



Jacques Andrew – Bucamp Théo – Kabongo Gregory – Couffe Lilian

Sommaire :

Partie Game Design (p.5):

- Intentions (p.6)
- Inspirations (p.7)
- Pitch/Breakthrough (p.8)
- Gameplay (p.9)
- 3Cs (p.10)
- Condition de défaite/victoire (p.14)
- Boucles de gameplay (p.15)
- Mécaniques de jeu (p.20)
- Ingrédients (p.26)
- LD patterns (p.40)
- Signes, feedbacks, tests utilisateurs (p.54)

Partie Direction Artistique (p.57) :

- Direction Artistique Visuelle (p.60)
- Direction Artistique Sonore (p.82)
- UI (p.84)

Game Design

Intentions :

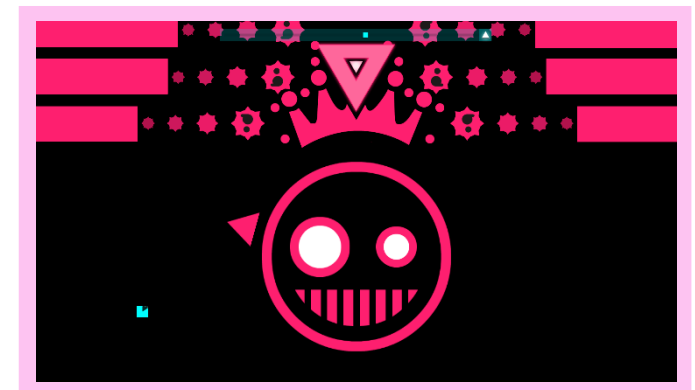
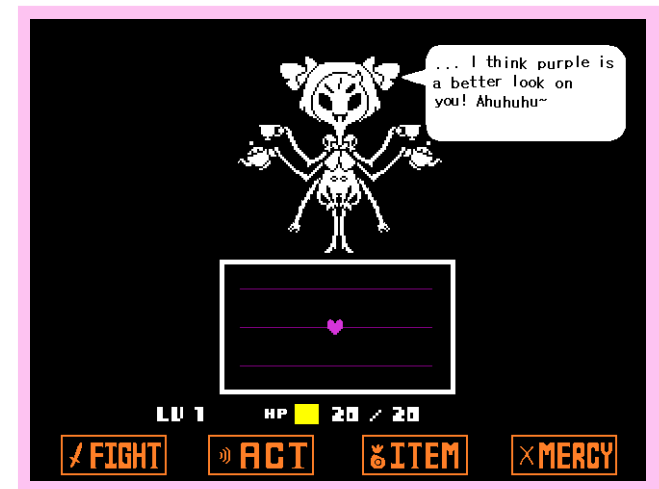
Tout d'abord, nous voulions un jeu **basé sur la musique**, plus précisément nous voulions que le **gameplay soit rythmé sur le tempo** de la musique. Nous voulions que le jeu challenge le joueur dans le **timing** de ses actions alors nous avons introduit une mécanique de **restriction de déplacement**. Nous avons directement trouvé intéressant l'idée de limiter les mouvements du joueur.

Nos intentions se sont alors concrétisées en un shmup où le joueur se déplace seulement sur **4 lignes horizontales**. Il peut se déplacer de gauche à droite avec le joystick sur les lignes et se déplacer entre les lignes en inclinant le stick vers le haut ou le bas. Il peut également envoyer des projectiles avec le stick droit, ces projectiles sont **envoyés dans le même angle** que celui de l'inclinaison du stick. De plus, l'apparition d'ennemis ainsi que les projectiles qu'ils envoient sont **en rythme avec la musique** jouée lors du niveau.

Inspirations :

Nous nous sommes principalement inspirés de deux jeux. Tout d'abord nous avons pris comme référence **Undertale**, plus spécifiquement du combat contre Muffet pour **la mécanique de déplacement**.

De plus, **Just Shape and Beats** a également été une influence notamment pour notre intention d'avoir un **gameplay calé sur le tempo** de la musique.



Pitch :

Wilfried est un jeune **apprenti magicien**, qui occupe tout son temps libre en faisant du **skate** sur ses propres rampes et musique dans ses oreilles. Cependant son hobby ennue beaucoup **son voisinage**, qui n'hésitera pas à l'**attaquer** pour le faire arrêter. Wilfred ne s'arrêtera jamais avant d'avoir terminé son parcours.



Breakthrough :

Les niveaux sont construits avec un **système de Boucles**. En effet, ceux-ci sont divisés en **différentes phases**, et chacune d'entre elles comporte ses **propres vagues d'ennemis**. Le joueur doit vaincre **tous les ennemis** d'une phase pour passer à la suivante et ainsi de suite jusqu'à la fin du niveau.

Elles sont appelées Boucles car en plus d'avoir plusieurs ennemis, chaque phase à sa **propre musique** (phase musicale) qui ont le même fonctionnement que les ennemis ayant un pattern régulier jusqu'à la phase suivante. Tant que le joueur n'est pas passé à la phase suivante, la **musique boucle** au même moment et ne continue réellement que lorsque le joueur tue **toutes les vagues d'ennemis** à l'écran.

Gameplay :

SK8 beat est un **shoot'em** up en 2D avec une caméra en **vue du dessus** (top view), dans lequel le joueur incarne un jeune **magicien skateur** en route chez lui. Dans ce jeu, le joueur se déplace sur **4 barres horizontales** et peut **tirer à 360°** autour de lui.

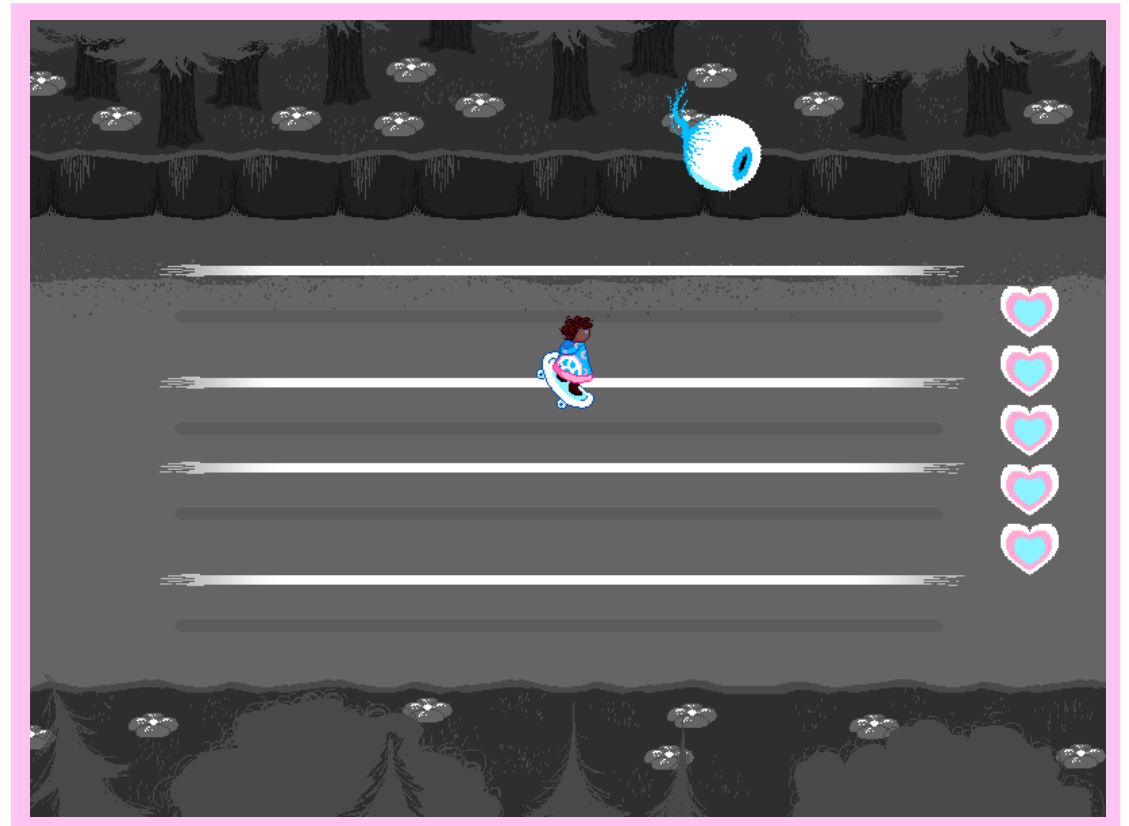
Le niveau est divisé en différentes phases et le joueur doit **tuer tous les ennemis stagnants** de celle-ci pour passer à la suivante. Et ainsi de suite jusqu'au bout du niveau.

L'aspect visuel du niveau (apparition, pattern et timing d'attaque des ennemis) est **mappé / lié au rythme de la musique** (notamment aux percussions du morceau).

3c's :

Caméra

La caméra de taille 800x600 est en **plongée** et couvre l'ensemble de l'espace de jeu. Elle est **fixe** sur le centre de l'espace de jeu et n'effectue ni zoom ni rotation.



Controls :

Déplacement : Incliner le **stick gauche horizontalement** pour se déplacer le long d'une barrière. Incliner le stick gauche **verticalement** pour déplacer d'une barrière à l'autre, tant que les barrières sont adjacentes.

Tir : Incliner le **stick droit** pour tirer. L'angle du stick droit donnera la direction du tir.



Character :

Le personnage est un **magicien skateur** qui peut se déplacer sur **4 rails**. Sur celle-ci, il peut se déplacer de **droite à gauche** et peut **sauter d'une ligne** adjacente à l'autre. Le personnage peut **tirer des projectiles** en 360°.



Condition de victoire/défaite :

Condition de victoire :

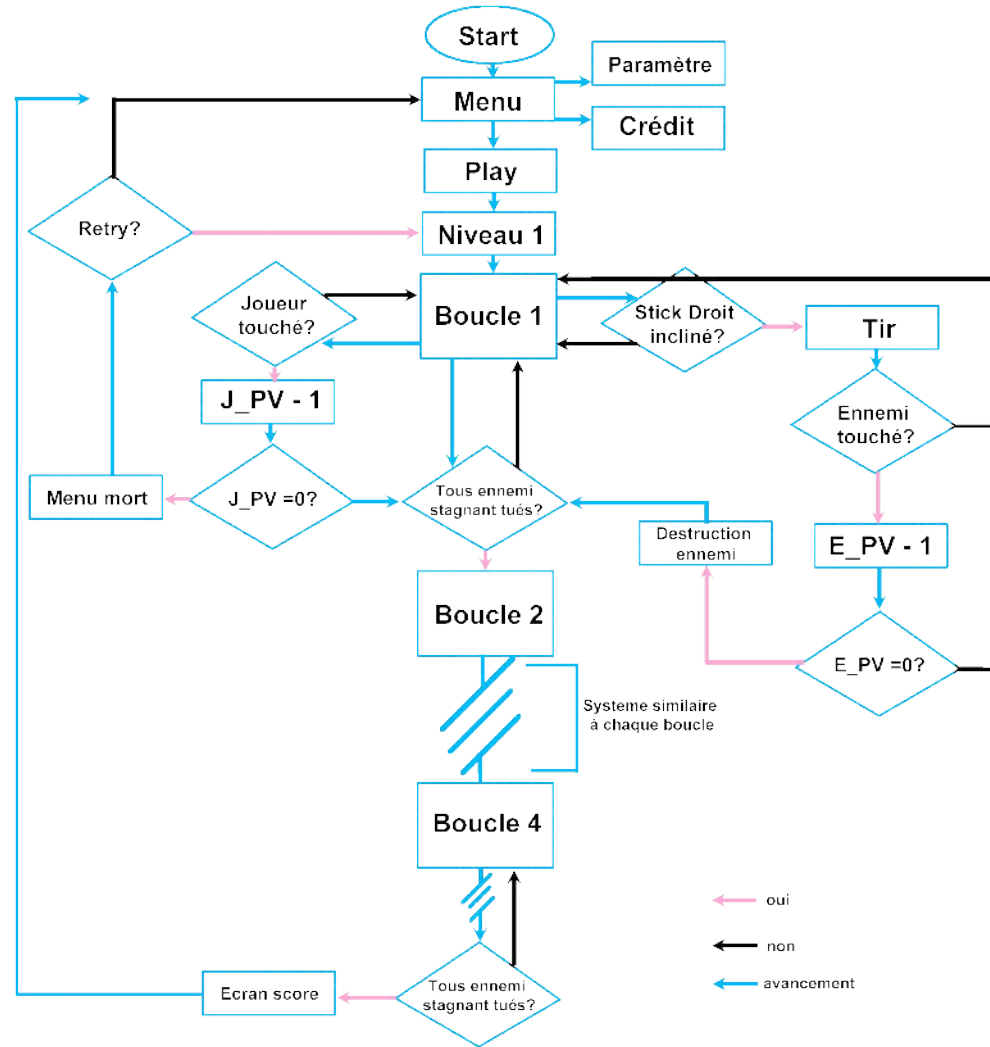
Détruire tous les ennemis obligatoires des phases du niveau avec une **vie supérieur à 0**.

Condition de défaite :

La **vie** du joueur **descend à 0**.

