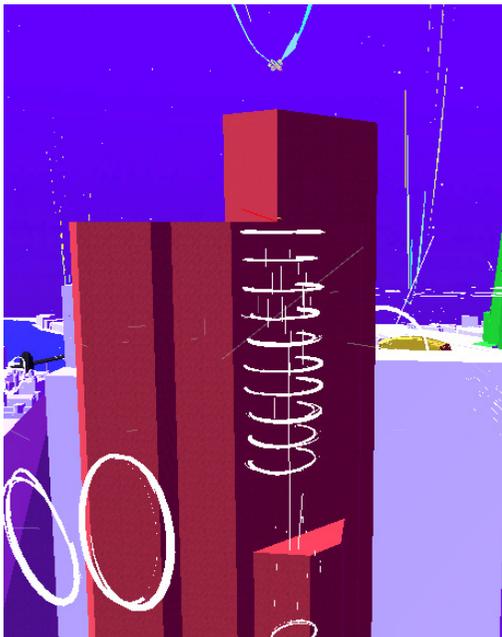


chemin pour se rapprocher derrière la 122 bishopsgate, composé d'un vent qui repose le joueur à la fin pour le récompenser de tout le dur chemin à accomplir, et lui indiquer la dernière ligne droite avant le plus haut bâtiment du quartier.

30 minutes de Gameplay

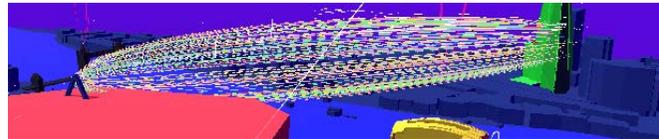


Chemin pour passer sur l'escalier du 122 Bishopsgate, avec une forte punition en cas d'échec (recommencer à partir du scalpel), composé pratiquement de bonus flap.



4 : 122 Bishopsgate

traverser le vent qui nous pousse vers le bas à l'aide des bonus flap, il faut donc spammer les flaps, une fois arrivé, étoile garantie sur le plus haut point du quartier.

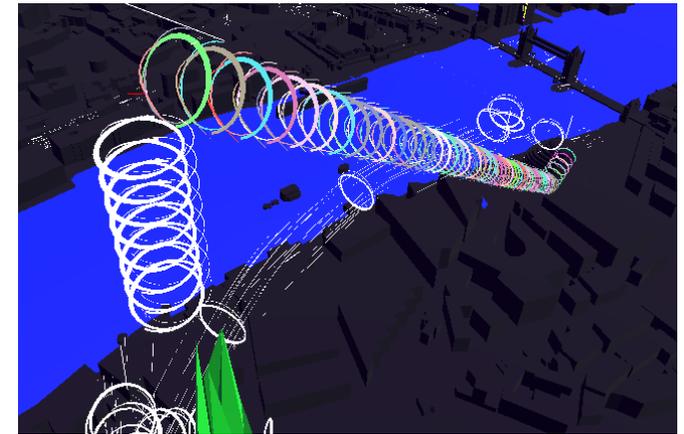


tyrolienne pour aller en bas de The shard.

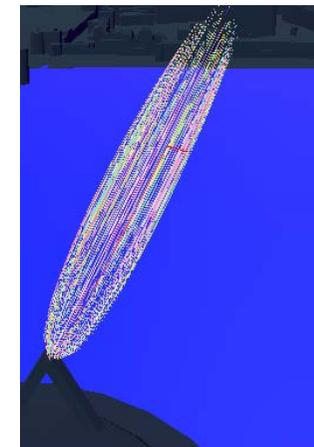


5 : The Shard

Plus haut bâtiment de Londres, on doit repartir du bas pour gravir ce building emblématique, tout en offrant un chemin difficile et un chemin facile mais plus long. Avec un peu d'aide en et une maîtrise de la verticalité, la 4ème étoile du chemin critique et le déblocage de la fin nous est offerte.



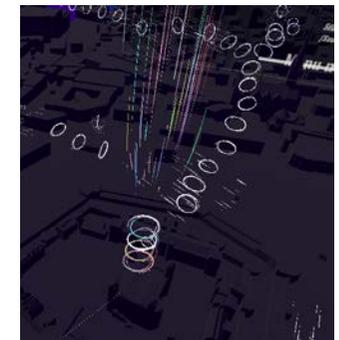
Chemin coloré apparaît, sans difficulté, le chemin se dessine pour retourner au Spawn.



6 : FIN

S'envoler au dessus du Spawn.

Tyrolienne pour retourner au spawn en traversant la tamise..



Progression et Apprentissage

Input joueur	Orientation	<u>Flapflap</u>	Fermeture
<u>Gameplay element</u>	Mur	vent	

<u>Gameplay element</u>	Mur	Vent contraire
Micro challenge	Précision, timing, mesure, anticipation, navigation	Précision, navigation, mesure, observation

Progression et Apprentissage

mur				
challenge	<u>Tease</u>	<u>Learn</u>	<u>practice</u>	master
<u>Precision</u>	Mur dans le champs de vision du joueur	Mur en limite de l'atelier	Mur en tant que obstacle	Trou dans le mur ou le joueur dois passé
timing	Mur en limite de l'atelier	virage	Slalome	Mur mouvant
mesure	Mur dans le champs de vision du joueur	Mur en tant que obstacle	Slalome	Mur mouvant
anticipation	Mur en limite de l'atelier	Passer au dessus d'un mur	slalome	Mur mouvant
navigation	Mur en limite de l'atelier	virage	slalome	Mur mouvant

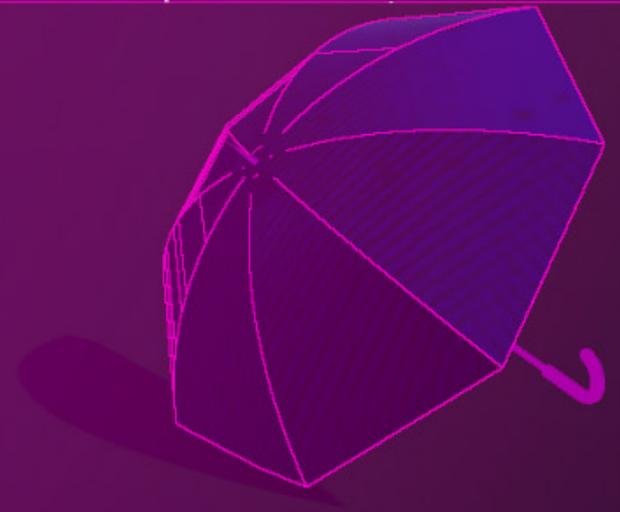
Progression et Apprentissage

Vent contraire				
challenge	<u>Tease</u>	<u>Learn</u>	<u>Practice</u>	Master
précision	<u>Appercevoir</u> la manche a air	Vent doux	Vent fort	Alterner les direction des vent fort
Navigation	Voir les <u>element</u> du niveau	<u>Legère perturbation</u>	Moyenne perturbation	Forte perturbation
Mesure	Manche a air	Légère perturbation	Moyenne perturbation	Moyenne perturbation <u>irégulière</u>



Programmation

- p76 - Mouvement Parapluie
- p78 - Camera Collision Wall
- p80 - Bonus de flap
- p81 - Vent
- p84 - Tornade
- p85 - Trigger Musique
- p86 - Etoiles



Programmation

Mouvement parapluie

- Le parapluie se déplace avec des impulsions en fonction de l'orientation de sa pointe.

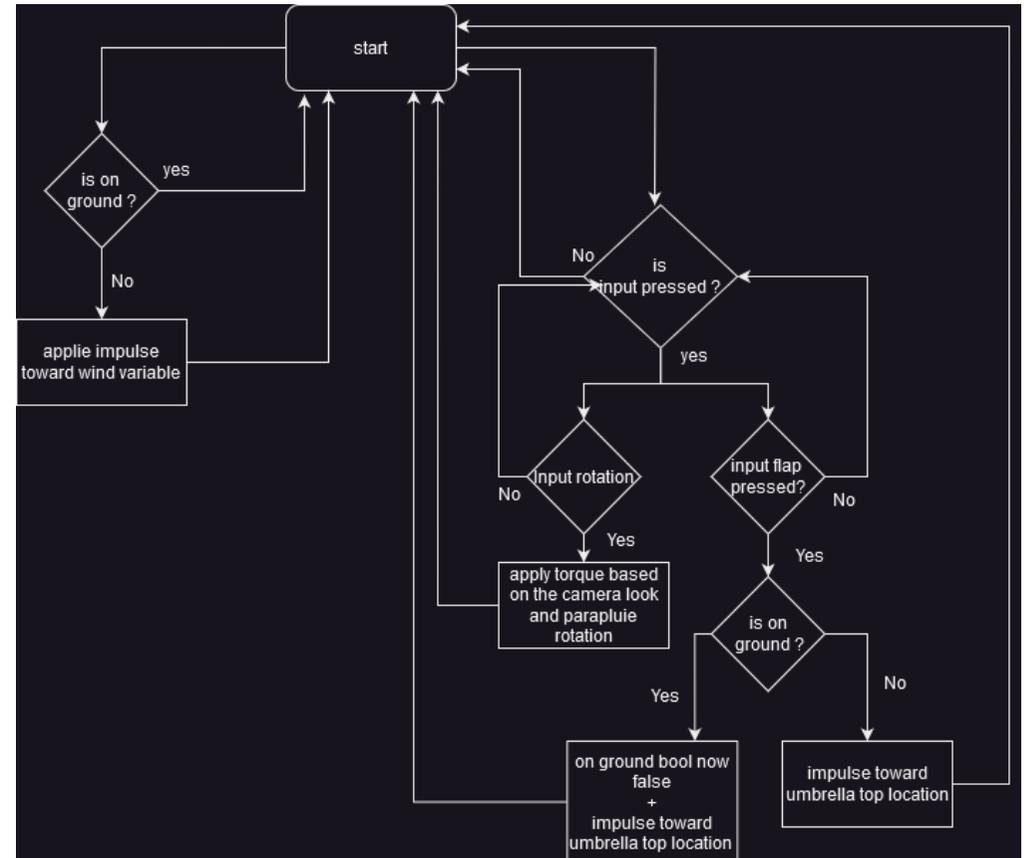
Pour l'orienter le parapluie reçoit un torque en fonction de l'orientation de la camera et des input utiliser (orientation joystick).

- Le joueur reçoit une impulsion continue en fonction des metrics, cette impulsion depend des metrics par default et de si le joueur est dans un trigger de vent.

- pour bloquer le joueur a la meme position lorsqu'il est au sol, on met sa dernière coordonné au moment ou il touche le sol dans une variable et met le boolean de blocage sur true. on assigne cette variable au transforme du parapluie tant que le boolean de blocage est true. lorsque le joueur flap, le boolean redeviens false. le blocage au sol desactive aussi toute influence du vent pour eviter d'avoir des bugs.

ont utilisait un tween a la base pour les rotation mais cela créait des probleme de mouvement qui se bloquait a certain axe, ont a donc choisi un torque pour faire cela car il sagit d'une impulsion que l'ont donne dans une direction qui du coup n'est pas sujet au blocage de si le parapluie est dans un certain angle.

le blocage au sol a été rajouter pour pouvoir mieux metriser le parapluie lorsque celui ci est au sol sinon celui ci retombe a chaque fois et il etait difficile de sortir de la position au sol

