

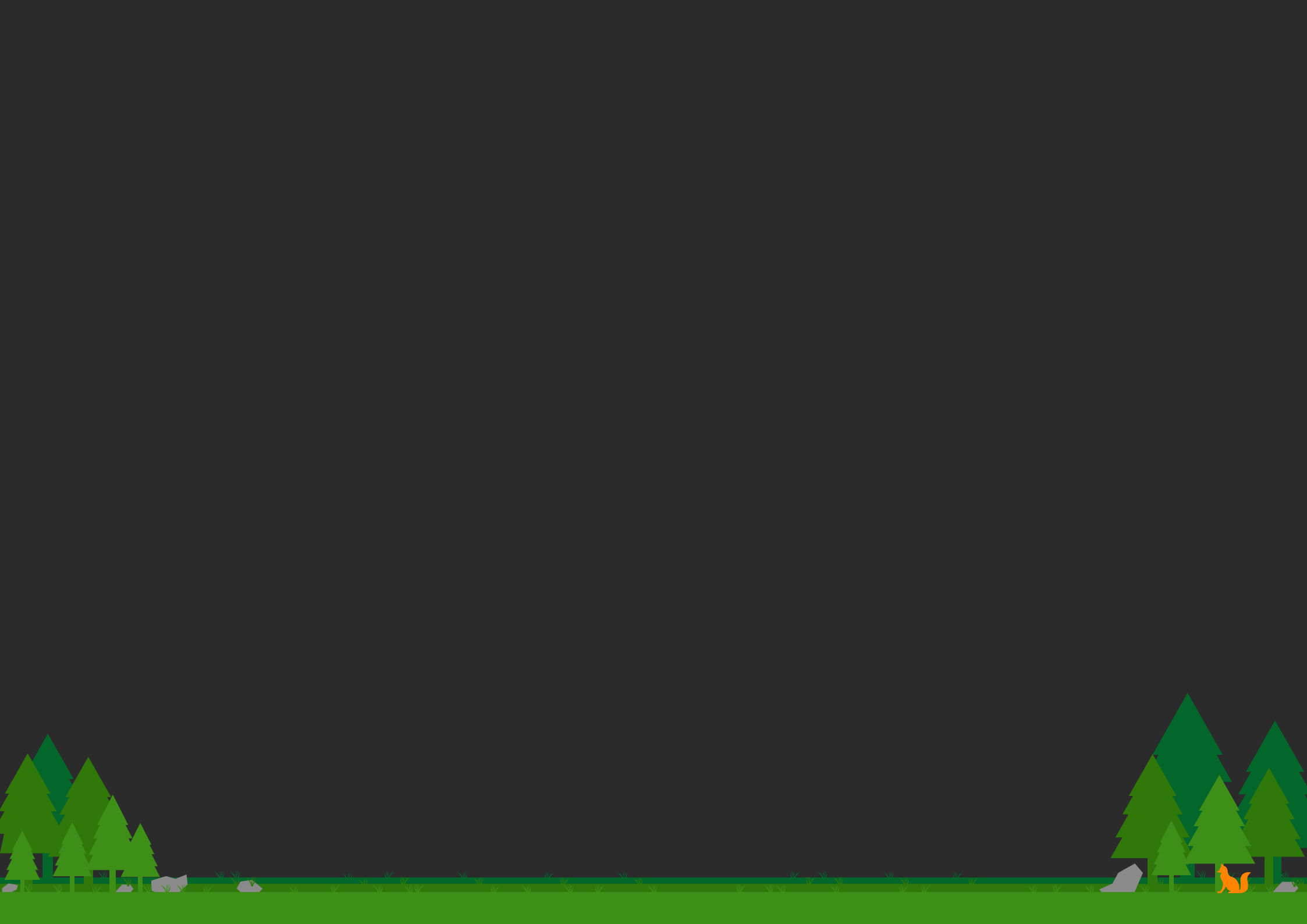


REBORN

Document de conception

Gonzalo Bourdieu, Guillaume Charier, Lucas Huriaux, Damien Saenz, Corentin Vignaud

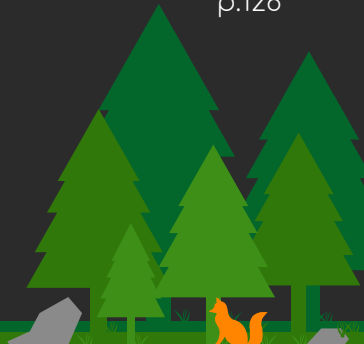
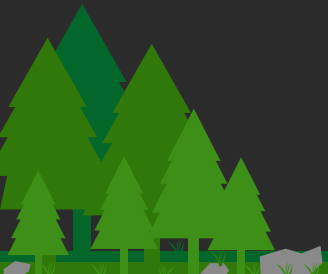




Sommaire

Sommaire

Game Overview	p.4	Phase de jour et de nuit	p.39	User Interface	p.80
Équipe	p.5	Attaque de nuit	p.40	Recherche en design	p.81
Fiche d'identité	p.6	Brouillard de guerre	p.42	Prototype	p.85
Références Game Design	p.7	Ingrédients supplémentaires	p.43	Changement du système du HUD	p.88
3C	p.8	Pilliers d'expérience	p.44	Explications générales	p.89
Caméra	p.9	Métaboucle de gameplay	p.45	Module jour / nuit	p.90
Character	p.10	Boucles de gameplay	p.47	Module d'énergie	p.91
Controller	p.11	Diagramme de Ventrice	p.50	Module mini-map	p.92
Game Experience	p.12	Modèle de Roger Caillois	p.51	Module sélection	p.93
Intentions générales	p.13	Heuristiques	p.52	Rendu final	p.101
Explication générales du gameplay	p.14	Objectifs	p.56	Direction sonore HUD	p.102
Nexus	p.15	Objectifs de niveau	p.57	Direction Artistique	p.103
Récolte d'énergie	p.16	Escorte de convoi	p.58	Intentions graphiques générales	p.106
Niveau du nexus	p.17	Survie	p.62	Intentions sonores générales	p.107
Création d'unités	p.18	Remplissage	p.63	Narration	p.109
Agents	p.19	Level Design	p.64	Direction graphique bâtiment	p.112
Unités alliés	p.23	Apprentissage	p.65	Direction sonore bâtiment	p.114
Direction graphique unités alliés	p.29	Niveau tutoriel	p.67	Annexe	p.115
Unités ennemis	p.30	Apprentissage de la difficulté	p.69	Timeline du projet	p.116
Direction graphique unités ennemis	p.32	RGD	p.70	Optimisation générale	p.119
Direction sonore unités	p.33	Challenges	p.73	Optimisation réunion	p.120
RGD	p.35	Biomes	p.75	Protocole de test	p.121
Rayon d'alimentation	p.36	Direction graphique environnement	p.76	Sound Design Document	p.123
Bâtiments désactivés	p.38	Direction sonore environnement	p.78	Intentions et recherches sonore	p.124
				Analyse sonore	p.126



REBORN

- Game Overview -



Équipe

Corentin Vignaud

Chef de projet - Game Designer -
Programmeur

Guillaume Charier

Game designer - Programmeur

Gonzalo Bourdieu

Game Designer - Level Designer

Lucas Huriaux

Game designer - Sound Designer -
Graphiste (3D)

Damien Saenz

Game Designer - Level Designer - UX/UI designer



Fiche d'identité

Pitch

Incarnez un ancien noyau d'énergie se réveillant après plusieurs siècles de sommeil. Seul survivant dans un monde en ruine vous devrez combattre les larsans, un peuple se nourrissant d'énergie vitale, pour réanimer votre civilisation éteinte. Créez des unités et affrontez les dangers qui vous attendent.

Genre

Reborn est un jeu en 3D avec une caméra vue de haut et des éléments de stratégie et de gestion.

Jeu de stratégie en temps réel PVE où le joueur doit gérer ses ressources, ses déplacements et ses unités.

KSP

- Approche nomade du RTS où le joueur déplace son bâtiment principal.
- Nexus qui alimente les unités et réveille les bâtiments inactifs.
- Explorer la carte pour récupérer de nouvelles unités et améliorations.
- Prise en main facile et complexification graduelle.

Cible

Public de joueurs intéressés par les jeux de stratégie avec des notions de gestion.

Support

Développé sur Unity 3D.
Disponible sur PC (Windows).

Nombre de joueurs

Un joueur.



Références Game Design

Jeu du genre RTS

Nous nous sommes principalement inspirés des mécaniques du genre qui sont communes à quasiment tous les RTS. De part la gestion de ressources, la création d'unités dans des bâtiments, la sélection et déplacement de celles-ci ainsi que la caméra et le HUD.



They Are Billions

Le point que nous trouvons intéressant dans ce jeu est le fait que ce soit un RTS PVE particulier. Le joueur ne se bat pas contre une IA identique à lui mais contre des ennemis répartis sur la carte et des vagues d'ennemis se dirigeant vers sa base à intervalles réguliers. Cela allège également la production car cela ne nécessite pas d'avoir une IA complexe.



Frostpunk

Le point intéressant dans ce jeu vient du réacteur. Celui-ci alimente les bâtiments alentour avec de la chaleur et consomme régulièrement de l'énergie. Nous avons décidé de reprendre ce système pour notre Nexus qui alimente les bâtiments et unités autour de lui.



REBORN

- 3C -



Camera

La caméra est une caméra de RTS classique vue de haut et légèrement tournée vers le bas. Elle est fixe et se déplace si le joueur bouge sa souris sur les bords de l'écran. Elle doit permettre au joueur de bien comprendre ce qui se passe sur son écran et sélectionner facilement ses agents. Son point de vue omniscient est nécessaire pour permettre la prise d'informations que nécessite le jeu. Elle ajoute également de la profondeur à l'angle de vue du jeu pour rendre le visuel moins plat et faire ressortir les modèles 3D.

Un HUD semi-permanent permet au joueur de rester informé sur ce qu'il se passe dans son environnement. La partie centrale du HUD apparaît uniquement lorsque le joueur sélectionne un objet interactif pour ne pas obstruer la vue en permanence. Celui-ci est également légèrement transparent pour que le joueur garde un visuel sur les zones occultées par celui-ci. Le joueur dispose également d'aide lorsqu'il passe sa souris sur différents éléments du HUD.



Character

Le joueur possède un point de vue omniscient sur la situation.

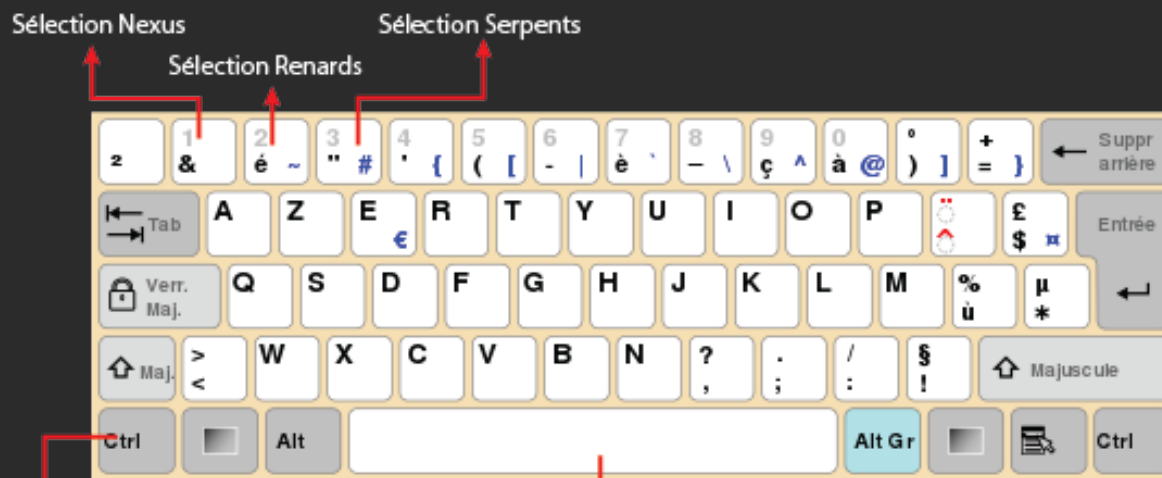
C'est par cet intermédiaire qu'il peut interagir avec son environnement. Le nexus représente sa base d'activité, nécessaire à son évolution dans le jeu. Son système d'énergie permet de recharger les unités autour de lui et d'activer les bâtiments à sa portée. Le nexus est également son point vital, la mort de celui-ci induisant la fin de la partie. Il doit donc être protégé par le joueur et pris en compte dans ses choix.

Ce nexus est plus lent que les unités et s'immobilise à intervalles réguliers en se faisant attaquer en fonction des phases de jeu.



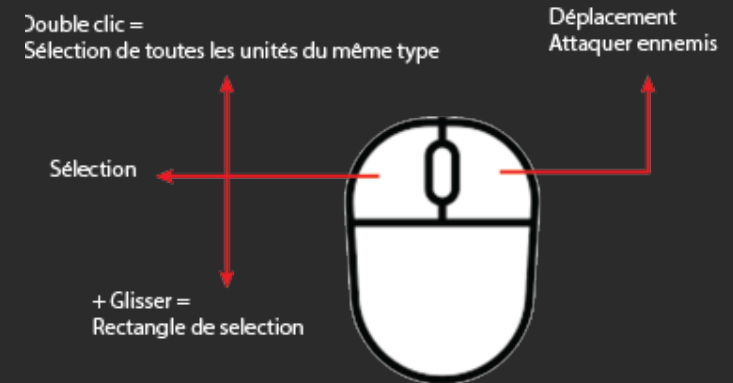
Controller

Notre jeu a été créé dans l'optique d'être jouable principalement à la souris. Les contrôles sont simples, le clic gauche servant à sélectionner ou cliquer sur des boutons du HUD et le clic droit servant à donner des instructions aux unités et placer des points. Un maintien du clic gauche fait apparaître un cadre de sélection plus pratique pour sélectionner des groupes d'unités. Des raccourcis clavier sont également présents pour faciliter la prise en main.



Recentrer la caméra sur le nexus

- + 1 = Ajouter/ supprimer Nexus du roster actuel
- + 2 = Ajouter/ supprimer Renards du roster actuel
- + 3 = Ajouter/ supprimer Serpents du roster actuel
- + Clic gauche = Ajouter/ supprimer sélection du roster actuel



REBORN

- Game Experience -



Intentions générales

L'intention derrière Reborn est de prendre une nouvelle approche du jeu de survie en le mêlant au genre du RTS. Ce genre présente habituellement un gameplay composé de bâtiments fixes et d'unités que le joueur utilise pour interagir avec son adversaire. Nous avons donc décidé de prendre à contrepied cette démarche et basé notre expérience sur un gameplay plus nomade. Pas de base-building ici et pas d'ouvriers, le joueur devra déplacer son bâtiment le plus important au cœur de l'action pour récolter des ressources et survivre au lieu d'envoyer des agents faire le travail à sa place.

Pour appuyer cette intention nous avons ajouté une mécanique d'alimentation. Le nexus recharge des bâtiments et les unités dans un rayon autour de lui. Ceux-ci se déchargent avec le temps lorsqu'ils sont hors de portée. Le nexus doit donc se rapprocher du champ de bataille pour les maintenir en vie, ajoutant plus de danger et de gestion pour le joueur.



Explication générale du gameplay

Dans notre jeu, le joueur contrôle son bâtiment principal, le nexus. Il peut le déplacer à l'aide de la souris. Ce nexus peut créer des unités contre un coût en énergie qui peuvent être déplacées sur la carte et combattre les ennemis répartis sur celle-ci. Cette énergie est la ressource principale du joueur et peut être acquise en se rapprochant des mines de ressources réparties sur la carte. Le nexus alimente les unités autour de lui et recharge leurs batteries qui se vident au cours du temps. Il consomme également de l'énergie dans le processus en fonction du nombre d'unités alimentés. Deux phases principales composent l'expérience de jeu. Une phase de jour où le joueur peut bouger librement et une phase de nuit. Cette phase de nuit a pour effet d'immobiliser le nexus qui se fera attaquer par des vagues d'ennemis. S'il parvient à défaire toutes ses vagues, la phase de nuit prend fin et le joueur peut de nouveau bouger.

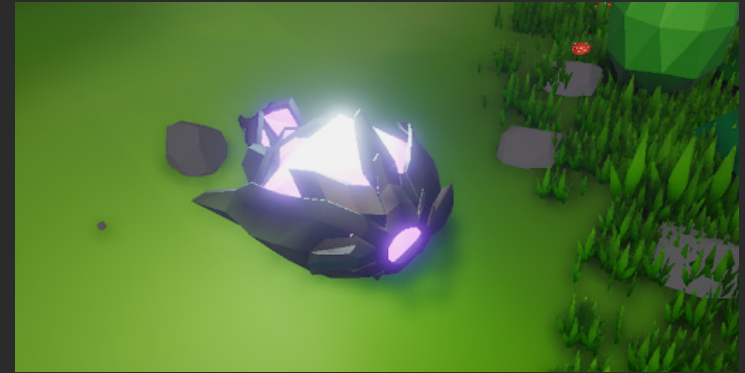
Exemple de gameplay : le joueur mine des ressources avec lesquelles il peut créer des unités. Une fois la mine vidée de ses ressources, le joueur part chercher un autre point de ressources en cherchant un chemin jusqu'à celui-ci en explorant les différentes possibilités avec ses unités et la caméra. En chemin, le joueur affronte des ennemis avec ses unités et sécurise un périmètre. Une fois la menace écartée, le joueur peut avancer jusqu'au point de ressource qu'il avait découvert.



Nexus

Le joueur incarne un nexus, seul survivant de son espèce après qu'une invasion de larsan ait détruit sa civilisation. Pour la réanimer, il devra réactiver la source d'énergie qui alimentait son peuple. Il aura pour ça quelques outils à sa disposition. Le nexus est la base mobile du joueur, c'est l'entité qui lui servira à créer ses unités et à les alimenter, mais il est fragile et le joueur perdra la partie si celui-ci est détruit.

Le nexus peut être sélectionné avec le clic gauche de la souris, et un ordre de déplacement peut lui être donné avec le clic droit.



Récolte d'énergie

Le nexus a besoin d'énergie pour survivre. Il pourra en récolter grâce aux mines de ressources réparties sur la carte. Cette récolte est automatique et nécessite uniquement que le nexus soit proche d'une mine. La valeur d'énergie actuelle est indiquée sur le HUD.



Les mines de ressources sont des regroupements de cristaux sortant du sol imprégnés d'énergie dont la civilisation de nexus se servait pour s'alimenter. Ces mines possèdent un stock d'énergie fini mais celui-ci se rechargera au fil du temps si la mine n'est pas récoltée pendant un certain temps. L'énergie restante dans une mine peut être évaluée grâce à une barre de ressources au-dessus de la mine et un changement de la couleur du cristal.

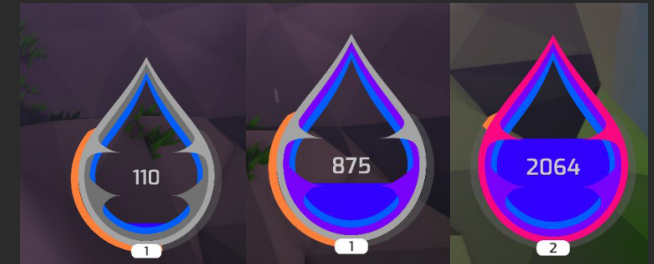


Niveau du nexus

Le nexus peut accumuler autant d'énergie qu'il le souhaite et ne possède pas de réserve maximale. Cependant s'il dépasse une certaine valeur d'énergie fixe, il montera de niveau et se verra octroyer des avantages et des inconvénients.

Sa vitesse, son rayon d'action et son champ de vision se verront augmentés mais il consommera passivement plus de ressources s'il passe au niveau supérieur.

Ce changement de niveau n'est cependant pas permanent, et le niveau du nexus pourra baisser si le joueur passe en-dessous de la valeur d'énergie requise.



L'apparence et la couleur du nexus changent en fonction de son niveau. Également, le niveau actuel du nexus et la valeur requise pour passer au niveau supérieure est indiquée sur le HUD grâce à des graduations sur la réserve d'énergie.

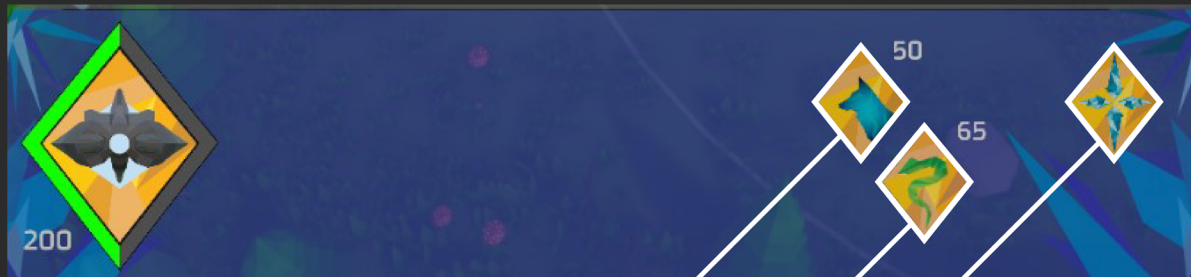


Création d'unités

Le nexus peut créer des entités d'énergie. Étant garant de l'écosystème, il s'en sert pour créer des représentation éthérés des animaux qui depuis, ont été annihilés par les larsan. Ces représentations gardent une partie de leurs comportements d'origine.

Si le nexus est sélectionné et possède assez d'énergie, il peut créer des unités en utilisant les icônes du HUD. Ces unités sont de différents types et peuvent être contrôlées par le joueur pour explorer le terrain ou attaquer et se défendre contre les larsan.

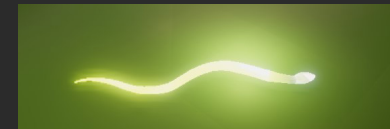
Le point d'arrivée des unités à leur création peut être modifié en cliquant sur l'icône de point de ralliement.



Icône de création du renard

Icône de création du serpent

Icône du point de ralliement



Agents

Les agents de notre jouet sont les unités alliées et ennemis. Elles répondent aux mêmes comportements à quelques différences près. Le joueur peut contrôler les unités alliées mais pas les unités ennemies, qui attaqueront à vue et devront être affrontés à l'aide d'unités alliées.

Trois états utilisés pour nos agents : Idle, Follow, Aggressive,

Idle State :

- Etat de base d'un agent
- Recherche une cible à attaquer
- En attente d'une nouvelle destination ordonnée par le joueur ou le système

Attack State :

- L'agent poursuit sa cible jusqu'à être à portée d'attaque
- Inflige des dégâts à sa cible quand il est à portée
- Si la cible se déplace après être à portée, l'agent le poursuit de nouveau
- Les ordres du joueur passent en priorité (exemple : changement de cible, ordonner de se déplacer à un endroit)
- Si la cible est détruite, l'agent en recherche immédiatement une autre à portée

Sous état possible : Super agressif

- Uniquement pour les agents ennemis
- Se dirige vers le Nexus
- Si une autre cible est à portée durant le trajet, attaque la cible
- Si la cible est détruite, se redirige vers le Nexus
- Si le Nexus est à portée, l'agent l'attaque jusqu'à destruction de celui-ci

Follow state :

- L'agent doit se déplacer à un endroit précis
- Arrivé à destination, l'agent recherche une cible à attaquer à sa portée

Idle State :

État de base d'un agent. Aucune action directe de l'agent, position immobile. Recherche une cible à attaquer ou un ordre de déplacement.

Priorité des actions :

- 1) Si agent allié, ordre de déplacement ou d'attaque du joueur
- 2) Réponds à une attaque
- 3) Fonction de recherche d'une cible à attaquer

Follow State :

Etat de déplacement d'un agent vers une destination. Utilisé pour les agents alliés et ordre de déplacement du joueur. L'agent se déplace jusqu'au point d'arrivée sans interruption (ignore attaque ennemi par exemple).