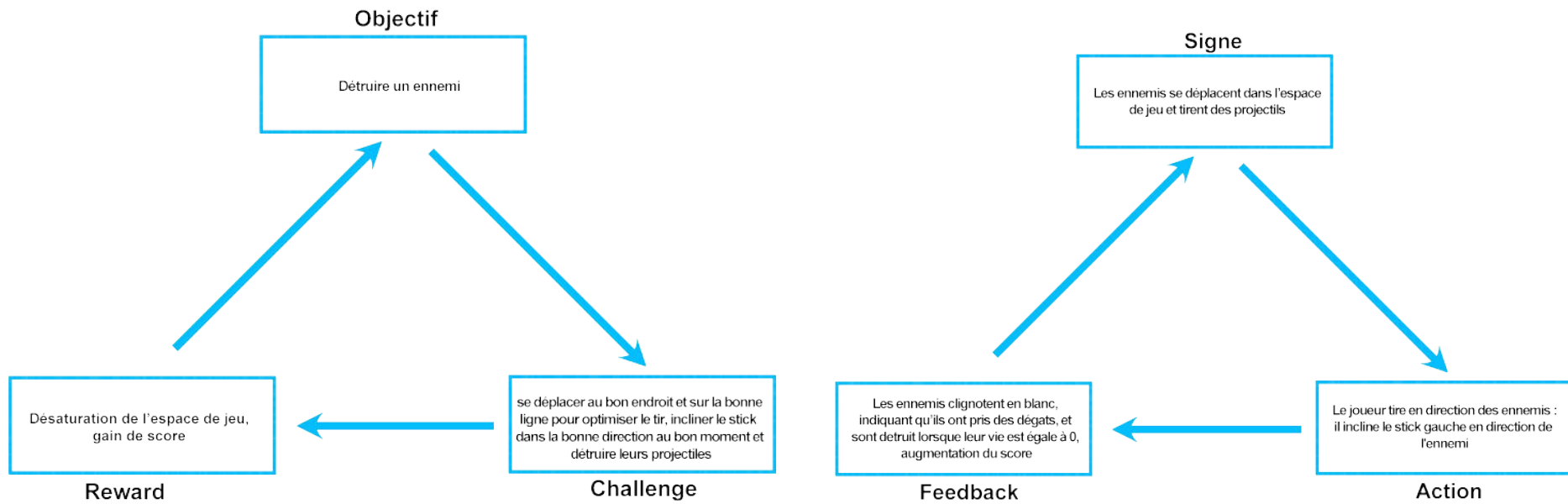


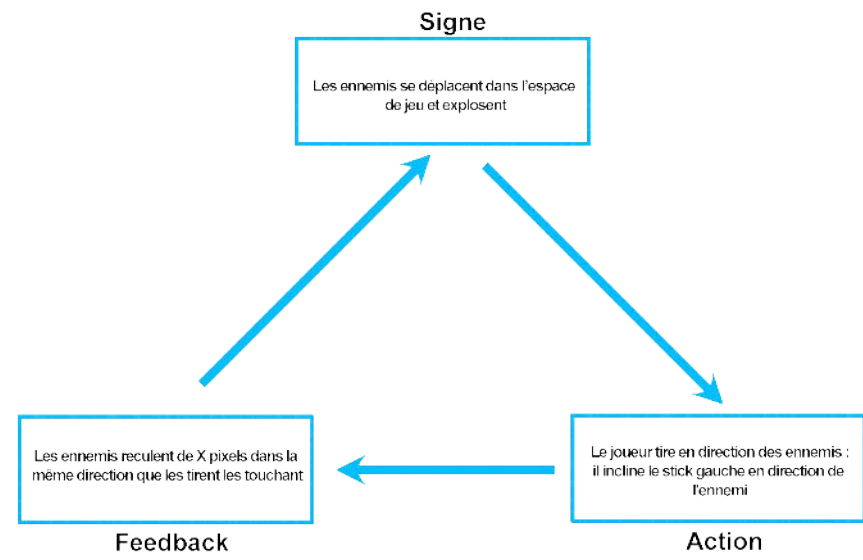
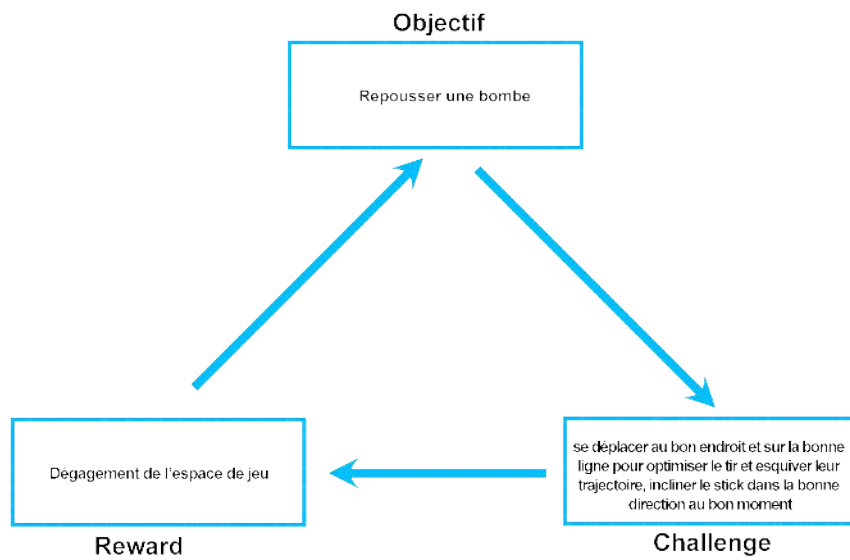
Boucles de gameplay (Schemas & Flowcharts):



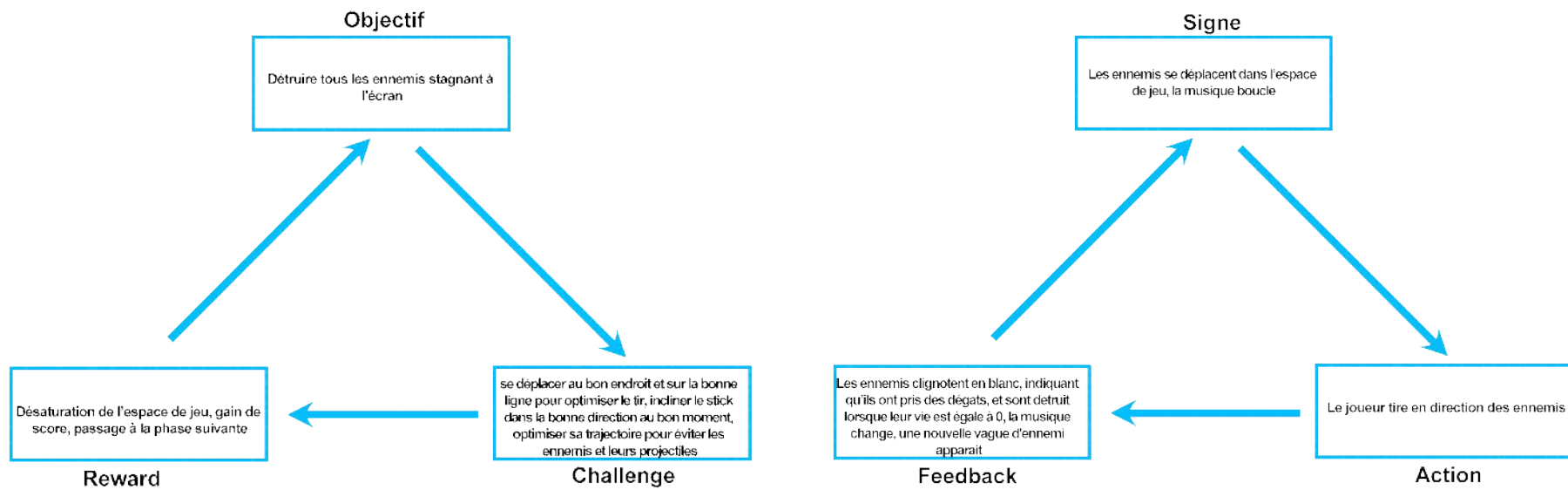
Objectifs :

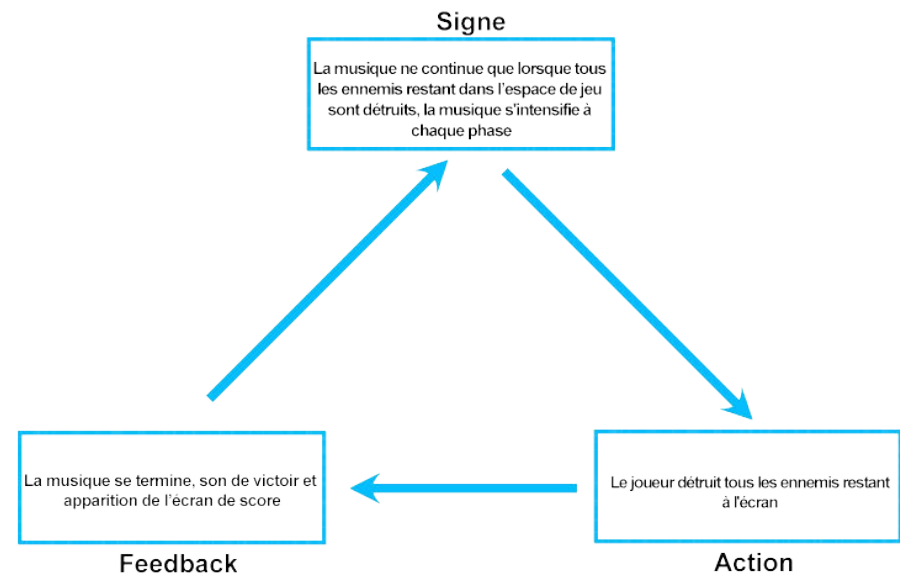
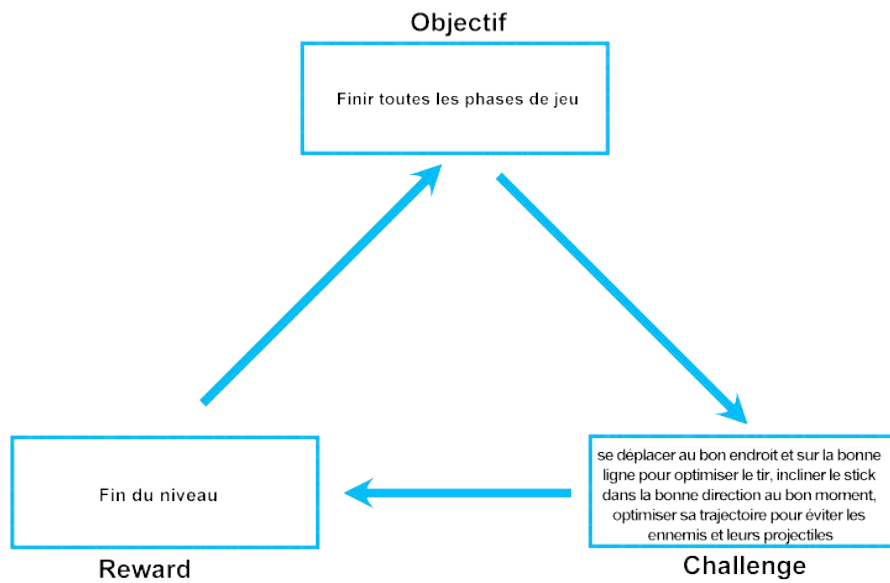
Micro :





Macro :





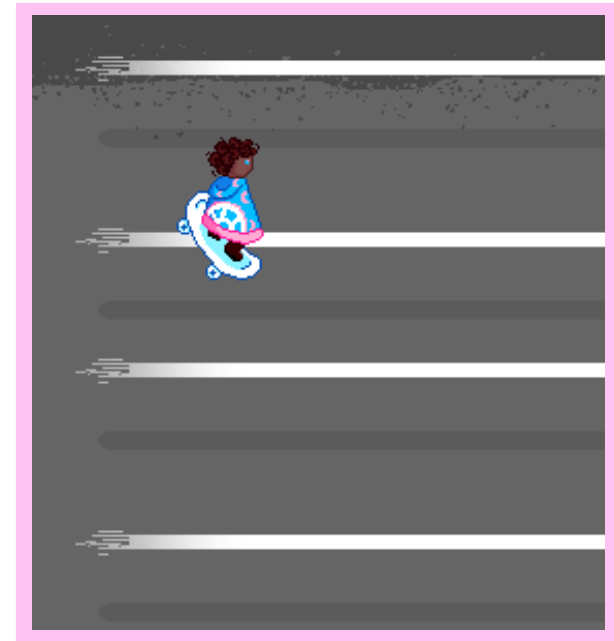
Mécaniques de jeu (Schémas & Flowcharts):

Déplacement :

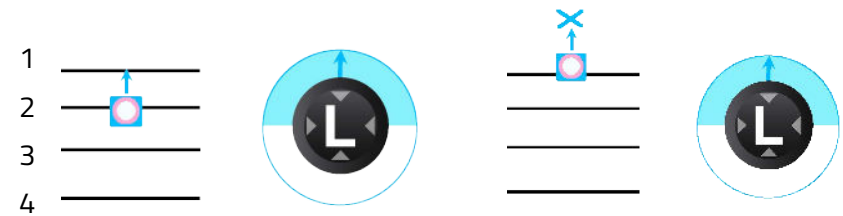
Metrix : Le joueur se déplace à une vitesse de **300 pixels/seconde**. Il atteint cette vitesse lors de son déplacement et **s'arrête immédiatement**.

Le joueur peut **changer de rail** toutes les **0.2 secondes** après son dernier changement de rail.

Player action : le joueur doit **incliner le stick** gauche.



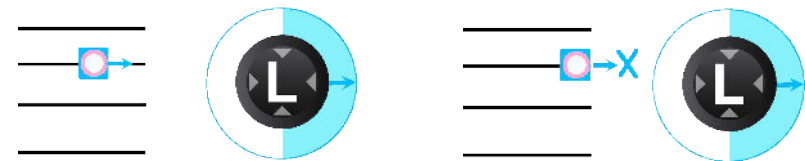
Rules : Le **déplacement** vers le **haut** ne se déclenche que lorsque le **rail** sur lequel le joueur est, est **supérieur à 1**.



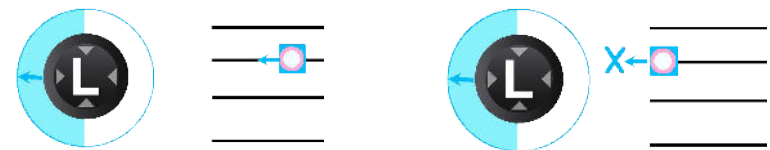
Le **déplacement** vers le **bas** ne se déclenche que lorsque le **rail** sur lequel le joueur est, est **inférieur à 4**.



Le déplacement vers la **droite** ne se déclenche que lorsque **la position X** le joueur est **inférieur au bord droit** des rails.



Le déplacement vers la **gauche** ne se déclenche que lorsque **la position X** le joueur est **supérieur au bord gauche** des rails.