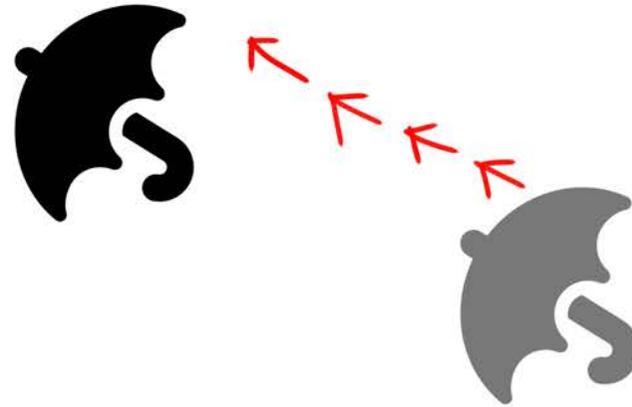


Programmation

Tween



Flap



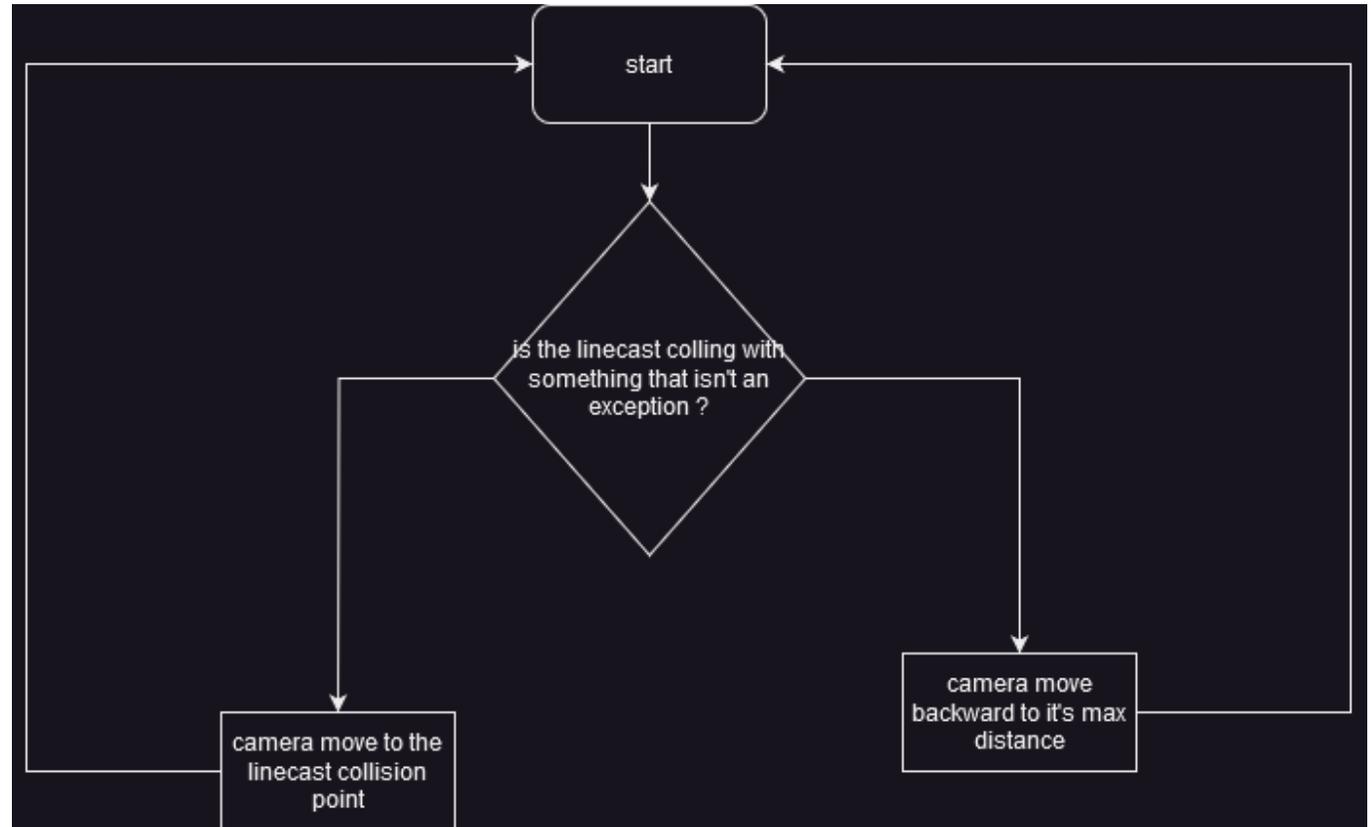
Torque



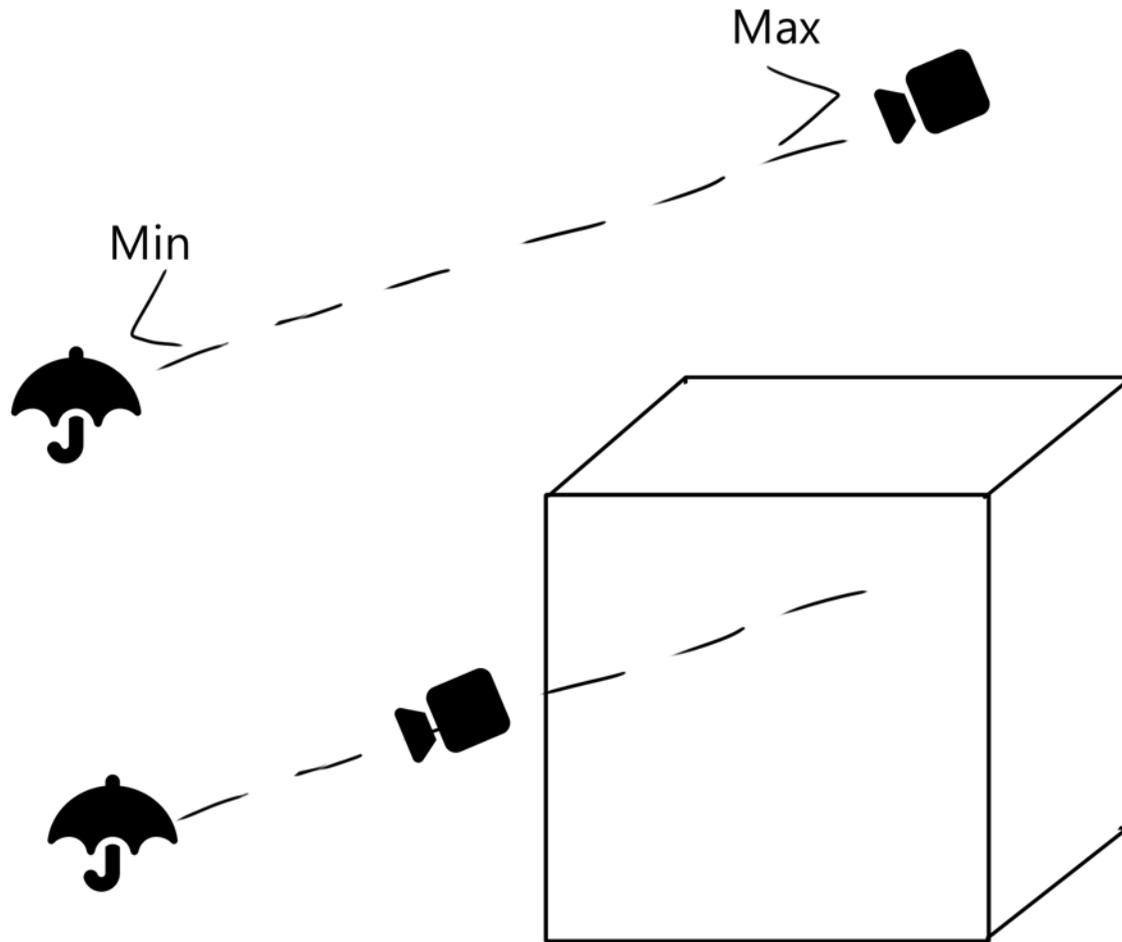
Programmation

Camera Collision Wall

- Le jeu verifie avec un line cast si entre le joueur et la camera il y a une collision qui ne fait pas partie des exception qui sont determiner par un tag.
- Si le linecast n'a pas de collision, il va reculer dans la direction oposé au parapluie jusqu'a atteindre le clamp.
- Si le linecast entre en collision avec un objet, la camera va se deplacer vers la collision en se metant un peu en avant pour ne pas rentrer a l'interieur.



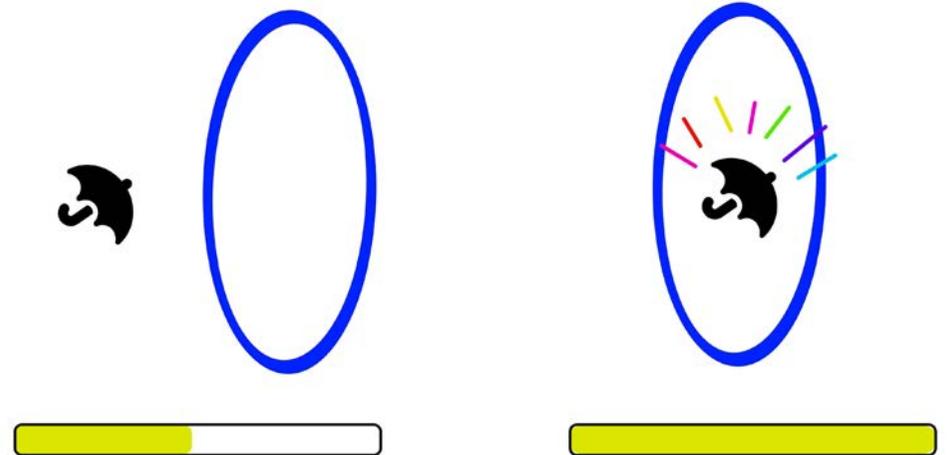
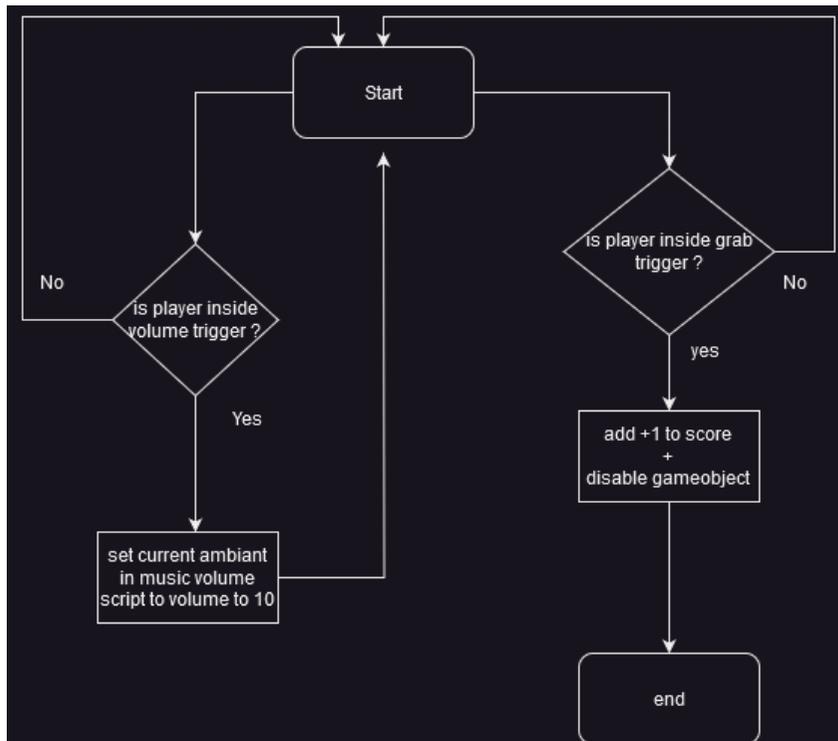
Programmation



Programmation

Bonus de flap

- lorsque le joueur rentre dans le trigger:
 - sa jauge de flap est remis a sa valeur complete
 - les particule du joueur s'active pour x sec
 - le collider et les particules sont desactiver pendant y seconde, apres ils se réactive.