Priorité des actions :

- 1) Si agent allié, ordre d'attaque ou de déplacement
- 2) Arrivée à destination
- 3) Réponds à une attaque

Aggressive State :

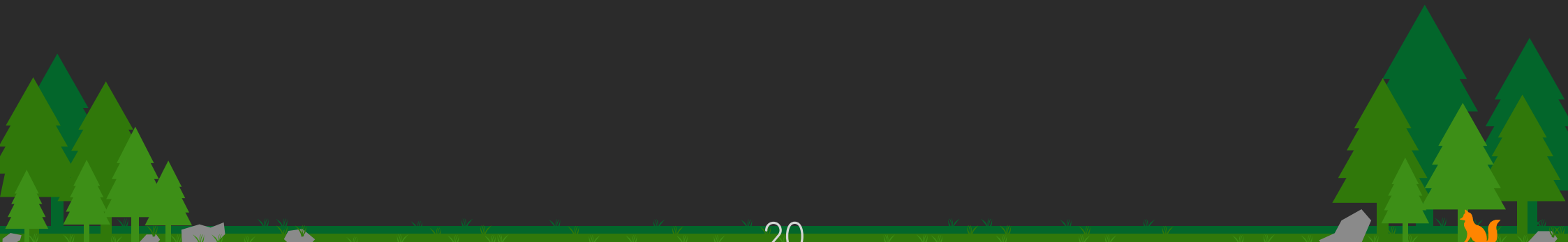
État d'attaque d'un agent. L'agent se déplace jusqu'à être à portée d'attaque de sa cible. Il la poursuit jusqu'à destruction de celle-ci. Si l'agent n'a plus de cible, il en recherche une autre à portée puis passe en idle state si rien n'est trouvé.

Sous état : Super aggressive

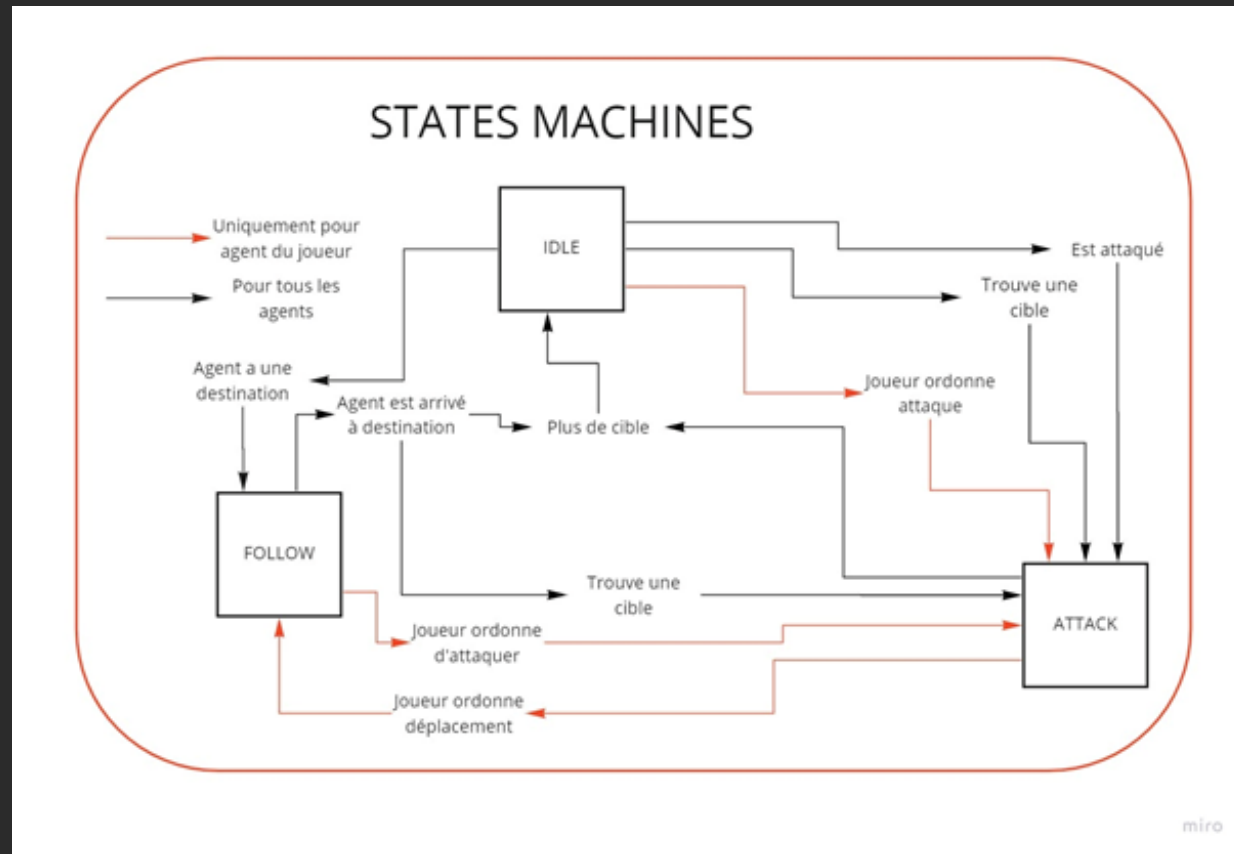
Uniquement pour agent ennemi lors d'une attaque de nuit. Les agents se dirigent toujours sur le Nexus. Si une cible (agent du joueur) est repérée, l'agent ennemi se dirige vers celle-ci. A la mort de la cible, l'agent se redirige vers le Nexus.

Priorité des actions :

Si agent allié, ordre de déplacement ou d'attaque sur une autre cible
Si agent ennemi Super aggressive, recherche d'une cible sur son chemin
Poursuite et attaque de la cible
Si agent ennemi Super aggressive, se dirige vers le Nexus



Comportement général :

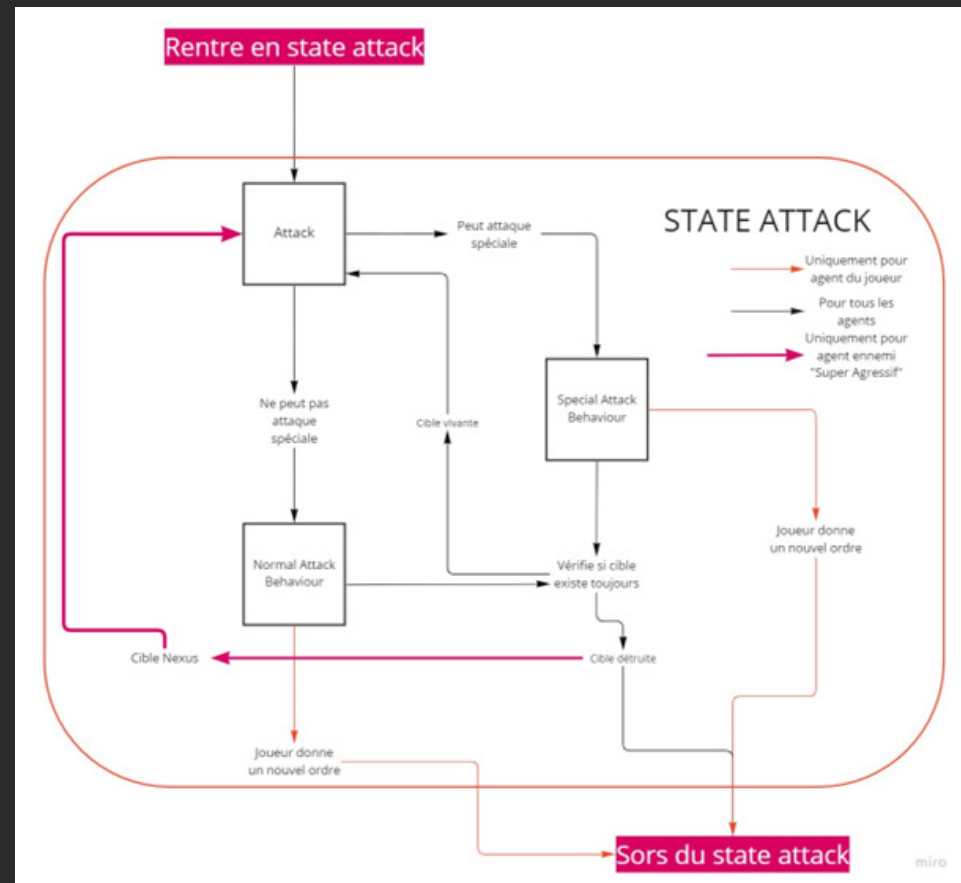


Agents

Il y a deux types d'agents dans le jeu : les unités alliées, créées par le nexus et contrôlées par le joueur, et les unités ennemis réparties sur la carte et attaquant le joueur la nuit.

Les agents répondent tous à la même state-machine qui détermine leur comportement. Il existe des variations entre les agents ennemis et alliés dues aux différences de ciblage pour les unités ennemies et la possibilité de donner des ordres pour les unités alliées.

Les unités communiquent également entre elles et enverront un signal de détresse si elles se font attaquer, appelant automatiquement les autres agents alliés ou ennemis à attaquer la cible.



Description State Attack

Unités alliés

Les unités alliées sont les agents que peut contrôler le joueur pour interagir avec le jeu. Elles sont au nombre de 4 et possèdent chacune des statistiques différentes et une compétence unique qui leur permettent d'être spécialisés en fonction des situations.

Les unités partagent 9 statistiques que l'on peut influencer, pour leur donner des avantages / inconvénients en fonction des situations, la barre tirant du rouge vers le vert donne une idée générale de l'équilibrage de l'unité en question.



Unités alliés

- Portée : Détermine la portée d'une unité, une petite portée détermine un corps à corps et une portée élevée définit un attaquant à distance.
- Coût : Détermine le coût en ressources d'une unité, un coût plus élevé induit un investissement en ressources plus sévère et une unité plus compliquée à produire en quantité.
- Entretien : Coût de l'alimentation de l'unité par le nexus. Plus cette valeur est élevée et plus l'unité demandera de l'énergie pour être alimentée à chaque tick, et induira une perte d'énergie plus importante sur le long terme. Cela donne un choix au joueur entre sécurité et pertes de ressources ou économie d'énergie et potentiel danger.
- Vision : Sert à déterminer le rayon du cercle de brouillard que l'unité révèle autour d'elle. On peut jouer sur cette valeur pour créer des unités à même de faire du repérage ou des unités en nécessitant d'autres pour naviguer sur la carte.



Renard

Le renard est l'unité de base du joueur. Ces statistiques sont moyennes et il ne possède pas de pouvoir. Son coût de création et d'entretien sont faibles.

L'intention derrière le renard est de créer une unité qui sert de valeur étalon pour le jeu et de première unité pour le joueur. Elle a donc été conçue comme une unité polyvalente pouvant être produite en quantité et pouvant être utilisées dans toutes les situations.

Cette unité ne possède pas de pouvoir.

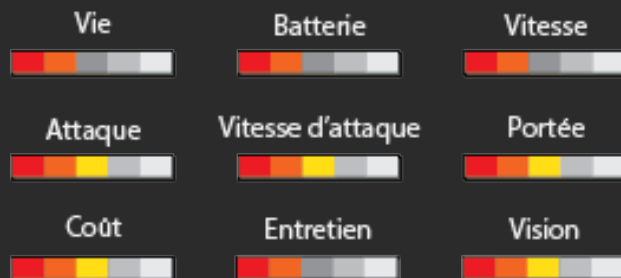


Serpent

Le serpent est une unité à distance pouvant se débarrasser facilement de groupes d'ennemis grâce à sa compétence. Elle possède une portée, vitesse d'attaque moyenne combinée avec de faibles points de vie et une vitesse légèrement inférieure à la moyenne.

L'intention derrière le serpent est de créer une unité efficace contre les groupes d'ennemis grâce à son pouvoir. Elle doit cependant se rapprocher dangereusement du combat pour attaquer et ses statistiques plus faibles que la moyenne lui donnent un désavantage au corps à corps.

Le serpent peut toutes les 15 secondes lancer une zone de poison au sol restant plusieurs secondes et infligeant des dégâts périodiques à tous les ennemis se trouvant dans celle-ci.



Ours

L'ours est unité défensive disposant tout de même d'un fort potentiel de dégâts. Il dispose d'un large nombre de points de vie et une attaque importante, compensés par une faible vitesse, portée et vitesse d'attaque. Ses coûts d'entretien et de création sont élevés.

L'intention derrière l'ours est de créer une unité pouvant être facilement envoyée en première ligne par le joueur pour concentrer et endurer les attaques ennemis, tout en possédant assez de dégâts et un pouvoir lui permettant d'être destructeur pendant les assauts.

Son pouvoir est un coup de griffe qu'il peut utiliser toutes les 20 secondes qui inflige de lourd dégâts dans un cône devant lui.



Référence de conception pour l'unité ours dans un genre low-poly

Panthère

La panthère est une unité servant au joueur à explorer son environnement. Ses capacités en combat sont très limitées mais elle possède une bonne vitesse et champ de vision. L'intention derrière cette unité est de donner un outil au joueur pour se repérer facilement sur la carte et l'explorer. Sa vitesse de déplacement et son pouvoir en font une unité qui peut leurrer facilement les ennemis. Après avoir été touchée par un ennemi, l'unité devient intouchable et perd l'agressivité des ennemis pendant quelques secondes.



Référence de conception
pour l'unité panthère
dans un genre low-poly

Direction graphique unités alliés

Intention graphique des unités alliés :

Le design des unités que le joueur peut contrôler est basé sur des références d'animaux réels que l'on peut trouver dans les différents biomes présents dans le jeu. Ce choix a été fait pour renforcer le côté de la narration où le joueur ravive un monde éteint. Dans ce but, le joueur crée des unités animales faites d'énergie grâce à des totems qui lui permettent de récupérer les données de l'unité correspondante. Le design graphique des unités a donc été fait pour coller à la narration. De ce fait, les unités alliés possèdent un aspect spectral formé d'énergie pour garder un lien avec l'énergie qui fait vivre le nexus. De plus, le design des unités a suivi le même processus d'affordance pour que le joueur puisse reconnaître facilement les unités.

Références graphiques unités alliés :

Nos références graphiques pour les unités sont principalement composées de Spirit of The North qui illustre clairement le côté mystique que nous avons donné à nos unités.

Une autre de nos références sont les patronus que l'on trouve dans la saga Harry Potter.



Unités ennemies

Les unités ennemies sont au nombres de 5 :

Balyjos

C'est l'unité la plus commune qu'affrontera le joueur sur la carte. Elle ressemble en tout point au renard du joueur mais avec des statistiques légèrement plus faibles pour lui donner un avantage lors de ces premiers combats.



Sactys

C'est une petite unité rapide, avec un fort potentiel offensif due à une grande vitesse, attaque et vitesse d'attaque, mais qui compense ses avantages avec une faible vitalité. Cette unité est très forte en nombres et un groupe de celles-ci parviendra facilement à tuer un groupe d'unités si elles possèdent un avantage numérique. Les dégâts de zone sont forts contre cette unité.



Unités ennemies

Skalam

C'est une unité possédant une grande portée et de forts dégâts d'attaque mais une mobilité et vitalité réduite. Cette unité est forte contre les unités uniques et peut décimer les rangs alliés si elle est laissée sans surveillance. Sa faible mobilité est son défaut et un groupe d'unités arrivant à percer les lignes adverses en viendra facilement à bout.



Barsak

C'est l'unité la plus robuste des ennemis. Elle possède une bonne robustesse et des dégâts élevés mais une vitesse d'attaque et mobilité réduite. Elle possède également une attaque de zone qui inflige de lourds dégâts en cercles autour d'elle. Un bon groupe d'unités et du micromanagement seront nécessaires pour en venir à bout.



Direction graphique unités ennemis

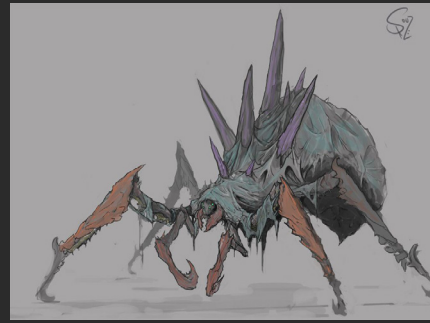
Intention graphique des unités ennemis :

Concernant les unités ennemies, notre principale objectif était que le joueur puisse comprendre facilement qu'il s'agissait de cibles hostiles à sa survie. Dans ce but, les unités ennemies ont un design qui se lie à la corruption qu'ils apportent avec eux. De plus, pour dissocier ces deux types d'unités, leur design se rapproche majoritairement d'insectes avec une tendance arachnide pour appuyer le côté répulsif et dangereux.

Les couleurs de nos unités sont donc en lien avec la corruption et le fait que ce peuple se nourrit d'énergie cristalline qui a peu à peu changé leurs constitutions.

Références graphiques unités ennemis :

Les références de nos unités ennemies se base principalement sur des concept art proche de la fantaisie et de la science fiction pour leurs aspects mélangeant à la fois insecte et alien.



Direction sonore unités

Intentions sonores des unités :

Dans le même objectif que nos intentions générales, le design sonore de nos unités a longtemps été réfléchi pour avoir une maximum d'affordance en fonction de leurs actions. Il fallait que le joueur puisse comprendre ce qu'il se passe à tout moment lors des combats, de manière à ajuster sa stratégie en conséquence. Pour que cela se passe correctement, nous avons établi des intentions claires quant à ce que devait faire une unité et ce qui devait en ressortir aussi bien au niveau du son que de la partie graphique. Un travail a alors été fait pour synchroniser au mieux les feedbacks sonores aux attaques, compétences, déplacements ou morts des unités. Cela a été rendu possible directement grâce aux animations auxquelles nous pouvions coupler le lancement des sons voulus pour coller au mieux entre ce qui se passait visuellement et au niveau sonore.



Références sonores unités alliés :

Malgré le fait que les unités alliées soient inspirées d'animaux réels, leur design sonore est inspiré tout comme le côté mystique de nos bâtiments de Spirit of the north. Créé par de l'énergie, il fallait que cela puisse se ressentir lorsque le joueur les manipulait. Dans ce but, leur design sonore est composé de sons énergétiques qui reprennent des caractéristiques propres aux animaux dont ils sont inspirés.



Références sonores unités ennemis :

Le design sonore des unités ennemies devait avoir un design majoritairement différent des unités alliées pour ne permettre aucune confusion lors des combats. Le design graphique des unités ennemies étant inspiré de créatures arachnides, nos références se sont majoritairement tournées vers des sons d'insectes et d'aliens.



RGD des unités

Voici un tableau récapitulatif de nos valeurs d'équilibrage des différents ingrédients et unités dans le but de modeler avec précision les challenges donnés au joueur.

Unit Name	Ressource	Life	Base speed	Creation time (secs)	Damage	Rate of Fire	Sight	Range	Projective Speed	Reload value	Resource Pickup speed	Total resources (Before LVL up)	Abilities
Renard	50	105	10	3	30	2.7	5	2	0	0	0	0	
Jersen	N/A	110	8	N/A	34	3	10	3	0	0	0	0	
Turret	N/A	N/A	N/A	2	40	25	N/A	10	25	1	0	0	Shoot Shoot
Nexus	N/A	150	8	N/A	0	0	12	0	0	0	10	800	
Zerg	N/A	40	10	N/A	14	2.5	10	3	0	0	0	0	
Serpant	65	80	10	3	25	2.7	5	4	0	10 (Before new zone)	0	0	8 Constant poison damage
Bear	75	300	6	3	55	3	5	3	0	3	0	0	65 attack zone
Panther	45	105	12	3	20	2.2	12	2	0	0	0	0	Invisible after getting first hit

Rayon d'alimentation

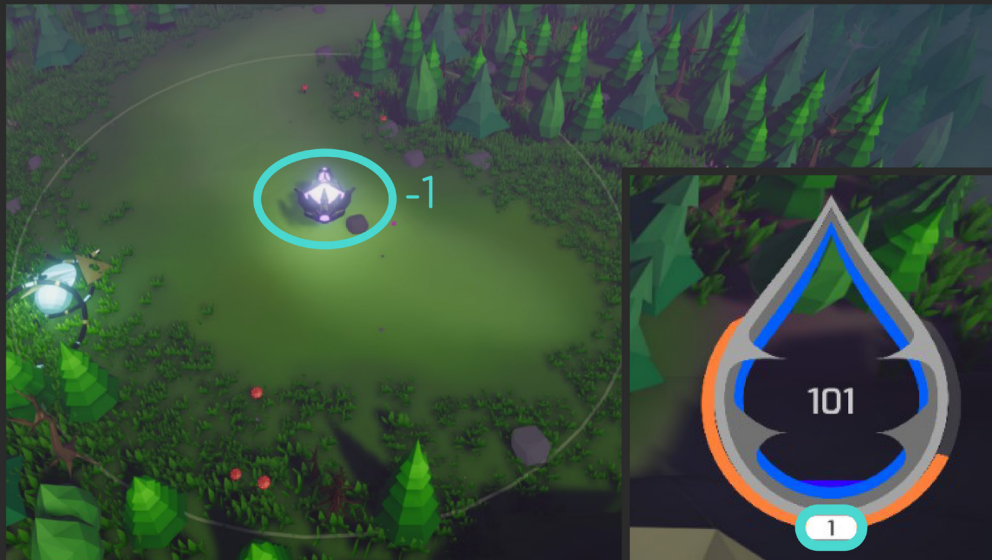
Le nexus alimente passivement les unités et structures autour de lui en utilisant son énergie. Ses alimentations sont ponctuelles et se produiront toutes les 5 secondes.

Les unités alliées et bâtiments possèdent tous deux des barres d'énergies. Quand la barre d'énergie des bâtiments est vide, ils se désactivent, quand celle des unités est vide, ils disparaissent. Si ces entités se trouvent dans le rayon d'action du nexus lors d'un tick d'alimentation, leur batterie sera rechargée d'un montant fixe.

Le nexus consomme de l'énergie lors de chacune de ses alimentations. Chaque unité et structure alimentées ajouteront un coût fixe qui sera déduit à l'énergie totale du joueur.

Une barre de chargement permet de savoir quand le prochain tick d'alimentation se produira et une valeur numérique affiche la quantité d'énergie qui sera consommée pendant celle-ci.

Ce système a pour but de forcer le joueur à bouger pour accumuler de l'énergie et encourager l'action autour du nexus et le rendre plus vulnérable, le joueur devant ramener ses unités périodiquement près de celui-ci même pendant les combats.



Exemple In-game d'alimentation du nexus avec un coût de 1



Exemple In-game d'alimentation du nexus avec un coût de 5



Exemple In-game d'alimentation du nexus avec un coût de 8

Bâtiments désactivés

Des bâtiments désactivés, vestiges de la civilisation disparue sont dispersés sur la carte. Le nexus peut les alimenter pour bénéficier de leur effet, mais ils redeviendront des ruines quand leur barre d'énergie s'épuisera.

La tour de défense est une structure pouvant attaquer les ennemis à portée. Elle se différencie en deux catégories :

- Une tour attaquant un seul ennemi à la fois avec une vitesse d'attaque moyenne.
- Une tour attaquant en zone autour d'elle avec une vitesse d'attaque faible.

La tour de vision est une structure de repérage dispersant le brouillard de guerre dans une large zone autour d'elle quand alimentée.



Tour de défense désactivée



Tour de défense activée



Tour de vision désactivée



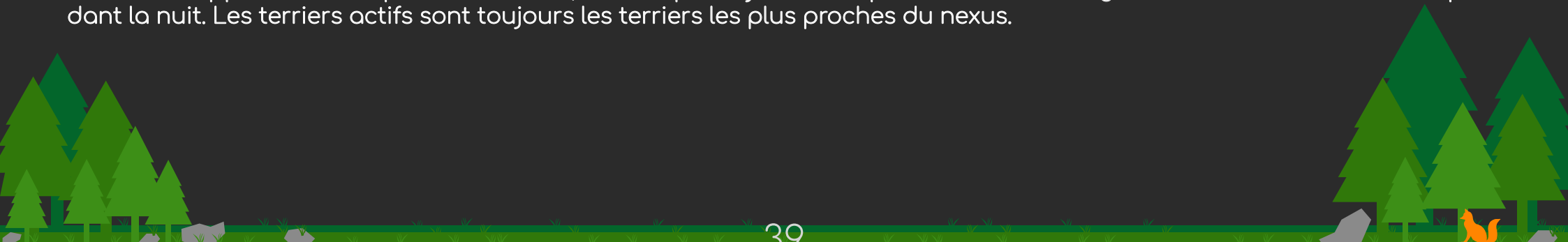
Tour de vision activée

Phase de jour et de nuit

La carte est soumise à un rythme jour/nuit qui modifie le comportement du nexus selon la période de la journée. La journée, le nexus est libre dans ces actions, mais la nuit, le nexus est immobilisé et ne peut plus se déplacer tout en préservant ses anciens comportements. Les larsan lancent également une offensive contre le nexus pendant la nuit. Cette offensive est générée procéduralement en fonction de la nuit en cours mais peut également être scriptée. La nuit arrive après 3 minutes de jour, affichées par une barre de progression en haut de l'écran. La nuit s'arrête après que la vague d'attaque soit battue et le nexus pourra de nouveau se déplacer.



Les larsan apparaissent depuis des terriers, visibles par le joueur et qui s'éclairent en rouge si des unités vont en sortir pendant la nuit. Les terriers actifs sont toujours les terriers les plus proches du nexus.