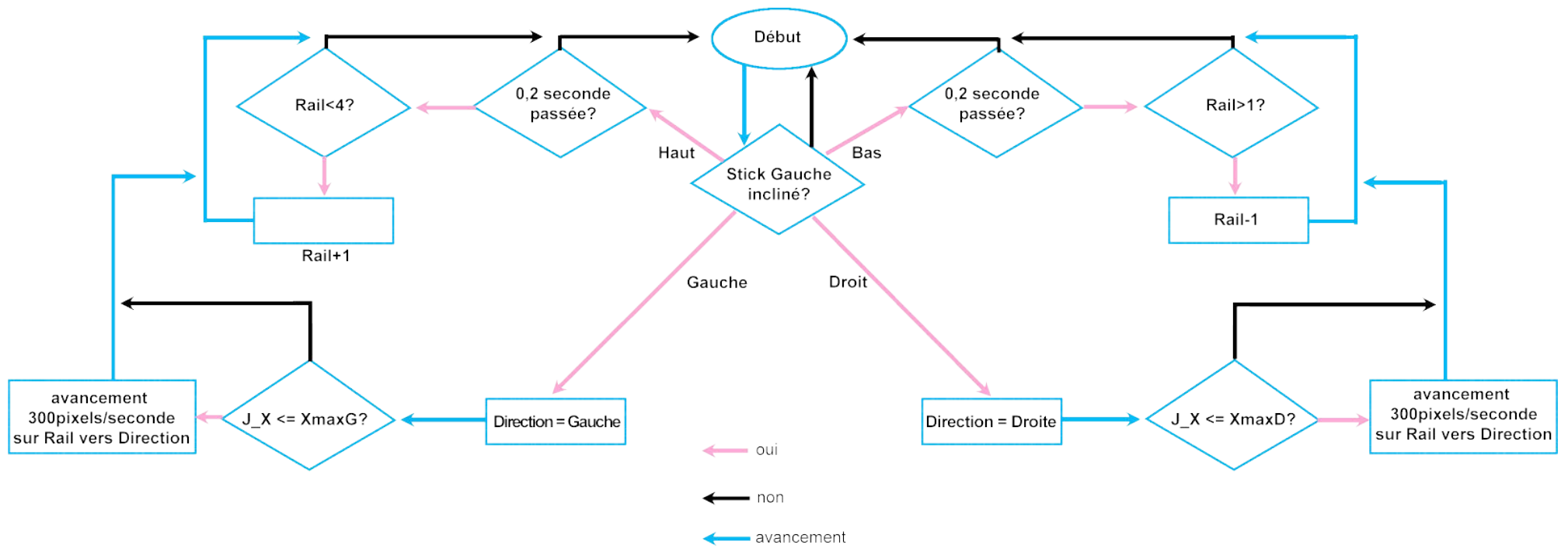


Tokens : le joueur.

State Change : position Y et position X du joueur.

Feedback : déplacement du joueur en fonction de l'inclinaison du stick gauche.

Flowchart :



Tir :

Metrix : Les **projectiles** du joueur ont une vitesse de **800 pixels/seconde** et une **cadence de 0.22 secondes** entre chaque tir. Un tir fait perdre 1 point de vie.

Player action : le joueur doit **incliner le stick Droit.**

Rules : il doit s'être **écoulé 0.22seconde** au minimum après le dernier tir.

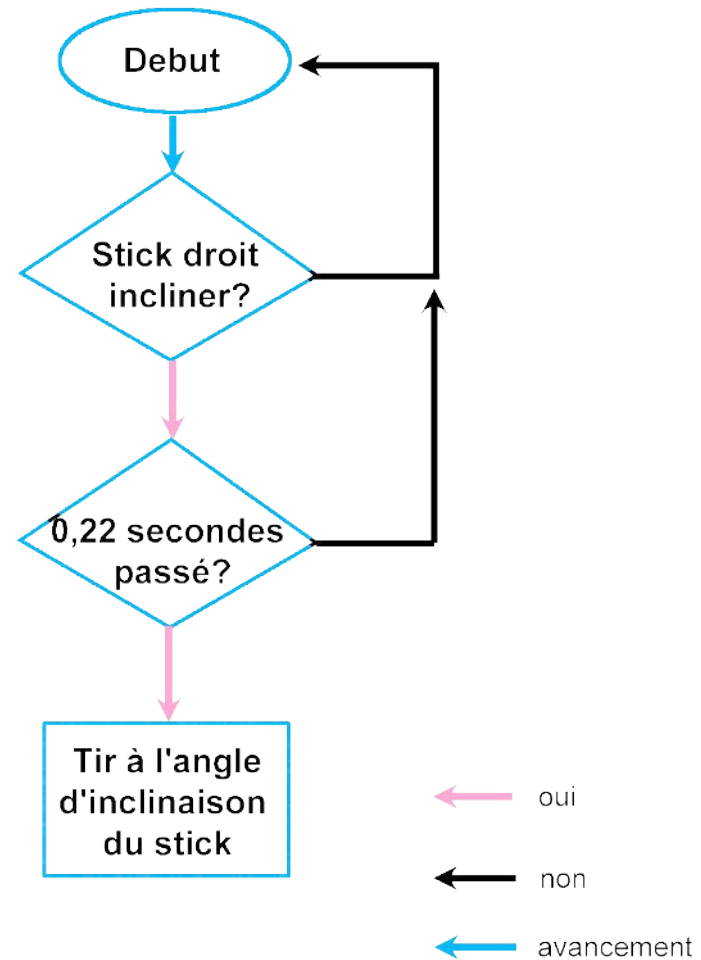


Tokens : joueur, tir, ennemi.

State Change : création d'un tir, application de la direction du tir, application d'une vitesse au tir.

Feedback: projection d'un tir depuis le sprite du joueur en fonction de l'inclinaison du stick droit.

Flowchart :

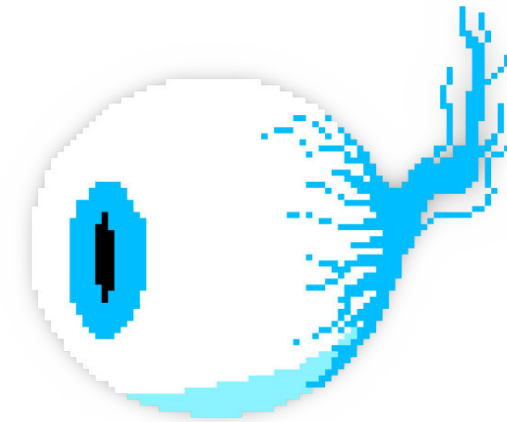


Ingrédients RGD :

Œil :

Description :

Ennemi **traversant l'écran** tout en **changeant de direction** à intervalles réguliers. Son angle alterne entre 35° et -35° de son angle de direction lorsqu'il est calé sur les beats, et son sprite pulse à 1.2 fois sa taille. L'alternance se fait **tous les 2 beats**. Lorsqu'il n'est pas calé sur le beat, il suit simplement son angle de direction. Il a 3 points de vie.



Micro Objectif :

Détruire l'ennemi.

Micro-Challenge :

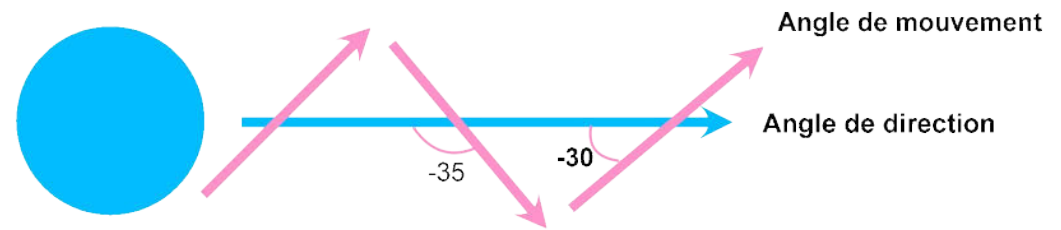
Incliner le joystick gauche pour éviter l'ennemi tout en **inclinant le joystick droit** dans la direction de l'œil.

Input :

Joystick droit et gauche.

Compétence :

Précision.



Ange :

Description :

Ennemi qui **apparaît en haut ou en bas de l'écran** et qui **se déplace de gauche à droite**. Il **lance des projectiles** à intervalles réguliers. Son pattern lui fait effectuer des **allers retours de gauche à droite** de l'écran. Son trajet dure X beats, puis arrivé à une extrémité, il s'arrête pendant X beats, avant de tirer et de se déplacer vers l'autre extrémité.

Cet ennemi **stagne à l'écran** et est donc nécessaire à détruire pour finir une phase.

Les projectiles des Anges **avancent dans la direction du joueur** au moment auquel il a été lancé pendant 2 beats avant de **se stopper** pendant 2 beats, puis de **s'orienter vers le joueur et d'avancer** pendant 2 beats et de continuer son pattern ainsi.

Il a 5 points de vie.



Micro Objectif :

Détruire l'ennemi.

Micro-Challenge :

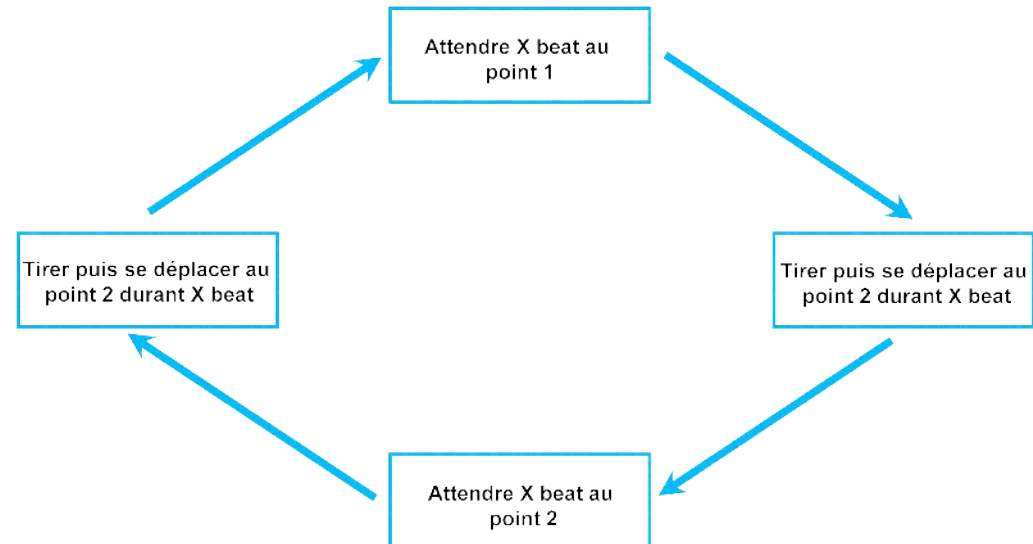
Incliner le joystick gauche lorsque les projectiles sont en mouvement tout en **inclinant le joystick droit** dans la direction de l'ennemi.

Input :

Joystick droit et gauche.

Compétence :

Précision.



Plante :

Description :

Ennemi qui **apparaît en bas de l'écran** et qui **suit horizontalement** les déplacements du joueur pendant X beat avant de s'arrêter X beat et de **tirer** puis recommencer à suivre le joueur. Il projette à intervalle régulier un nombre de **projectile en arc de cercle**.

Cet ennemi **stagne** à l'écran et est donc nécessaire à détruire pour finir une phase.

Il a 7 points de vie.



Micro Objectif :

Détruire l'ennemi.

Micro-Challenge :

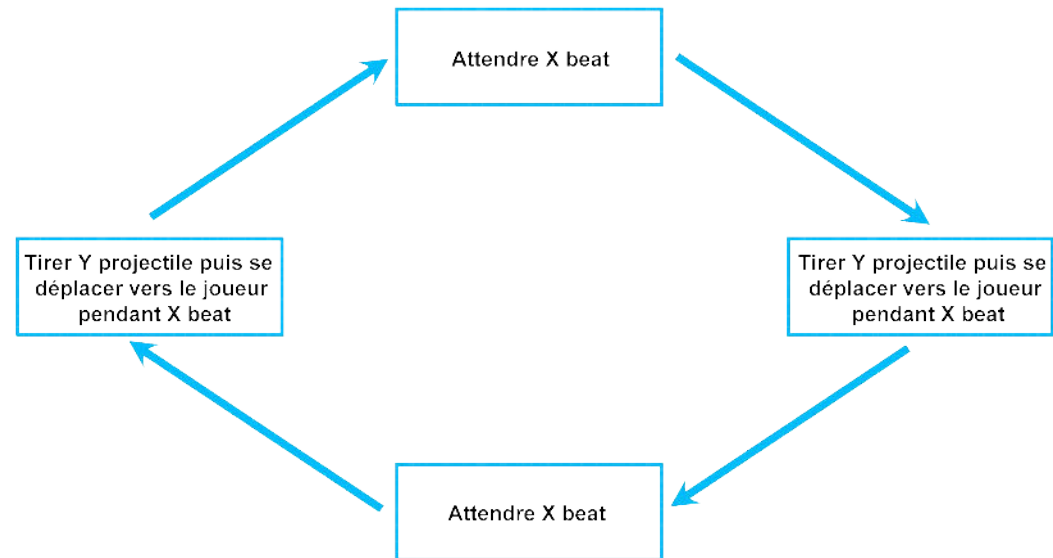
Incliner le stick gauche pour éviter les projectiles tout en **inclinant le stick droit** dans la direction de l'ennemi.

Input :

Joystick droit et gauche.

Compétence :

Précision.