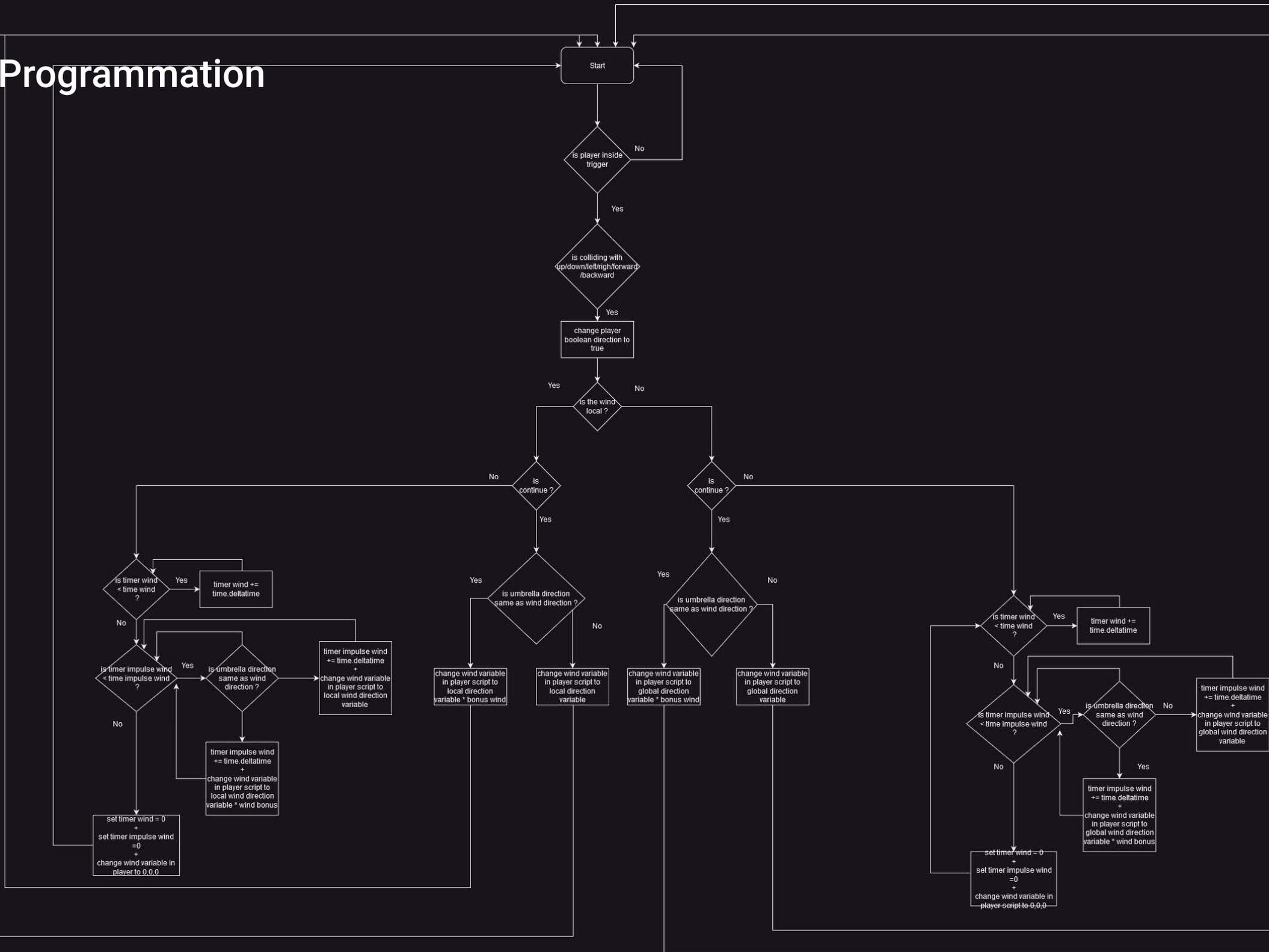
Programmation

Vent

- Il check si le vent est local ou global
- Il check si il est continue ou impulsion
 - Si le vent est continue :
 - Si le vent et local il change les variable de l'impulsion du parapluie de façon continue dans la direction indiqué par les metrics en fonction de l'orientation du trigger
 - Si le vent est global il change la variable de l'impulsion du parapluie de façon continue dans la direction indiqué par les metrics et de l'axe X,Y,Z du monde.
 - Si le vent est en impulsion
 - Si le vent est local, il change la variable d'impulsion du parapluie toutes les X second qui dure Y seconde en fonction des metrics et de l'orientation du trigger avant de revenir a la valeur par défaut
 - Si le vent est global il change la variable d'impulsion du parapluie toutes les X second qui dure Y seconde en fonction des metrics et de l'axe X,Y,Z du monde avant de revenir a la valeur par défaut
 - L'impulsion reçoit un boost si le joueur est orienter dans la direction du vent et un maluce si il est orienter dans la direction oposé

ont a choisi de mettre la variable d'impulsion sur le joueur afin de permettre de créer des situation ou le vent est deja present pour le joueur et les trigger influence ce vent. nous n'avons pas utilisé cette mecanique mais mettre la valeur sur le joueur plutot que le script de vent a été quand meme utile pour créer les tornade.

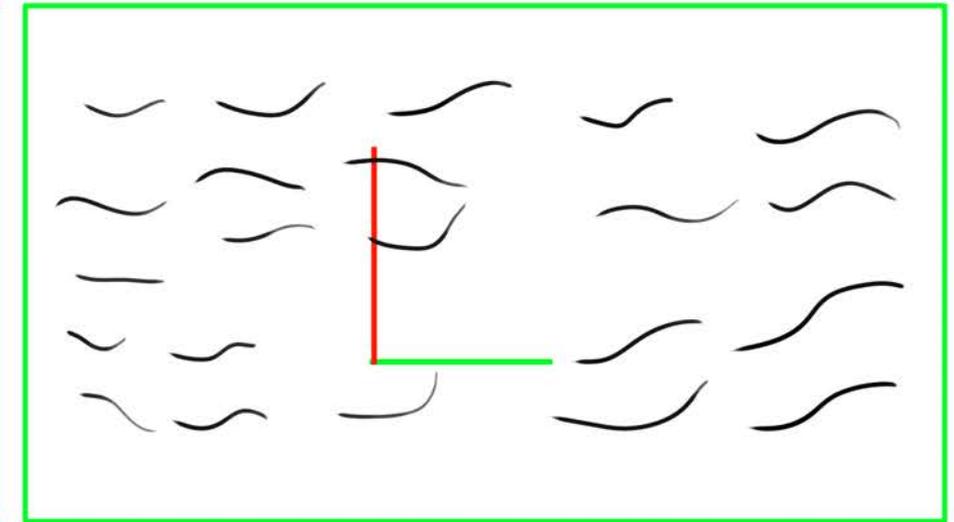
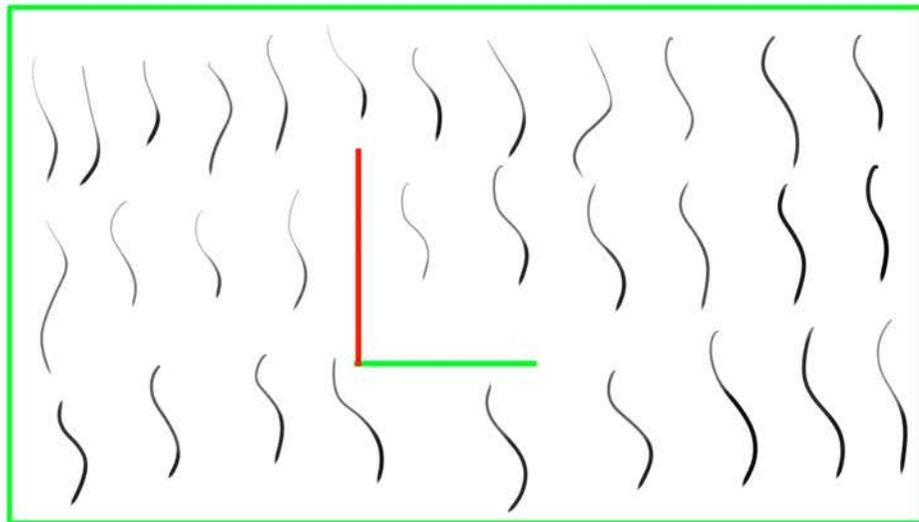
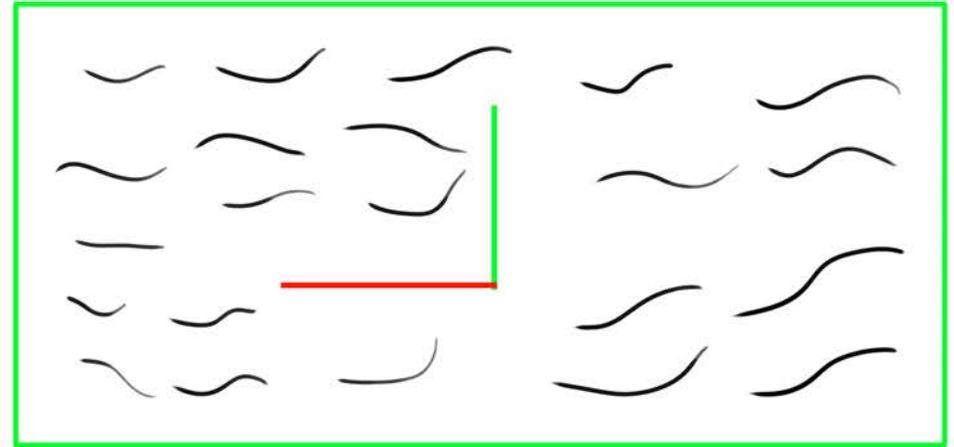
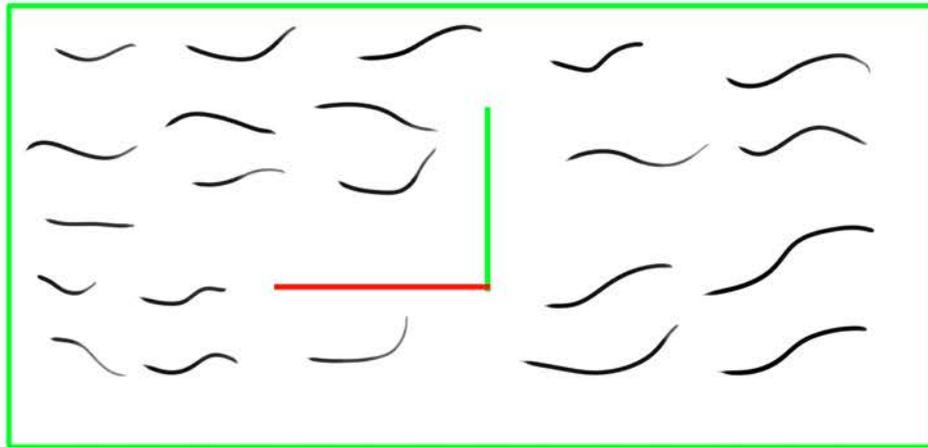
Programmation



Vent local

Programmation

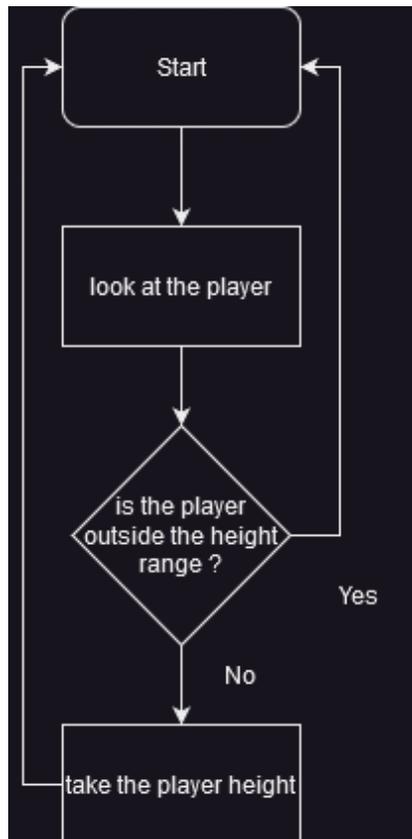
Vent global



Programmation

Tornade

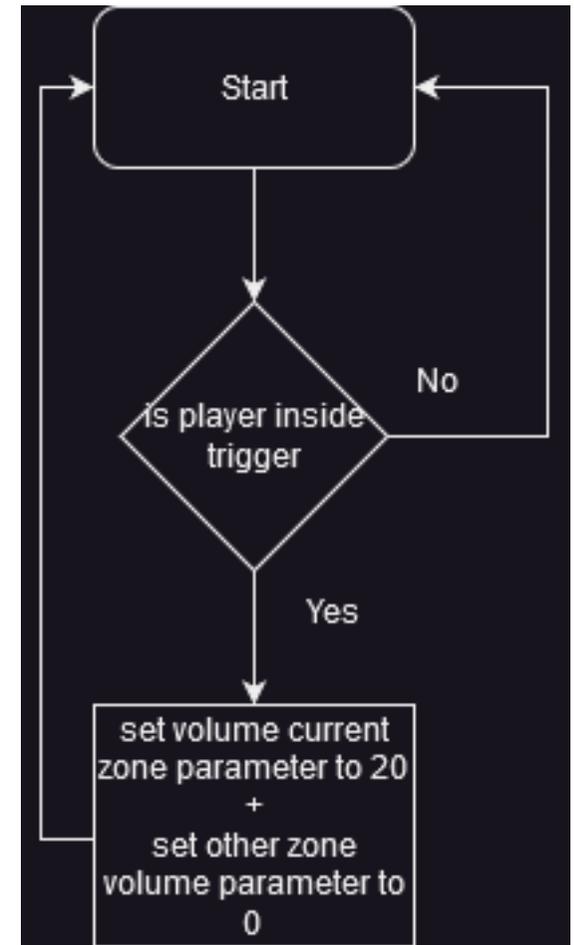
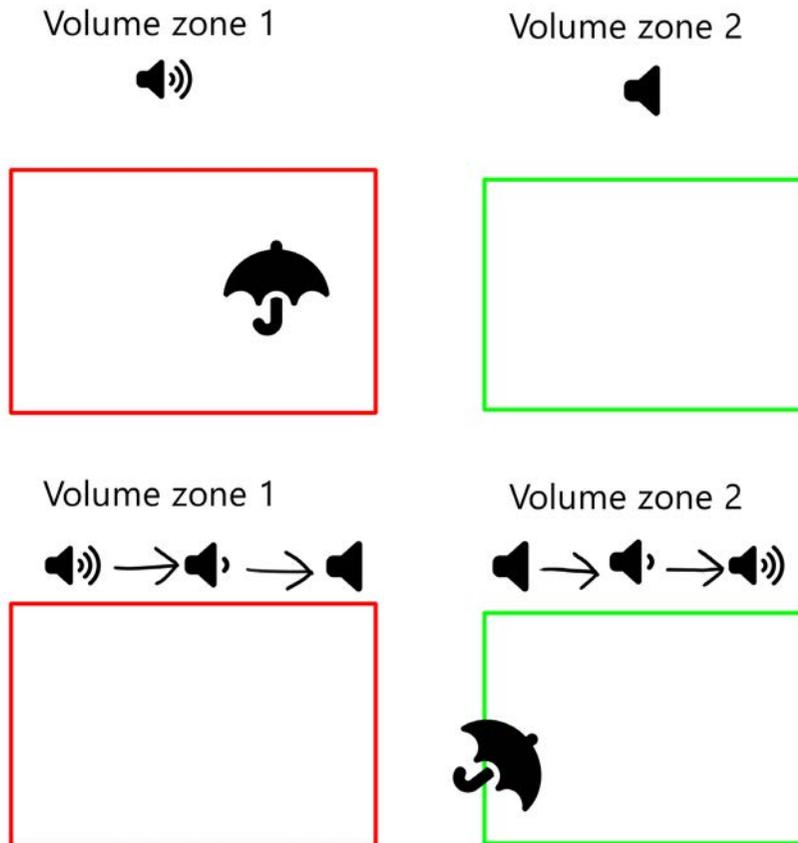
- le game object sera toujours orienté en direction du joueur grace a la fonction lookAt de unity.
- le game object va prendre la hauteur du joueur pour sa propre hauteur mais est limité a une valeur.
- pour déplacé le joueur il utilise un vent local.