Attaque de nuit

La nuit, les larsan attaquent le nexus, le sachant vulnérable, celui-ci passant dans un mode de repos pour économiser son énergie.

Un coût déterminé est attribué à chaque nuit ainsi qu'un nombre de portails actifs. Chaque portail actif va créer aléatoirement des unités larsan, possédant chacune un coût en fonction de leur difficulté, jusqu'à égaler ou excéder ce coût. On peut également empêcher certaines d'unités d'apparaître avant une certaine nuit pour offrir une courbe de difficulté au joueur. Certaines nuits peuvent également être scriptées en termes des unités apparaissant dans celles-ci.

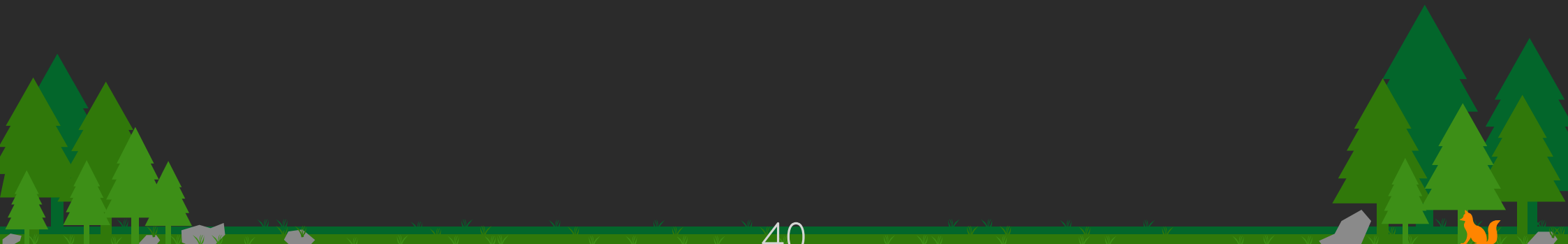
Ex : 3ème nuit, coût de 7, 2 portails actifs

Unités : Renard -> 2 de coût, Ours -> 5 de coût

Résultat : Portail 1 -> 4 renard = 8 de coût

Portail 2 -> 2 renard + 1 ours = 9 de coût

Ce système permet de s'assurer d'avoir un contrôle sur les attaques de nuit tout en laissant une part d'incertitude dans leur composition.



Nombres de portails actifs par nuit

Night0	Night1	Night2	Night3	Night4	Night5	Night6	Night7	Night8	Night9
1	2	2	3	4	4	4	4	4	4

Coût par portail par nuit

Night0	Night1	Night2	Night3	Night4	Night5	Night6	Night7	Night8	Night9
4	4	6	6	8	8	8	10	10	10

Blocage d'unité avant une certaine nuit

Unit0	Unit1	Unit2	Unit3
0	2	6	6

Custom Waves

	IsScripted	Unit0	Unit1	Unit2	Unit3
Night0	0	4	0	0	0
Night1	0	2	2	0	0
Night2	0	3	3	0	0
Night3	0	1	5	0	0
Night4	0	4	4	0	0
Night5	0	2	6	0	0
Night6	0	0	8	0	0
Night7	0	5	5	0	0
Night8	0	2	8	0	0
Night9	0	0	10	0	1

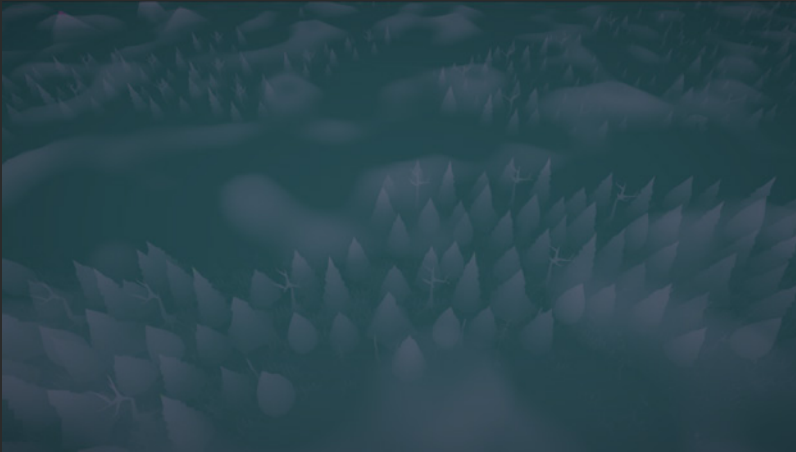
Cette phase de nuit est essentielle pour pousser le joueur à explorer, accéder à de nouvelles unités, accumuler de l'énergie et agrandir son armée. Elle le pousse à aller de l'avant en utilisant un challenge périodique d'une intensité grandissante. La composition de ses attaques est générée par un aléatoire restreint pour garder un équilibre mais privilégier la rejouabilité.

Brouillard de guerre

Le brouillard est un ingrédient créé pour ajouter un sentiment d'exploration du jeu en poussant le joueur à découvrir la carte au fur et à mesure, faire des choix de direction et perturber sa stratégie.

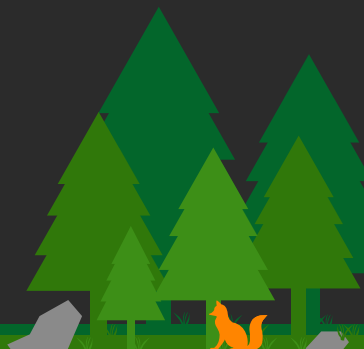
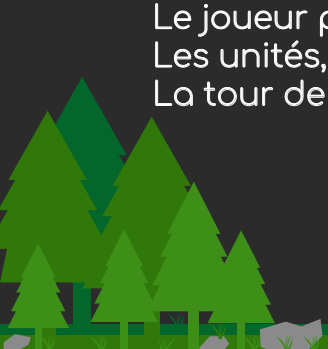
Le brouillard peut être séparé en trois catégories :

- Brouillard noir : zone non explorée par le joueur et recouverte par un brouillard opaque qui empêche de voir à travers.
- Brouillard gris : zone précédemment explorée par le joueur mais hors d'un champ de vision direct. Il laisse apparaître les mines, bâtiments et décors mais pas les larsan.
- Pas de brouillard : zone actuellement dans un champ de vision, tout est visible.



Le joueur peut dissiper le brouillard de plusieurs manières :

Les unités, nexus et bâtiments (si activés) possèdent un champ de vision passif.
La tour de vision permet de révéler une grande zone autour d'elle si alimentée.



Ingrédients supplémentaires

Totem :

Le totem sert à récupérer des unités. Si le nexus est proche de celui-ci, il va rapidement aspirer les données qui y sont contenues et qui lui donneront accès à un nouvel animal à créer. L'animal sur le totem sera celui récupéré par le joueur, et la découverte d'un totem d'une unité déjà possédée octroie un bonus d'énergie à la place.



Cristal Rouge :

Le cristal rouge est une mine de ressources corrompue par les larsen et qui, au lieu de produire de l'énergie, l'aspire. Si le nexus s'en approche, il se verra lentement drainé de celle-ci.



Piliers d'expérience

Exploration encouragée :

Le joueur est libre de prendre le chemin qu'il souhaite et l'exploration des recoins de la carte est récompensée par le gameplay et de manière visuelle.

Approches et stratégies variées :

Le joueur a du choix dans la manière dont il doit atteindre ses objectifs, que ce soit dans sa manière de résoudre les combats, les chemins qu'il prend ou sa sélection d'unités.

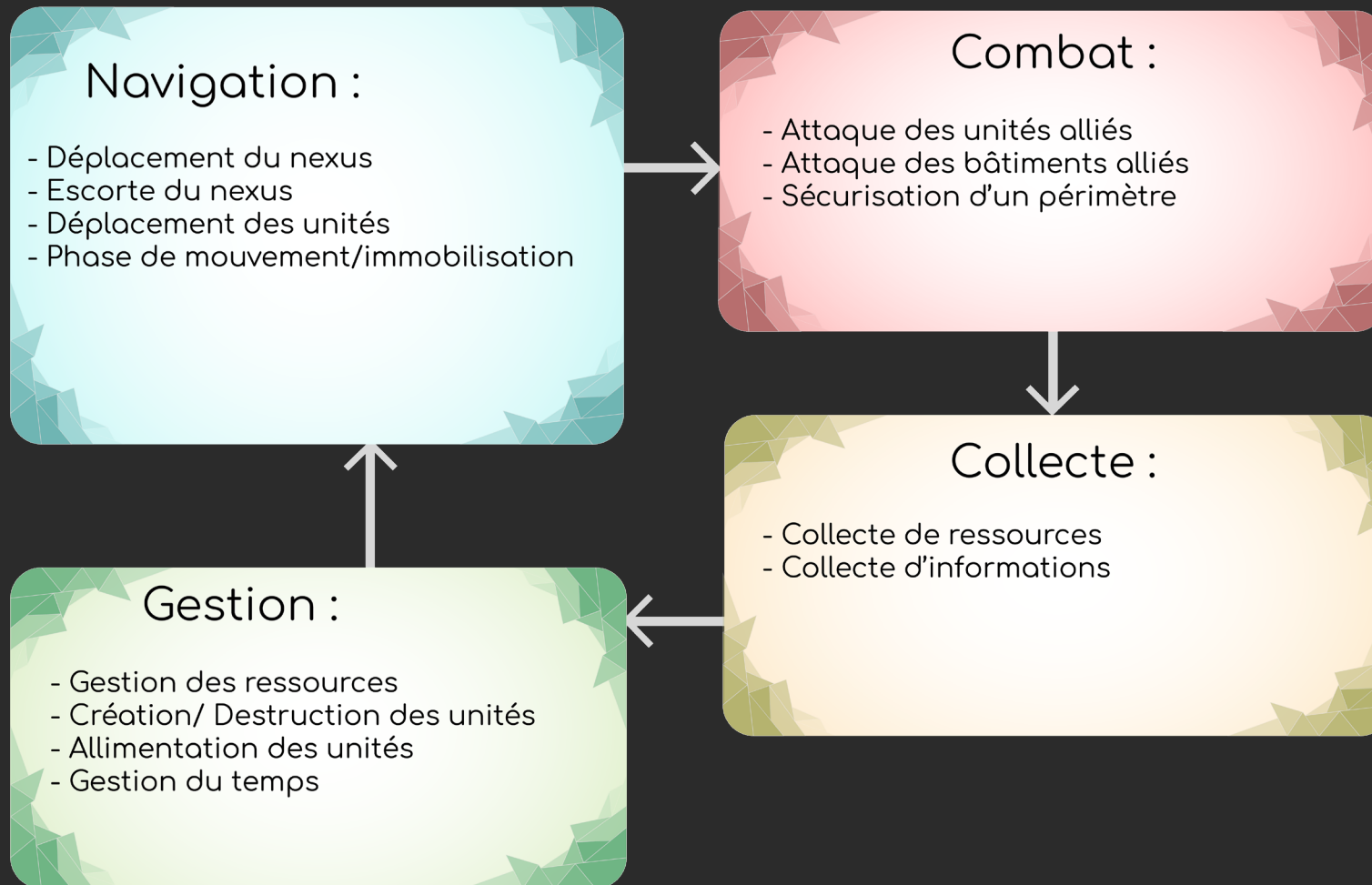
Gestion de l'énergie

Le joueur devra maintenir ses ressources qui se vident avec le temps et lui sont nécessaires pour survivre. Le joueur doit évaluer des choix et bien évaluer la situation pour ne pas gaspiller.



Métaboucle de gameplay

Méta boucle



Métaboucle de gameplay

Navigation :

- Déplacement du nexus : Sélection et déplacement du nexus dans l'espace.
- Escorte du nexus : Protéger le nexus pendant son déplacement.
- Déplacement des unités : Sélectionner et déplacer des unités ou groupe d'unités.

Combat :

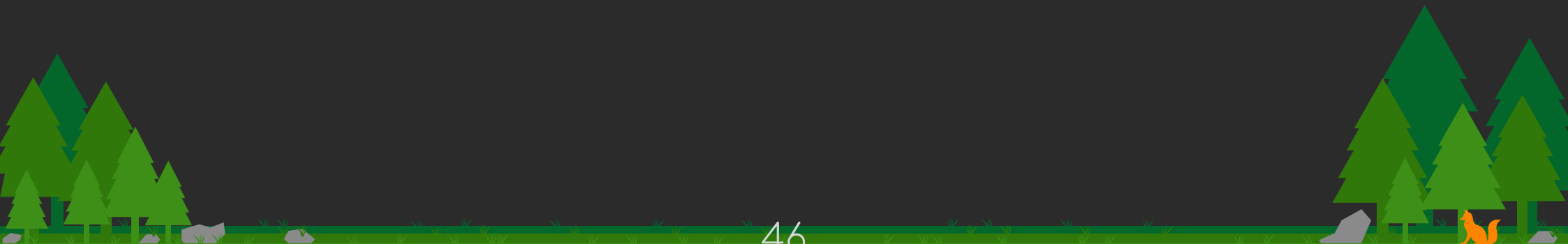
- Phase d'immobilisation : Défense du nexus pendant les attaques nocturnes.
- Attaque des unités alliées : Attaque avec les unités alliées.
- Attaque des bâtiments alliés : Alimentation et utilisation des bâtiments alliés offensifs.
- Sécurisation d'un périmètre : Défense d'un périmètre autour du nexus.

Collecte :

- Collecte de ressources : Aspirer de l'énergie des mines.
- Collecte d'informations : Explorer la carte et repérer les tracés et positions d'ennemis et bâtiments.

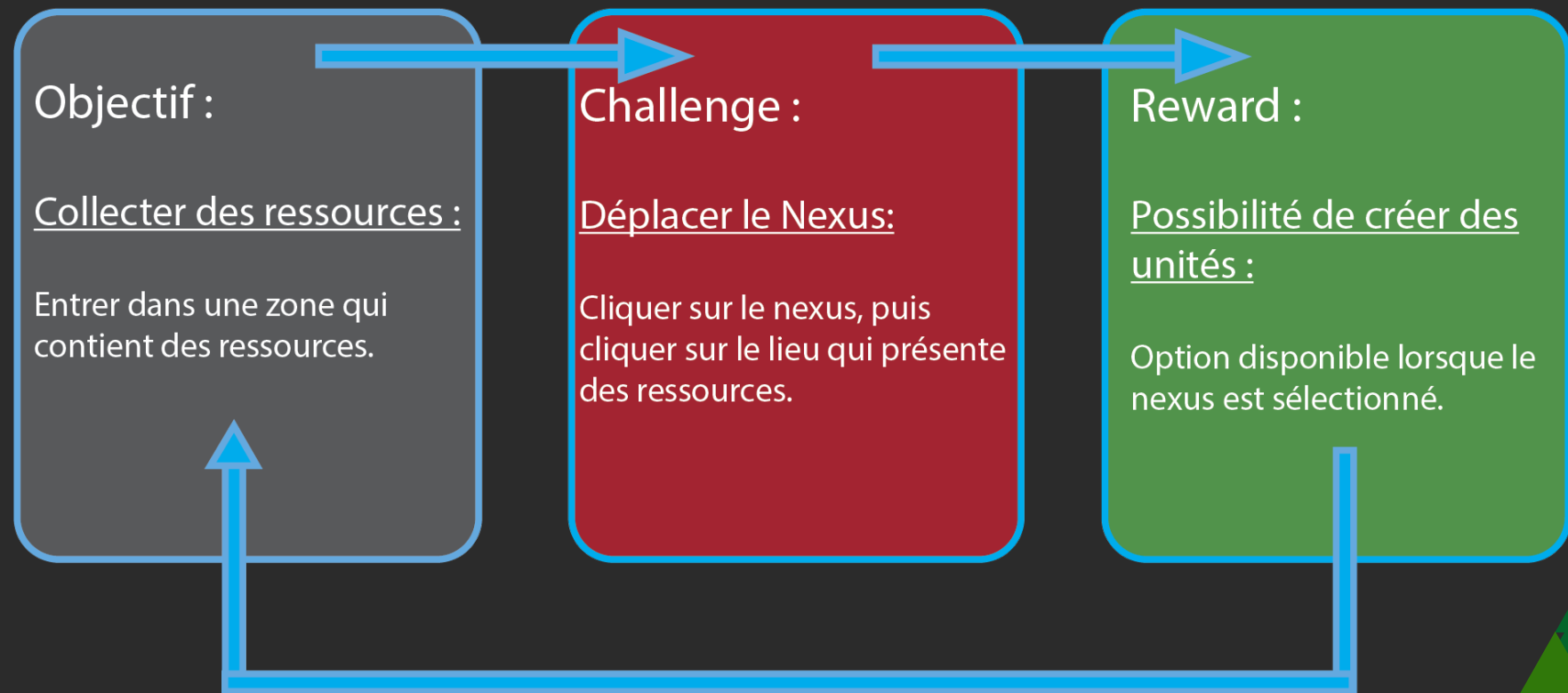
Gestion :

- Gestion de ressources : Gérer son énergie (création d'unités, entretien).
- Gestion des unités : Créer et gérer son armée.
- Gestion de l'alimentation des unités : Gérer quelles unités sont dans la rayon d'alimentation.
- Gestion du temps : Gérer le temps d'une journée pour ne pas être surpris pendant la nuit.

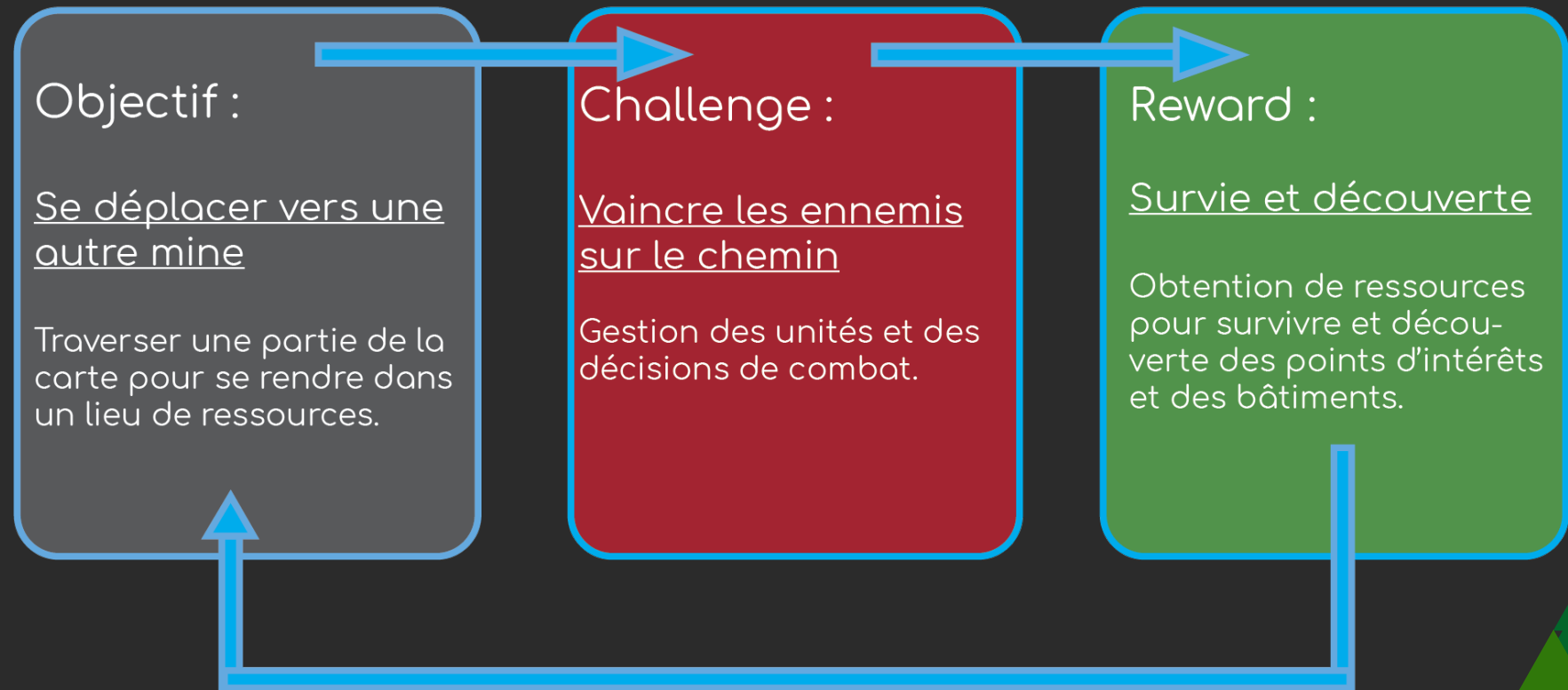


Boucles de gameplay

Micro



Méso



Macro

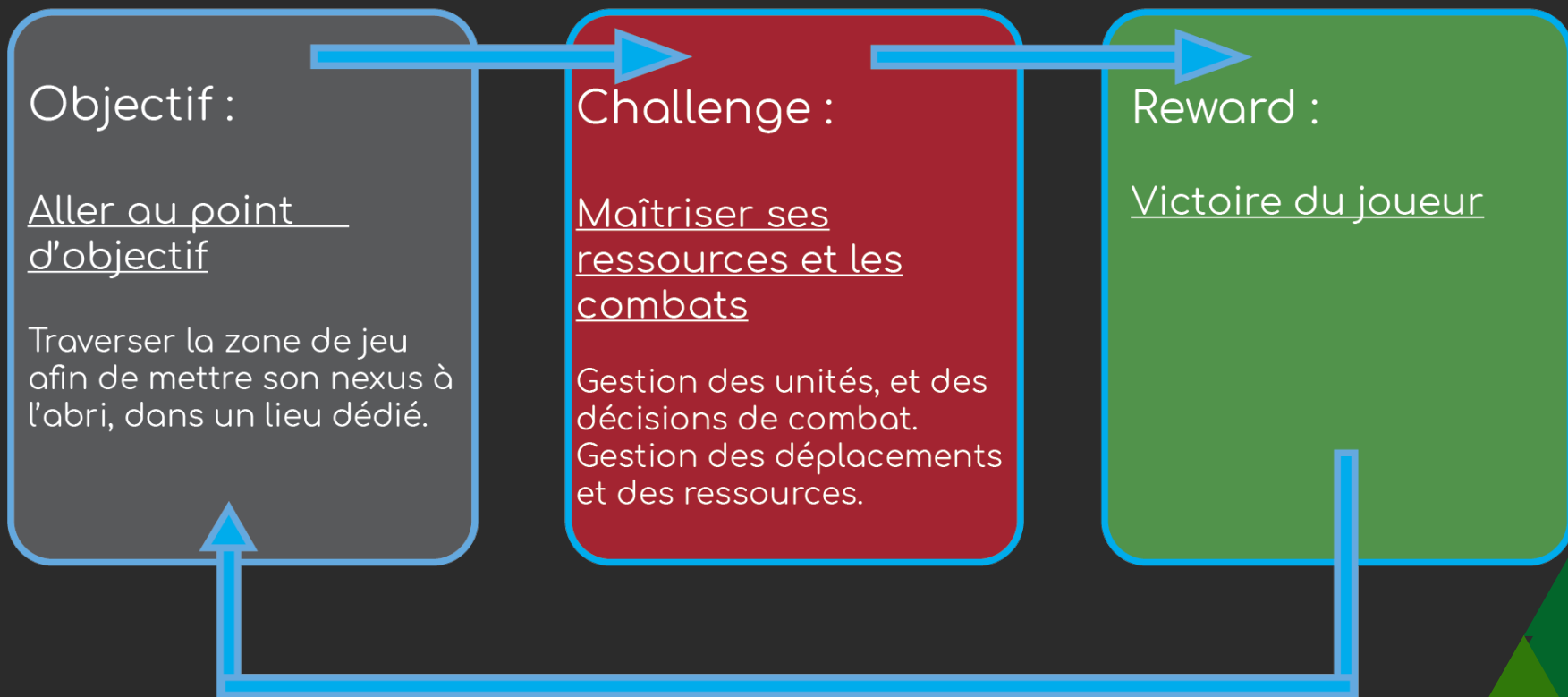


Diagramme de Ventrice

Concept

Déplacer votre Nexus pour trouver des ressources et développer votre armée

Skills

Gestion

Stratégie

Logique

Features

Gérer des ressources

Créer des unités

Déplacer ses armées

Mechanics

Collecte d'énergie

Production d'unité

Sélectionner des unités

Attaques des unités

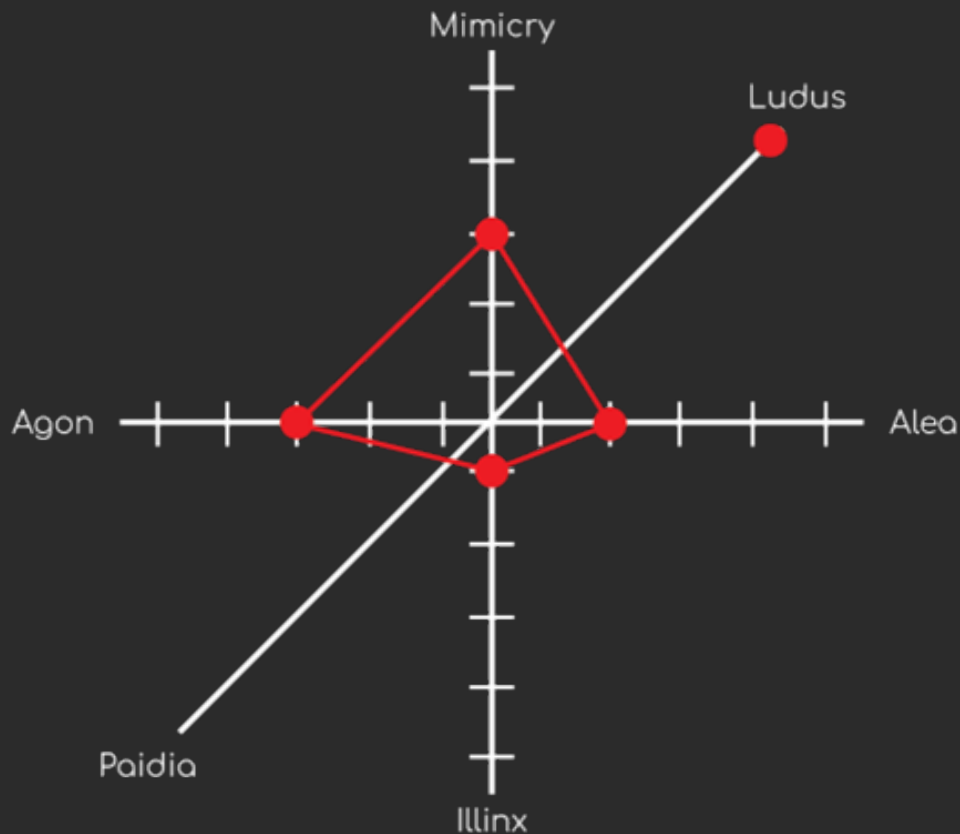
Déplacement de la caméra

Interface

Souris

Clavier

Modèle de Roger Caillois



Alea & Agon : Reborn est un jeu de stratégie en temps réel qui repose sur des règles précises. Il y a cependant de l'aléatoire et de marges de manœuvres dans les actions du joueur.