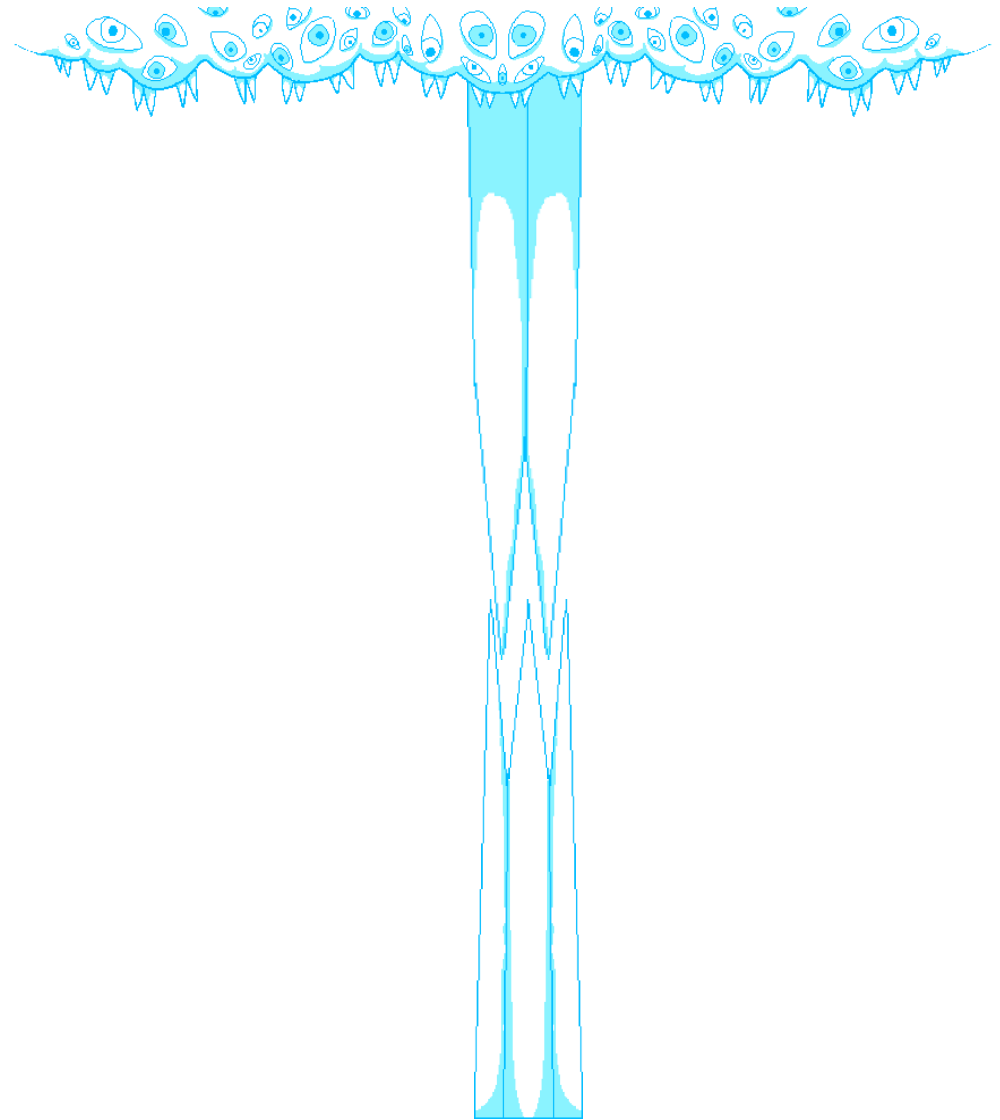


Porte :

Description :

Un **obstacle** qui prend **toute la hauteur de l'écran** et qui **se déplace vers la gauche** de l'écran. A intervalle régulier, **la porte s'ouvre** laissant une fenêtre **d'opportunité** au joueur pour **la traverser**.

La porte **s'ouvre** pendant X beat, puis **attend** X beat avant de **se fermer** sans transition.



Micro Objectif :

Passer entre les portes au bon moment.

Micro-Challenge :

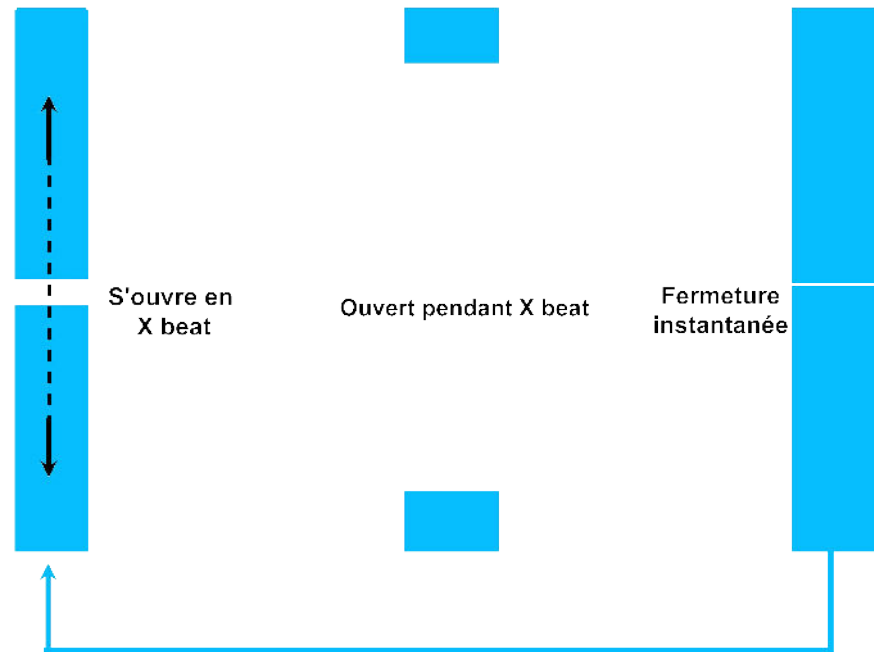
Incliner le stick gauche vers la droite au moment où **la porte est ouverte**.

Input :

Joystick gauche.

Compétence :

Timing.



Bombe :

Description :

Ennemi qui **se déplace constamment vers le joueur**. Au bout de X beat donné à chaque bombe, **elle explose**, ce qui **inflige des dégâts** au joueur s'il **en est proche**. Le joueur ne peut **pas la détruire** mais lorsqu'il lui tire dessus, la bombe **recule de Y pixel** dans la direction opposée à celle dont le joueur a tiré.



Micro Objectif :

Repousser l'ennemi.

Micro-Challenge :

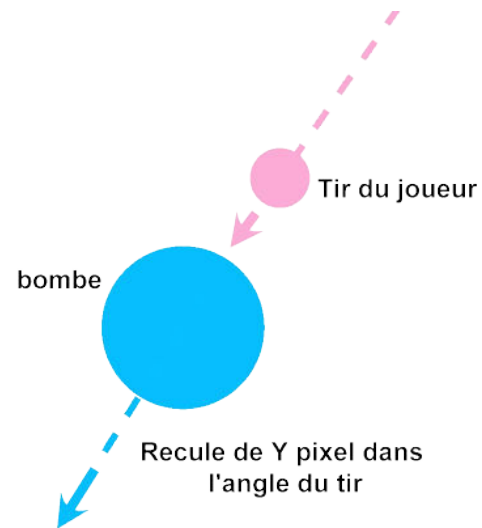
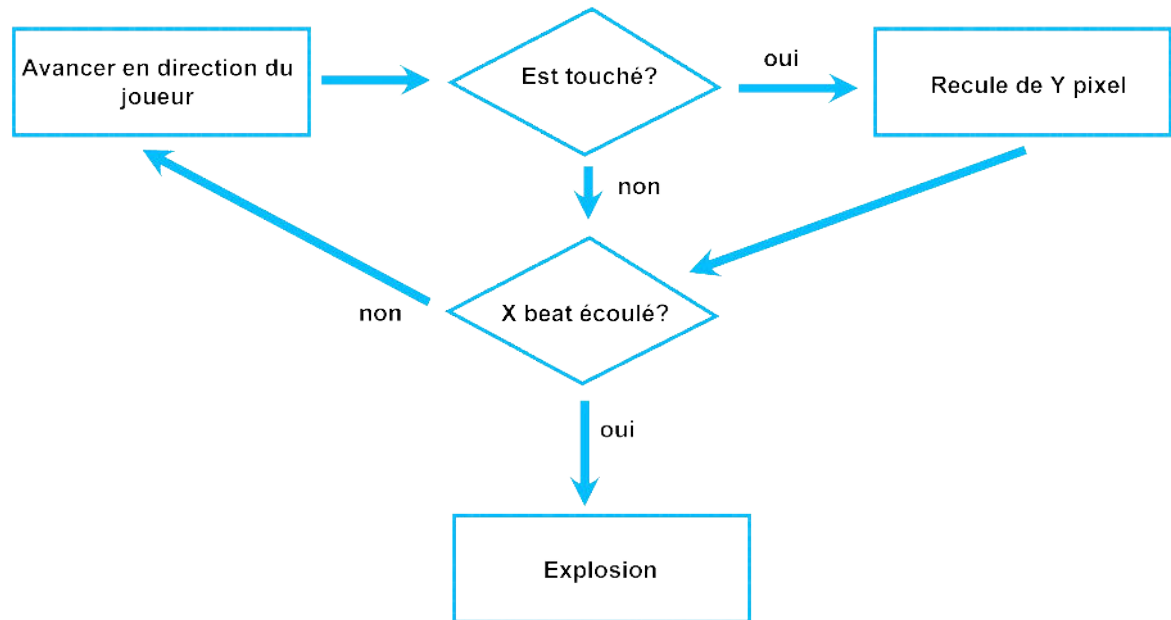
Incliner le stick droit dans la direction de l'ennemi.

Input :

Joystick droit.

Compétence :

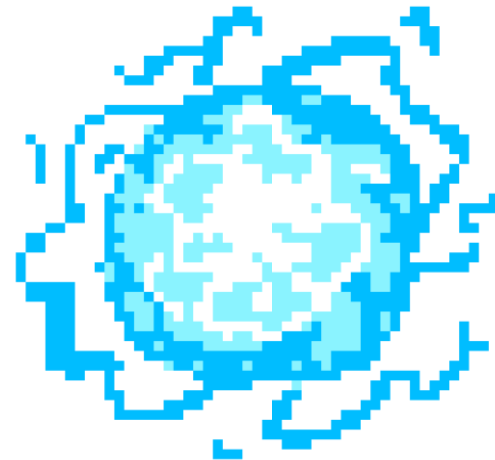
Observation, précision.



Laser :

Description :

Ennemi **projetant un laser** horizontalement **sur tout un rail** ou peut se déplacer le joueur. Avant **l'apparition du laser**, la ligne sur laquelle il apparaîtra est **indiquée par une bande bleue**. Après avoir tiré, l'ennemi à l'origine du laser **sort de l'écran**.



Micro Objectif :

Esquiver les lasers.

Micro-Challenge :

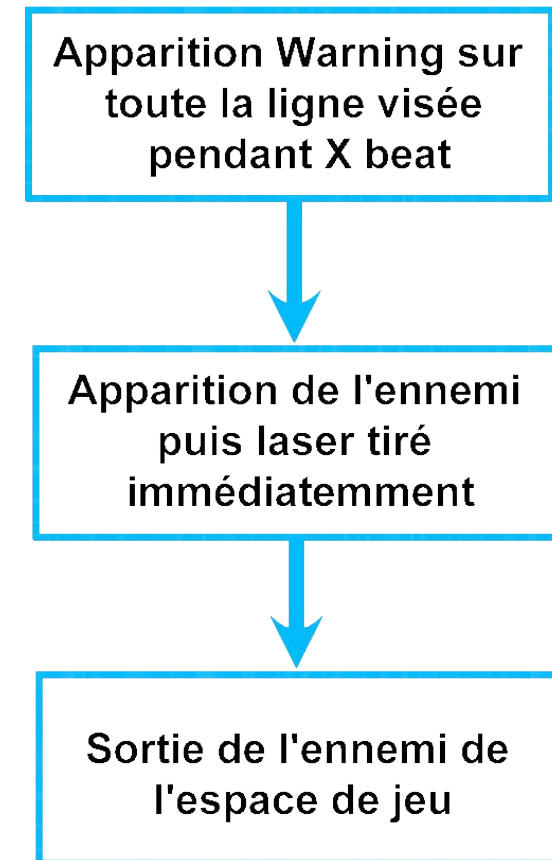
Incliner verticalement le stick
gauche avant l'apparition du laser.

Input :

Joystick gauche.

Compétence :

Timing, Réflexes.



Chauve-Souris :

Description :

Ennemi **apparaissant en haut ou en bas** de l'espace de jeu. A intervalle régulier, l'ennemi **visé le joueur** pendant X beat et **verrouille sa position** (faisant apparaître un faisceau rose). Au bout d'un lapse de temps, il **envoie ensuite un projectile** dans la direction verrouillée plus tôt et attend pendant X beat avant de viser de nouveau.

Cet ennemi **stagne** à l'écran et est donc nécessaire à détruire pour finir une phase.



Micro Objectif :

Détruire l'ennemi.

Micro-Challenge :

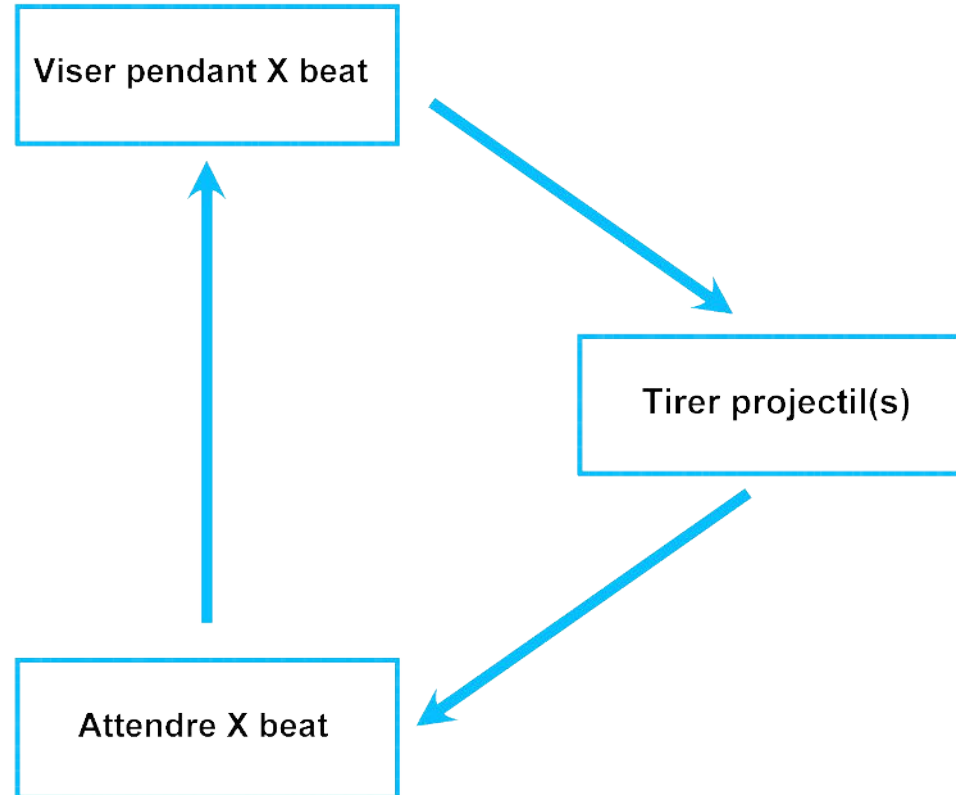
Incliner le stick gauche lorsque le feedback est sur le joueur et **incliner le stick droit** dans la direction de l'ennemi.

Input :

Joystick droit et gauche.

Compétence :

Observation, Précision, Timing.