Programmation

Trigger Musique

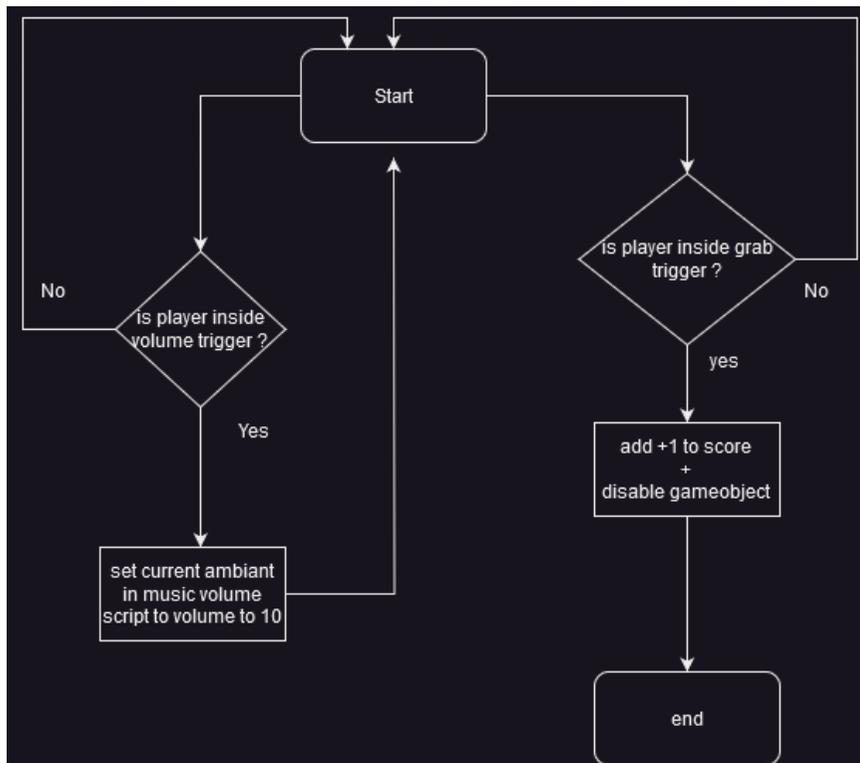
-lorsque le joueur entre dans un trigger musique, cela change progressivement le volume de la musique designer au trigger avec un lerp jusqu'a la valeur audio max
- cela reduit aussi le volume des autre musique progressivement a 0



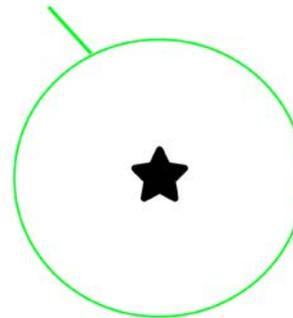
Programmation

Etoiles

- si le joueur est dans le trigger music, cela baisse la musique ambiante de moitié pour laisser le joueur entendre la musique du bonus.
- si le joueur entre dans le trigger de grab, cela ajoute +1 au score et disable l'objet



trigger volume



Volume bonus



Volume musique



Volume bonus



Volume musique

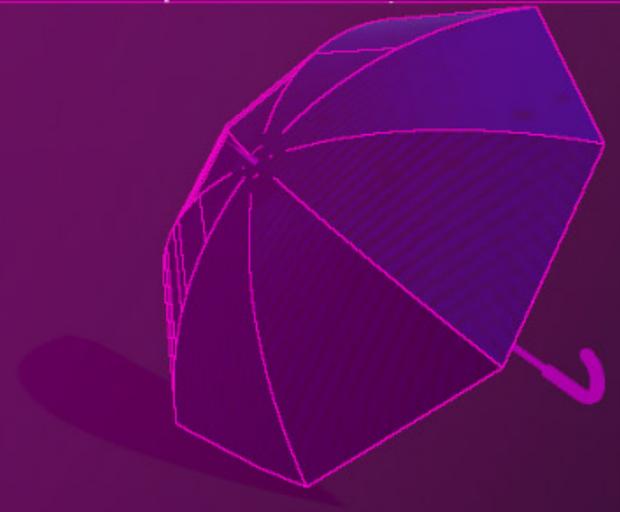




PARAPLUIE

= Direction Artistique Visuelle :

- p88 - Intentions
- p89 - Visuels
- p91 - Signes & Feedbacks
- p92 - Techniques Visuels
- p93 - Références



Intentions

Représentation poétique, ambiances marquées et codifiées

Pour aller avec ce controller atypique, nous avons pris comme décision de pousser visuellement (et auditivement) une ligne simple et alégorique.

L'esthétique est unique, mais garde les harmonies de la ville de londre, ainsi que les circulatoires des routes et des bâtiments pour remplacer l'image commune de londre, et la séparer en plusieurs zones.

L'utilisation des couleurs est devenu le pilier de la direction, et la codification un moyen de faire les rapprochement, de forcer la mémoire de Londres à chaque instant au joueur.



Visuels

Direction poétique assumée, couleurs unies et outlines marquées

Tout le jeu est formé principalement par 3 couleurs qui changent en fonction de la zone dans laquelle l'avatar se situe.

Une couleur pour les murs, le sol, les bâtiments.

Une couleur pour les outlines dessinés par les segments des géométries.

Une couleur pour le parapluie.

Le but est de rendre un aspect poétique, d'utiliser qu'un seul shader pour le jeu afin de rendre une esthétique unique, qui cohésionne avec le controller.

Le ciel et les particules visuelles de vent sont bleue et blanche, et tendant à 60% vers la couleur des bâtiments.

Certains bâtiments importants de la ville de Londres utilisés comme landmarks, ont une propre couleur.

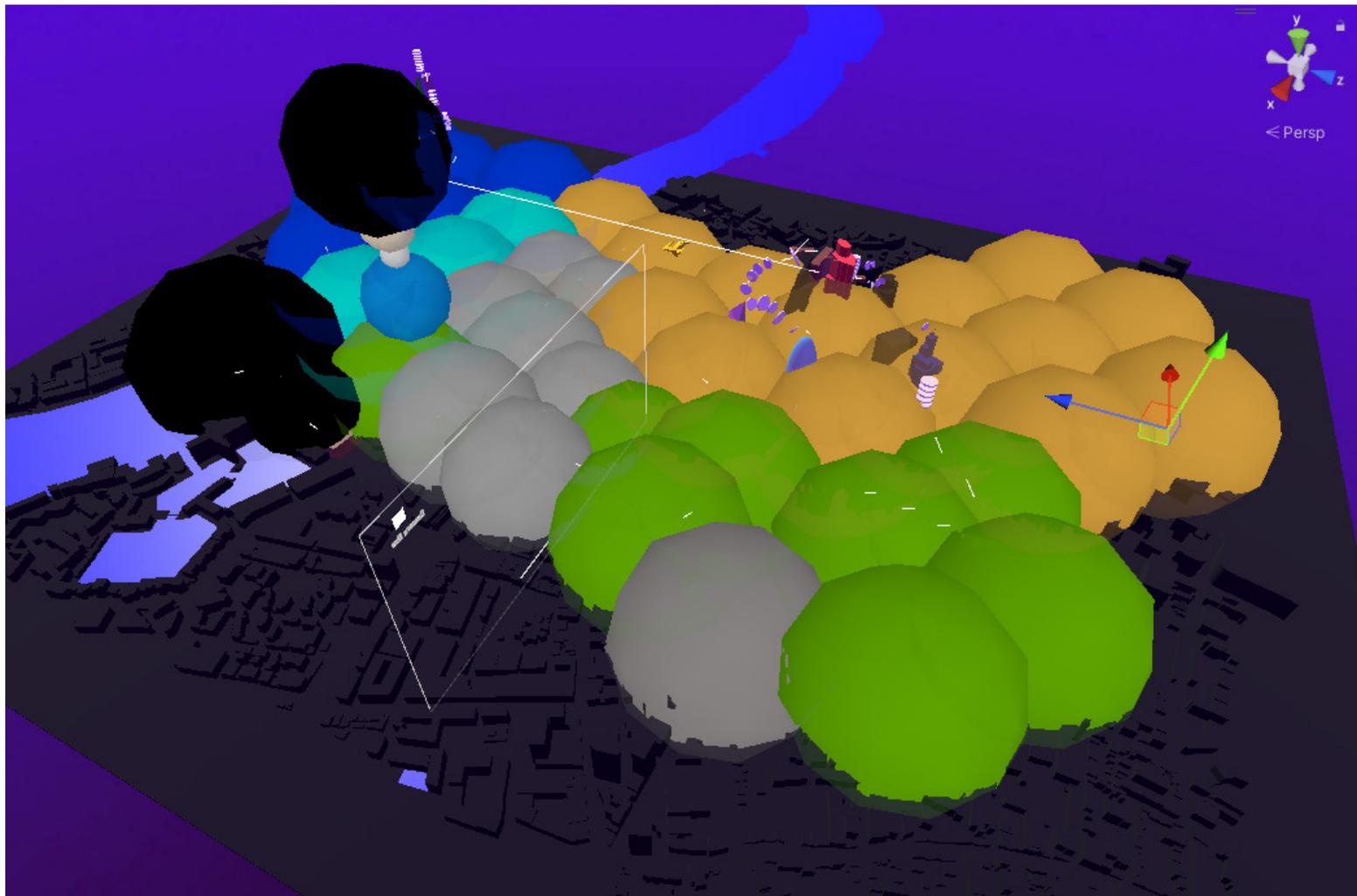


Visuels

Les zones de couleurs

Les 3 couleurs principales changent en fonction de la position de l'avatar dans la carte,

Chaque zones représentent une ambiance, un environnement, et surtout un type de gameplay lié à cet environnement.



Signes & Feedbacks

L'aide au joueur

Un grand défi dans Parapluie à été d'aider le joueur dans l'afordance et tout ce qui sert au gameplay. Le controller étant difficile nous devons être irréprochable sur le fonctionnement des mécaniques et des ingrédients. En plus du game feel satisfaisant de réussir les challenges et objectifs.

Perfect Flap

La barre d'énergie s'adapte au flap, une zone se situe au timing du perfect sur la barre d'UI, en plus d'être complétée intra dégiétiquement avec un anneau qui se referme sur le parapluie, pour aussi iniduer ce timing.

des particules de vents sont propulsés par le parapluie quand le joueur effectue un perfect, ces particules sont coloré, plus grandes et rapides lors d'un perfect méga flap.

Bonus Flap

Un anneau assez large et généreux en hitbox, on doit comprendre qu'il est là pour aider le joueur et tracer le chemin.

Le Feedback est coloré, et fait bien comprendre son utilité.

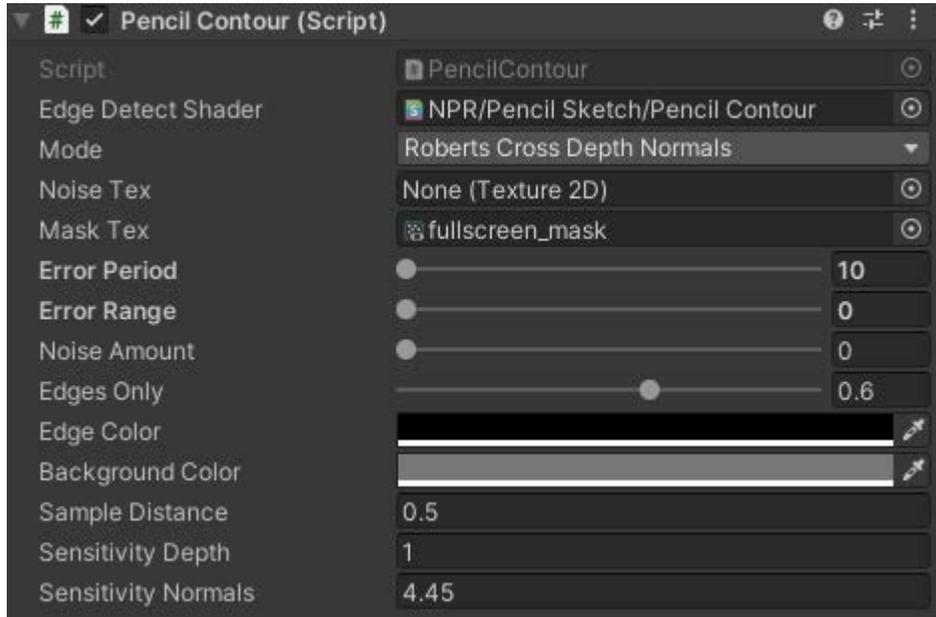
Etoiles & skyline

Les étoiles sont utiles et surtout des récompenses pour le joueur, elles sont marqués visible de loin et surtout repérable facilement, le chemin à parcourir est donc claire avec les signes des autres ingrédients.