Mimicry : Le joueur incarne un nexus dans un monde dévasté et corrompu ou il ne reste que quelques traces de son ancienne civilisation. Le joueur devra rentrer dans l'univers pour découvrir les décors et traces de cette ancienne civilisation.

Illinx : Le jeu n'a pas pour vocation de véhiculer des sensations fortes.

Ludus & Paidia : Le jeu est codifié et même si le parcours du joueur est libre, il se rapproche plus du ludus que de la paidia.

Heuristiques de l'engagement ludique

Choix :

Le joueur possède une liberté de choix dans l'ensemble des actions qu'il peut faire. Toutefois, ces choix sont guidés par les besoins et contraintes auxquels il fait face lors de l'expérience de jeu.

Choix de mouvement :

Le joueur possède une liberté de déplacement dans l'environnement contraint par la topologie de la carte qui peut évoluer. Le prototype confronte également le joueur à des événements qui peuvent faire varier ses choix de direction.

Le joueur doit donc faire des micro choix lors de ses déplacements en fonction des avantages et inconvénients qu'il y perçoit.

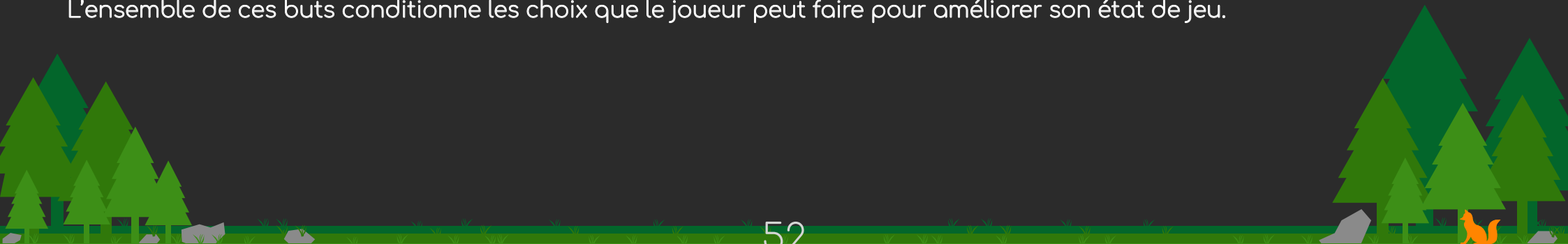
Ainsi le joueur peut prédire le choix qui lui sera le plus bénéfique ou le moins désavantageux en fonction de sa situation de jeu.

Paramètre de mouvement :

Son but peut être :

- Obtenir des ressources facilement.
- Obtenir des ressources plus difficilement mais qui peuvent être plus nombreuses.
- Se diriger vers une zone qui est plus sûre pour le joueur et lui permettre de stabiliser l'état de jeu dans lequel il se trouve.
- Explorer des nouvelles zones pour atteindre des points d'intérêts.

L'ensemble de ces buts conditionne les choix que le joueur peut faire pour améliorer son état de jeu.



Contraintes de déplacement :

Le joueur est également soumis à différents événements qui vont le contraindre dans ses actions et altérer les choix qu'il peut faire. Ces contraintes peuvent venir d'éléments extérieurs ou directement de l'état du joueur.

Les contraintes peuvent être :

- Contrainte de déplacement lié à la rotation des cycles d'immobilisation du nexus qui permet au joueur de se déplacer ou non.
- Contrainte de son état de jeu. Le joueur doit se préparer à passer certaines zones qui peuvent être difficile en fonction de l'état de ses ressources et unités.
- Contraintes du rayon d'alimentation qui limite le déplacement des unités.

Choix d'utilisation des ressources :

Le joueur possède un ensemble de ressources et unités qu'il peut utiliser pour effectuer différentes actions. Ses actions coûtent chacune un nombre de ressources différent. Le joueur doit donc gérer ses ressources en fonction des besoins qui se présentent à lui. Ses besoins sont liés à son état de jeu qui évolue en permanence.

Les ressources sont donc utilisées par le joueur pour différentes actions qui peuvent être centrées sur :

- La création d'unités.
- L'attaque de nouveaux points d'intérêts.
- La défense du nexus.

Toutefois le joueur reste contraint dans l'utilisation de ses ressources par les coûts des actions souhaitées et son état de jeu.



Variété :

Le joueur va rencontrer dans sa session de jeu plusieurs événements qui vont diversifier les situations auxquelles il peut faire face.

La variation de ces situations de jeu se fait par les différents cycles qui vont venir changer l'état et les propriétés des éléments de jeu.

Cette variété est observable dans le prototype par le cycle d'immobilisation du nexus qui vient modifier les actions que le joueur peut réaliser lors de ces différentes phases.

La variété des situations est aussi entretenue par le système de vagues ennemis qui vont être différent en fonction de l'endroit où se trouve le joueur.

Dans des versions futures du projet, d'autres éléments comme le système de marée vont faire varier la topologie de la carte ce qui influencera les décisions du joueur.

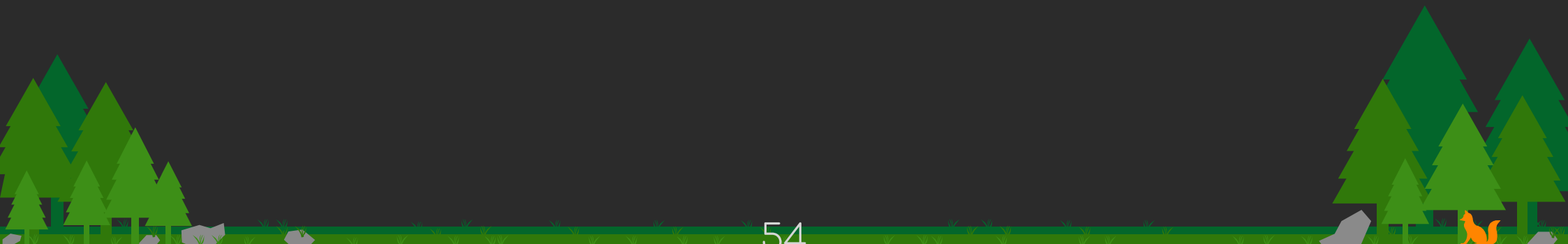
Conséquences :

Les choix effectués par le joueur amènent à changer son état de jeu vers un état qui peut être positif ou négatif.

Les conséquences de ses choix se feront ressentir sur la situation de jeu du joueur et vont l'affecter à court, moyen et/ou long terme.

Cela peut se caractériser lors d'une conséquence négative par un manque de ressources ou d'unités lors d'une phase de défense qui induit une plus grande difficulté à maintenir un état stable lors de cette phase.

Au contraire, une conséquence positive suite à un choix, pourra être ressentie par une aisance accrue à gérer une situation de défense ou la possibilité de construire davantage de bâtiments ou d'unités.



Predictibilité :

La prédictibilité est amenée de différentes manières.

Le joueur peut prédire ce qui va se passer dans des situations qui viendraient à se répéter, de par son expérience de jeu, comme le comportement des ennemis ou le nombre de ressources dans un dépôt.

Des signes comme sa future consommation d'énergie et la prochaine phase d'immobilisation du nexus sont clairement affichés pour montrer l'état du système au joueur.

Cela permet au joueur d'orienter ses choix en fonction de ses prédictions et de limiter les conséquences d'une situation négative sur son état de jeu.

Incertitude :

Le joueur lors d'une phase d'immobilisation va se retrouver dans une situation où il manquera d'informations, sur le nombre d'ennemis et l'endroit où ils apparaîtront. Cette incertitude peut le conduire à une mauvaise prédiction de l'état de jeu qu'il doit atteindre pour passer cette phase. Si le joueur ne prépare pas assez sa défense, il peut se retrouver dans une situation difficile liée à son état de jeu instable. Au contraire, s'il prépare trop sa défense, le joueur peut se retrouver avec un déficit de ressources pour les prochaines phases.

Cette incertitude peut également se voir lors des déplacements du joueur, le joueur pouvant mal prévoir les ressources nécessaires à son trajet.

Satisfaction :

La satisfaction du joueur sera principalement liée à son état de jeu :

- La réussite d'une phase de défense avec des pertes minimales.
- La réussite d'une phase de déplacement vers la zone fixée comme objectif.
- Atteindre un état de jeu stable.



REBORN

- Objectifs -



Objectifs de niveau

L'objectif final du joueur est d'atteindre la source d'énergie qui se trouvera au bout du dernier niveau. Cependant pour y arriver, il devra passer tous les niveaux précédents.

Le jeu final se compose de 12 niveaux divisés en trois catégories correspondant à l'objectif de celui-ci :

- Escorte de convoi
- Survie
- Remplissage

Le but étant d'offrir aux joueurs différents types de challenges et de changer les habitudes des joueurs, en modifiant sa manière de se déplacer, d'interagir avec des ennemis et la gestion d'énergie. En sortant des missions linéaires et similaires, nous cherchons à offrir des missions où le joueur sera plus propice à l'échec et devra repenser sa stratégie. Cela permet également d'enlever une partie de la redondance et de capter l'intérêt du joueur.

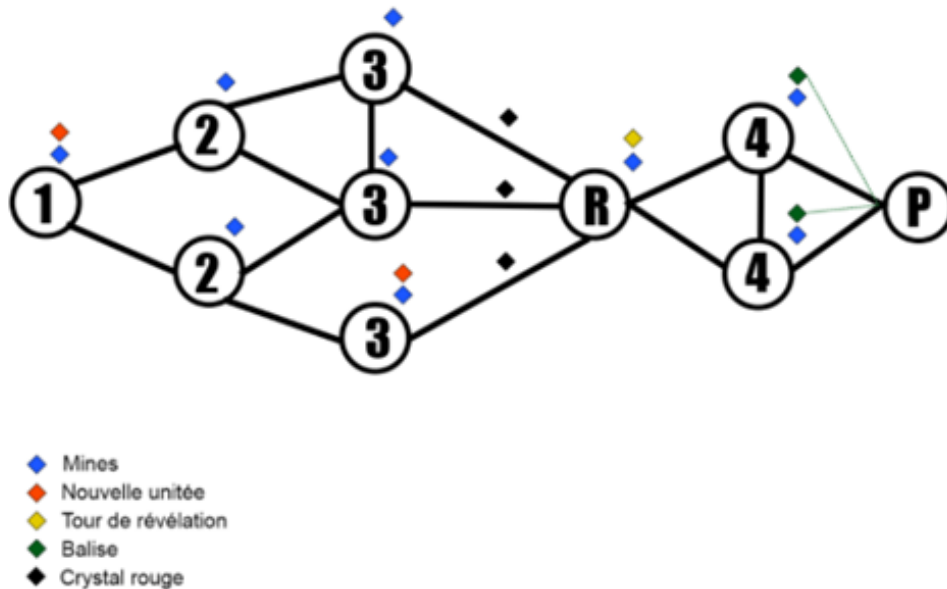


Escorte de convoi

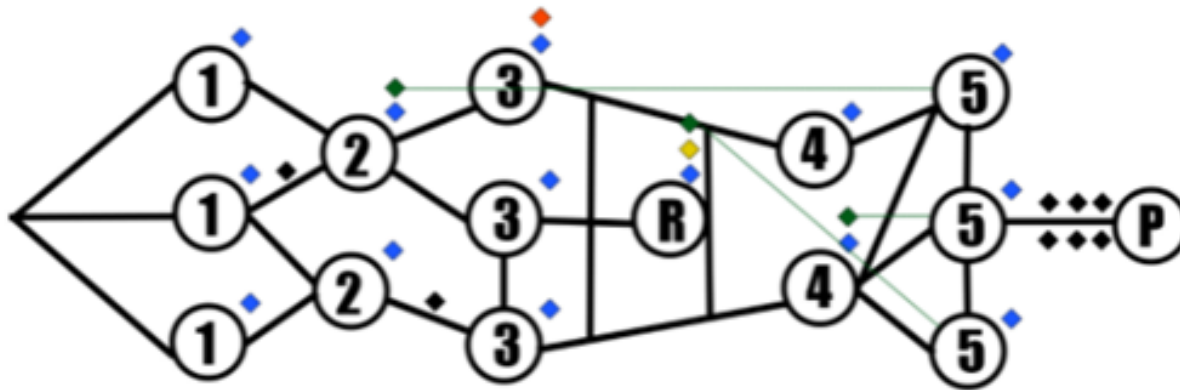
L'escorte de convoi est le premier style d'objectif que le joueur rencontrera, et le plus simple pour les débutants. Le style de jeu est assez lent, propice à apprendre de nouvelles mécaniques, et peu punitif en cas d'erreurs.

Cette mission consiste à amener le nexus d'un point A vers un point B. Ces missions sont plus guidées que les deux autres de plus le joueur traverse une expérience narrative avec des aspects stratégiques.

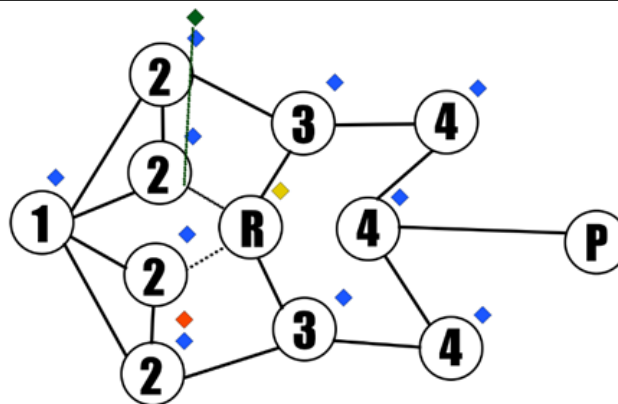
Une mission de type escorte de convoi se structure avec la création du chemin entre les différents points d'intérêts pour jouer avec le rythme jour/nuit imposé par le style de jeu, une transition entre deux points sur le schéma correspondant au passage du jour à la nuit.



Exemple de structure de mission de convois : facile

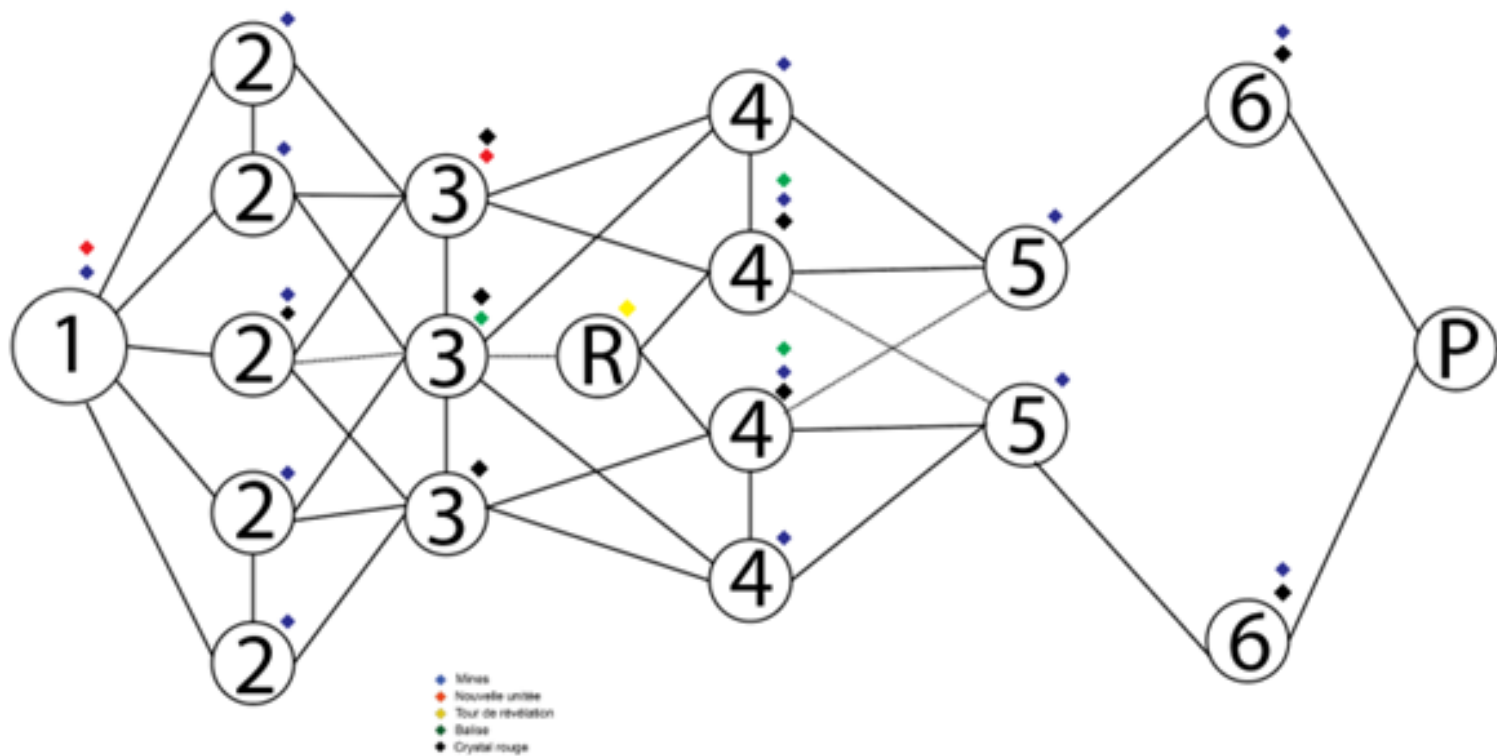


Exemple de structure de mission de convois : moyen



Exemple de structure de mission de convois : moyen

- ◆ Mines
- ◆ Nouvelle unité
- ◆ Tour de révélation
- ◆ Balise
- ◆ Crystal rouge

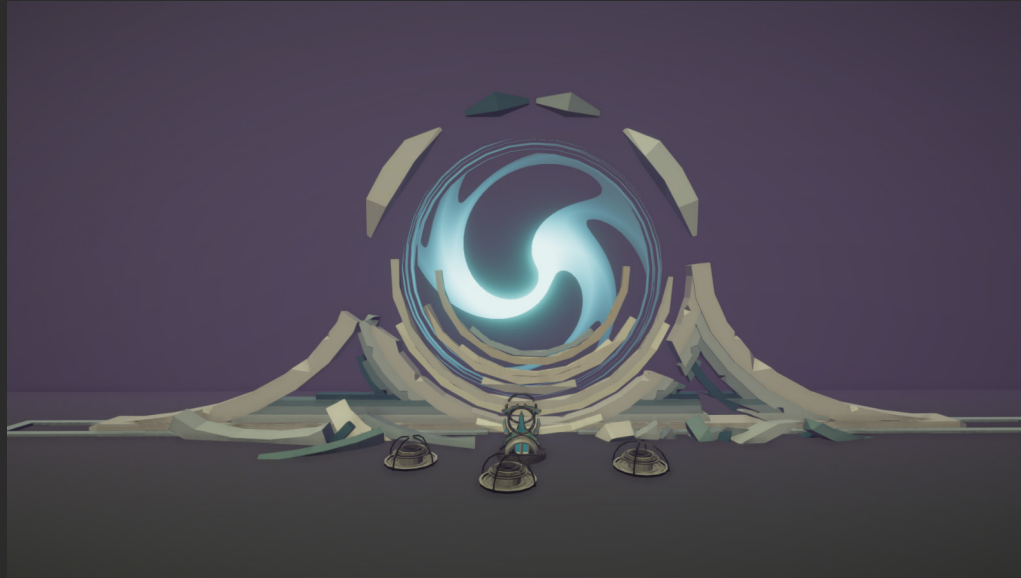


Exemple de structure de mission
de convois : difficile

Ce type de niveau introduit également un ingrédient spécifique, les balises et la porte.

C'est un système de défense conçu par la civilisation des nexus pour empêcher les invasions. Mais comme on peut le voir, cela n'a pas servi à grand-chose. Le nexus est cependant bloquée par celle - ci et devra l'ouvrir pour passer.

Une balise est un objet que le joueur devra alimenter une fois pour l'activer. Plusieurs balises sont reliées à une porte et quand toutes les balises connectées sont activées, la porte s'ouvre. un trait sur le sol entre la balise et la porte marque le lien de parenté entre les deux objets et change de couleur quand alimentée.



Ces balises peuvent être de trois types différents :

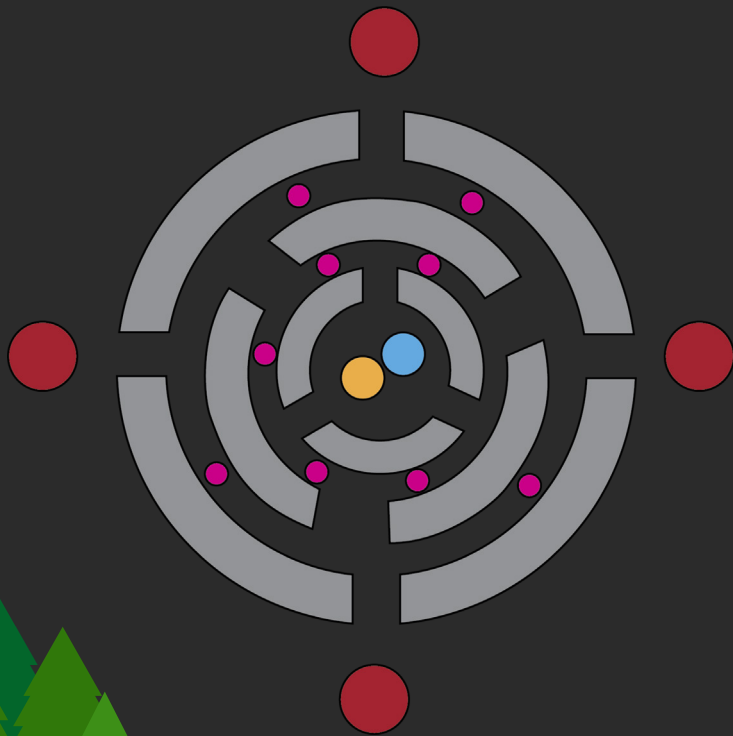
- classique : les balises activées le resteront.
- simultanées : les balises doivent être activées au même moment.
- compte à rebours : la balise se désactivera après une certaine durée.

Survie

Le deuxième style de mission est la survie.