LD Patterns (RLD):

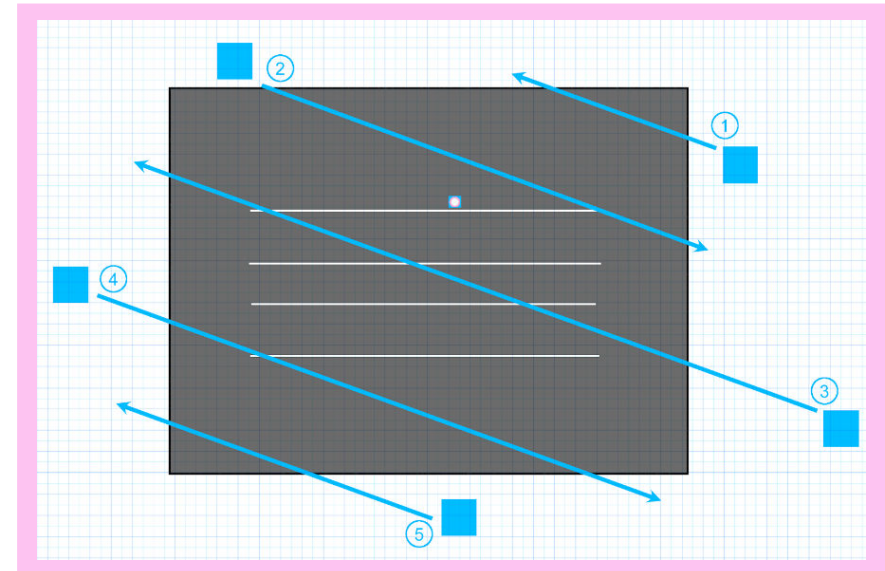
Le level design est **scripté** dans l'entièreté du niveau et ne comporte **pas d'aléatoire**. A cette fin, nous avons créé **unité de temps**, le **Beat(b)** qui équivaut à **0.333 secondes**. Cette unité correspond donc aux beats de la musique et nous permet alors de **caler les patterns** sur le **rythme** de la musique.

Phase 1, 0b-66b (boucle à 65b):

Intro, 0b-33b :

Ces Yeux ne sont **pas calés** sur le rythme et se déplacent à une vitesse de **500 pixels/seconde.**

Le premier passage d'un ennemi se passe **en dehors la zone de déplacement du joueur**, puis le suivant sur **l'extrémité droite**. Ainsi le joueur peut utiliser pour la première fois **les contrôles de tir, sans être en danger**. Le 3eme passe **en diagonale** au centre de la zone de déplacement du joueur, le forçant donc à **se déplacer**.



1. **Œil** : à **6b** = spawn à (820 pixels,80 pixels) à **200°**
2. **Œil** : à **12b**=spawn à (200 pixels, -20 pixels) à **20°**
3. **Œil** : à **18b**=spawn à (820 pixels,400 pixels) à **200°**
4. **Œil** : à **24b**=spawn à (-20 pixels,300 pixels) à **20°**
5. **Œil** : à **30b** = spawn à (300 pixels,620 pixels) à **200°**

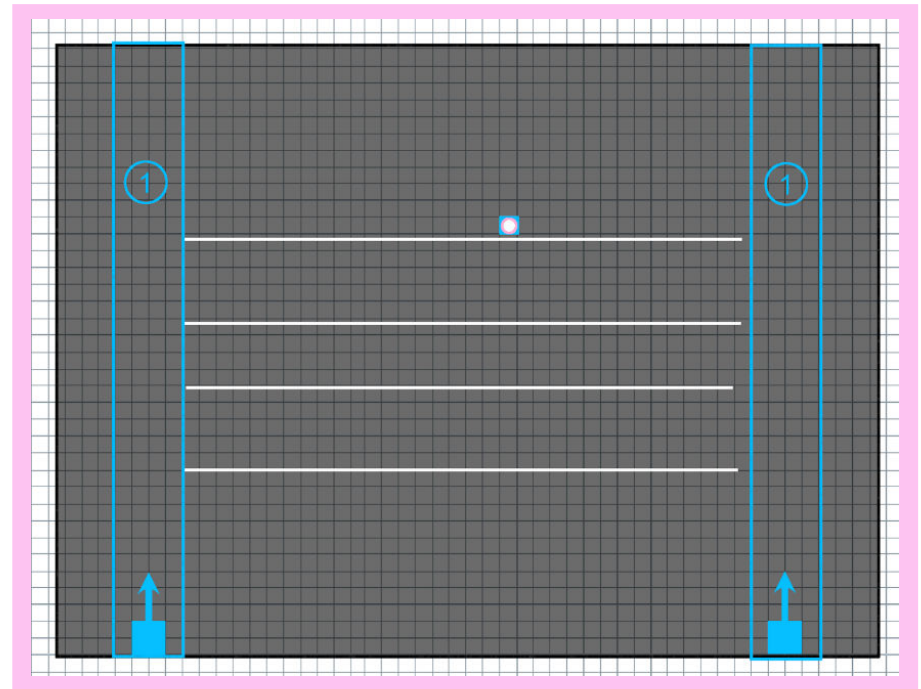
Boucle 1 33b-66b :

1. Petit Œil : A partir de 33b à 66b, un œil est créer à (60 pixels,630 pixels) et un autre à (700 pixels,630 pixels) toutes les 8 beats.

Ces Yeux sont en rythme. Leur angle alterne donc entre 35° et -35° par rapport à leur angle de direction qui est de -90° , et pulsent, tous les 2 beats.

Ils ont une taille $\frac{1}{2}$ fois inférieure à la taille des yeux d'origine.

Ils ont une vitesse de 500 pixels/seconde.

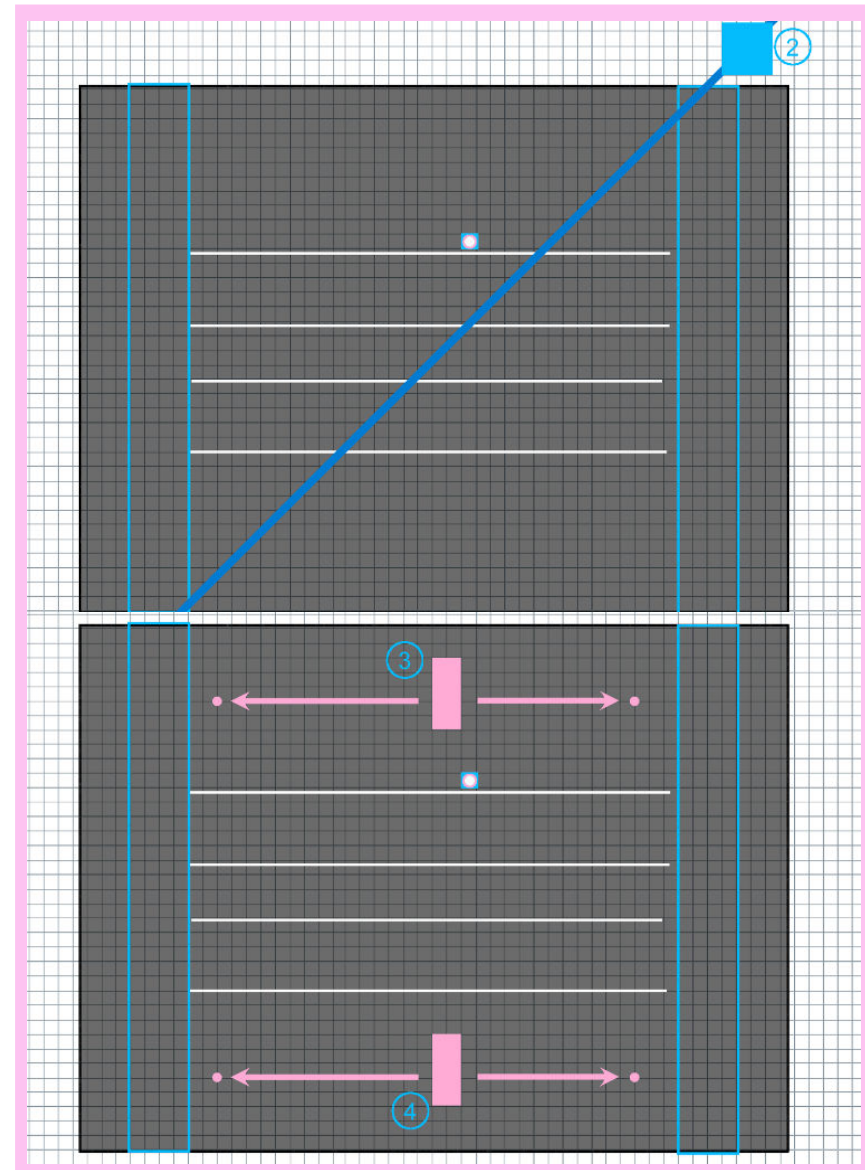


2. Œil : A 36b, 2 yeux sont créés à (800 pixels, -20pixels) avec 4 beats d'écart.

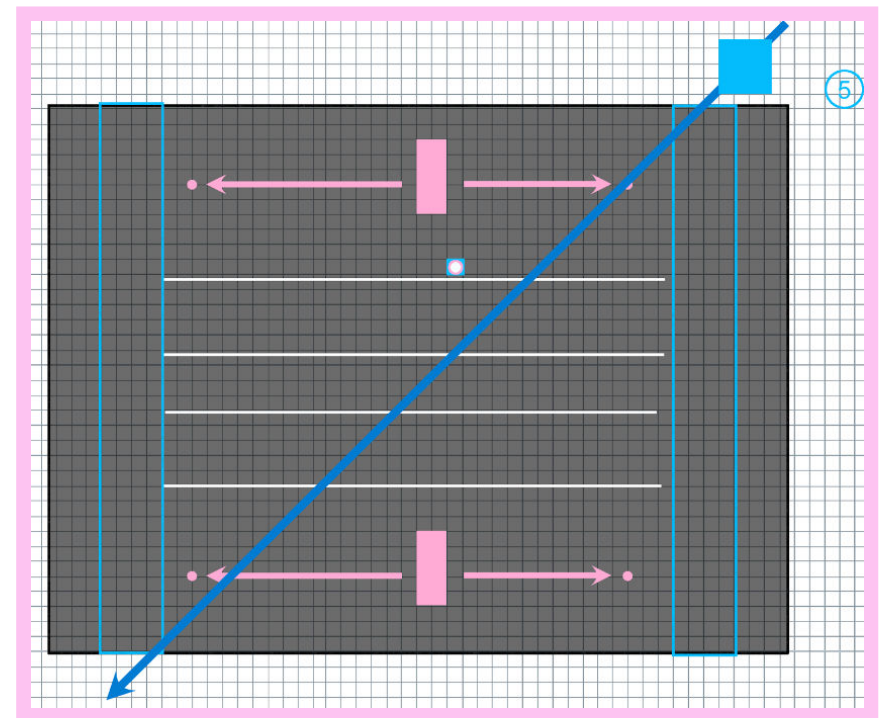
Les Yeux passant sur ce pattern sont tous d'une taille 1. Ils sont **en rythme** et ont un angle de direction de **135°** et une vitesse de **200 pixels/ seconde**.

3. Ange : à 50b = spawn à (400 pixels, -10pixels)
4. Ange : et 24b après 3. = spawn à (400 pixels, 610 pixels)

Tout le pattern des Anges est **calé sur 2 beats** (pairs). Ainsi, ils **attendent** 2 beats à droite ou à gauche, **tirent** puis **se déplacent à l'opposé**, déplacement qui dure 2 beat, puis attendent 2 beats et le pattern se répète ainsi.

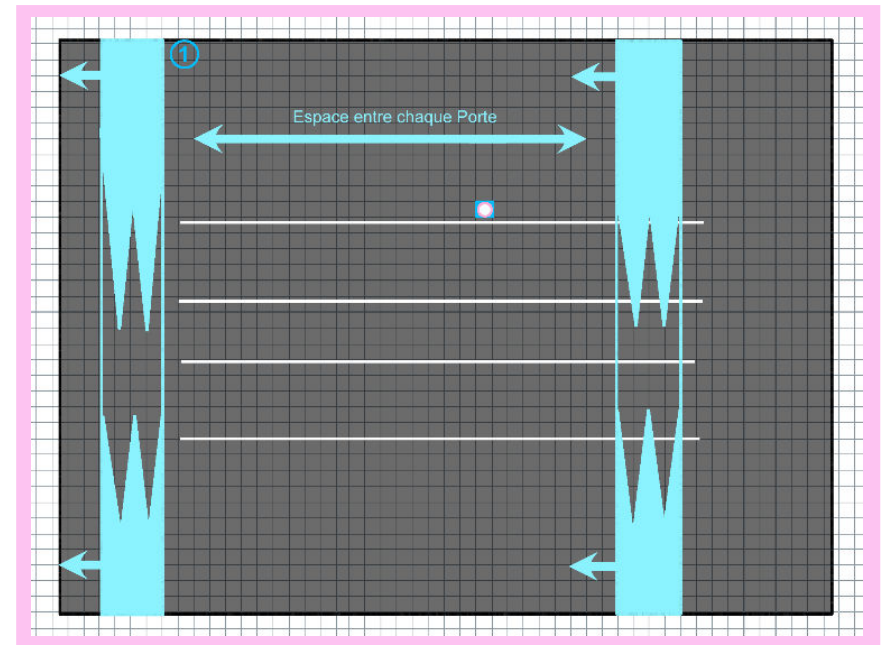


5. Boucle Œil: A partir de 60 b et tous les 24b jusqu'à la fin de la boucle, 2 Yeux créés à (800 pixels, -20pixels) avec 4 beats d'écart.



Phase 2, 67b-131b (boucle à 130b):

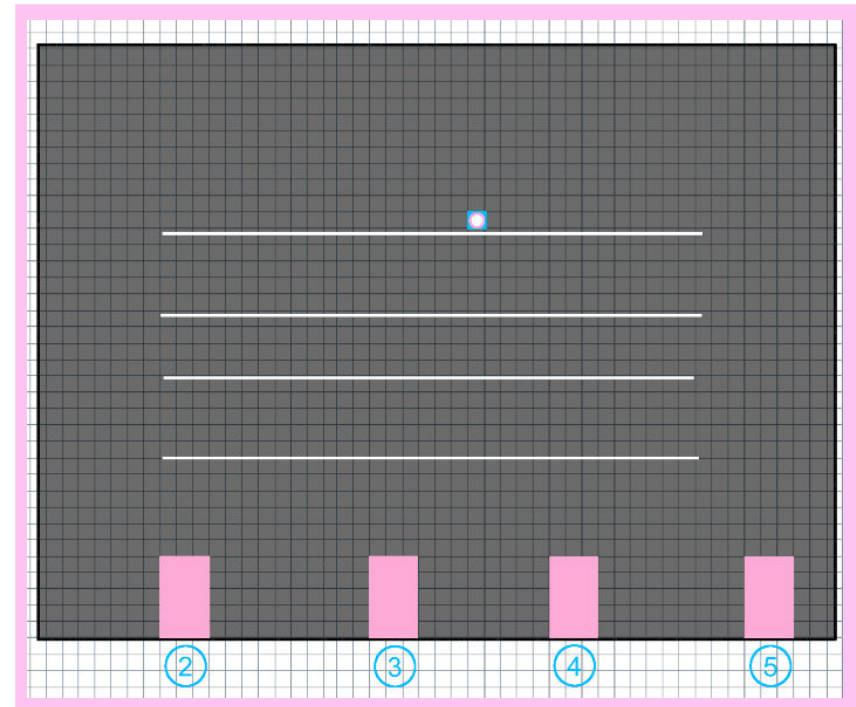
1. Porte : A partir de **72b** jusqu'à **131b**, une porte apparaît aux coordonnées (900 pixels, 500 pixels) tous les **8 beats**. La porte se referme tous les **4 beats**, puis se rouvre en **2 beats** et attend le 4eme beat pour se refermer et donc répéter son pattern. Cet ennemi est calé sur les Snare (beats impairs).
Les portes se déplacent à une vitesse de 200 pixels/seconde.