La skyline se dessine avec la couleur des landmarks, on comprend rapidement le rapprochement, et l'utilité qui est de voir la complétion du niveau.



Visuels - Technique



Exception:

Il est important de réduire les problèmes de visibilité, donc de ne pas confondre les landmarks, le parapluie et la zone de jeu.

Nous avons créé un mask permettant de retranscrire en noir et blanc les objets visibles : noir le parapluie et landmarks, blanc le reste. Le post process agit sur l'image dessinée en blanc.

Avec cela les landmarks ne changent jamais de couleurs, et le parapluie qui change de couleur selon la zone sur la carte, ne sera pas influencé par Edge Only.

Post-Process

La caméra utilise un shader avec ces paramètres :

OUTLINE : les outlines sont plusieurs lignes dessinés avec plus ou moins d'épaisseur, avec un paterne régulière.

Sample Distance : dessin des outlines plus ou moins épaisses

Error Period : dessin des outlines plus ou moins régulières

Error Range : épaisseur du paterne des outlines

COULEUR :

Edge Color : couleur des outlines

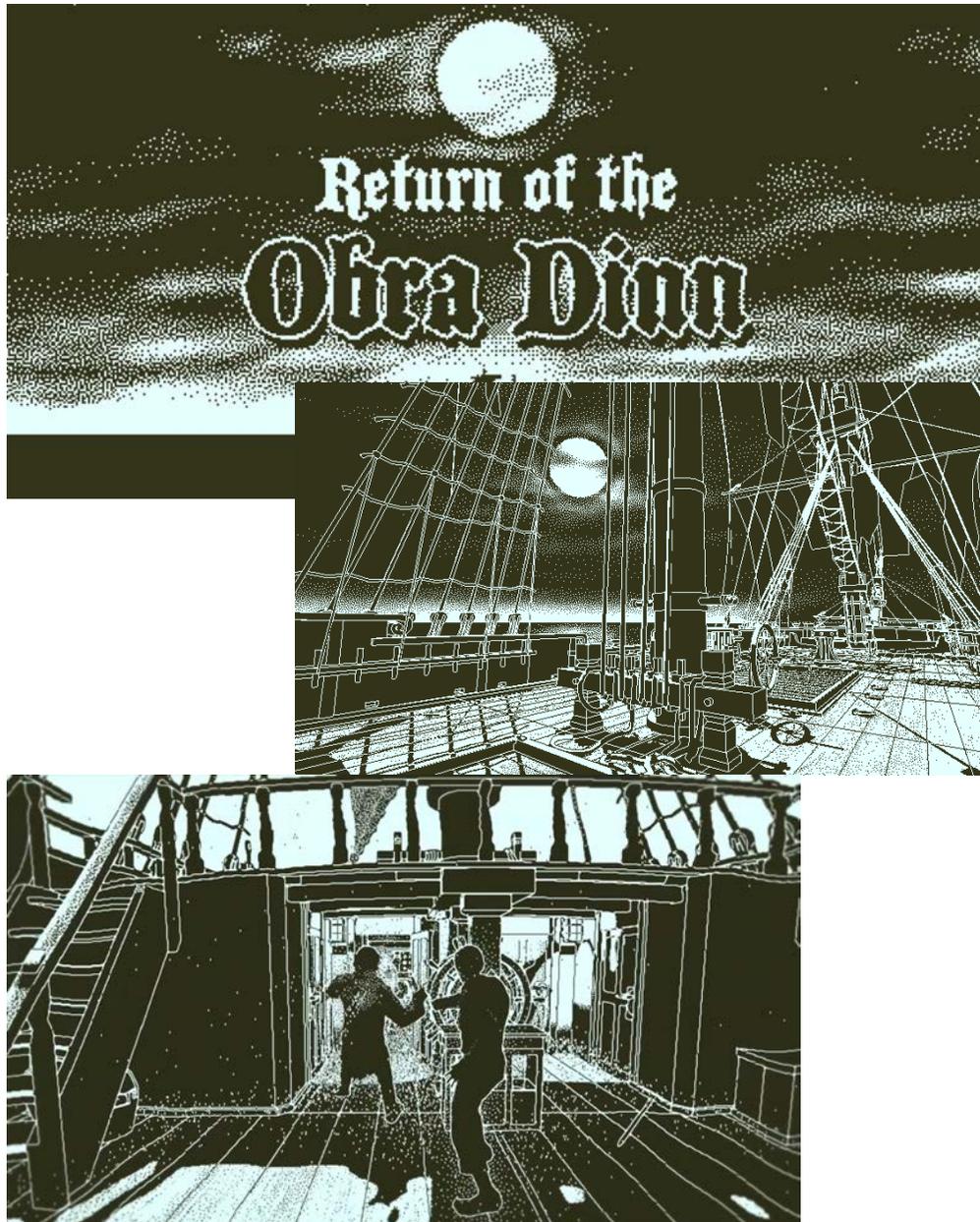
Background Color : couleur unie

Edge Only : Coefficient (entre 0 et 1) sur lequel les couleurs à l'écran se rapprochent à Background Color

Le material des bâtiments dans le jeu prend la même couleur que Background color.

0.6 est le paramètre optimal, il permet, d'accentuer les couleurs et les ombres des bâtiments et de bien changer les couleurs de la skybox sans donner le même teint unis des bâtiments.

Références



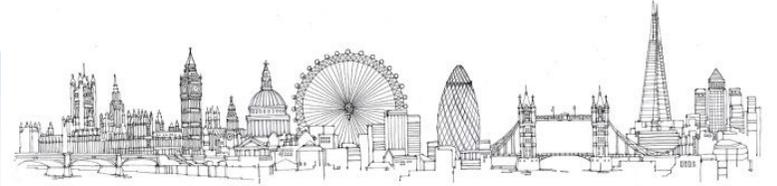
Return of the Obra Dinn :

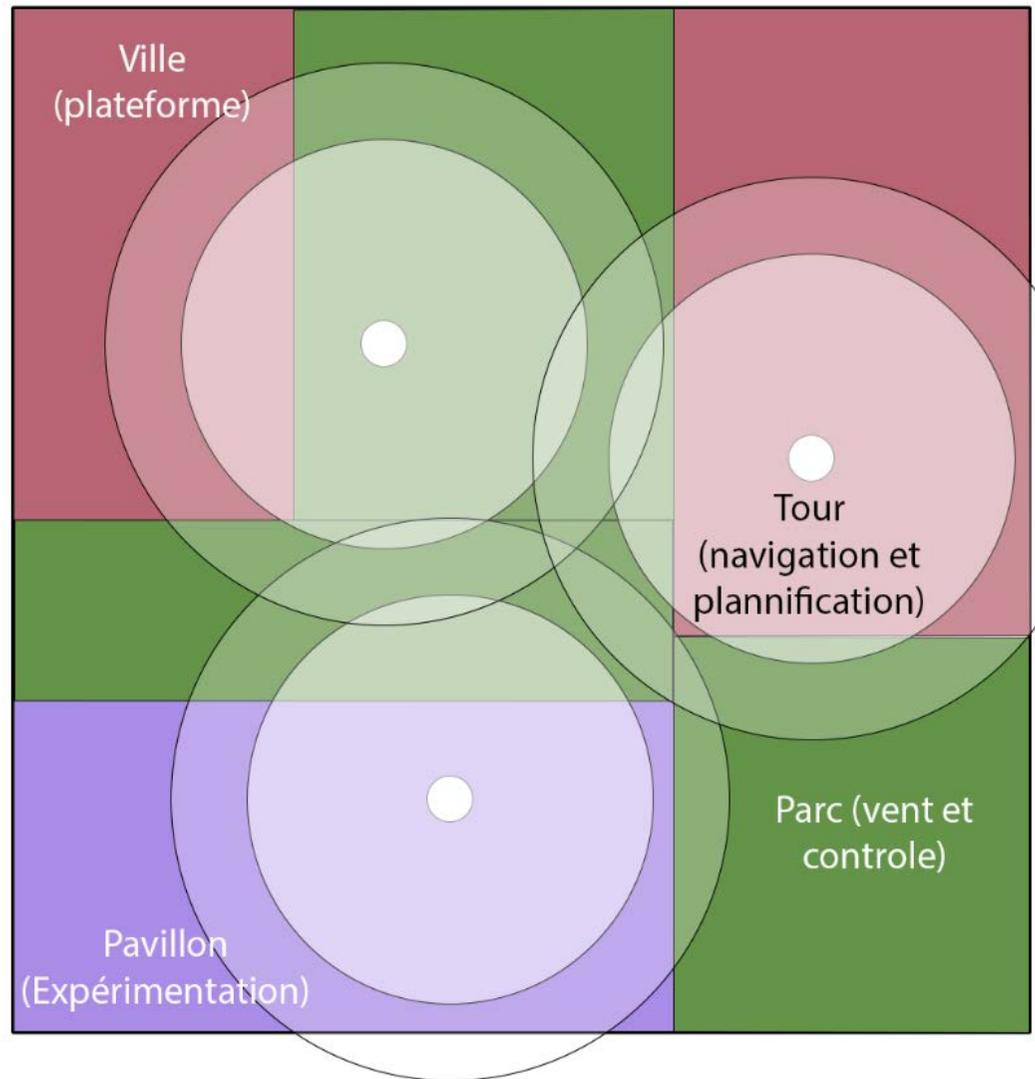
Ce jeu utilise que des nuances de noir et blanc, mais reprends esthétiquement les couleurs unis, l'ombrage et les outlines de manière intéressantes, on en a extrait ces contrastes qui fonctionnent bien. et le fait d'utiliser la hauteur et les couleurs nous permettent de ne pas confondre tout les éléments.

ECHOPLEX :

L'utilisation d'une couleur unie par zone est un élément fort de ce jeu, on s'inspire surtout des ombrages et de la délimitation entre les zones pour clarifier nos situations de jeu.









Direction Artistique Sonore :

- p97 - Intentions
- p98 - Références
- p100 - Overview des sons
- p102 - Production des sons
- p105 - Tracking et Avancement
- p107 - EventList
- p110 - Intégration
- p114 - Répartition Musique



Intentions

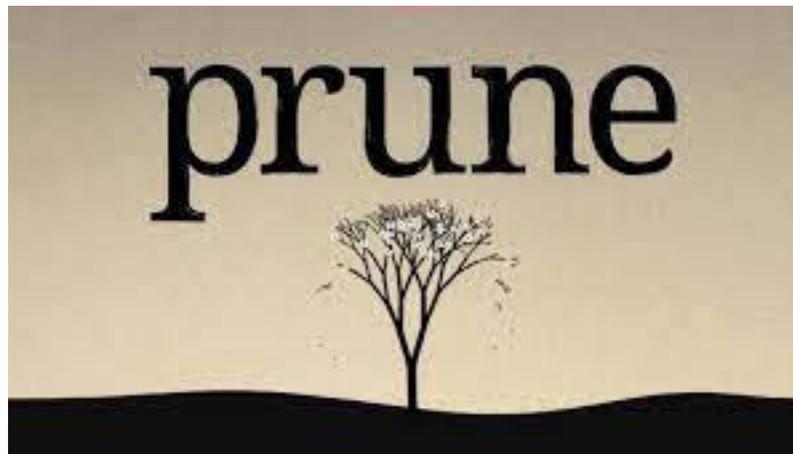
Parapluie se focalise sur un contrôleur reprenant les caractéristiques d'un parapluie fantasmé capable de voler. Le fun et toute la profondeur du gameplay se font par la maîtrise de ce contrôleur de la part du joueur et sur comment cette maîtrise est quantifiée et mesurée. C'est pourquoi les sons de Parapluie doivent servir ce propos et agrémenter ce fantasme en offrant des feedbacks sonores que le joueur imagine instinctivement lorsqu'il s'imagine un parapluie volant (impact du vent sur la toile, sensation de vitesse et de liberté etc..)

De plus, les feedbacks sonores doivent faire comprendre au joueur la situation dans laquelle ce dernier se trouve. En effet, le joueur doit comprendre lorsqu'il touche un obstacle ou lorsqu'il régénère sa jauge de flap.

Pour ce faire, notre Sound Design délimite 2 catégories de sons :

- les sons réalistes pour le parapluie. Cela permet au joueur de comprendre les actions et réactions du personnage suivant les commandes réalisées. Si le joueur touche un obstacle, le son entendu doit correspondre au son que peut faire un parapluie s'il touche un obstacle.
- les sons en synthèse pour le système. Cela permet au joueur de comprendre que les actions réalisées n'ont pas un impact direct sur le personnage mais sur le système de jeu. En effet, lorsque le joueur récupère un bonus de flap, le parapluie n'est pas affecté par ce bonus. En revanche, le système comprends que le joueur peut faire plus de flap.