Dans celui-ci une nuit éternelle s'est installée laissant le nexus immobile pendant celle-ci. Dans le rayon du nexus se trouvent des bâtiments et mines de ressources qui lui donneront un capital de départ et un moyen de défense. La partie se termine quand le joueur aura survécu un certain temps.

La carte est construite autour de l'emplacement du nexus et son objectif sera de survivre à plusieurs vagues d'ennemis. Des objectifs secondaires l'encouragent à prendre des décisions, comme détruire des terriers ennemis pour empêcher l'apparition d'ennemis à certains endroits.



Légende :
Rouge = Spawner à ennemi
Rose = Terriers
Jaune = Nexus
Bleu = Mine



Alimentation

Le dernier type de mission est le alimentation.

Dans celui-ci, l'objectif du joueur est d'accumuler des ressources sur la carte pour les amener à un réservoir commençant vide ou centre de la carte qui pompera automatiquement les ressources du nexus quand il s'en approche. La partie se termine quand le réservoir est plein.

Le joueur pourra accumuler de l'énergie dans les mines réparties sur la carte tout en conservant une partie pour créer des unités le défendant des ennemis sur la carte et des attaques nocturnes visant le réservoir, chaque coup le vidant légèrement.

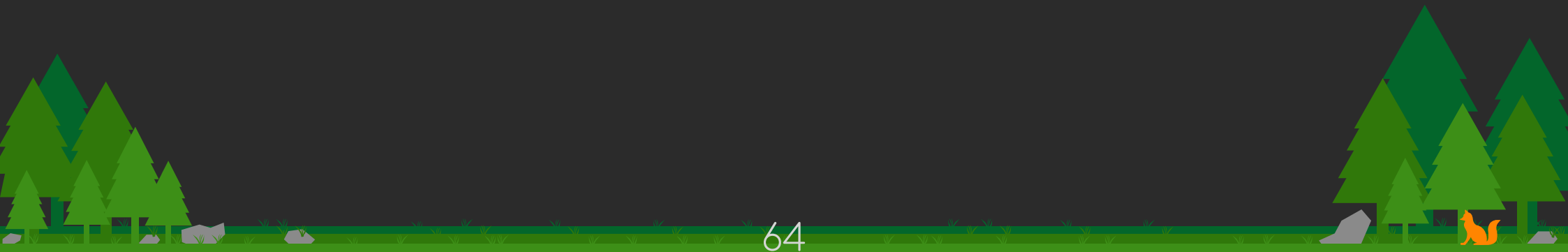
Un nombre de nuit maximum est lié à cet objectif et la partie se terminera par un game over si le joueur dépasse cette limite. Les cartes sont créées autour du réservoir qui sera situé au centre.

Le niveau final, consistant à réactiver la source d'énergie du nexus sera ce type de niveau.



REBORN

- Level Design -



Apprentissage

Inputs joueur	Déplacer	Récolter	Produire	Attaquer
Gameplay elements	Arbres	Mines	Nexus	Unités

Le premier input que doit apprendre le joueur est le déplacement du nexus car il est la base du jeu. Les joueurs doivent aussi se voir dicter un apprentissage autour des fonctions du nexus, celles-ci étant diverses et parfois difficiles à comprendre pour un joueur novice. Le nexus peut récolter de l'énergie dans les mines mais peut s'en voir prélevées par les unités qu'il alimente, il est donc impératif de bien faire comprendre ces concepts aux joueurs. Les joueurs doivent également apprendre le déroulement du cycle jour/nuit et des éléments qui en découlent.

Pour aider les joueurs à prendre en main les différents inputs et les aider à mieux comprendre les différentes fonctions de chaque éléments, et les incidences qu'ils ont dans une partie de jeu, un niveau tutoriel allié à un handboarding est désigné pour les accompagner dans le début de leur aventure.



Le handboarding se présente au joueur comme une succession de quêtes que le joueur peut suivre ou non. Les indications apparaissent en haut à gauche de l'écran et peuvent être dépliées par le joueur pour plus de précisions en cas de difficulté. Le joueur possède avec cette série de quêtes des explications sur les différents éléments de gameplay et leurs interactions entre eux. Il pourra ensuite avec les bases obtenues pendant ce niveau évoluer seul et acquérir de manière plus autonome ses prochaines connaissances en jouant.

A screenshot of a game environment with a dark, misty background and a path leading towards a small structure. The text '► Déplacez votre Nexus' is overlaid in orange.

► Déplacez votre Nexus

A screenshot of a game environment similar to the previous one, but with a blue arrow pointing to a specific location on the path. The text '▼ Déplacez votre Nexus' is overlaid in orange.

▼ Déplacez votre Nexus

Sélectionnez le Nexus Avec un clic droit.
Choisissez ensuite la destination avec
clic gauche

Niveau tutoriel

Dans un premier temps, on apprend au joueur les contrôles de bases de son nexus, c'est-à-dire la sélection de celui-ci, son déplacement, la création d'unités puis le contrôle de celles-ci pour mener une attaque.

Les challenges sont ensuite adaptés pour créer une progression et aider le joueur à prendre en main facilement les contrôles de son nexus et de son armée. Ensuite, le joueur sera confronté à l'apprentissage de la gestion de l'énergie, et l'arrivée des premières nuits et vagues d'ennemis force le joueur à perdre des unités au combat et se retrouver en difficulté au niveau de ses ressources. Les unités composant l'armée du joueur ayant un coût de production et un coût d'entretien, le challenge de combat évolue en adéquation avec le challenge économique du joueur.

A ce challenge économique et offensif s'ajoute un challenge défensif lors des attaques nocturnes où le joueur devra défaire des groupes d'ennemis tout en préservant son nexus. La progression de ce dernier challenge se fait sur trois axes qui dépendent du nombre de jours auxquels le joueur à survécu : le nombre d'ennemis sortant des terriers, le nombre de terriers actifs et la variante d'ennemi sortant de ces portails.



On a donc en paramètre de progression des différents challenges pour le jour :

- le nombre d'ennemis présents dans une zone
- la distance entre les mines d'énergies
- le nombre de ressources dans chaque mine
- les différents bâtiments présents sur la carte et leurs coûts d'entretien

Et pour la nuit :

- le nombre de chemin autour de chaque mine
- le nombre de portails autour de chaque mine
- le nombre de portails actifs
- le nombre d'unités créées
- la diversité des ennemis

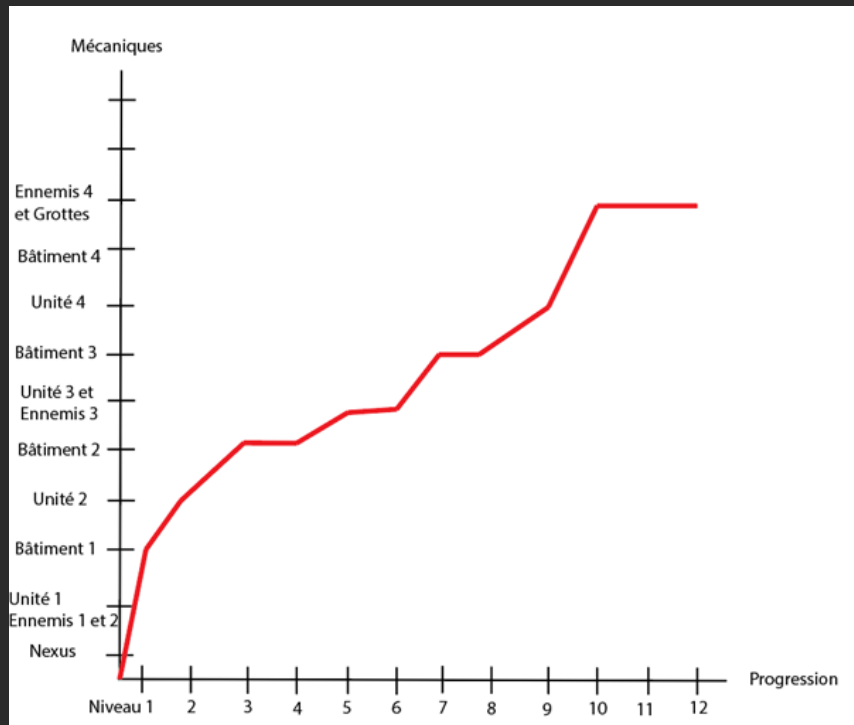
Nombres de portails actifs par nuit									
Night0	Night1	Night2	Night3	Night4	Night5	Night6	Night7	Night8	Night9
1	2	2	3	4	4	4	4	4	4

Coût par portail par nuit									
Night0	Night1	Night2	Night3	Night4	Night5	Night6	Night7	Night8	Night9
4	4	6	6	8	8	8	10	10	10

Blocage d'unité avant une certaine nuit			
Unit0	Unit1	Unit2	Unit3
0	2	6	6

Custom Waves					
	IsScripted	Unit0	Unit1	Unit2	Unit3
Night0	0	4	0	0	0
Night1	0	2	2	0	0
Night2	0	3	3	0	0
Night3	0	1	5	0	0
Night4	0	4	4	0	0
Night5	0	2	6	0	0
Night6	0	0	8	0	0
Night7	0	5	5	0	0
Night8	0	2	8	0	0
Night9	0	0	10	0	1

Apprentissage et difficulté



Graphique de rationalisation de l'apprentissage

Niveau 1	Niveau 2	Niveau 3	Niveau 4
Tous les inputs			
Unité 1 & Bâtiment 1	Unité 2	Bâtiment 2	

Niveau 5	Niveau 6	Niveau 7	Niveau 8
Input capacité unité 3			
Unité 3 & Ennemis 3		Bâtiment 3	

Niveau 9	Niveau 10	Niveau 11	Niveau 12
Input capacité unité 4			
Unité 4	Bâtiment 4, Ennemis 4 & Terrier		

Introduction des éléments au fil des niveaux

RGD

Documentation RGD des niveaux.

L'échelle va de F = facile à D = difficile en passant par N = normal.

On peut utiliser des + ou des - pour préciser encore plus la valeur de difficulté.

Bat = bâtiment et En = ennemi.

	Niveau 1				Niveau 2				Niveau 3				Niveau 4			
TLPM	T	L	P	M	T	L	P	M	T	L	P	M	T	L	P	M
Bât 1		F -	F -			F +	N -			N	N	F +			D -	N +
Bât 2										F	F +				N	
En 1		F -	F				F +	F +			N	N			D +	D -
En 2		F -					F				F +				N	N



Documentation RGD des niveaux.

L'échelle va de F = facile à D = difficile en passant par N = normal.

On peut utiliser des + ou des - pour préciser encore plus la valeur de difficulté.

Bat = bâtiment et En = ennemi.

	Niveau 5				Niveau 6				Niveau 7				Niveau 8			
TLPM	T	L	P	M	T	L	P	M	T	L	P	M	T	L	P	M
Bât 1			N				N-				N	N+			D-	D+
Bât 2			N-								D-				N	
En 1			N				N	D-			N	N+			D+	D-
En 2			N-				N				D-				N	N
Bât 3										F-	F	N-			N-	N
En 3		F-	F				F+				N-	F+			N-	N

Documentation RGD des niveaux.

L'échelle va de F = facile à D = difficile en passant par N = normal.

On peut utiliser des + ou des - pour préciser encore plus la valeur de difficulté.

Bat = bâtiment et En = ennemi.

	Niveau 9				Niveau 10				Niveau 11				Niveau 12			
TLPM	T	L	P	M	T	L	P	M	T	L	P	M	T	L	P	M
Bât 1			N				N	D-			D-	N+			D	D
Bât 2			D								D-					D+
En 1			N				N	D-			N	N+			D+	D-
En 2			N-				N				D-				N	N
Bât 3				N			N				N	N+			N+	D+
En 3			N-				N				D-	N+			D	N+
Bât 4						F-	F				F+	N-			N-	N
En 4		F-	F				F+				N-	F+			N-	N

Challenges

Différents challenges proposés au joueur en fonction de la mécanique.

Élément de Gameplay	Input joueur	Micro Challenge Physique			
		Reflex	Timing	Précision	Mesure
Timer de Nuit	Déplacement Nexus		Vitesse Nexus Position des zones sûres	Position Mine Position Tour Taille Rayon	
Réserve de ressources	Déplacement Nexus		Vitesse Nexus Position des zones sûres	Position Mine Position Tour Taille Rayon	Consommation
Unités ennemies	Attaque grâce aux unités	Vitesse Ennemis Position Ennemis	Temps de création des Unités. Vitesse des Unités	Placement capacités unités	Nombre d'unités pour gagner

Challenges

Différents challenges proposés au joueur en fonction de la mécanique.

Élément de Gameplay	Input joueur	Micro Challenge Mental			
		Stratégie	Tactique	Observation	Déduction
Défense nocturne	Déplacement Nexus et Unités		Placement Nexus Bâtiments utilisés	Choix emplacement mines et bâtiments	
Consommation	Déplacement Nexus et Unités	Chemin choisi par le joueur en fonction de la mission	Placement Nexus Bâtiments utilisés	Position Mines	
Attaque	Déplacement des unités		Placement Nexus Bâtiments utilisés Choix unités Placement unités	Trouver le meilleur angle d'attaque grâce a des unités.	

Biomes

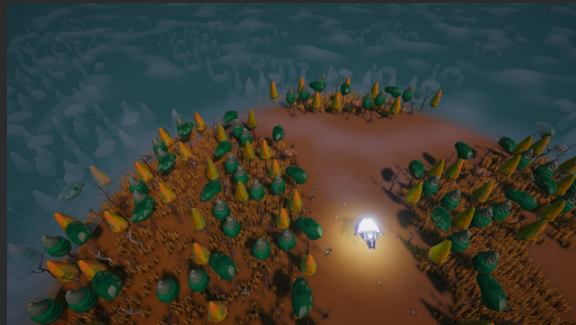
Le joueur en traversant les niveaux va évoluer dans trois biomes différents ayant chacun leurs spécificités de gameplay.

Il y a trois biomes et chacun de ceux-ci comprend 4 niveaux.

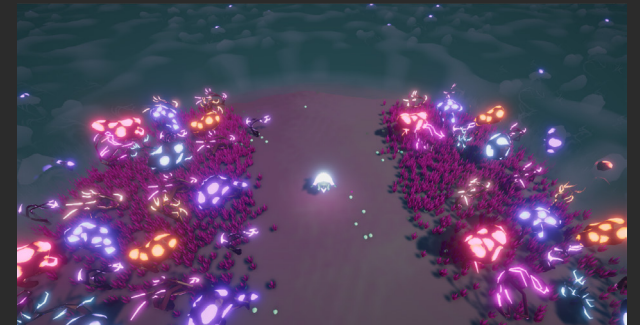
Ces biomes sont de plus en plus corrompus au fil des niveaux pour montrer que l'on s'approche de la source de la corruption, là où les larsen ont frappé le plus fort, là où il y avait le plus d'énergie.



Taiga : pas de modification



Savane : zones plus ouvertes et chemins plus larges



Forêt corrompue : chemins étriqués, peu de marges de manœuvres

Direction graphique environnement

Intentions graphiques environnement :

L'environnement de Reborn a été réfléchi pour le jeu final qui en découlera. Comme le joueur parcourt plusieurs niveaux pour ranimer son ancienne civilisation, il fallait que chaque biomes qu'il allait traverser puisse avoir une identité forte qui lui est propre. Dans ce but, nous avons érigé plusieurs lignes directrices en fonction de chaque biomes et de ce qu'ils devaient apporter au jeu en termes de sensation. Un autre lien avec notre narration a vu le jour au fur et à mesure de l'avancement du projet. Reborn étant un jeu de survie, nous avons fait le choix d'intégrer la narration directement dans l'environnement que le joueur pourra explorer. Ainsi, nos biomes présents permettent de suivre l'histoire du jeu en les traversant pour se rendre compte de l'ampleur de l'invasion des larsans, qui représentent les ennemis du joueur. Ce peuple apportant avec lui de la corruption, il était important pour nous que le joueur puisse ressentir cela grâce à notre environnement. Dans ce but, l'environnement évolue en conséquence de l'invasion des larsans.

En lien avec notre choix graphique, le low poly, nous avons pu créer de fort contraste entre chaque biomes et ainsi créer des transitions environnementales qui permettent au joueur de suivre son avancée dans l'histoire.



Références graphiques environnement :

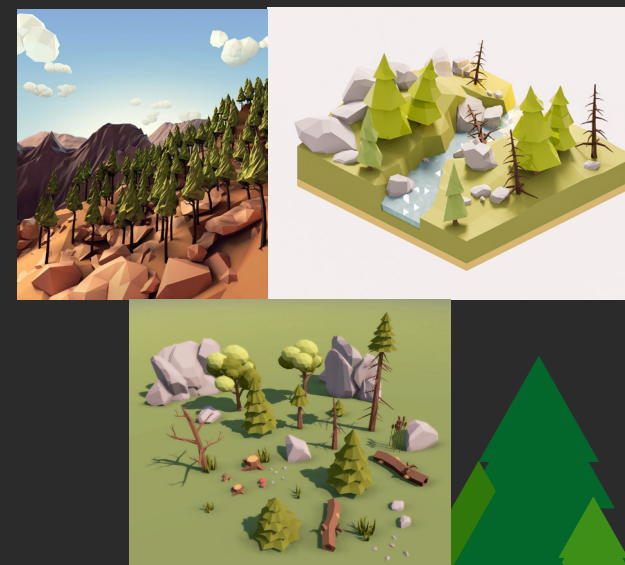
Astroneer : Une de nos références principales en termes de direction graphique a été Astroneer pour sa DA low poly. Ce jeu nous a largement inspiré pour la gestion des couleurs de nos biomes dans le but de pousser chacune de ces identités afin que le joueur puisse facilement les identifier.



Monument Valley : Dans le même cadre que le jeu Astroneer, ce jeu a été une inspiration pour ses gammes chromatiques et l'ambiance qui pouvait découler de ces choix de couleur. Il a notamment été utile lors des transitions entre les phases de jour et de nuit pour changer l'ambiance de notre jeu. Ce choix a été important pour faire comprendre au joueur les enjeux à chaque moment de la partie et lui faire ressentir différentes émotions en fonction de chacune d'entre elles.



Enfin, une grande partie de nos références graphiques s'est tournée vers des concepts art dans des banques d'images que nous nous sommes appropriés pour les adapter et les rendre cohérents à notre style graphique ainsi qu'à nos intentions environnementales.



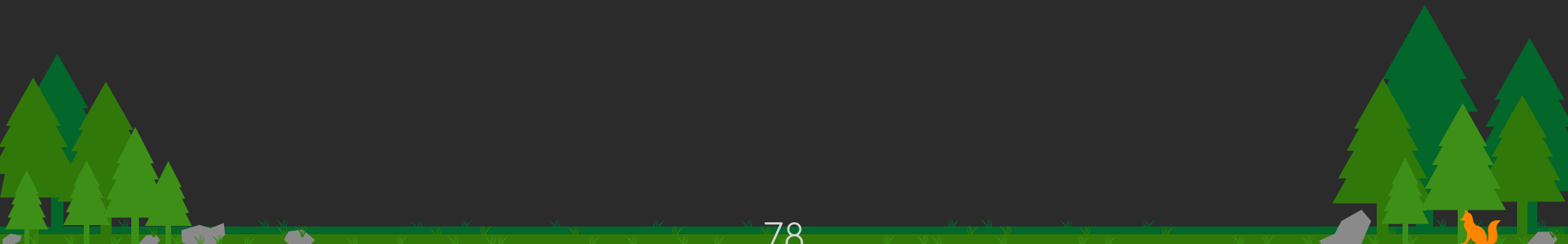
Direction sonore environnement

Intentions sonores environnement :

Notre jeu étant composé de plusieurs environnements distincts, notre intention principale était de créer des ambiances sonores uniques et évolutives en fonction des différentes phases de jeu.

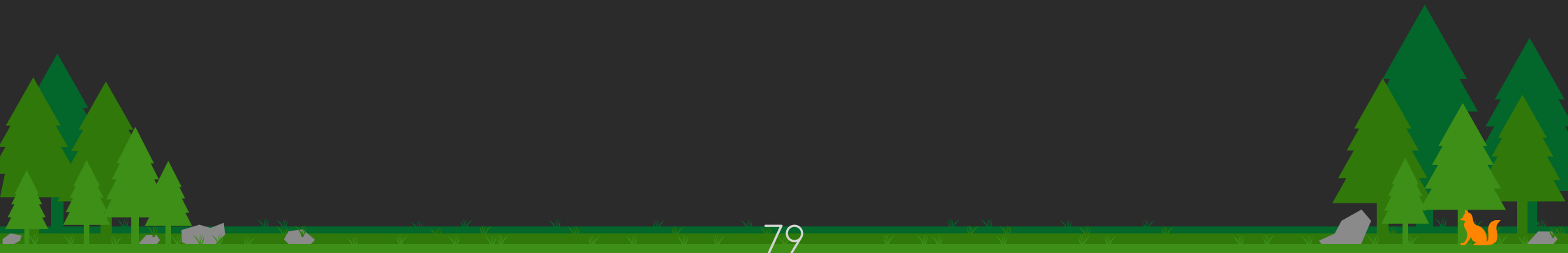
Un de nos enjeux majeurs pour le design des sons environnementaux a été de casser les différentes boucles qui se jouent continuellement. Pour cela, nous nous sommes inspirés de la façon de faire de league of legends. Notre façon de procéder était de créer différents layers principaux pour les sons environnementaux et d'ajouter des sons sur les différents objets de la carte. De cette manière le joueur entend différents sons en fonction du placement de sa caméra ce qui crée différents layers sonores et cassent la monotonie qu'on pourrait entendre.

Un autre point important du sound design environnemental était de créer deux ambiances bien distinctes entre les différentes phases de jour et de nuit. Pour cela, nous nous sommes questionnés sur les émotions que l'on devait faire ressortir lors de la phase de nuit en accord avec notre direction artistique. Cette phase étant plus rapide que la phase de jour, un sentiment de stress et d'anxiété devait être perçu par le joueur. Pour mettre en place cette situation, un snapshot a été ajouté au son de la nuit dans le but d'accentuer les bruits aigus qui sont principalement sur les ennemis. De plus, l'ambiance de nuit change entièrement pour passer d'une forêt calme à une forêt plus austère et dangereuse avec différents bruits d'animaux nocturnes.



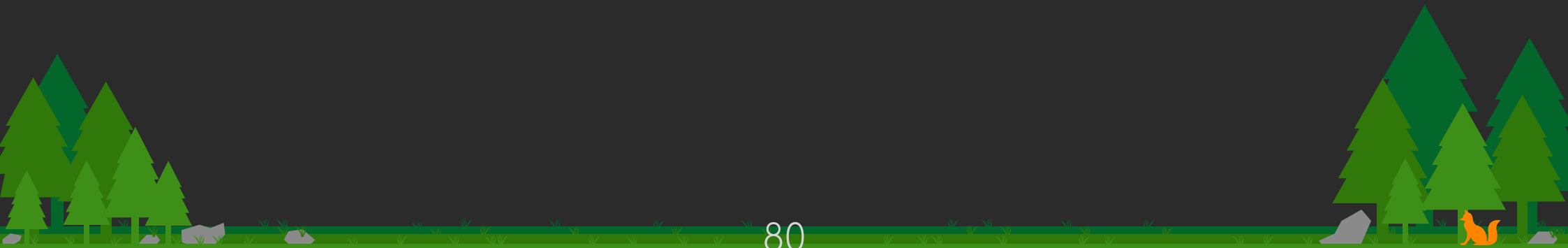
Références sonores environnement :

Pour aider à l'immersion du joueur dans notre univers, nos références sonores pour les sons environnementaux sont proches des sons réalistes que l'on retrouve dans les forêts et taïgas. Pour cela, nous nous sommes directement inspirés de différentes forêts avec des loops sonores qui correspondaient à l'ambiance visée.



REBORN

- User Interface -



Recherche en design :

Les HUD dans les RTS Je vais étudier à travers l'exemple de RTS références les différents HUD, leurs points communs et différences afin de pouvoir créer le HUD le mieux optimiser pour notre jeu. Notre jeu n'ayant pour le moment aucune base de DA et pas de liste de mécaniques précise je ferais un HUD provisoire qui évoluera à la cour du projet. Les RTS que nous avons pris comme références sont: Starcraft 2, Warcraft 3, Age of empires 3 et Northgard. Nous allons commencer l'analyse par un listage des différents éléments de chacun des HUD des RTS pris comme exemple Warcraft 3.



Gameplay Warcraft



Gameplay Warcraft

On observe d'abord que le HUD est découpé en 5 parties toujours présentes à l'écran, ces parties ont chacune un rôle précis, et sont ou nom différent en fonction des entités sélectionnées par le joueur.