2. Plante : à 70b = spawn à (100pixels,700pixels)
3. Plante : à 86b = spawn à (350pixels, 700pixels)
4. Plante : à 102b = spawn à (550pixels,700pixels)
5. Plante : à 118b = spawn à (750pixels,700pixels)

Les Plantes tirent tous les 4 beats, puis se déplacent vers le joueur pendant 2 beats et répètent ce pattern. Cet ennemi est calé sur les kicks (beats impairs). Elles

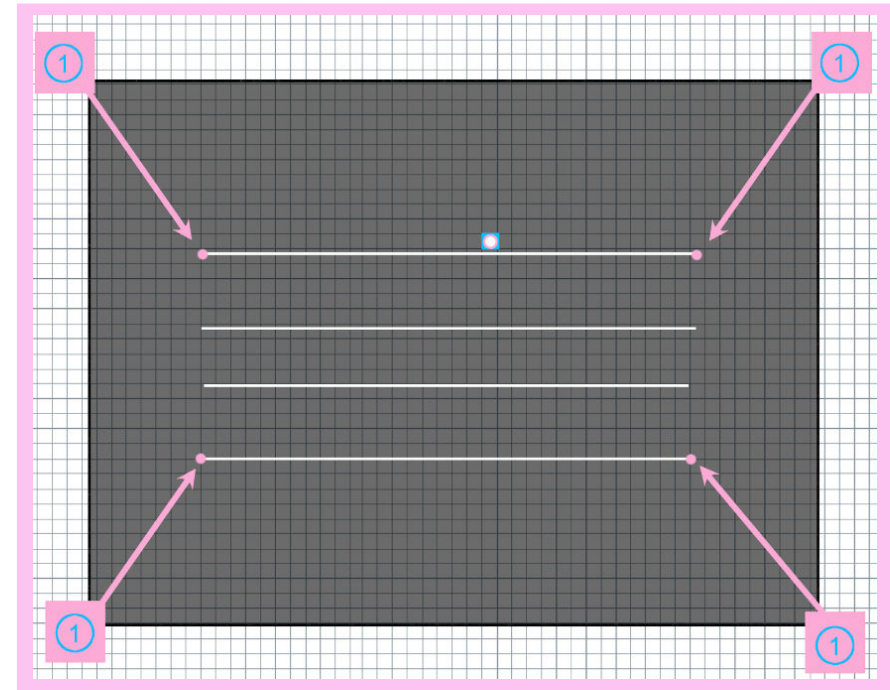
tirent 5 projectiles à la fois.



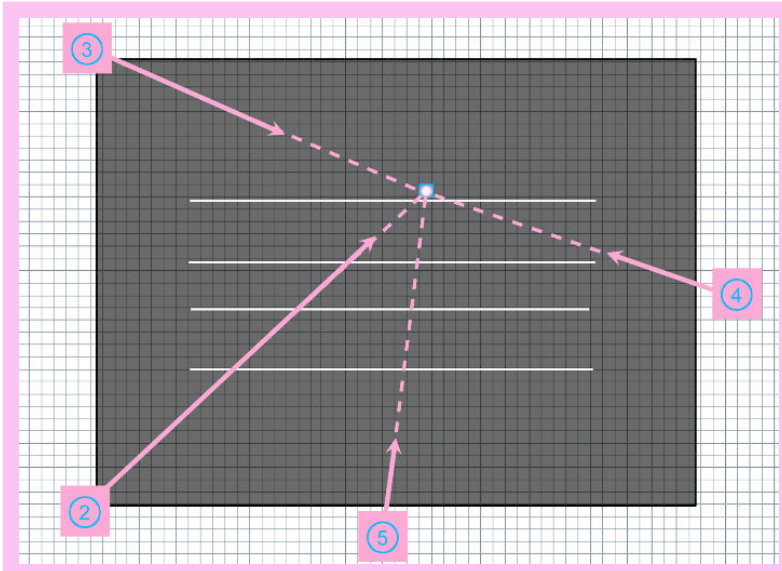
Cette phase fait alterner les plantes et les portes, challenge de précision et de timing.

Phase 3, 132b-189b :

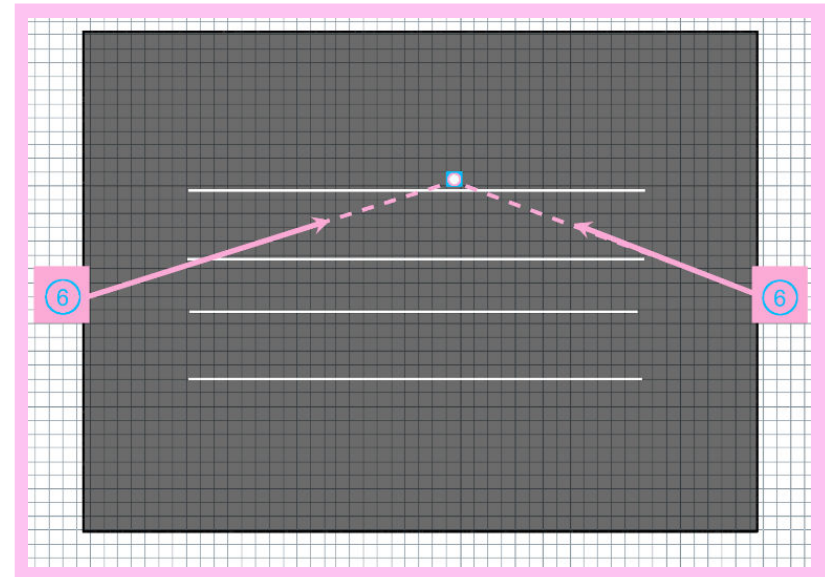
1. Bombe : à 134b = spawn à
 - (-20pixels, 610 pixels) qui explose à 147b
 - (-20pixels, -10 pixels) qui explose à 151b
 - (810 pixels, -10pixels) qui explose à 155b
 - (810 pixels,610 pixels) qui explose à 159b



Ces Bombes apparaissent aux 4 coins de la zone du joueur et se déplacent vers le point leur étant assigné.



2. Bombe : à 158b = spawn à (-20 pixels,610 pixels) qui explose à 163b
3. Bombe : à 162b = spawn à (-20 pixels,-20 pixels) qui explose à 167b
4. Bombe : à 166b = spawn à (820 pixels,300k pixels) qui explose à 171b
5. Bombe : à 170b = spawn à (400 pixels,620 pixels) qui explose à 175b



6. Bombe : à 174b = spawn à
 - (-20 pixels,300 pixels) qui explose à 179b
 - (820 pixels,300 pixels) qui explose à 183b

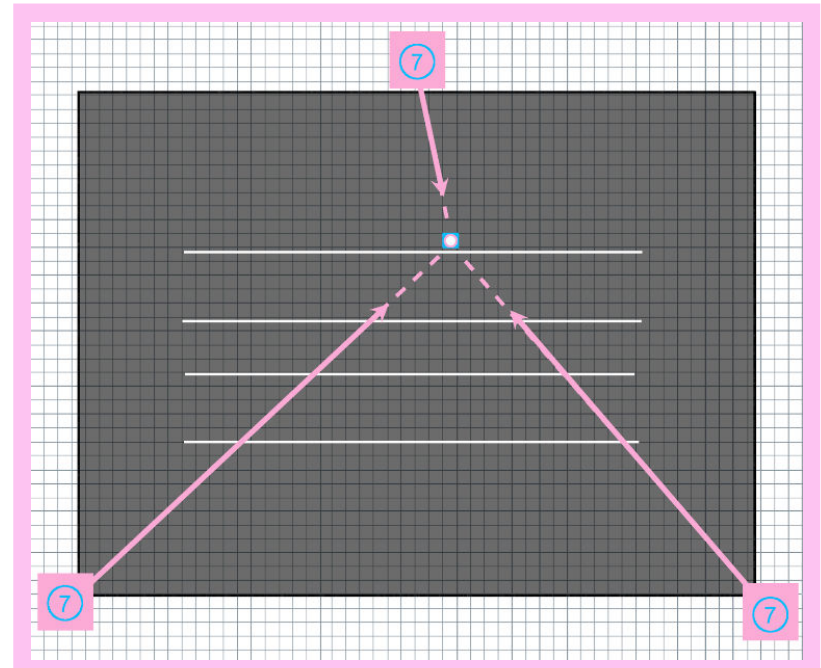
7. Bombe : à 180b = spawn à
- (400 pixels, -20 pixels) qui explose à 187b
 - (600 pixels, -20 pixels) qui explose à 187b
 - (820 pixels, 600 pixels) qui explose à 187b

Hormis les 4 premières bombes, les bombes se dirigent toujours vers le joueur à une vitesse de 150 pixels/seconde.

Lorsqu'une bombe est touchée, elle recule de -40 pixels dans la direction où elle a été touchée.

Les bombes explosent automatiquement à la valeur qui leur a été assignée.

Cet ennemi est calé sur les Snares (tous les 4 beats, calé sur les beats impairs).



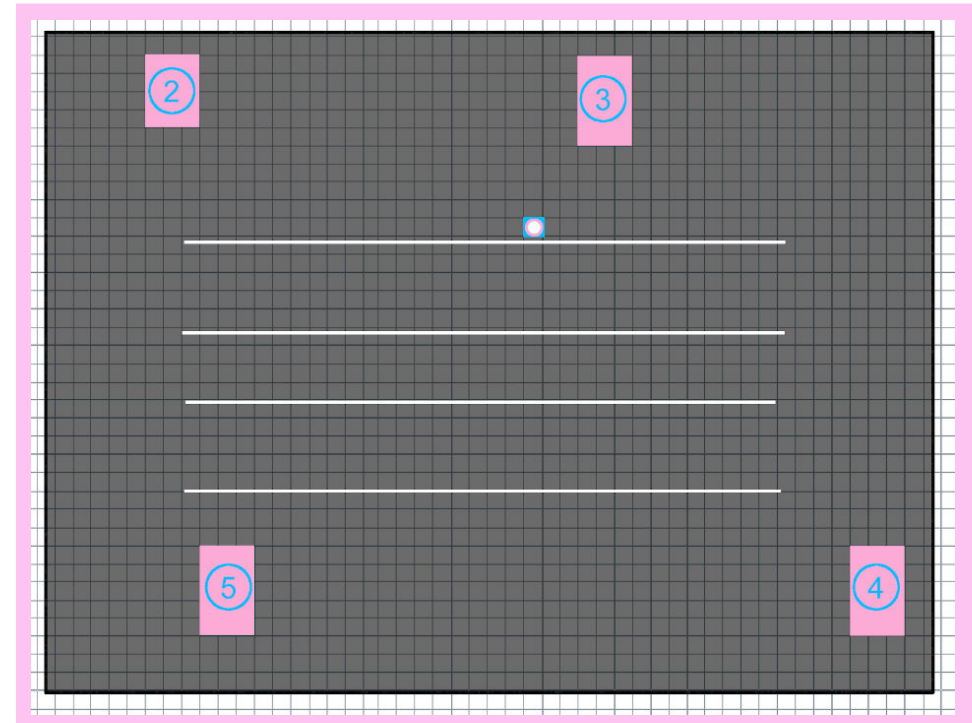
Phase 4, 190b-260b (boucle à 259b):

1. Laser : Ils apparaissent tous les 4 beats à partir de 191b jusqu'à 253b. D'abord le Warning apparaît pendant 2 beats, puis le laser est tiré au 3eme beat et l'ennemi se retire de l'écran, avant que de nouveau(x) Warning(s) apparaissent aux 4eme beat. Cet ennemi est calé sur les Kick (beats impairs).

Pattern apparition laser par numéro d'apparition :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RAIL 1	X		X		X	X				X	X		X		X	X
RAIL 2									X			X		X		X
RAIL 3								X				X	X		X	
RAIL 4		X		X		X	X				X			X	X	X

1. Chauve-Souris : à 200b = spawn
(50 pixels, -50 pixels)
2. Chauve-Souris : à 229b = spawn
(600 pixels, -50 pixels)
3. Chauve-Souris : à 235b = spawn
(800 pixels,650 pixels)
4. Chauve-Souris : à 257b = spawn
(100 pixels,650 pixels)



Les Chauve-Souris restent immobile à l'écran. Elles rentrent en mode visée pendant **2 beats**, puis tirent au 3eme et attendent le 4eme pour recommencer leur pattern. Cet ennemi est calé sur les **Snares (beats pairs)**. La vitesse de leurs tirs est de 400 pixels/seconde.

6. Laser (Boucle):

De 255b jusqu'à l'outro, les lasers, ayant la même cadence d'apparition que les précédents mais avec ce pattern de position se répétant :

Rail 1	1		1	
Rail 2		2		2
Rail 3	1		1	
Rail 4		2		2

Le pattern d'apparition des Yeux, Portes et Lasers effectue une boucle, soit dès le début de la phase, soit lorsque la phase commence à se boucler sur elle-même car le joueur n'a pas encore éliminé les bons ennemis.

Cela permet, lorsque le joueur finit de détruire les ennemis roses alors qu'une nouvelle boucle vient de commencer, de le garder actif, même si la fin de cette boucle sera un moment de jeu plus calme.