Pour finir, notre jeu se veut poétique. De ce fait, le Sound Design de l'environnement ne doit pas être surchargés et prendre le dessus sur les sons du personnage principal. Il doit être assez doux et simple, comme dans Prune.



Références

Avatar



Pour établir les sons du personnage, nous nous sommes penchés sur des jeux utilisant le même système de déplacement. En effet, nous pouvons donc citer Zelda Breath of the Wild ainsi que Genshin Impact. Ces deux jeux permettent au joueur de se déplacer dans les airs.

Il est important de noter que ces modes de déplacement offrent une expérience auditive réaliste et une sensation de vol au joueur. Il est donc intéressant de nous inspirer de ces deux cas pour créer les sons de notre parapluie.

Références

Environnement

Notre parapluie se déplace dans la ville de Londres.

De ce fait, nous pouvons donc imaginer des sons correspondant à une ambiance de ville, comme la foule ou encore la circulation.

En plus de ces sons caractéristiquent à une ville, nous avons opté pour des musiques d'ambiance qui se joueront suivant la zone où se situe le joueur. Cela donnera une réelle identité à chaque zone et renforcera le côté poétique.



Overview des sons

EventList : Character

Pour savoir quels sons créer pour le personnage, il suffit de nous questionner sur les actions de ce dernier. De ce fait, nous pouvons déterminer l'ouverture, la fermeture, le vol, le flap et la chute du parapluie. Cependant, nous ne pouvons nous contenter des sons précédents. En effet, suite à cela, le parapluie peut se poser.

EventList: Environnement

Pour garder notre côté poétique, nous avons créer 4 types de zones au sol :

- les bâtiments de petite taille
- les bâtiments de taille moyenne
- les bâtiments de grande taille
- l'eau

De plus, afin de montrer au joueur son ascension, nous avons rajouté des zones dans le ciel. Cela permet donc au joueur de faire la différenciation entre le sol, le ciel et l'espace.

Nous pouvons cependant rajouter des sons réalistes comme le klaxon des voitures ou encore une foule afin de donner un côté plus vivant de notre ville.

EventList : Système

Les sons systèmes représentent les menus ainsi que les sons qui ne sont pas directement produit par le parapluie ou l'environnement. De ce fait, nous pouvons donc rajouter aux sons des menus les sons liés à la jauge du flap, comme le bonus de récupération de flap.



Processus de Production

Recherches

Avant de commencer la production des sons, nous faisons des recherches afin de trouver des références intéressantes et/ou des procédés de créations.

Pour déterminer notre sujet de recherche, nous établissons notre point d'entrée sur l'un des ingrédients de notre jeu.

Par exemple, nous nous sommes penchés sur le parapluie. A partir de cet ingrédient, nous avons cherché des références sur les potentiels sons à produire, puis sur des moyens de recréer les sons.

Prenons le cas du "flap". En effet, il s'agit de la seule action du parapluie lui permettant de s'envoler.

Dans le monde réel, nous pouvons prendre exemple sur les oiseaux qui s'envolent en battant des ailes. De ce fait, grâce à cette recherche, nous pouvons en conclure que le son du "flap" peut ressembler à celui d'un battement d'aile.

Suite à cette conclusion, nous pouvons partir sur une recherche de production.

Grâce aux résultats de la recherche, nous pouvons nous baser sur la méthode de Libertivi. Il nous suffit de prendre un morceau de tissu et de l'agiter dans nos mains.

Productions des sons

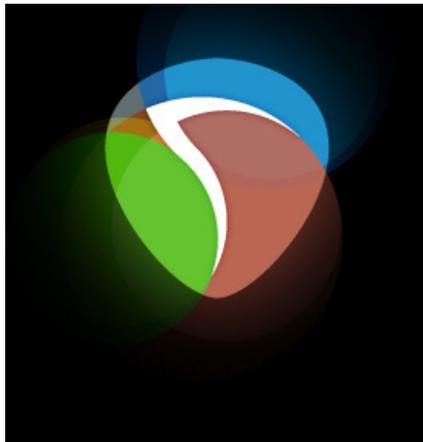
Personnage

Pour produire les sons, nous utilisons Reaper. Lorsque les sons sont créés, nous les insérons dans un Fmod afin qu'ils puissent être joués dans Unity.

La production se lance après plusieurs recherches concluantes, afin de ne pas oublier nos intentions.

Concernant le personnage, nous avons plusieurs moyens de production :

- La première consiste à créer les sons en intégralité. Par exemple, nous pouvons citer l'ouverture et la fermeture d'un parapluie (action réalisée sur un vrai parapluie), ou encore l'atterrissage (manche en bois



fmod

Production des sons

Environnement

Pour les sons caractérisants la ville, nous avons utilisés de réels sons, avec le même processus que pour le parapluie.

Néanmoins, nous avons eu un processus différent pour produire les musiques d'ambiance. En effet, s'agissant principalement de musique d'ambiance et n'étant pas compositeur, nous avoins donc eu recours à l'intelligence artificielle «beatoven.ai». Cette intelligence artificielle nous demande le genre et le tempo de la musique souhaitée.

Cependant, pour l'eau, nous avons recouru au carillon kochi aqua. En effet, les carillons nous ont permis de garder l'effet de musique, tout en rappelant le son de l'eau.



beatoven.ai

Productions des sons

Système

Pour les sons du système, nous avons eu plusieurs approches.

La première concerne les sons non mélodiques, comme la collecte d'étoile. En effet, nous avons utilisé jsfxr, un outil permettant de créer des sons 8-bit, soit en synthèse. Le site permet à l'utilisateur de créer des sons par thèmes, comme la collecte de bonus.

La deuxième concerne les sons mélodiques, comme ceux du menu. Pour les créer, nous avons utilisé 1BITDRAGON, un logiciel permettant la composition de musique avec un léger son de synthèse.



Tracking et avancement

Pour connaître l'avancement du Sound Design, nous avons mis en place une EventList. En effet, ce tableau nous permet de voir le rangement des sons dans le Fmod, tout en sachant leur utilité, leur utilisation et leur référence, si elle existe. De plus, ce tableau comporte deux colonnes qui permettent de déterminer si les sons et leurs placeholders sont à créer, ou intégrés dans le Fmod ou Unity. Cela nous permet donc de définir les tâches à effectuer.

Tracking et avancement

Pour connaître l'avancement du Sound Design, nous avons mis en place une EventList.

En effet, ce tableau nous permet de voir le rangement des sons dans le Fmod, tout en sachant leur utilité, leur utilisation et leur référence, si elle existe.

De plus, ce tableau comporte deux colonnes qui permettent de déterminer si les sons et leurs placeholders sont à créer, ou intégrés dans le Fmod ou Unity. Cela nous permet donc de définir les tâches à effectuer.

En plus des données précédentes, nous avons intégré une colonne "Suggestions". Cette dernière nous permet d'indiquer les modifications à effectuer sur les sons produits mais qui possèdent des défauts. Ces suggestions représentent donc le refus du son et donc de la remise en production de ce dernier.

Suggestions	Ref	Placeholder Etat	Son Final Etat
le son se joue un peu tard, il doit etre instantané		intégrer dans unity ▾	a faire ▾

	Sous-dossier	Event Name	Link	Priorité	Type	
Player	Player action	open		Prio 0	oneshot 2D	Son d'ouverture d
		close		Prio 0	oneshot 2D	Son de fermeture
		flap		Prio 0	oneshot 2D	woosh
		perfect_flap		Prio 1	oneshot 2D	woosh
	player_reaction	fall		Prio 0	loop 2D	chute à travers le
		landing		Prio 0	oneshot 2D	manche en bois p
		hit		Prio 3	oneshot 2D	manche en bois la
		fly		Prio 0	loop 2D	Vent + bruit de tiss
		bonus_flap		Prio 0	oneshot 2D	son de bonus
		tyrolienne		Prio 1	loop 2D	descente d'une ty
System	UI	No flap		Prio 0	oneshot 2D	son erreur
		étoile idle		Prio 2	loop 3D	musique en 8 bit
		étoile collect		Prio 1	oneshot 2D	son de bonus
	Menu	barre_progression		Prio 3	oneshot 2D	jauge qui se rempl
		navigation_menu		Prio 2	oneshot 2D	note de piano
		Select		Prio 1	oneshot 2D	notes de flutes
	Return		Prio 2	oneshot 2D	notes de flutes	
Ville	musique_ambiance	ambiance_petit		Prio 3	loop 3D	musique d'ambian
		ambiance_petit_ciel		Prio 3	loop 3D	musique d'ambian
		ambiance_moyen		Prio 3	loop 3D	musique d'ambian
		ambiance_moyen_ciel		Prio 3	loop 3D	musique d'ambian
		ambiance_gratte-ciel		Prio 3	loop 3D	musique d'ambian
		ambiance_gratte-ciel_ciel		Prio 3	loop 3D	musique d'ambian
		ambiance_eau		Prio 3	loop 3D	musique d'ambian
	ambiance_sonor	ambiance_espace		Prio 3	loop 3D	musique d'ambian
		general		Prio 3	loop 3D	son d'une ville en
		bird		Prio 3	oneshot 3D	cri de différents oi
	bird_aile		Prio 3	oneshot 3D	battement d'ailes c	
	voiture_klaxon		Prio 3	oneshot 3D	différents sons de	

	Description	Conditions de Trigger
○ ▾	Son d'ouverture du parapluie	Le joueur ouvre le parapluie.
○ ▾	Son de fermeture du parapluie	Le joueur ferme le parapluie.
○ ▾	woosh	Le joueur fait un flap.
○ ▾	woosh	Le joueur fait un perfect flap.
▾	chute à travers le vent	Le joueur tombe en étant fermé.
○ ▾	manche en bois posée sur le sol	Le joueur se pose sur le sol.
○ ▾	manche en bois lâchée sur le sol	Le joueur touche un obstacle.
▾	Vent + bruit de tissu (drapeau)	Le joueur plane en étant ouvert.
○ ▾	son de bonus	Le joueur traverse un anneau de bonus de flap.
▾	descente d'une tyrolienne	Le joueur utilise une tyrolienne.
○ ▾	son erreur	Le joueur veut réaliser un flap mais sa jauge est vide,
▾	musique en 8 bit	Lorsque le joueur s'approche de l'étoile, il peut entendre la musique.
○ ▾	son de bonus	Le joueur récupère l'étoile.
○ ▾	jauge qui se remplit	Le joueur récupère l'étoile.
○ ▾	note de piano	Le joueur navigue dans les menus.
○ ▾	notes de flutes	Le joueur sélectionne dans les menus.
○ ▾	notes de flutes	Le joueur retourne dans le menu précédent.
▾	musique d'ambiance	Le joueur est dans la zone "petits bâtiments" et proche du sol,
▾	musique d'ambiance	Le joueur est dans la zone "petits bâtiments" et dans le ciel,
▾	musique d'ambiance	Le joueur est dans la zone "moyens bâtiments" et proche du sol,
▾	musique d'ambiance	Le joueur est dans la zone "moyens bâtiments" et dans le ciel,
▾	musique d'ambiance	Le joueur est dans la zone "gratte-ciel" et proche du sol,
▾	musique d'ambiance	Le joueur est dans la zone "gratte-ciel" et dans le ciel,
▾	musique d'ambiance	Le joueur est dans la zone "eau"
▾	musique d'ambiance	Le joueur est dans la zone "espace" lorsqu'il a fini le niveau,
▾	son d'une ville en général	Se joue automatiquement et s'entend lorsque le joueur est proche du sol.
○ ▾	cri de différents oiseaux	Se joue aléatoirement
○ ▾	battement d'ailes de différents oiseaux	Se joue aléatoirement
○ ▾	différents sons de klaxon	Se joue aléatoirement et lorsque le joueur est proche du sol,

Nom du paramètre	Ref	Placeholder Etat	Son Final Etat	Report Bug	Sug
	Zelda BotW	intégrer dans unity ▾	intégrer dans un ▾		
	Zelda BotW	intégrer dans unity ▾	intégrer dans un ▾		
	oiseau / woosh	intégrer dans unity ▾	intégrer dans un ▾		
	oiseau / woosh	intégrer dans unity ▾	intégrer dans un ▾		
	Zelda BotW	intégrer dans unity ▾	intégrer dans un ▾		
	un parapluie	intégrer dans unity ▾	intégrer dans un ▾		
	un parapluie	intégrer dans unity ▾	intégrer dans un ▾		
	Zelda BotW	intégrer dans unity ▾	intégrer dans un ▾		
	Bonus jeux 8 Bits	intégrer dans unity ▾	intégrer dans un ▾		
	une tyrolienne	intégrer dans unity ▾	intégrer dans un ▾		
	erreur windows	intégrer dans unity ▾	intégrer dans un ▾		
	Etoile Mario Galaxy	intégrer dans unity ▾	intégrer dans un ▾		
	Bonus jeux 8 Bits	intégrer dans unity ▾	intégrer dans un ▾		
	Jauge jeux 8 Bits	intégrer dans fmod ▾	intégrer dans fm ▾		
	instruments symbolisant le vent / la légèreté	intégrer dans unity ▾	intégrer dans un ▾		
	instruments symbolisant le vent / la légèreté	intégrer dans unity ▾	intégrer dans un ▾		
	instruments symbolisant le vent / la légèreté	intégrer dans unity ▾	intégrer dans un ▾		
hauteur		intégrer dans unity ▾	intégrer dans un ▾		
hauteur		intégrer dans unity ▾	intégrer dans un ▾		
hauteur		intégrer dans unity ▾	intégrer dans un ▾		
hauteur		intégrer dans unity ▾	intégrer dans un ▾		
hauteur		intégrer dans unity ▾	intégrer dans un ▾		
hauteur		intégrer dans unity ▾	intégrer dans un ▾		
	Carillon Kochi Aqua	intégrer dans unity ▾	intégrer dans un ▾		
		intégrer dans unity ▾	intégrer dans un ▾		
hauteur	Paris	intégrer dans unity ▾	intégrer dans un ▾		
	Différents oiseaux	intégrer dans unity ▾	intégrer dans un ▾		
	Différents oiseaux	intégrer dans unity ▾	intégrer dans un ▾		
hauteur	Différents Klaxon de voiture	intégrer dans unity ▾	intégrer dans un ▾		