La partie Rouge correspond à la carte et aux interactions avec celle-ci, c'est une partie fixe qui n'a en variable de modification seulement la carte comprise à l'intérieur, et les parties dévoilées ou non de celle-ci. La partie bleue est variable, elle s'adapte à/aux éléments sélectionnés par le joueur. Elle y comprend toujours en sa gauche les stats et la photo de l'entité sélectionnée / l'entité la plus forte de la sélection. En son centre sont affichées les tâches en cours si c'est un bâtiment, ou bien le nombre et la vie des unités de la sélection si c'est une armée. Si une seule unité est sélectionnée, cette zone affichera son attaque et sa défense. La zone verte est une zone disponible ou non en fonction de l'entité sélectionnée par le joueur, elle est disponible si celle-ci est un Héros, et elle en contient ses items. La partie orange est une partie « action », elle est donc variable en fonction de l'unité sélectionnée, elle affichera les possibilités propres à l'unité ou à la sélection. Par exemple, c'est dans cette zone que le joueur demandera à un bâtiment de créer des soldats ou à un paysan de fabriquer un bâtiment. La dernière zone, la zone rose est une zone de menu et d'information, elle est constante, elle affiche les trois ressources que possède le joueur, ainsi qu'un menu.



On observe aussi que le statut de santé de ses héros est toujours affiché en haut à gauche de l'écran. Starcraft 2 :



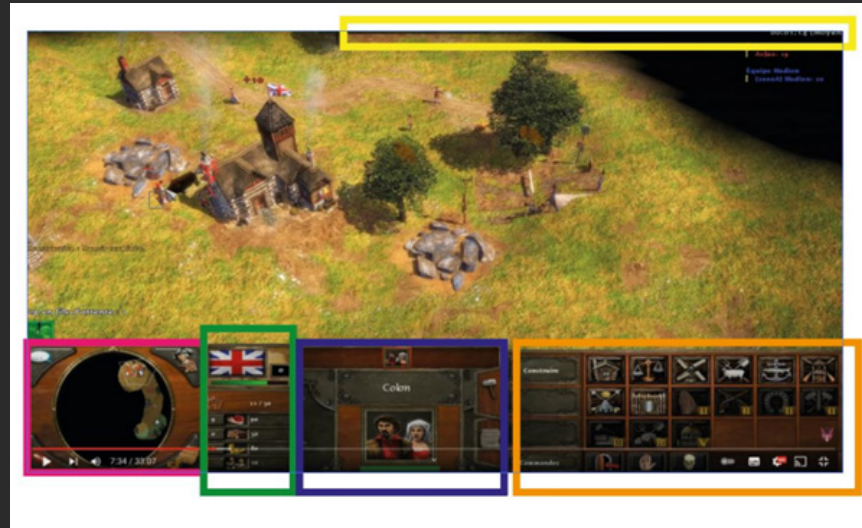
Gameplay StarCraft



Gameplay StarCraft

Encore une fois on observe un HUD composé de plusieurs parties constantes ou variables. La partie gauche fixe avec la carte affichée, la partie centrale avec la/les troupes ou le bâtiment sélectionnés ainsi que l'action en cours, exemple : créer des soldats. Une partie à droite d'action, où l'on peut gérer les actions de ses troupes, ouvriers et bâtiments. Nous retrouvons aussi en haut de l'écran la zone d'affichage des ressources du joueur.

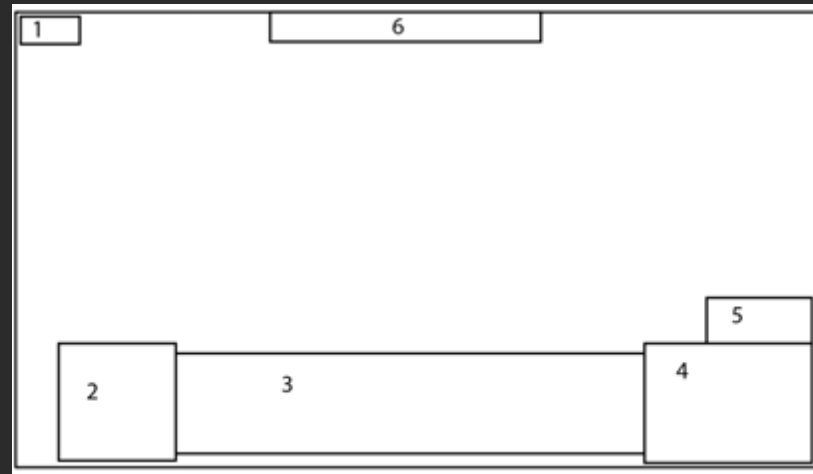




Gameplay Age Of Empire III

Toujours le même système à l'exception des ressources qui sont affichées en bas de l'écran. On observe donc que dans nos RTS de références les HUD sont très similaires, il y a un certain code propre au RTS, le but sera donc pour nous de constituer un HUD qui a sa personnalité sans trop casser les bases pour ne pas perdre les joueurs habitués au RTS. Nous allons alors créer aussi des zones constantes et d'autres variables.

Prototype :



La zone 1 correspond au bouton de menu.

La zone deux correspond à la carte et aux interactions avec celle-ci.

La zone 3 affiche les stats et l'état des unités ou bâtiments sélectionnés.

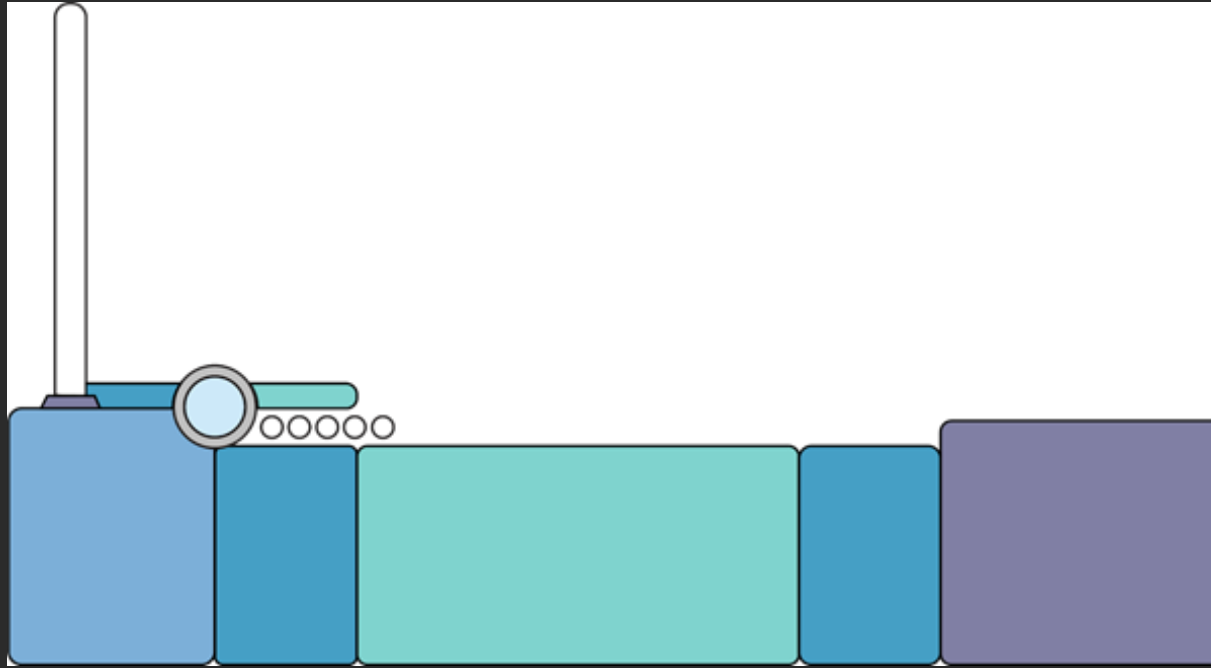
La zone 4 est la zone d'action pour construire, etc.

La zone 5 affiche les ressources, celle-ci a été placée juste au-dessus de la zone qui demandera au joueur de dépenser ses ressources pour une meilleure visibilité sur celles-ci.

La zone 6 informera le joueur du temps écoulé depuis le début de la partie.

Après itérations et décisions de Game Design, certains aspects de Gameplay ont légèrement modifié certains éléments du HUD. Le résultat du premier prototype de HUD intégré à notre expérience de jeu nous donne de façon claire toutes les informations dont le joueur a besoin.

Prototype :



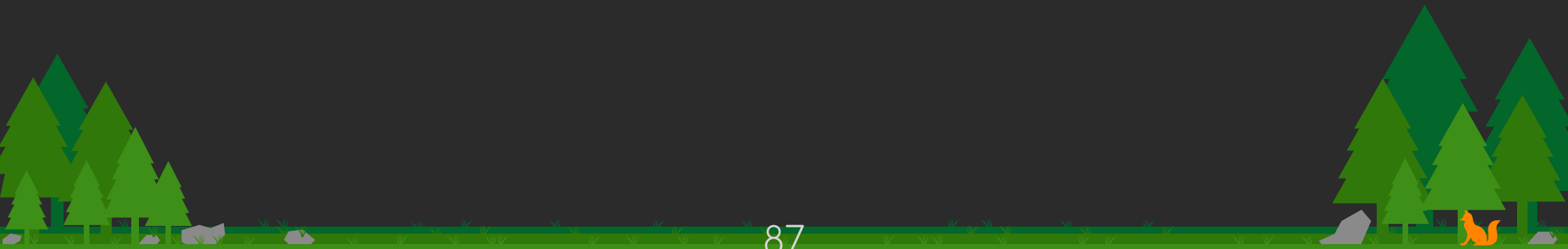
On y observe un réservoir à gauche servant à montrer graphiquement la quantité de ressource du joueur. On trouve aussi un cercle/chargement indiquant le timer avant un changement de phase de jeu pour le joueur. Démonstration prototype du HUD en jeu, il manque encore l'intégration de plusieurs éléments tels que la mini map.

On a observé dans la majorité des RTS un HUD toujours présent à l'écran et masquant jusqu'à un tiers de la zone de jeu. Nous avons pris la décision de mettre un paramètre de transparence à notre HUD pour perdre moins de visibilité sur l'espace de jeu sans pour autant perdre en informations.

Prototype :



Après des PlayTests, il est ressorti que ce HUD n'était pas clair, les joueurs ne faisaient pas le lien entre les valeurs numériques et leurs utilités. Une deuxième version plus claire a donc été désignée dans l'intention de pallier aux problèmes de lisibilité.

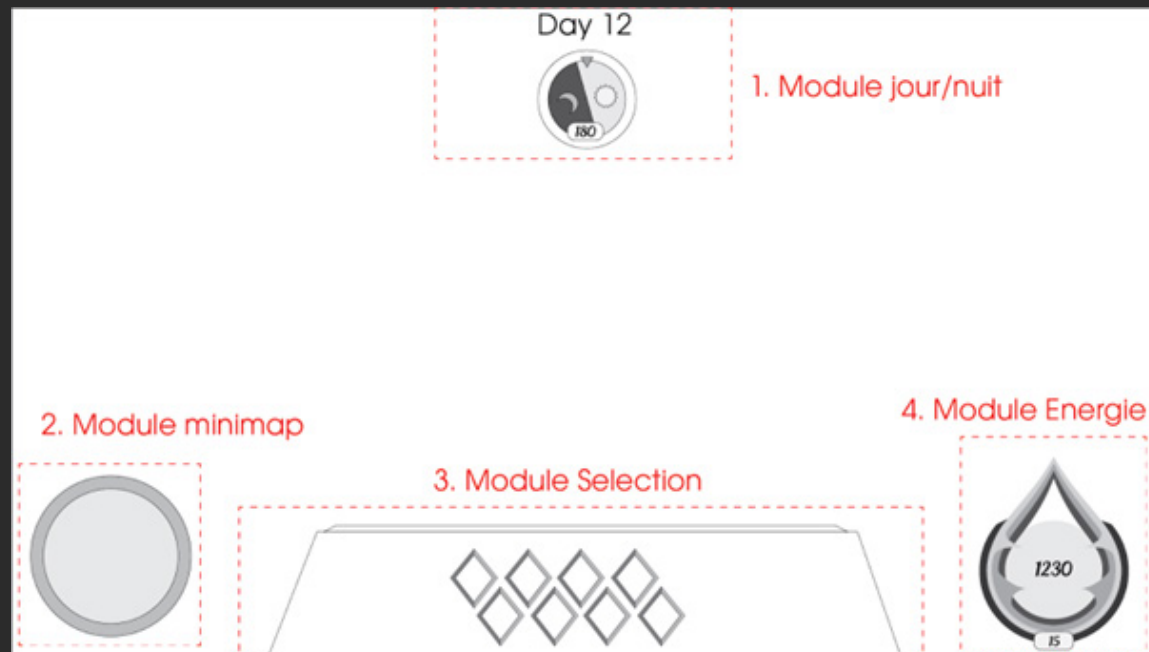


Changement du système du HUD

Après de nouveaux playtests, nous avons conclu que la nouvelle version du HUD était plus lisible et plus appréciée des testeurs. En revanche nous n'étions pas satisfaits du design et de la place qu'il prenait constamment sur l'écran même quand il n'affiche aucune information. Nous avons donc décidé de refaire une nouvelle version du HUD en gardant en tête les anciennes erreurs pour ne pas le refaire. Le nouveau HUD est en cours de test et nous attendons les retours pour pouvoir y faire des corrections.



Explications générales



L'interface en jeu a été conçue pour donner un maximum d'informations aux joueurs tout en les gardant claires. Pour cela nous avons séparé l'interface en 4 modules distincts sur le critère de leur fonctions :

- le module jour / nuit affiche la période de la journée actuelle
- le module énergie affiche l'énergie, la consommation et le niveau du nexus
- le module minimap qui sert à afficher la carte
- le module sélection rétractable qui affiche les actions disponibles en fonction de l'élément sélectionné.

Cette interface est semi-transparente pour permettre au joueur de discerner ce qui se passe dans les recoins qu'elle recouvre et dispose de mouse-over pour donner des précisions sur les éléments de celui-ci en jeu.

Module jour/nuit

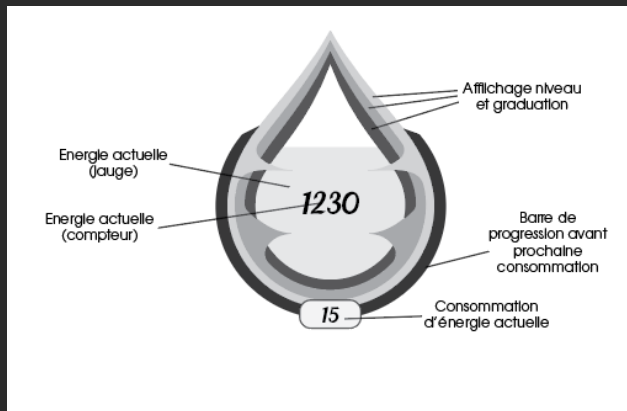
Le module jour/nuit a pour but de montrer les informations de la période actuelle de la journée et le temps de progression jusqu'à celle-ci.

Nous avons opté pour un logo central représentant une lune ou un soleil en fonction du moment de la journée et une barre de progression circulaire pour représenter la progression de la journée.

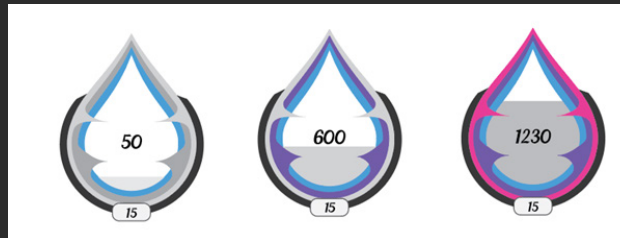


Module d'énergie

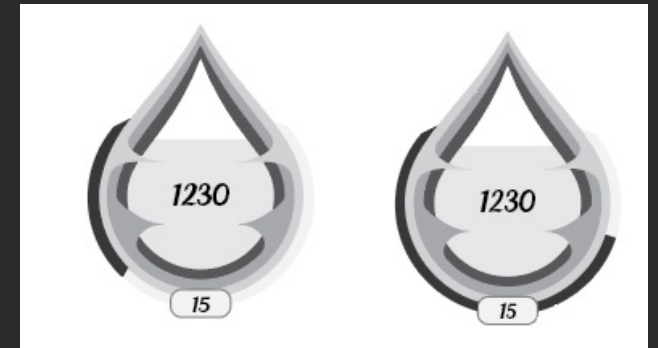
Le module énergie doit afficher trois informations : le niveau du nexus et les paliers de ceux-ci, l'énergie actuelle du joueur et sa consommation d'énergie.



L'énergie actuelle du joueur est affichée par la partie intérieure du module se remplissant au fur et à mesure et changeant de couleur ainsi qu'une valeur numérique. Autour d'elle se trouve trois couches représentant des niveaux avec des graduations pour indiquer le niveau de remplissage à atteindre pour passer au niveau supérieur. Elles changent également de couleur quand le nexus monte de niveau.



La consommation d'énergie est affichée à la base du module avec une barre de progression et une valeur numérique.



Module mini-map

Le module minimap a pour but de montrer un récapitulatif de la carte au joueur avec les zones révélées dans le brouillard de guerre.

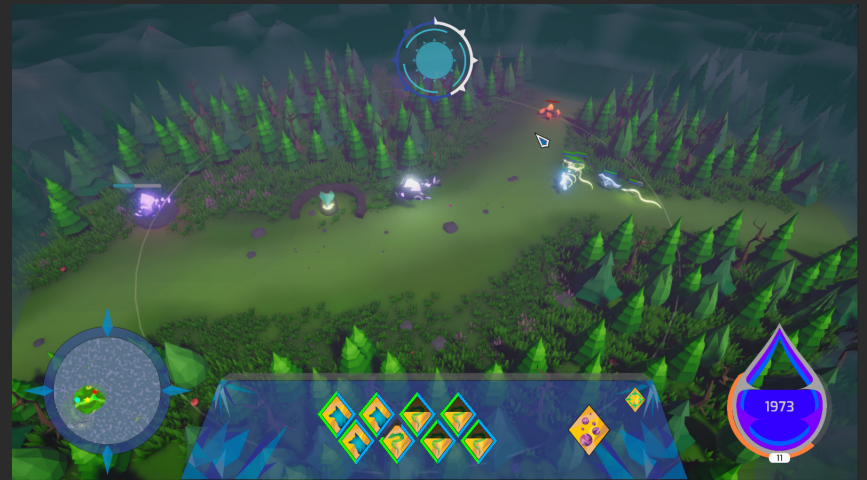
Le joueur peut naviguer dans cette carte avec la souris en maintenant le clic gauche enfoncé et faisant des zooms avec la molette.

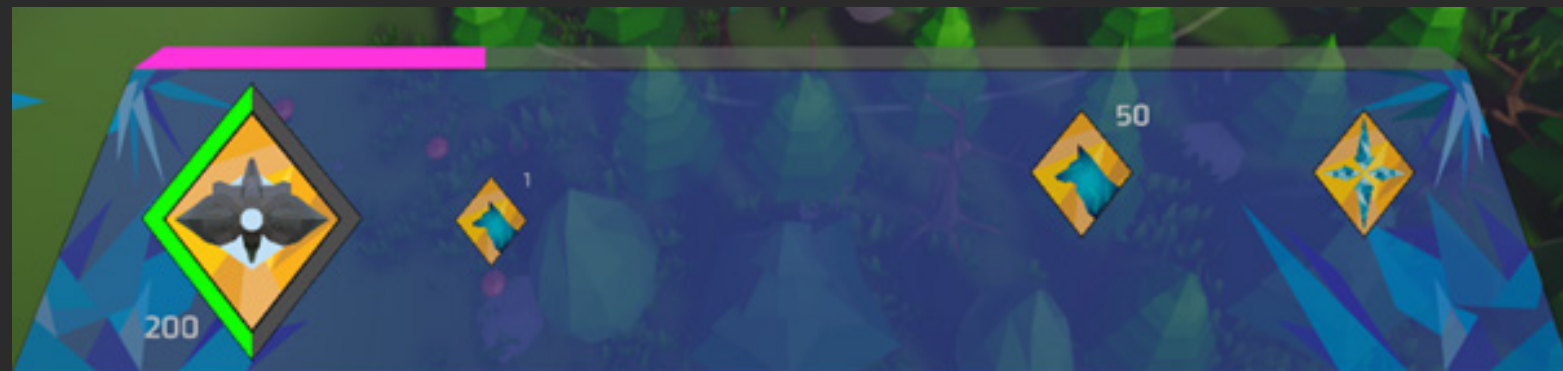
Des icônes sont également présentes pour que le joueur repère facilement les structures et agents autour de lui.



Module sélection

Le module Selection est le module central qui évolue en fonction de ce que le joueur a sélectionné.
Si le joueur n'a rien sélectionné, le module se rétracte pour laisser plus de place au champ de vision du joueur tout en laissant la barre de production d'unités visible.



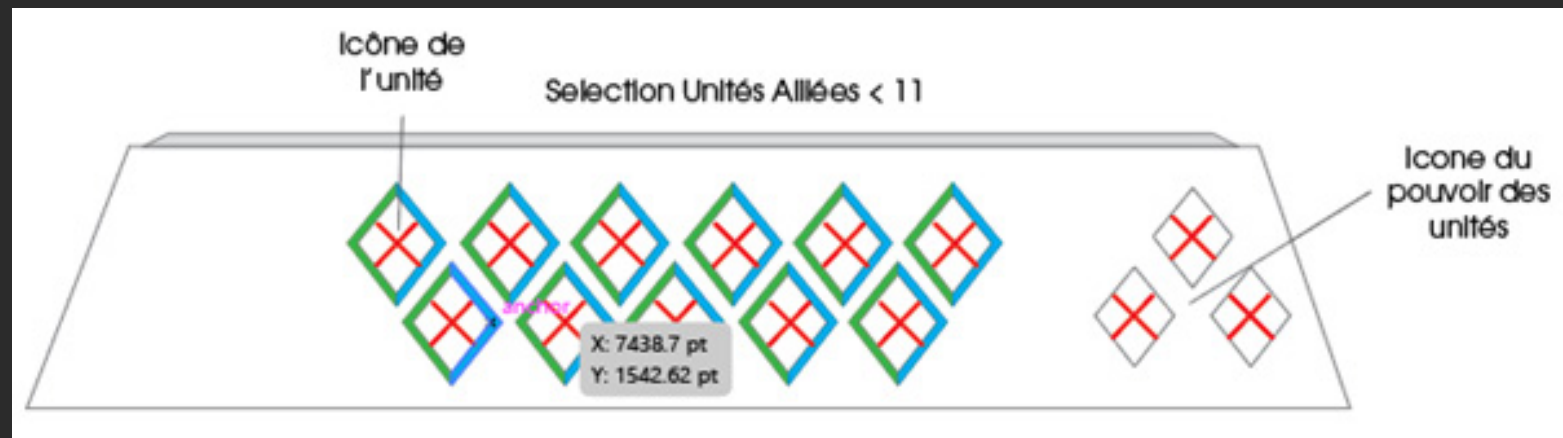


Sélection du Nexus

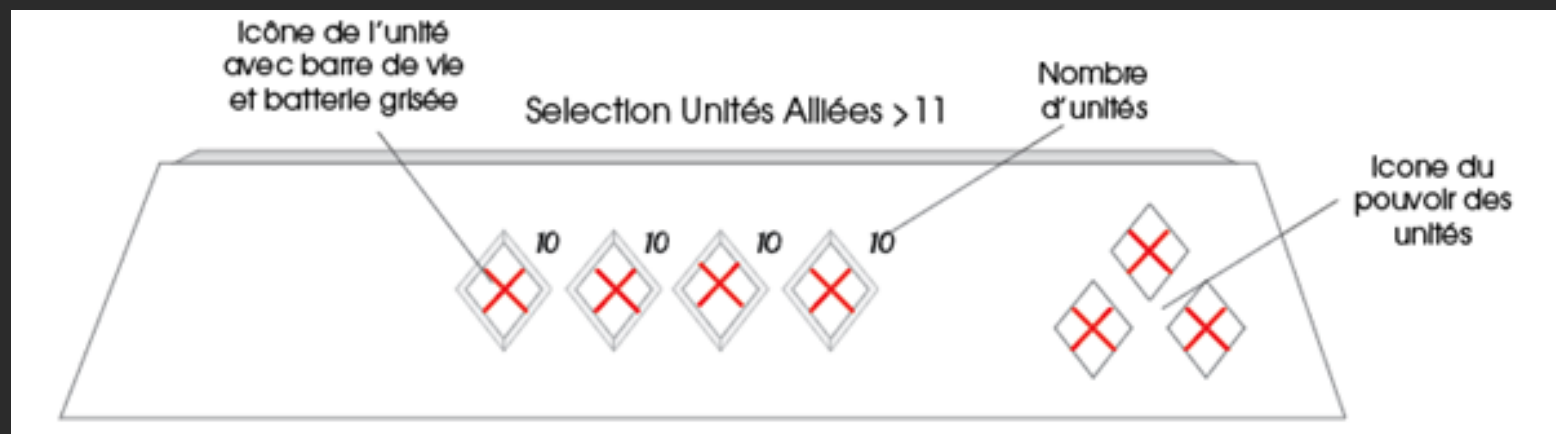




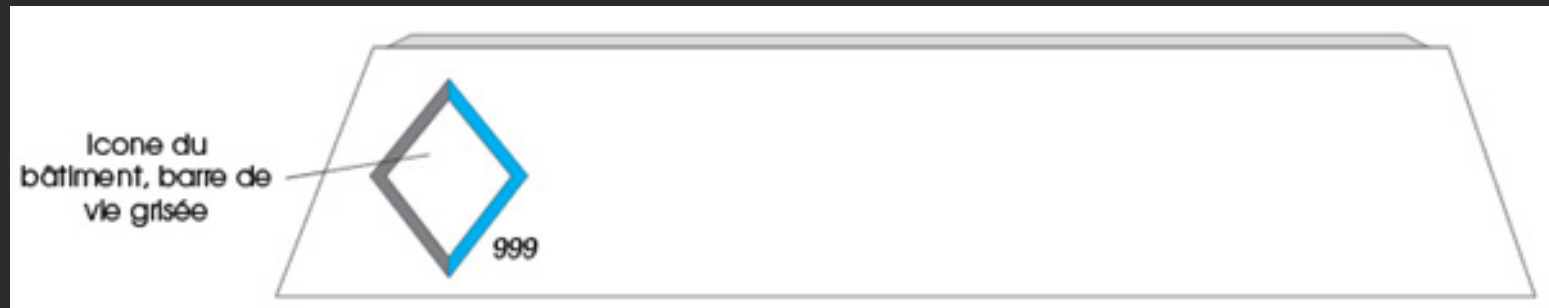
Sélection d'une unité alliée unique, la partie grise dans l'icône est une barre de rechargement de compétence