Signs & feedback, tests utilisateurs :

Signs & Feedback

	Feedbacks visuels	Feedback sonore
Sur tout le niveau:	Le joueur est touché Ennemi touché Ennemi tué	Le joueur est touché Bruit ennemi touché Bruit ennemi tué
User Interface	Bouton play Bouton classique (flèches, boutons...) Transition entre niveaux (woosh) Bouton Press Start	Bouton play Bouton classique (flèches, boutons...) Transition entre niveaux (woosh) Bouton Press Start
Animation (Intro)	Animation quand le joueur lance le niveau pour la première fois	Musique étouffée lorsque le casque est mis
Musiques		Musique du menu Musique du niveau (6 phases + transition) Jingle mort Jingle victoire

Phase musicales	Ennemis	Feedbacks visuels	Feedback sonore
Phase 1 (34b - 66b)	Oeil (tt 7.999s)	Dégâts et destruction oeil	Dégâts et destruction oeil
	Petit Oeil (tt 2.33s)	Dégâts et destruction petit oeil	Dégâts et destruction petit oeil
	Ange (50t / 63t)	Dégâts et destruction Ange Dégâts et destruction Follower	Dégâts et destruction Ange Dégâts et destruction Follower
Phase 2 (67b - 130b)	Plantes(70t/86t/102t/118t)	Tir (1.333s) Destruction plantes	Tir des plantes Destruction plantes
	Murs (tt 2.666s)	Fermeture et ouverture	Fermeture mur
Phase 3 (131b - 189b)	Bomb (tt 2t)	Destruction des bombes (tt 1.333s)	Explosion des bombes
Phase 4 (190b - 273b)	Boules électriques (alt Warn 1t / tir 1t)	Laser	Tir laser
	Chauves souris (191t/199t/221t/239t/249t)	Destruction chauve souris	Destruction Chauve Souris

Jusqu'à la fin du développement, les **playtest** et **retours utilisateurs** ont permis d'identifier les problèmes les plus récurrents du jeu :

On a du mal à identifier sur quels instruments sont rythmé les ennemis :

- Tous les ennemis sont maintenant **codés sur les percussions** du niveau, qui sont les instruments le plus facilement

Le jeu est peu lisible :

- En effet, la grande quantité de feedback faisait clignoter tout l'écran. Trouver un équilibre entre les **feedbacks musicaux** et les **feedbacks du Shoot em' up** a été très difficile.

On peut finir le niveau sans tuer personne / les ennemis s'accumulent rapidement :

- Notre niveau durait 1min30, précisément le temps de la musique. Ce qui fait que peu importe ce que fait le joueur, le niveau avance, et dans le cas où il ne détruit pas les ennemis, ils s'accumulent sur l'écran. Nous avons donc créé une **nouvelle mécanique** pour notre niveau, **les boucles**. Nous avons **divisé les phases** du niveau pour faire que le joueur doit **tuer tous les ennemis** de celles-ci pour passer aux suivantes. La **musique** à également été **divisée**, afin qu'elle **boucle** tant que le joueur n'est pas passé à la partie.

Manque de feedbacks visuels et sonores :

- Les sprites ennemis réagissent quand ils se prennent un **coup** et quand ils **meurent**. Pour créer une cohérence entre la musique et les effets sonores, les sons que font les ennemis quand ils meurent proviennent de la musique du niveau.

La difficulté est dérégulée :

- Les joueurs jugent la difficulté comme "dérégulée", surtout par rapport au **nombre d'ennemis** sur l'écran et on a décidé de simplifier un grand nombre d'éléments, tout en gardant un niveau de jeu exigeant. Notre jeu est donc devenu un "**die and retry**" avec une **difficulté plus maîtrisée et mesurée**, qui demande au joueur de rejouer au jeu pour **apprendre les patterns** des ennemis, trouver des stratégies et finir le niveau. **L'intensité** du niveau **varie avec la musique**, avec une partie plus calme avant la phase la plus compliquée pour laisser respirer le joueur.

Direction artistique

Direction Artistique visuelle :

Thèmes envisagés :

Thème musical : Le personnage incarné était un **maître de partition** qui se **déplacerait** sur les **lignes de partitions** musicales. Les ennemis étaient de forme simple, **inspirés de notes de musique** auxquelles ils auraient correspondu.

Thème magie et musique : Le thème original a ensuite **évolué** vers un **magicien contrôlant de la musique** et se battant contre de la musique.



Cependant ces deux thèmes plaçaient les **ennemis** comme étant la **source de la musique**, ce qui n'était **pas cohérent** avec **l'évolution du gameplay** du jeu. En effet, le pattern des ennemis est **influencé et calé** sur le rythme mais ce n'est **pas eux qui créent la musique**. Nous avons néanmoins **gardé le thème du magicien**. Nous avons aussi gardé le **même système de barre**, inspiré des cordes d'un instrument et partition.

Thème gardé :

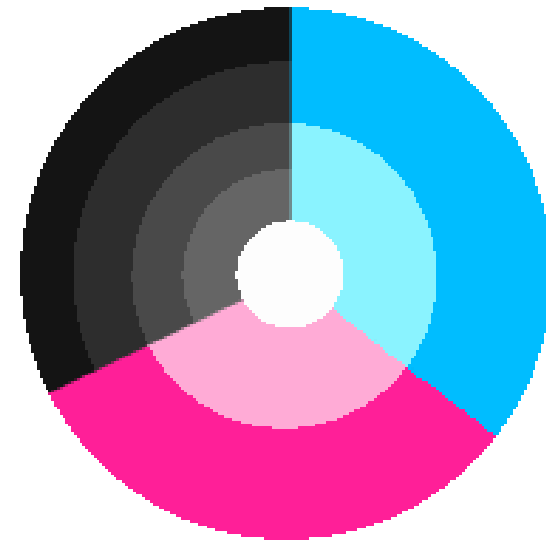
Le personnage incarné par le joueur est un **jeune magicien** faisant du **skate** sur **4 rampes** qu'il crée grâce à sa magie. Pendant son chemin, il **écoute de la musique** sur laquelle tout le monde autour de lui va s'en trouver affecté, notamment tout son **voisinage de monstres** et **d'êtres magiques** qui n'apprécient guère son hobby et qui **vont l'attaquer**.



Palette de couleur :

Pour illustrer ce thème, nous avons choisi du **pixelart**, dans une fenêtre de 800 par 600 pixels, afin de coller à notre **inspiration rétro**. Pour la palette de couleur de l'environnement, nous avons choisi **8 différentes couleurs** ainsi que le blanc. Il y a 4 nuances de gris, utiliser pour les différents fonds, qui sont sombre pour faire **opposition** au éléments importants qui sont en 2 nuances de bleu et/ ou de rose ainsi que blanc. Seul Wilfred, le personnage contrôlé par le joueur, possède des couleurs supplémentaires.

La palette des ennemis dépend de s'ils doivent être obligatoire à tuer afin de compléter une phase et de terminer la boucle. Ainsi, les ennemis devant être tuer, ceux qui stagnent à l'écran, sont dans la **palette rose et blanc**, tandis que les ennemis secondaires sont dans la **palette bleu et blanc**.



Wilfred :

Wilfred est un jeune magicien, il est donc inspiré de l'**image classique des magiciens**, portant une **large cape** descendant jusqu'à ses pieds. Cependant, Wilfred est un magicien **modernisé**, ainsi sa cape comporte une **capuche de sweat**, des **couleurs lumineuses** et des **tatouages**.

Les **tatouages** et **motifs** présents sur Wilfred sont inspirés de **cercles magiques** et rappelant des **motifs de lune et de soleil**, qui sont deux éléments ayant une place importante dans la **magie et la sorcellerie**, avec les différentes phases de lune et éclipse solaire symbole de force pour la pratique de rituels.



**Exemple typique de magicien:
Merlin L'Enchanteur**



Concept art de Wilfred

L'animation de Wilfred le représente en train de slider sur un rail, ayant un mouvement de rotation léger de son skate.

Son animation de mort le représente chutant de son skate, celui-ci toujours sur le rail.

Sprite sheet de Wilfred



Ennemis :

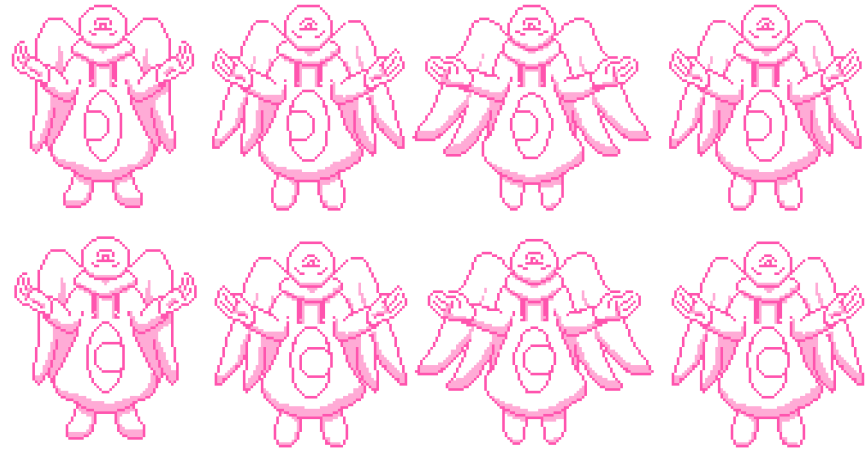
Ange :

Les Anges sont des **créatures magiques** inspirées par un mélange des **dépictions des anges** dans les textes bibliques et la représentation de ceux-ci dans les **médias modernes**. Cela résulte ainsi en un **ange humanoïde** composé d'un **œil sur son front**, deux yeux fermés, et d'un dernier **sur son ventre**, qui **suit du regard** la position du joueur.



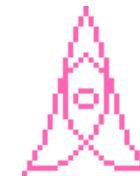
Reference de " biblically accurate angel"

Son **animation** le représente **oscillant entre ailes et bras déployés** et repliés.



Sprite sheet d'Ange

Son **projectile** représente **une flèche** contenant un **œil**.



Sprite projectile

Œil:

L'œil est un monstre inspiré de l'univers **horifique** notamment présent à **Halloween**.

Ce monstre est seulement composé d'un **globe oculaire** et une **partie de nerf**.

Sprite sheet d'Œil



Plante:

Cet ennemi est inspiré des **dionées attrape-mouche**, qui sont des **plantes insectivores**. Dans cet univers, ces plantes vivant dans la forêt sont agressives et capables **d'expulser des graines de leurs bouches** pour attaquer les passants.



Son **animation** l'a représenté **gesticulant** et **refermant** légèrement **la bouche**.



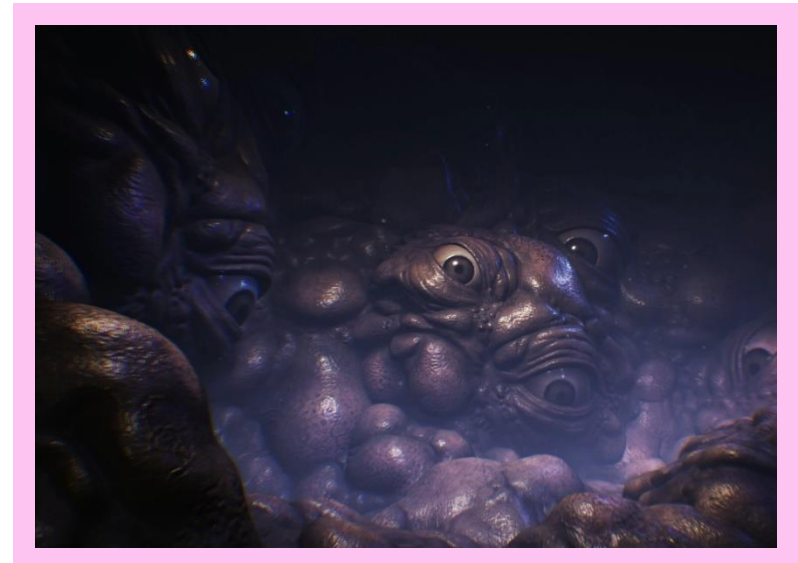
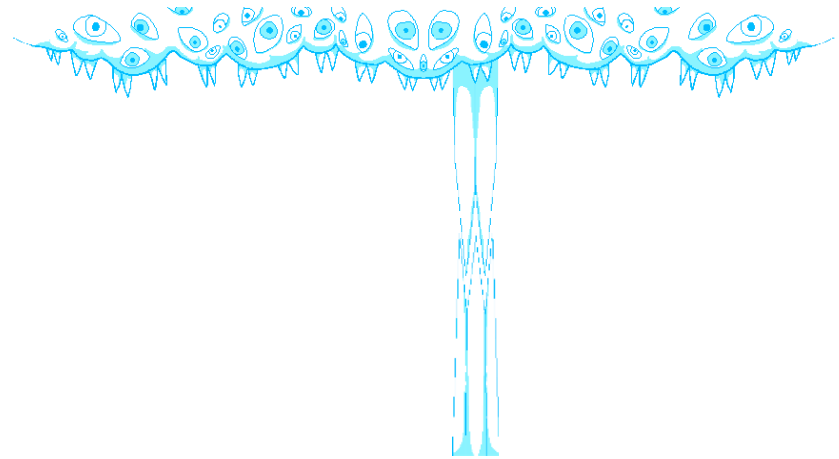
Sprite sheet de Plante



Sprite de tir

Porte:

Les portes sont un nuage constitué d'un **amas de chair et d'œil** qui **referme ses dents** sur le joueur pour l'attaquer. Ce monstre est inspiré d'un style **soft de body horror**, notamment inspiré par les **"flesh wall"** de **Little Nightmares II**.



Bombe:

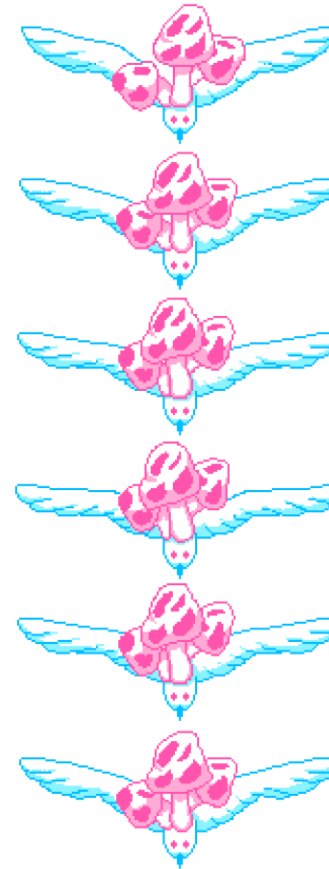
Les bombes sont formées d'une espèce de **champignons** qui **ont pris possession d'oiseaux**, les utilisant afin d'atteindre d'autres espèces pour les **infectés de leurs spores**, en **explosant près d'eux**.