Intégration

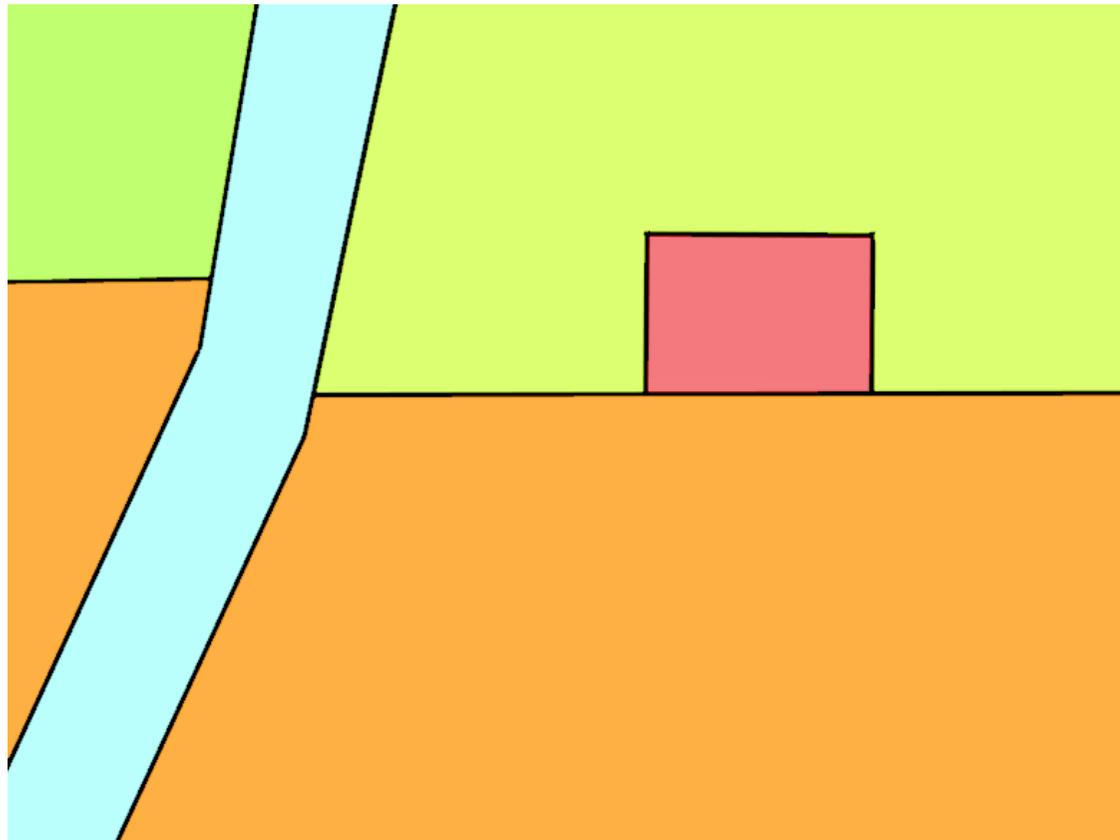
Nom Event	Fmod	Unity
open	Le son est dans un event 2D.	Le son se joue lorsque que le joueur ouvre le parapluie.
close	Le son est dans un event 2D.	Le son se joue lorsque le joueur ferme le parapluie.
flap	Le son est dans un event 2D. De plus, l'événement est un multi-instrument.	Le son se joue lorsque le joueur fait un flap.
perfect_flap	Le son est dans un event 2D. De plus, l'événement est un multi-instrument.	Le son se joue lorsque le joueur fait un flap avec le bon timing.
fall	Le son est un event 2D.	Le son se joue lorsque le parapluie est fermé et qu'il perd de l'altitude.
landing	Le son est un event 2D.	Le son se joue lorsque le parapluie touche le sol.
hit	Le son est un event 2D.	Le son se joue lorsque le parapluie touche un obstacle.
fly	Le son est un event 2D.	Le son se joue lorsque le parapluie est ouvert et qu'il flotte.
bonus_flap	Le son est un event 2D.	Le son se joue lorsque le joueur récupère un bonus de flap.
tyrolienne	Le son est un event 2D.	Le son se joue lorsque le joueur utilise une tyrolienne.

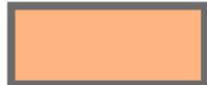
Nom Event	Fmod	Unity
no_flap	Le son est un event 2D.	Le son se joue lorsque le joueur fait l'action de flap mais que la jauge est vide.
etoile_idle	Le son est un event 3D. De plus, une loop est appliquée.	Le joueur entend le son lorsqu'il entre dans le collider. Plus il s'approche de l'étoile, plus le son sera fort. Le son s'arrête si le joueur sort du collider ou récupère l'étoile.
etoile_collecte	Le son est un event 2D.	Le son se joue lorsque le joueur récupère l'étoile.
barre_proges- sion	Le son est un event 2D.	Le son se joue lorsque le joueur accomplit un objectif. Cependant, les objectifs étant les étoiles, le son n'est pas joué.
navigation_menu	Le son est un event 2D.	Le son se joue lorsque le joueur se déplace dans le menu.
select	Le son est un event 2D.	Le son se joue lorsque le joueur sélectionne une option dans le menu.

Nom Event	Fmod	Unity
ambiance_petit	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur est dans le collider de la zone et que la hauteur du parapluie est inférieure à une certaine valeur. Sinon, le son ne se joue pas.
ambiance_petit_ciel	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur rentre dans le collider de la zone et que la hauteur du parapluie est supérieure à une certaine valeur. Sinon, le son ne se joue pas.
ambiance_moyen	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur rentre dans le collider de la zone et que la hauteur du parapluie est inférieure à une certaine valeur. Sinon, le son ne se joue pas.
ambiance_moyen_ciel	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur rentre dans le collider de la zone et que la hauteur du parapluie est supérieure à une certaine valeur. Sinon, le son ne se joue pas.
ambiance_gratte_ciel	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur rentre dans le collider de la zone et que la hauteur du parapluie est inférieure à une certaine valeur. Sinon, le son ne se joue pas.
ambiance_gratte_ciel_ciel	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur rentre dans le collider de la zone et que la hauteur du parapluie est inférieur à une certaine valeur. Sinon, le son ne se joue pas.
ambiance_eau	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur rentre dans le collider de la zone. Lorsqu'il sort de la zone, le son s'arrête.
ambiance_espace	Le son est un event 2D. De plus, une loop est appliquée.	Le son se joue lorsque le joueur finit le niveau.

Nom Event	Fmod	Unity
general	Le son est un event 3D. De plus, une loop est appliquée.	Le joueur entend le son lorsqu'il entre dans le collider.
bird	La liste de son est un event 3D. De plus, cette liste est contenu dans un scatterer.	Le joueur entend le son lorsqu'il entre dans le collider. Si la hauteur du parapluie est supérieure à une certaine valeur, les sons d'oiseaux seront plus présents.
bird_aile	La liste de son est un event 3D. De plus, cette liste est contenu dans un scatterer.	Le joueur entend le son lorsqu'il entre dans le collider. Si la hauteur du parapluie est supérieure à une certaine valeur, les sons d'oiseaux seront plus présents.
voiture_klaxon	La liste de son est un event 3D. De plus, cette liste est contenu dans un scatterer.	Le joueur entend le son lorsqu'il entre dans le collider. Si la hauteur du parapluie est supérieure à une certaine valeur, les sons de voitures seront moins présents.

Schéma de la répartition des musiques d'ambiance



-  Petit Bâtiment
-  Moyen Bâtiment
-  Gratte-ciel
-  Eau

Gestion des sons avec la hauteur

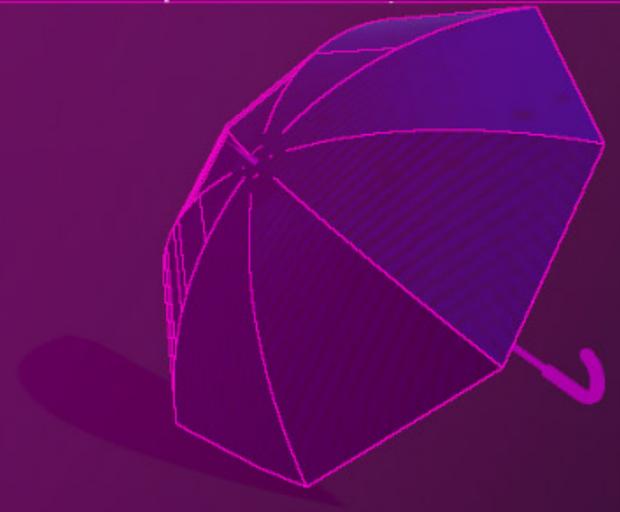
Le paramètre de la hauteur se règle en fonction de la position y du parapluie dans le niveau. De plus, si y est supérieur à une valeur définie, le parapluie sera considéré dans le ciel. Cela nous permet donc de gérer les sons du ciel et du sol comme suit le schéma suivant :

Valeur de la hauteur délimitant le sol du ciel	Musique d'ambiance du ciel	Majoritairement des oiseaux
	Musique d'ambiance du sol	Mélange de voitures et d'oiseaux
	Musique d'ambiance	Son d'ambiance



UI et UX :

- p117 - Intentions
- p118 - Références
- p120 - UI Wireframe
- p121 - UI Feedback et HUD
- p122 - Moodboard
- p123 - Main Menu et éléments HUD
- p124 - Pause Manu et bouton



Intentions

UI - La clarté avant tout/UX - Fluidité et progression encadrée.

Notre projet a une tendance à être chaotique, parfois même déroutants dans les situations les plus difficiles. Aussi, nous souhaitons faire en sorte que la clarté soit mise à l'honneur pour ce qui est de notre UI afin de ne pas ajouter au désordre du joueur paniquant avec les contrôles.

Pour ce faire nous utilisons des éléments d'interface sans texte afin qu'ils soient compréhensibles sans effort, et suffisamment épurés pour qu'ils soient lisibles en un minimum de temps afin d'alléger autant que possible la charge cognitive du joueur.

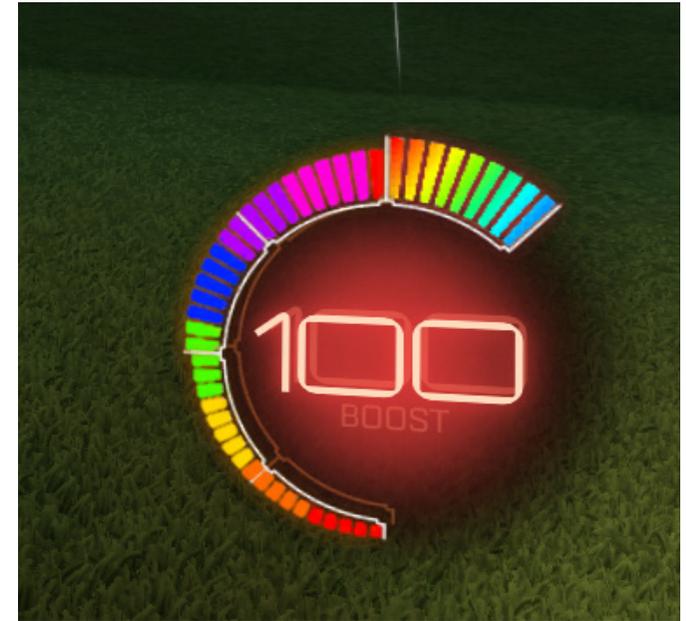


Références

La jauge de Boost de Rocket League

Une de nos références principale est rocket league et plus particulièrement sa jauge de boost. Simple, clairement visible, elle permet au joueur de savoir à tout moment où il en est de sa ressource.

Nous avons repris la même idée mais comme nous manipulons des quantités plus faibles de ressources, nos « Flaps » sont indiqués par des petits marqueurs colorés plutôt qu'une jauge colorée. Les marqueurs sont placés en arc sur le côté du cercle contenant le nombre restant de « Flaps » et disparaissent/apparaissent à mesure que le joueur utilise et regagne ceux-ci.