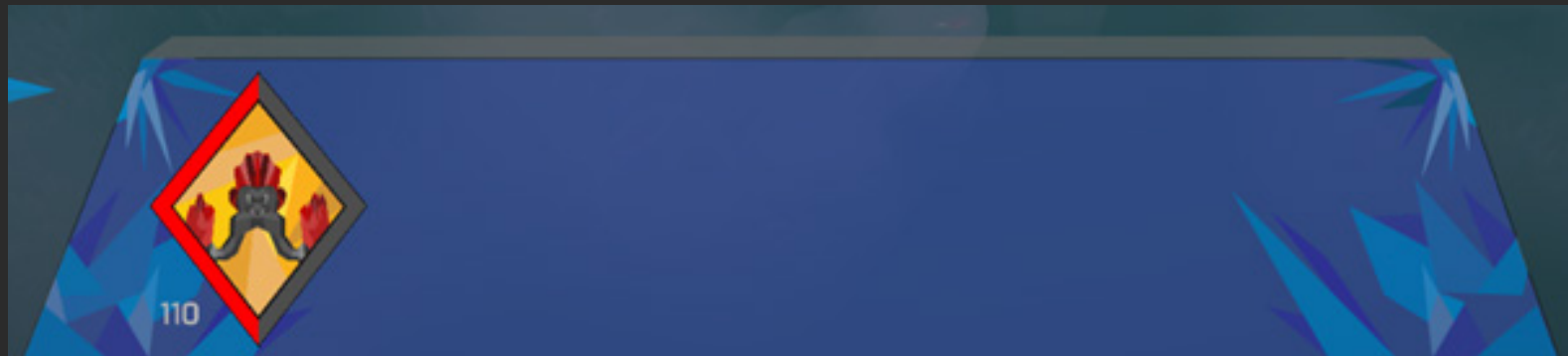
Sélection d'unités alliées dont le nombre est inférieur à 11



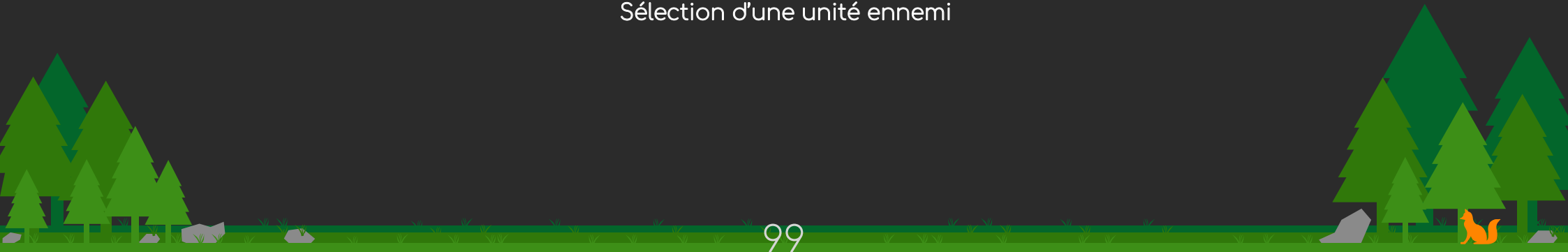
Sélection d'unités alliées dont le nombre est inférieur à 11

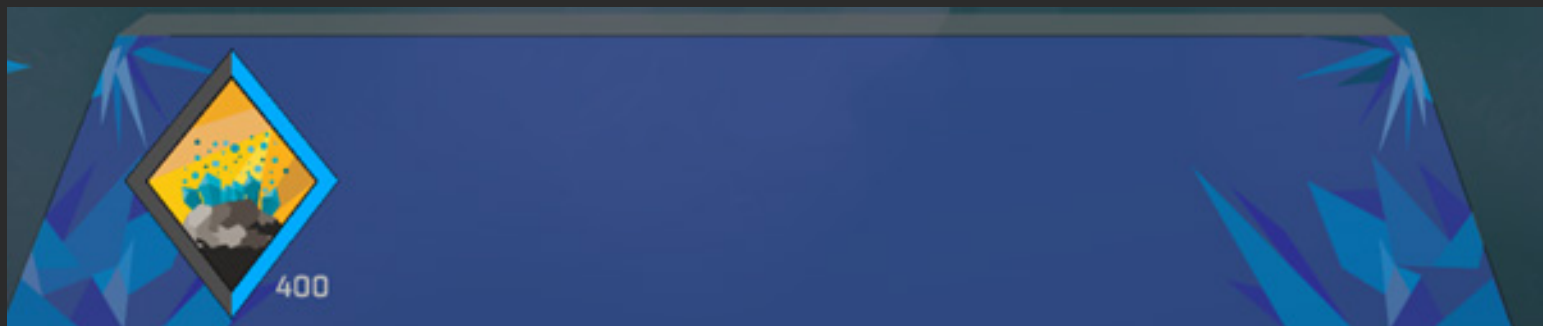


Sélection d'unités alliées dont le nombre est inférieur à 11



Sélection d'une unité ennemi





Sélection de ressources



Rendu final



Direction sonore HUD

Intentions sonores HUD :

Concernant notre HUD notre design sonore a été entièrement réalisé en 2D. Étant en permanence visible par le joueur, il était impératif que le joueur ait toujours accès aux feedbacks sonores.

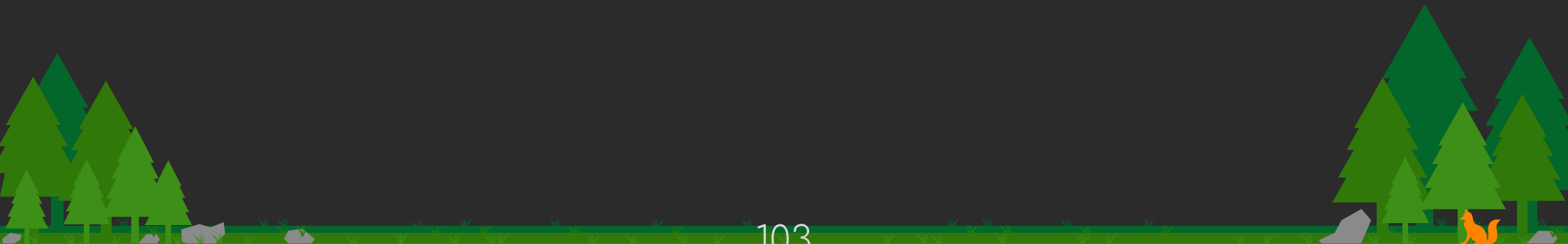
Références sonores environnement :

Pour nos références concernant l'HUD, nous nous sommes principalement inspirés des sons de notifications mobiles. Ce choix nous a permis de facilement mixer ces sons pour les rendre cohérents à notre direction sonore futuriste, sans pour autant qu'ils soient trop complexes, ce qui aurait noyé les sons du jeu lors de son utilisation. Pour la gestion de nos sons nous nous sommes servis de références comme Starcraft 2.



REBORN

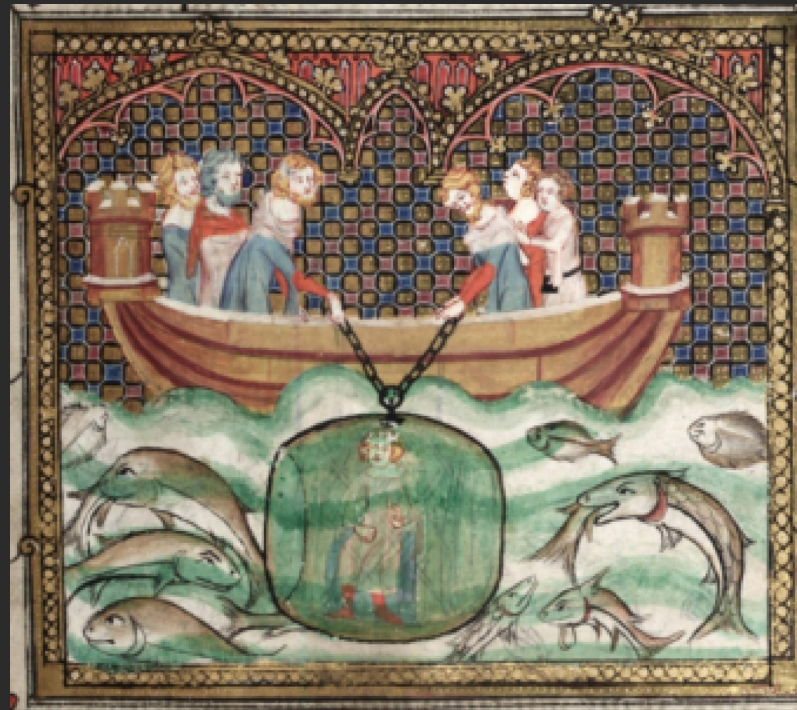
- Direction Artistique -

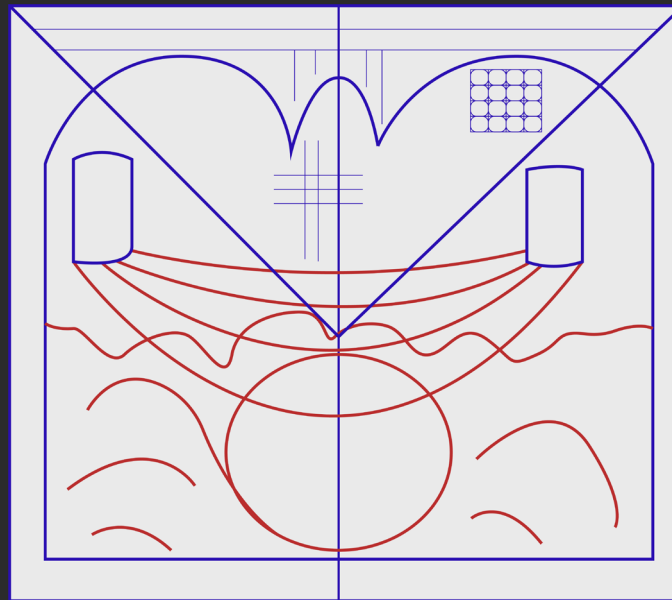


Direction Artistique :

Analyse du référent :

À la suite de l'image de référence, nous avons décomposé ce qu'elle pouvait représenter pour nous. D'abord, dans une optique de gameplay qui nous a donné le genre actuel, nous avons ensuite abordé cette image d'un point de vue artistique.





— Lignes correspondantes à la partie structurée

— Lignes correspondantes à la partie organique

Cependant, nous nous sommes écartés de la thématique que nous présentait l'image pour analyser dans un premier temps la composition.

Il nous est donc apparu deux plans principaux, contrastés par la formation des lignes de construction. Le premier plan placé en haut est soumis à une forte géométrie avec des lignes symétriques et des formes cubiques, rectangulaires et rondes. Le second plan situé dans la partie basse de l'image est lui beaucoup plus organique et libre de formes. On y comprend la différence entre deux mondes opposés. L'un connu, symbolisé par des lignes de construction ordonnées et architecturées, l'autre inconnu, laissé à des lignes libres et courbes, encore brutes.

Intentions graphiques générales :

Intimement liée à notre genre de jeu qui était au départ dans le style du RTS, notre choix de direction graphique s'est porté sur du Low poly. Ce choix est lié aux nombreux éléments qui allaient être présents en permanence à l'écran. Pour éviter la confusion du joueur et donc de le faire sortir de l'immersion, nous avons voulu créer un environnement facilement analysable. Notre ligne directrice a donc été de créer des assets graphiques en low poly avec comme intention de toujours faire ressortir les formes les plus simples pour faciliter la compréhension du joueur lors de ses parties. Ce choix s'est aussi présenté à nous au vu d'un déficit de compétences centré sur la direction graphique dans notre groupe. Le low poly nous permet donc de nous concentrer sur le fond dans un premier temps avant de donner la forme, en fonction de notre volonté finale pour chaque assets.



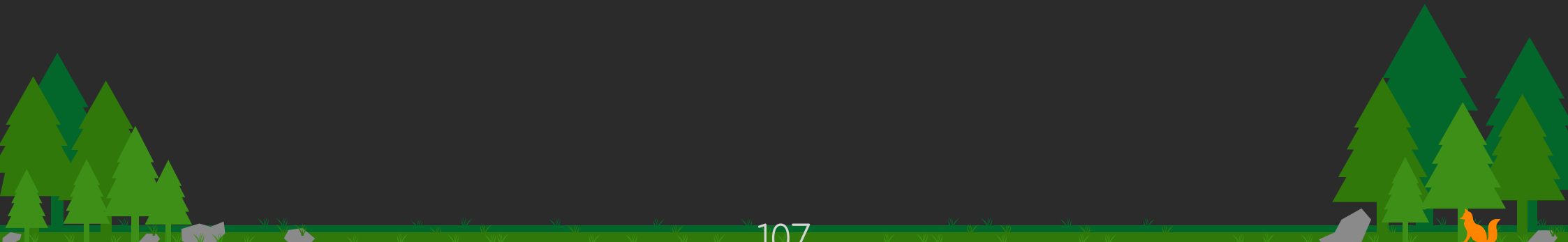
Intentions sonores générales :

Nos intentions sonores pour notre jeu est de donner au joueur différents environnements sonores en accord avec ce qu'il se passe en jeu.

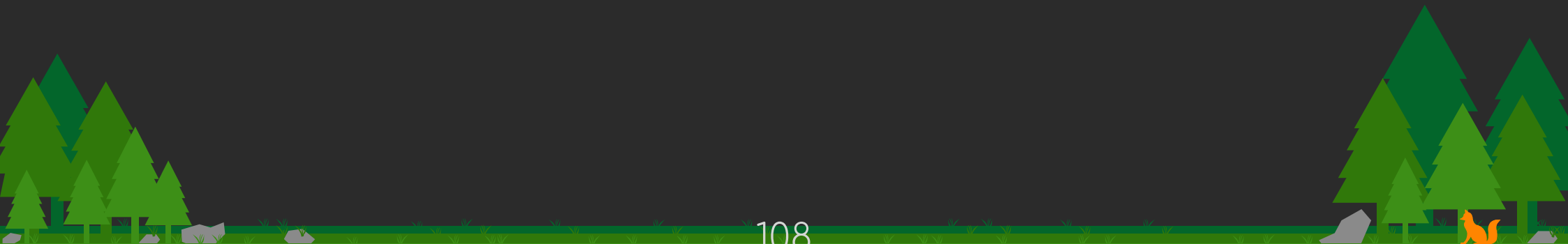
Reborn est un jeu possédant deux phases principales, jour est nuit, il est important de bien symboliser ces deux phases séparément pour plonger le joueur dans différents états.

Pour permettre une immersion complète, le sound design a été pensé pour être entendu majoritairement avec un casque. Ce choix vient du design spatialisé d'une majeure partie de nos sons. L'intérêt de ce choix vient du nombre de sons que le joueur peut entendre en même temps. Reborn étant inspiré des mécaniques d'un RTS, beaucoup d'éléments peuvent se trouver à l'écran et interagir entre eux. Pour faciliter la compréhension est ne pas perdre le joueur, plusieurs lignes directrices ont été définies dès le début de la création de notre sound design.

Dû au nombreux sons présents dans le jeu, l'affordance s'est trouvée être notre point principale en termes de contrainte de design. Tout en restant claire dans nos références pour ne pas nous éloigner de l'ambiance visée, chaque son a été pensé pour être le plus affordant possible dans le but de ne pas perdre le joueur lors des moments à haute intensité sonore. Pour cela, un travail de design a eu lieu sur chaque son de façon à ce qu'il s'intègre au mieux aux actions et feedbacks du jeu. Pour aider à cet effet, le son a également été synchronisé sur les différentes animations de notre jeu pour aider à ce que le feedback aussi bien visuel que sonore soit le mieux perçu par le joueur.

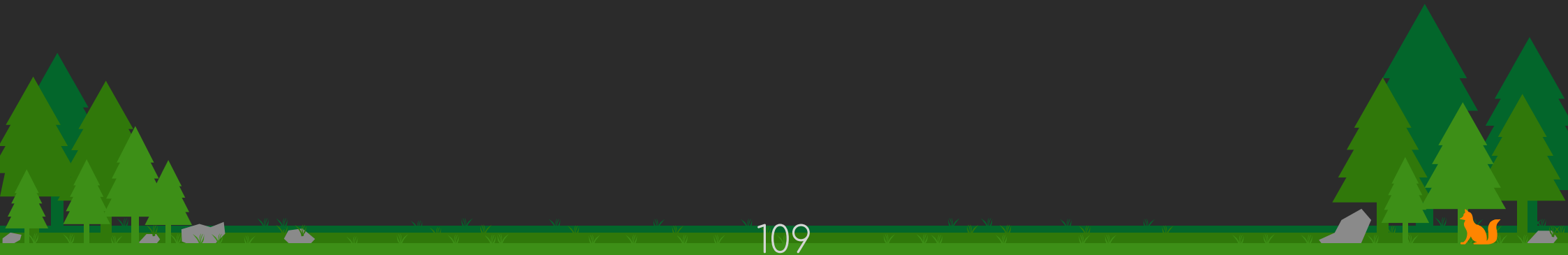


Comme dit précédemment, nous avons également fait le choix de spatialiser une partie de nos sons. Grâce au fonctionnement de notre listener ancré sur la caméra, le joueur peut garder en permanence un lien entre ce qu'il voit et ce qu'il entend. Reborn permettant au joueur une liberté totale au niveau de la caméra, nous ne voulions pas créer de dissonance en plaçant le listener sur le player. Ce choix permet également de créer un sentiment de stress lorsque le joueur ne regarde plus son nexus. N'ayant plus de feedbacks sonores de ce qu'il pourrait se passer, éloigner la caméra du nexus représente un danger parfois nécessaire si le joueur veut récolter des informations un peu plus loin. Un autre point plus micro de la spatialisation des sons a été de faciliter la compréhension de ce qui pouvait se passer à l'écran, notamment lors des combats qui représentent des moments de forte intensité. Le joueur peut grâce à ça, comprendre ce qu'il se passe aussi bien à gauche qu'à droite de son écran et réagir en conséquence.



REBORN

- Narration -



Narration

La narration dans notre jeu prend deux formes

La première plus classique est une narration linéaire dans la forme d'une bande dessinée interactive présente au début du jeu qui présente au jeu les bases du scénario : le nexus se réveille après plusieurs siècles d'inactivité, il se rappelle que sa civilisation a été envahi par les larsan, il regarde autour de lui et tout est en ruines, c'est le seul nexus survivant et la source d'énergie de son peuple est tarie. Il va donc décider de la réactiver en accumulant de l'énergie au fur et à mesure de son périple.

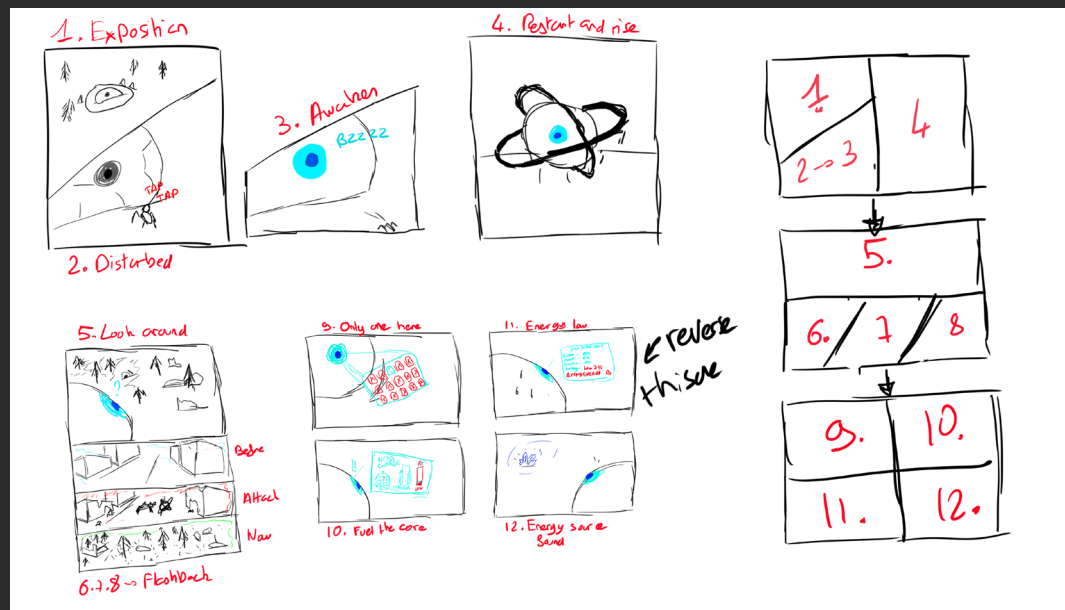
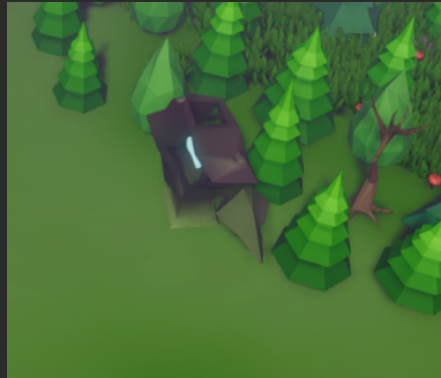


Schéma préliminaire du story board de la bande dessinée introductive

La deuxième forme est la narration environnementale. Des ruines sont disposées dans le décor et aide à déterminer le but et l'utilité qu'en avait la civilisation du nexus.