Ils sont inspirés du **Ophiocordyceps unilateralis**, une espèce de champignons qui **infecte les fourmis**. Ses couleurs sont bleues et roses car, à contrairement des autres ennemis, **sa destruction est nécessaire** pour passer à la phase suivante mais il **se détruit de lui-même**.

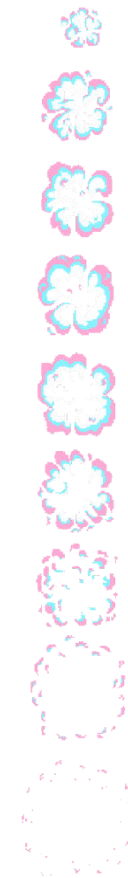


Son animation le représente entrain de **planer vers le joueur.**

Son explosion représente un **nuage se dispersant** petit à petit à l'instar d'un **nuage de spore.**



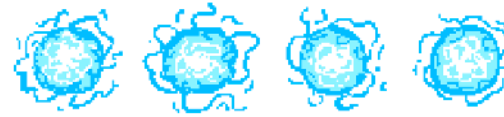
Sprite sheet de bombe



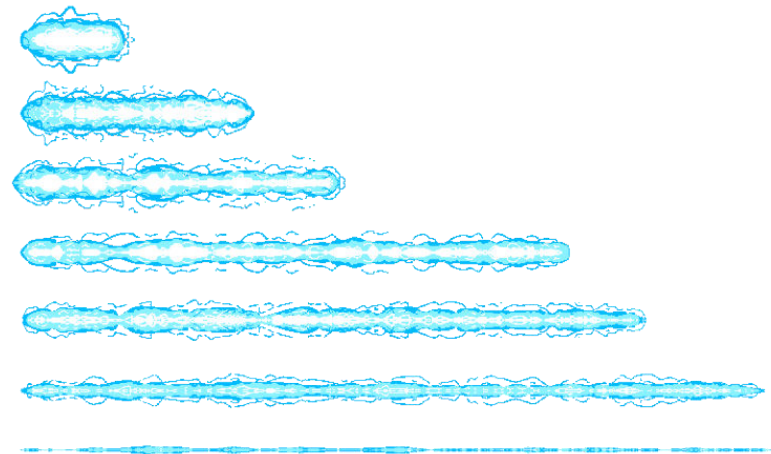
Sprite sheet de l'explosion

Laser:

Les lasers sont créés par des **êtres électriques** présents sous forme de **boule bleuté** se déchargeant sur les rails sur lesquels se déplace le joueur.



Sprite sheet de boule électrique



Sprite sheet de laser

Chauve-Souris :

Cet ennemi est inspiré des **chauves-souris** de la famille des **oreillards**, normalement des animaux de petite taille, ici étant **géants** et ayant une **énorme bouche affamée** sur le ventre qui tire des projectiles sur le joueur.



Son projectile
représente une onde,
faisant écho à
l'**écholocalisation** des
chouves-souris.



Sprite sheet de chauve-souris

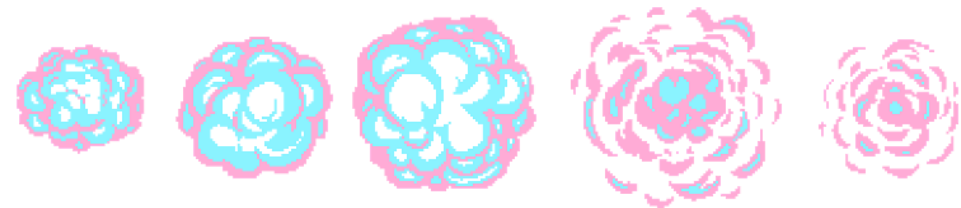


Sprite de projectile

Explosion:

Cette explosion existe en **deux tailles**, une **normale** et une **2 fois plus petite**. La normale apparaît à la **mort de chaque ennemi**, tandis que l'autre apparaît à la **destruction de projectile**, comme ceux des anges.

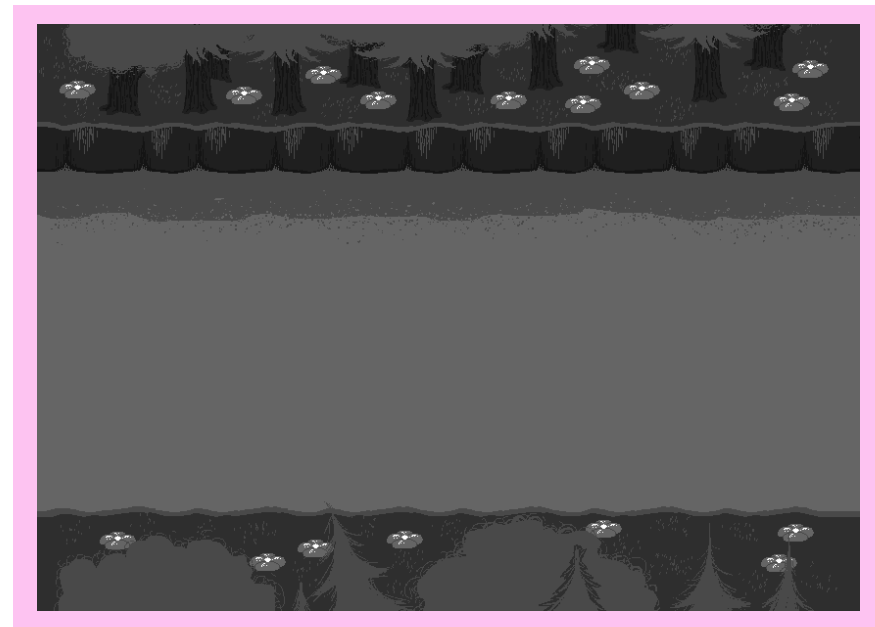
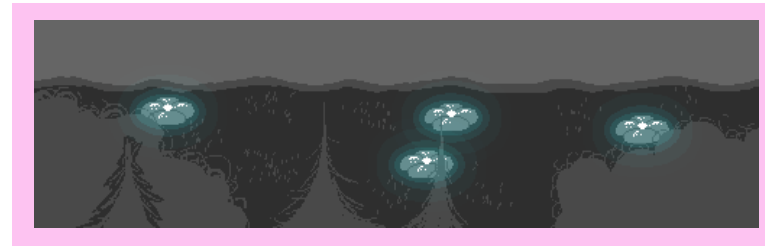
Toutes les destructions d'ingrédients ennemis font apparaître cette explosion à part les bombes, qui ont leur propre animation.



Sprite sheet de l'explosion

Fond :

Le fond du niveau représente un **chemin en bord de forêt** en **nuances de gris** afin de mieux faire **ressortir le personnage et les ennemis**. La seule couleur est une surbrillance bleue sur les fleurs et les rails qui pulse en rythme avec la musique.



Animation :

Afin d'**introduire** brièvement le joueur dans **l'univers du jeu**, nous avons décidé d'avoir une **introduction au niveau animé**. Ceci permet de montrer au joueur que **Wilfred crée les rampes** grâce à sa magie, puis **lance la musique** et **enfile son casque** audio. Lorsqu'il appuie sur **Play**, le volume de la **musique augmente** jusqu'à atteindre son volume normal lorsque le casque est mis sur ses oreilles. Après ceci, **les couleurs de l'environnement transitionnent** jusqu'à devenir les couleurs qui apparaissent une fois le niveau lancé.

Avec cette animation du **personnage principal mettant un casque**, cela **incite aussi le joueur** à en utiliser un afin d'être plus immergé dans la musique.

Lien : <https://youtu.be/Xd0Z1MzujG0>

Direction Artistique sonore :

Le style rétro de la musique :

Comme indiqué dans le titre : "SK8-beat" notre intention était de créer un SHMUP dans un **style rétro** et **électronique**.

Toutefois, nous voulions créer un jeu qui **se démarque** des SHMUP classiques où l'on joue un vaisseau dans l'espace. Nous voulions mettre en avant le **game feeling** de SK8-Beat, et ce par la **musique**. C'est pourquoi on s'est beaucoup investi pour créer un système de jeu différent et entraînant.

Le style électro de la musique :

Nous voulions que les joueurs **aient envie de rejouer** au jeu, malgré leur défaite, malgré la difficulté du niveau. C'est pourquoi nous avons décidé de créer un SHMUP avec une **musique électronique entraînante**. Celle-ci est très rythmée, comme le niveau.

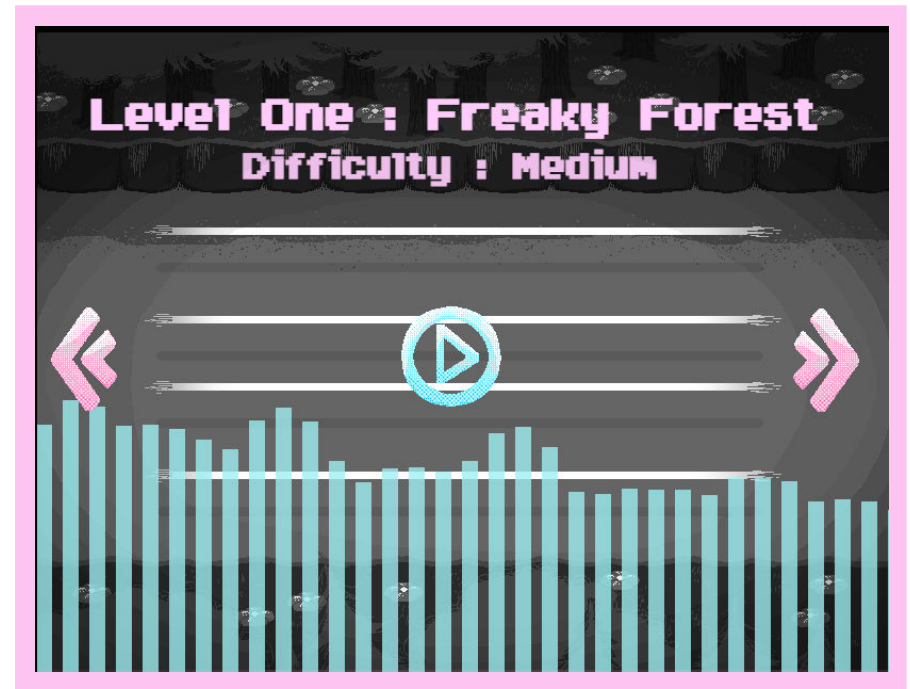
Pour éviter d'avoir une musique "papier peint" redondante, comme on peut en trouver dans des jeux d'arcades, un style électronique permet de rendre la **musique plus vivante, intense** et surtout **dynamique** ! La musique en elle-même est plutôt courte et elle est divisée, comme le niveau, en plusieurs phases très distinctes.

Au niveau sonore et visuel, le joueur n'a pas l'impression que le niveau se répète car il évolue, avec de nouvelles vagues d'ennemis, de nouveaux morceaux musicaux. Ceci lui donne envie d'aller au bout du niveau, pour entendre la musique en entier et découvrir tous les ennemis.

UI :

A propos de l'interface utilisateur, nous sommes de manière globale partis sur une esthétique reprenant le **code couleur du niveau** c'est à dire principalement composé de **rose** ainsi que de **bleu**.

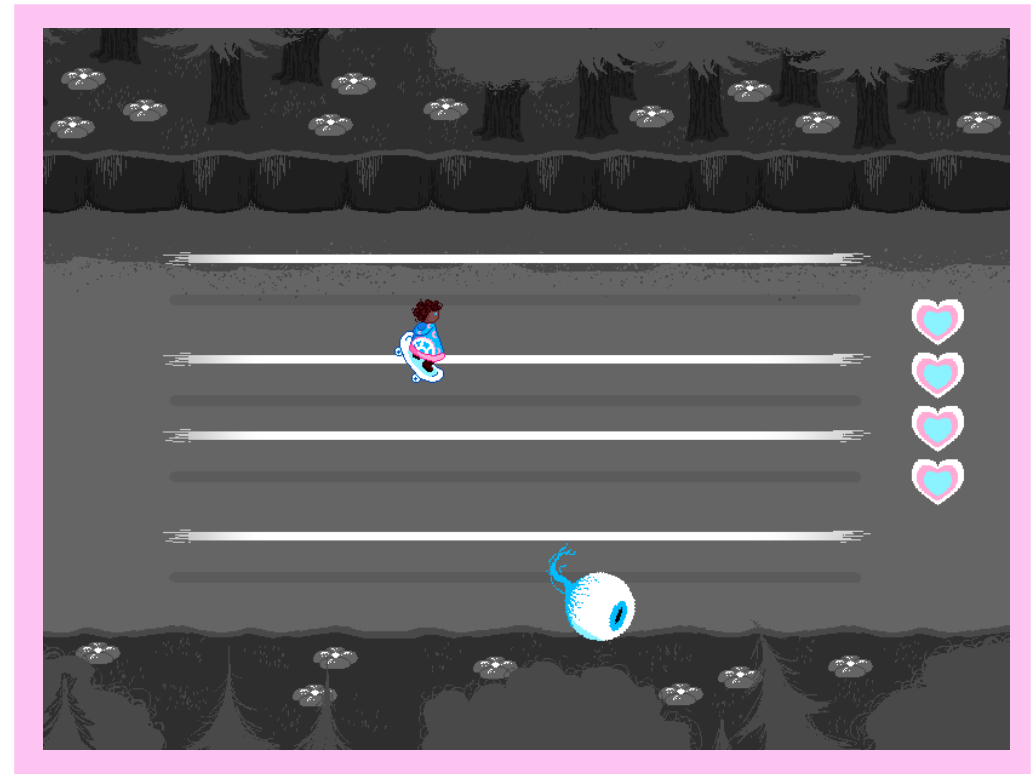
Pour l'interface du choix du niveau, nous avons décidé de partir sur une **interface intra diégétique** où le joueur choisit son niveau sur un **baladeur numérique**. Nous sommes partis sur cette idée afin de **rapidement immerger le joueur** en captivant son attention, il a l'impression **d'incarner l'avatar** choisissant sa musique.



Au niveau de l'interface **dans le niveau**, nous avons décidé d'une interface **très simple** ne montrant **seulement les points de vie** du joueur afin de ne **pas obstruer la vue du joueur**.

Dû à la **saturation** (en éléments à l'écran) de l'écran **souvent élevée**, nous avons placé les **points de vie sur le côté** de l'écran afin qu'il ne soit **simple de regarder** ses points de vie **en pleine partie**.

De plus, les **cœurs** sont **bleus et roses** avec un **contour blanc** afin de bien pouvoir le **distinguer des potentiels ennemis passant derrière**.



Lien du proto de SK8 Beat :

<https://kahnetop.itch.io/sk8-beat>