Le Timer des Jeu de Combat 2D

Dans les jeux de combat 2D dans le style de Mortal Kombat ou Street Fighter, le timer est au centre de l'écran afin d'être toujours au moins dans la périphérie de la vision des joueurs.

Nous souhaitons que le joueur puisse à tout instant facilement savoir où il en est dans le niveau aussi notre timer va se placer en haut au centre de l'écran afin de maximiser sa visibilité.

Références

New Super Mario Bros 2



Dans New super mario bros DS, le timer en haut à droite de l'écran du haut permet de savoir le temps restant. Il change de couleur à mesure que le temps s'amenuise. Nous souhaitons reprendre ce principe dans notre projet.

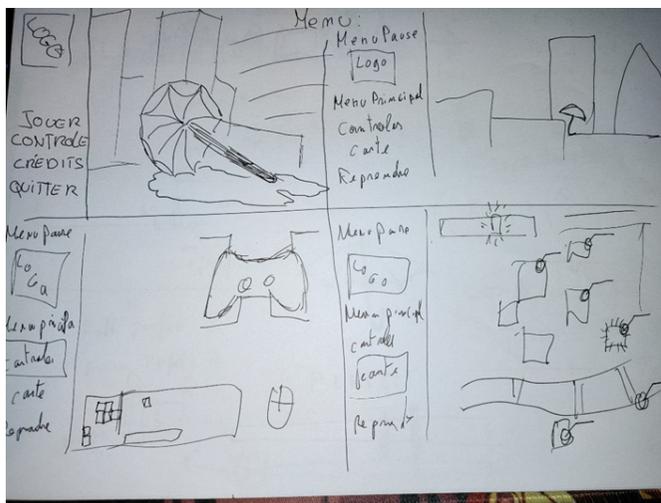
Le joueur doit battre un niveau dans la limite d'un temps imparti s'il veut gagner une étoile supplémentaire, le timer changera de couleur peut avant que ce temps imparti ne soit écoulé, puis une fois que le temps sera écoulé, afin que le joueur soit toujours informé de ses performances et puisse aviser en temps réel de sa situation.

De plus nous souhaitons aussi mettre un indicateur de progression de niveau qui, à l'instar de celui de New super mario bros DS, indiquera au joueur constamment sa progression dans le niveau afin qu'il puisse juger à l'aide du timer comment il s'en sort.

UI & Wireframe

Mécanique	éléments liés	Besoins	Nature (interface, feedback graphique, feedback sonore)	Type	Forme	
Flap, moyen de déplacement du joueur, fonctionne comme une méduse	Jauge de Flap --> Chaque flap consomme une partie de la jauge. Lorsque la jauge est vide le joueur ne peut plus flapper. La jauge se régénère automatiquement au fil du temps. Lorsque le joueur flap dans le bonbon timing, il effectue un flap parfait (plus puissant) et améliore sa régénération de flap	Le joueur doit pouvoir comprendre qu'il vient de flap	Feedback visuel	diégétique	animation de flap	
				Feedback sonore	diégétique	son du flap
		Le joueur doit comprendre qu'un flap lui coûte de l'énergie	Interface	non diégétique	animation de décrémentation de la jauge de flap	
		Le joueur doit pouvoir voir que son énergie remonte	Interface	non diégétique	animation de régénération de la jauge de flap	
		Le joueur doit pouvoir voir quand il doit rappuyer pour faire un flap parfait	Feedback visuel	diégétique	glow léger du parapluie dans la couleur du perfect flap	
			Interface	non diégétique	apparition d'un marqueur de la couleur du perfect flap sur la jauge de flap	
		Le joueur doit comprendre qu'il vient de faire un flap parfait	Feedback visuel	diégétique	animation et particule flap parfait	
			Interface	non diégétique	apparition du texte "perfect" au dessus de la barre de flap	
		Le joueur doit pouvoir comprendre qu'il ne lui reste plus qu'un flap	Feedback sonore	diégétique	son du flap parfait	
		Le joueur doit comprendre que sa jauge de flap est vide et qu'il ne peut plus faire de flap	Interface	non diégétique	clignotement du fond de la jauge de flap blanc --> rouge	
			Interface	non diégétique	La barre de flap devient rouge	

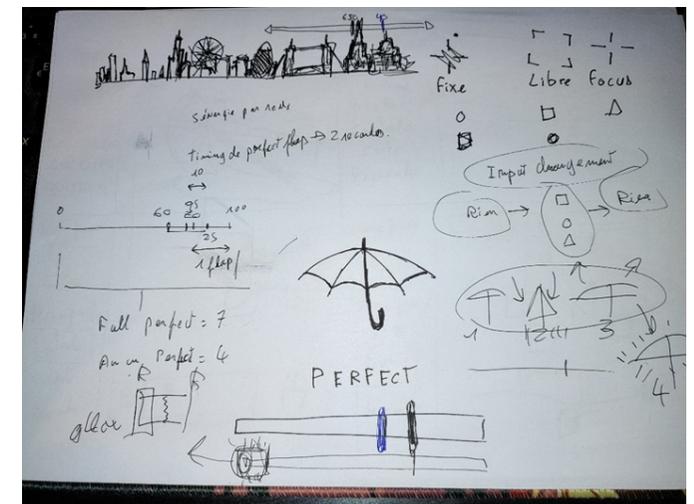
Menus :



Changement vis-à-vis du wire frame :

Inversion des boutons reprendre et menu principal \square moins de chance de retourner au menu principal par mégarde.
Ajout d'un raccourci sur select pour afficher le menu de navigation rapide + la carte

Interface In-Game :



UI - Feedbacks & HUD

Changement de caméra	L'angle de la caméra qui fonctionne de manière automatique : focus -> regarde un point de la map, Libre -> ne bouge pas, Fixe -> Regarde vers l'avant du parapluie.	Le joueur doit savoir dans quel angle de caméra il se trouve	Interface	non diégétique	icone carré, rond, et triangle + texte qui disparaît 2 secondes après avoir changé la caméra.
Visiter tous les bâtiments	Lorsque le joueur visite un bâtiment sur la carte et qu'il obtient le collectible lié à ce bâtiment, son pictogramme se colorera sur la frise en haut de l'écran dans la couleur de sa zone	Le joueur doit comprendre que le bâtiment fait parti des bâtiments à visiter	Feedback visuel	diégétique	correspondance entre la forme générale du bâtiment et l'ombre d'un des bâtiments de la frise
			Feedback visuel	diégétique	Landmark fort (bâtiment plus imposant que les alentours, niveau de détails plus élevé)
			Interface	non diégétique	correspondance entre la forme générale du bâtiment et l'ombre d'un des bâtiments de la frise
		Le joueur doit comprendre que c'est un bâtiment qu'il n'a pas encore visité	Feedback visuel	diégétique	cube collectible encore visible (non obtenu) + colonne de lumière --> indicateur visible de loin, aide à la navigation
			Interface	non diégétique	ombre du bâtiment blanche sur la frise (absence de couleur)
		le joueur doit comprendre que sa visite du bâtiment a été comptabilisée	Feedback visuel	diégétique	disparition du collectible et de la colonne de lumière
			Feedback sonore	diégétique	son d'obtention de collectible
			Interface	non diégétique	mise en couleur du bâtiment correspondant sur la frise
Menus	Comprendre les fonctions du menu et qu'une action à été effectuée	Comprendre qu'il clique sur un bouton	Interface	non diégétique	highlight du bouton
		Comprendre qu'il effectue un retour en arrière	Feedback sonore	non diégétique	son de retour en arrière
		Comprendre qu'il quitte les menus	Interface	non diégétique	animation de fermeture du menu, retour en jeu
Menu/Navigation rapide sur la carte	Comprendre le fonctionnement de la navigation rapide	Associer les points de téléportation disponible à leur position sur la carte	Interface	non diégétique	lorsque le joueur hover un point de téléportation, le bâtiment correspondant est highlight dans la frise présente dans le menu

PARAPLUIE - Moodboard

3GDB - Couffin Yumi - Vo Thomas - Prudhomme Kyle - Boulais Sébastien

Caméra : 3 types



Rocket League : Ball cam



Icones :

Mobilité - Stabilité - Dynamisme

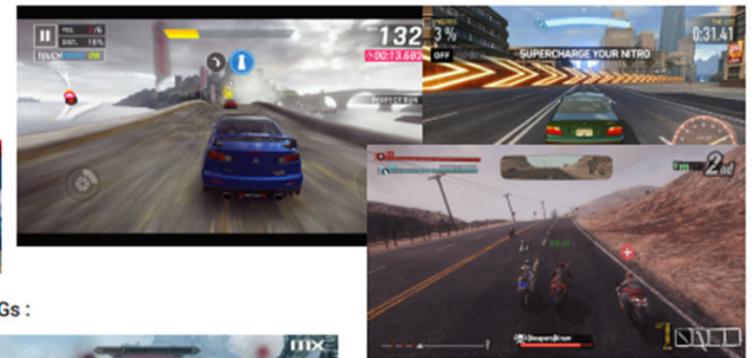


Energie : Barre d'énergie

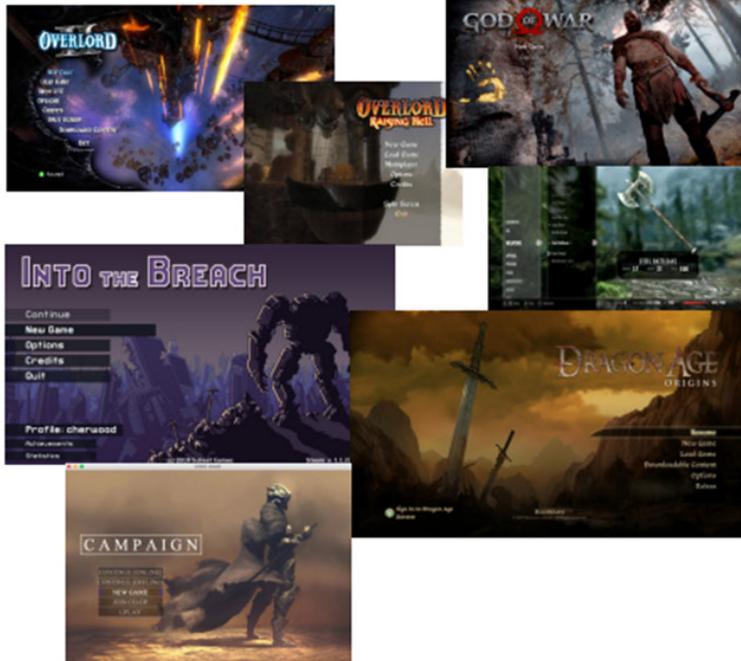
Jeux de baston :



Jeux de course :



Menu : de coté



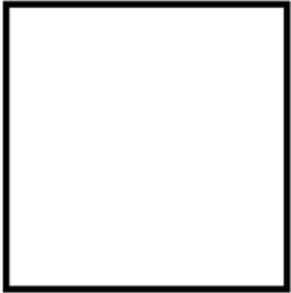
RPGs :



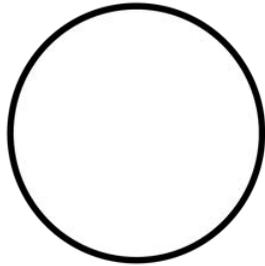
Carte : Progression



UI - Éléments & Menu principale



Caméra fixe, carré = stable. Contour noir permet de démarquer l'icone du jeu et de créer une séparation intra vs extra diégétique.



Caméra libre, rond = décomplexé, sans limites. Contour noir permet de démarquer l'icone du jeu et de créer uen séparation intra vs extra diégétique.



Un dégradé en arc de cercles reprenant les couleurs de l'écran d'accueil.

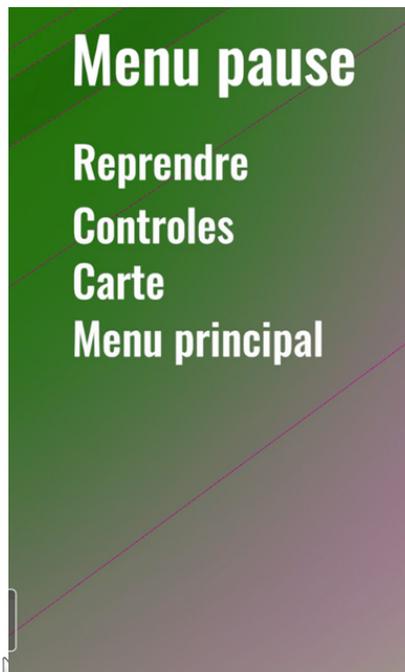
Sobre et en accord avec les couleurs de la scène.

Met en avant les boutons Blancs

UI - Menu pause & boutons

Un dégradé en arc de cercle reprenant le motif du menu principal.

Le motif possède une opacité moyenne permettant de tirer profit des changements de couleurs du shader et d'accorder le menu de pause avec la couleur de l'environnement actuel du joueur.



Boutons blancs neutres

Permet d'avoir du contraste avec toutes les combinaisons de couleurs de menu choisies.