Direction graphique bâtiment

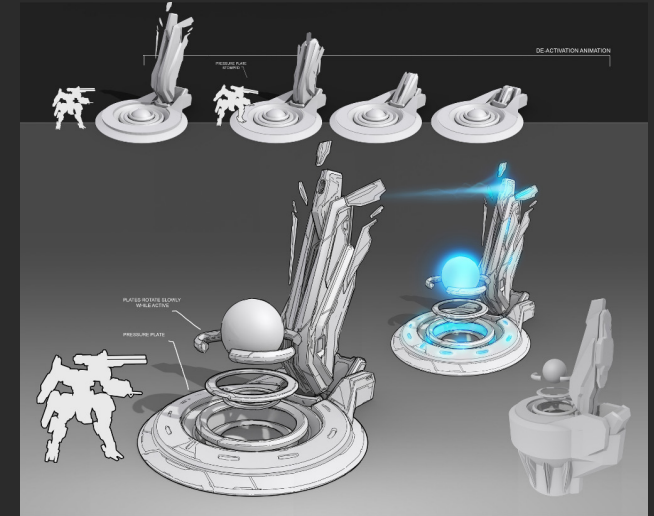
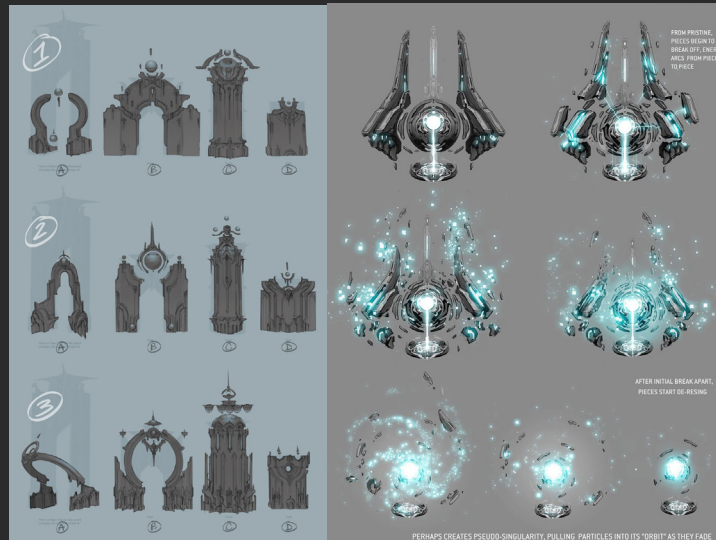
Intentions graphiques bâtiment :

Nos intentions concernant nos bâtiments ont grandement évoluées lors de la mise en place de la narration de notre jeu. Dans ce but, un large travail a été réalisé pour créer une ligne directrice claire de ce que nous voulions concernant le design de nos bâtiments. Les bâtiments qui servent aux gameplays ont toutefois été designer en correspondance de leurs utilités au sein du jeu pour garder le principe d'affordance qui nous est cher. Cependant, d'autres bâtiments qui ne rentraient pas dans cette catégorie ont eu pour ligne directrice, de servir principalement à la narration environnementale de notre jeu. Pour ces designs, notre ligne directrice a été de créer une utilité à nos bâtiments en se demandant quels étaient les besoins de la civilisation maintenant éteinte du nexus. Le design graphique des bâtiments a donc été réfléchi autour de ces axes-ci en ajoutant une architecture qui permet de renforcer la narration de notre jeu. Le joueur peut d'ailleurs voir que les bâtiments "narratifs" réagissent à son approche en regagnant un peu de leur énergie passé. Cela se manifeste par des lumières qui s'allument lorsque le nexus est proche. Ce choix a été fait pour renforcer le rôle premier du nexus qui doit, à lui tout seul, réactiver sa civilisation éteinte.



Intentions graphiques bâtiment :

Nos références pour les bâtiments ont majoritairement été des concepts arts à dominante Sci-fi pour coller au maximum à la direction graphique que nous voulions pour tout ce qui englobe la civilisation du nexus.



Direction sonore bâtiment

Intentions sonores bâtiment :

Dans Reborn, plusieurs bâtiments peuvent influencer le gameplay du joueur. Nous avons donc fait en sorte que le design sonore de ces bâtiments puisse coller au mieux à leurs fonctions lorsqu'ils sont activés. La plupart des bâtiments possédant deux formes, une activée et une désactivée, il fallait pour le joueur, en plus du changement graphique, qu'il puissent identifier cela par le son. Dans ce but et en lien avec notre ligne directrice concernant la spatialisation des sons, nous avons créé des zones 3D qui émettent des sons en fonction des bâtiments. Ce design permet de changer l'ambiance environnementale lorsque le joueur approche d'un bâtiment particulier ou lorsqu'il l'active pour accompagner les feedbacks visuels.

Références sonores bâtiment :

En lien avec notre narration, nos bâtiments tout comme l'entièreté de la civilisation du nexus se trouvent en ruine. Il était donc important pour nous de percevoir à travers le son, que le joueur réactive une civilisation éteinte. A cet effet, nous nous sommes inspirés des temples et mécanismes présents dans Shadow of the tomb raider qui caractérisent bien notre intention d'anciens bâtiments qui s'activent de nouveau après de nombreuses années.

Ajouté à cette intention sonore, nous avons également travaillé sur un autre layer qui prenait en compte toute la partie sonore qui attrait au mystique. Pour cela, une de nos références principales était Spirit of the north qui comme Shadow of the tomb raider, met en scène des anciens monuments et bâtiments qui se réactivent au passage du joueur. Seulement, dans Spirit of the north une dimension mystique et spirituelle est bien plus présente tout comme dans Reborn. Nous nous sommes donc grandement inspirés de leur façon de procéder notamment lors des réveils de ces bâtiments éteints.



REBORN

- Annexe -



Time line du projet

Noyau système et mécaniques de bases
HUD Simple
Mécanique de selection
Ennemis comportements basiques
Système de ressources

2 Décembre 2021



IA et comportement unités plus poussés
Feedbacks simples
Système alimentation et consommation
d'énergie
Premier LD
Intégration de certains décors et
modèles 3D

15 Janvier 2020



Comportement de groupe et raffinage de l'IA des agents
Intégration de la tour
Changement du système d'énergie et sa consommation
Cycle jour-nuit et attaque nocturne
Intégration sonore et visuelle plus poussé
Intégration du post-process
Feedbacks Essentiels
Level Design retravaillé
Intégration écran tutorial
Refonte HUD

2 Février 2022



Intégration d'un brouillard de guerre
Intégration du raton de vision des
unités et bâtiments
Intégration de la tour de vision

3 Mars 2022



Amélioration brouillard de guerre
Intégration crystal rouge
Amélioration tour de vision
Intégration des interrupteurs
Création ébauche LD niveau facile

17 Mars 2022

Communication entre les unités
Changement graphique brouillard de guerre

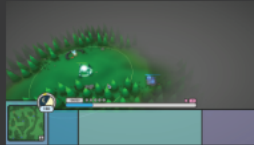
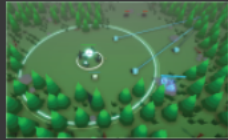
24 Mars 2022

Changement champ de vision unité
Intégration placeholder de map
Changement esthétique terrain et décors

31 Mars 2022

Intégration 4 unités ennemies et alliées
Intégration nouveau système de niveau
Intégration nouveau système d'attaque de nuit
Intégration modèle 3D
Finalisation LD premier niveau

10 Avril 2022

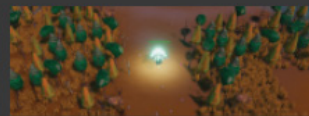


Intégration modèle 3D unités
Shader attaque serpent
Toggle attaque auto compétences
Rework lighting and décors
Changement Sound Design
Création de deux biomes supplémentaires

07 Mai 2022

Changement Light et post process
VFX Attaque ennemis et alliés
Raffinage scène LD
Intégration nouveau HUD
Intégration icônes minimap
Intégration décors

20 mai 2022

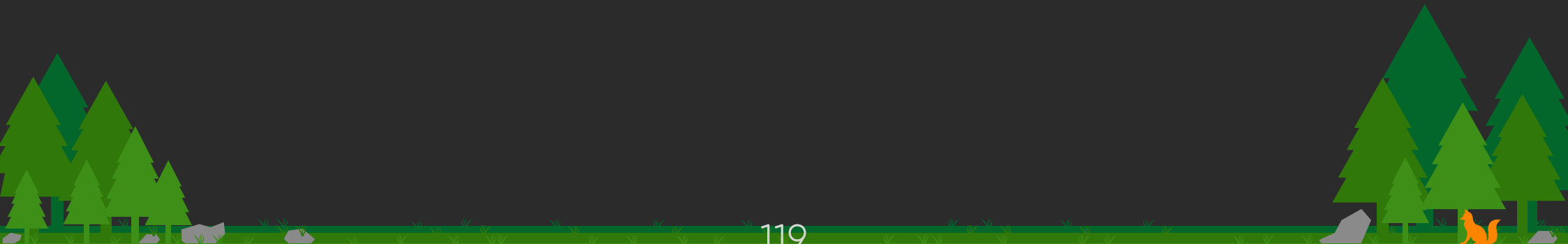


Optimisation générale

Pour optimiser la production nous avons réparti l'équipe en rôles selon les spécialités de chacun.

- Lucas : Direction Artistique, Modélisation 3D et Sound Design.
- Guillaume : Lead Programmeur Gameplay, VFX et intégration.
- Corentin : Chef de Projet, Programmeur Gameplay et intégration, Game Design.
- Damien : UX/UI Designer, Game Design, Level Designer.
- Gonzalo : Game Design, Level Designer.

Les rôles et les tâches attribués à chaque personne ont pu légèrement varier mais la majorité du temps, c'est en fonction de ses rôles que les missions de chacun ont été réparties.



Optimisation réunion

Pour organiser la production et faire en sorte que le projet soit en constant avancement tout en gardant tout le monde à la page sur les modifications apportées au projet, nous faisons des réunions hebdomadaires avec toute l'équipe qui se déroulent comme ceci :

- tour de table où tout le monde présente le travail qu'il a effectué en le comparant avec ses objectifs de la semaine.
- discussion sur l'ordre du jour préparé en amont par le chef de projet.
- discussion sur des points libres soulevés par le groupe (suggestions de changements).
- répartition des tâches de la semaine en fonction des rôles et des disponibilités de chacun et rédaction du planning.
- rédiger un récapitulatif de réunion en bilan.

Pendant la semaine, chacun poste son travail au fur et à mesure sur le discord et l'équipe peut ensuite faire des retours, qui font office de petites validations avant la réunion, qui elle viendra trancher.

Groupe 3GDA	04/03 - 11/03	HotWheels	Lucas	Damien	Guillaume	Gonzalo	Corentin
SP7	Intégrations	Amélioration de la diversité	2/16h	2/16h	2/16h	2/16h	2/16h
	<input type="checkbox"/>	Ajout des unités supplémentaires	/	/	2.75	/	/
	<input type="checkbox"/>	- Script permettant compétence spécial (ce doit d'être permissif)	/	/	1	/	/
	<input type="checkbox"/>	- Intégrer compétence Serpent	/	/	0.5	/	/
	<input type="checkbox"/>	- Intégrer compétence Scout	/	/	0.5	/	/
	<input type="checkbox"/>	- Intégrer compétence Tank	/	/	0.5	/	/
	<input type="checkbox"/>	- Intégrer Serpent	/	/	0	/	/
	<input type="checkbox"/>	- Intégrer Scout	/	/	0	/	/
	<input type="checkbox"/>	- Intégrer Tank	/	/	0.25	/	/
	<input type="checkbox"/>	- Moodboard Serpent	0	/	/	/	/
	<input type="checkbox"/>	- Moodboard Scout	0	/	/	/	/
	<input type="checkbox"/>	- Moodboard Tank	0	/	/	/	/
	<input type="checkbox"/>	- Modèle 3D Serpent	0	/	/	/	/
	<input type="checkbox"/>	- Modèle 3D Scout	0	/	/	/	/
	<input type="checkbox"/>	- Modèle 3D Tank	0.5	/	/	/	/
	<input type="checkbox"/>	- Animation Serpent	0	/	/	/	/
	<input type="checkbox"/>	- Animation Tank	0	/	/	/	/
	<input type="checkbox"/>	- Animation Scout	0	/	/	/	/
	<input type="checkbox"/>	Création minimap	/	0.25	1	0.25	/
	<input type="checkbox"/>	- Patch des problèmes de fps	/	/	1	/	/
	<input type="checkbox"/>	- "OnePage" comportement minimap avec mockup (écran en jeu) (illustrator)	/	0.25	/	0.25	/
	<input type="checkbox"/>	Finir balise / porte	1.25	/	0.5	/	0.5
	<input type="checkbox"/>	- Créer modèle 3D porte	0.5	/	/	/	/
	<input type="checkbox"/>	- Colorisation Balise	0.5	/	/	/	/
	<input type="checkbox"/>	- Animation Porte	0.25	/	/	/	/
	<input type="checkbox"/>	- Intégration sous Unity	/	/	/	/	0.5
	<input type="checkbox"/>	- Shader porte	/	/	0.5	/	/
	<input type="checkbox"/>	Finir spawner	0.75	/	/	/	0.25
	<input type="checkbox"/>	- Créer modèle 3D spawner	0.5	/	/	/	/
	<input type="checkbox"/>	- Animation Spawner	0.25	/	/	/	/
	<input type="checkbox"/>	- Colorisation Spawner	0.25	/	/	/	/
	<input type="checkbox"/>	- Intégration sous unity	/	/	/	/	0.25
	<input type="checkbox"/>	Faire attaque de nuit	/	/	/	/	1
	<input type="checkbox"/>	- Refaire le script d'attaque de nuit	/	/	/	/	1
	<input type="checkbox"/>	Créer système d'amélioration	1	/	/	/	1.25
	<input type="checkbox"/>	- OnePage système d'amélioration	/	/	/	/	0.25
	<input type="checkbox"/>	- Arbre d'amélioration renard	/	/	/	/	0.25
	<input type="checkbox"/>	- Arbre d'amélioration serpent	/	/	/	/	0
	<input type="checkbox"/>	- Arbre d'amélioration tank	/	/	/	/	0
	<input type="checkbox"/>	- Arbre d'amélioration scout	/	/	/	/	0
	<input type="checkbox"/>	- MockUp UI Amélioration	1	/	/	/	0.5
	<input type="checkbox"/>	- Créer bâtiment d'amélioration placeholder	/	/	/	/	0.25
	<input type="checkbox"/>	- Intégration nouvelles ressources energie orange	/	/	/	/	0
	<input type="checkbox"/>	- Intégration système d'amélioration	/	/	/	/	0
	<input type="checkbox"/>	- Lien entre système d'amélioration et variables unités	/	/	/	/	0
	<input type="checkbox"/>	Level Design	/	3.5	/	3.5	/
	<input type="checkbox"/>	- Optimiser le layout pour qu'il prenne la forme d'un carré	/	0.25	/	0.25	/
	<input type="checkbox"/>	- Intégrer les spawners sur les cartes (et bâtiments si pas déjà mis)	/	0.25	/	0.25	/
	<input type="checkbox"/>	- Continuer l'intégration des cartes sous unity (prendre en compte les suggestions narra de Lucas)	/	1.5	/	1.5	/
	<input type="checkbox"/>	- Tester à fond et repérer les bugs ou pb de design	/	1.5	/	1.5	/
	<input type="checkbox"/>	Réalisation planning et récapitulatif	/	/	/	/	0.5
		Total	2	3.75	4.25	3.75	3.5
		▼ BONNE CHANCE À TOUS ! ▼					

urgent
planifié
optionnel

Protocole de test

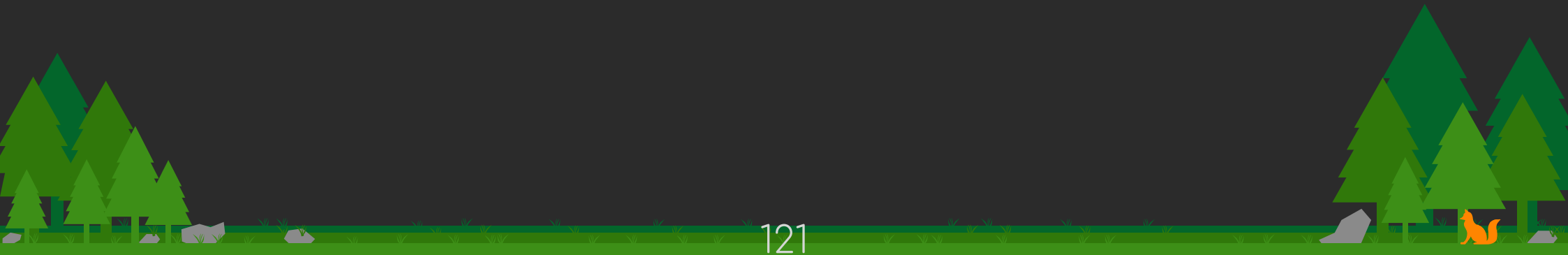
Afin de récolter des données dans le but de corriger et d'améliorer l'expérience utilisateur, un protocole de test a été pensé.

Dans un premier temps nous laissons le joueur jouer, et nous observons sa façon d'interagir avec le jeu sans en aucun cas prendre contact avec le joueur.

Ensuite nous posons directement une liste de questions généralistes au joueur en essayant de ne pas l'influencer. Puis enfin nous demandons aux joueurs de remplir un formulaire plus précis, en anonyme est à l'abri des regards afin de récolter les informations les plus pures et moins influencées possible.

Une fois les résultats reçus, nous mettons en commun tous les résultats pour essayer d'identifier les endroits où on peut observer des pics de perturbation pour l'utilisateur.

Une fois le/les problème/s identifié/s, nous effectuons des recherches dans le but de corriger les problèmes en question puis mettons en place de nouveaux tests, afin de pouvoir observer si le problème est résolu ou non.



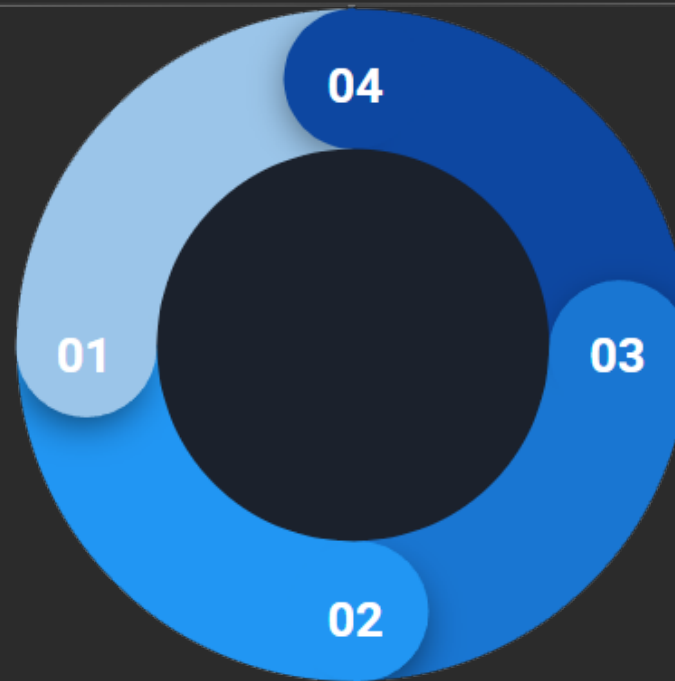
Protocole de Test

Observation

Le joueur joue sans interruption pendant qu'un membre de l'équipe observe et prend des notes.

Discussion Oral

L'examineur échange à l'oral sur les points qui ont pu poser problème à l'aide d'une liste de questions et en essayant de ne pas influencer le testeur.



Analyse

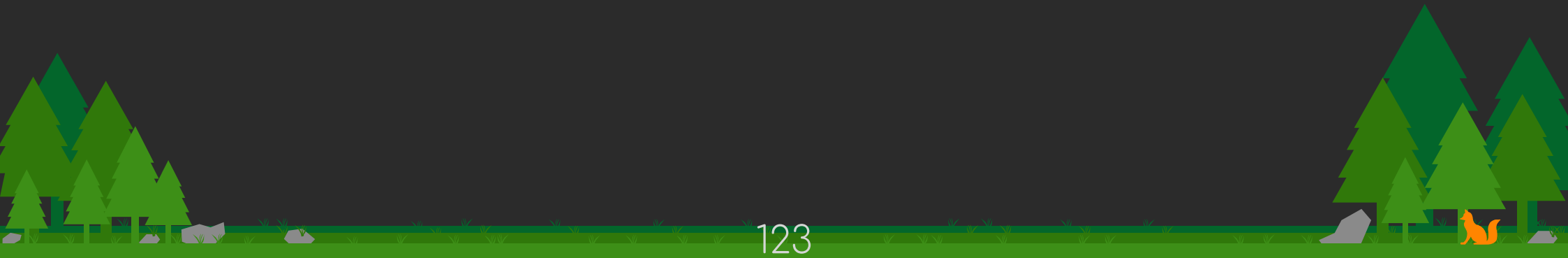
Analyse des données reçues.

Questionnaire

Les testeurs reçoivent un questionnaire anonyme qu'ils doivent remplir en toute intimité.

REBORN

- Sound design document -



Intentions et recherche sonore :

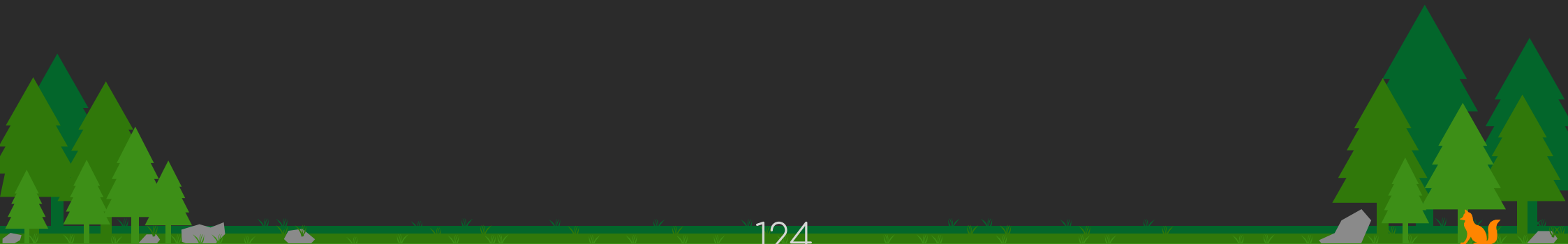
Nos intentions en design sonore sont de fournir au joueur un univers qui pourra lui faire ressentir différentes émotions lors de son aventure. Notre jeu étant constitué de différents biomes, nous avons pour objectif en lien avec la direction graphique, de fournir au joueur des sons environnementaux uniques à chacune de ses zones. Chaque zone à une identité qui lui est propre et qui permettra au joueur de comprendre dans quel environnement il se trouve.

Cependant, ces sons environnementaux ne devront pas pour autant surcharger le jeu qui devra rester fluide. Le genre de notre jeu demande également de nombreux events et feedbacks cohérents aux actions du joueur. Les sons environnementaux devront donc être vite identifiables sans, pour autant, prendre toute l'importance du son que nous devons répartir entre les différentes feedbacks.

Quant à l'ensemble des feedbacks du jeu, nous avons fait le choix de privilégier des sons 3D pour permettre une plus grande profondeur au gameplay et permettre au joueur d'avoir également un impact sur les sons qu'il peut entendre selon ses actions.

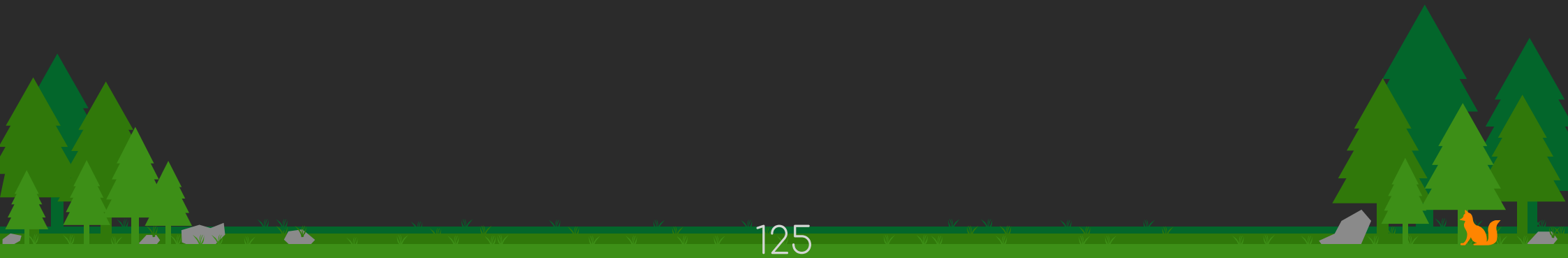
Nos intentions concernant la création sonore sont de former des sons en lien avec les thèmes qui émergent de notre jeu. Ainsi, la sphère environnementale est construite autour de références réelles avec des sons reprenant des environnements comme des forêts ou des savanes. Toutefois, une autre sphère viendra se superposer à la première en apportant un layer beaucoup plus fantastique qui sera centré sur l'énergie et le mysticisme, ceux-ci étant un des principaux thèmes de notre projet.

Nos recherches sonores se sont donc articulées autour des sphères décrites plus haut pour permettre de créer des sons et transitions environnementales qui s'intégreront parfaitement avec le reste des sons. Avec notre deuxième thématique, il sera important de gérer les sons en lien avec l'énergie qui est un élément capital du gameplay.



Nous avons, au cours de l'année, décidé de concentrer nos analyses sur le fonctionnement des transitions et les différentes manières auxquelles nous pourrions nous référer pour créer les sons environnementaux les plus profitables à notre projet. Pour cela, ils nous a paru cohérent d'analyser des jeux dans lesquels se trouvaient plusieurs catégories de transitions environnementales.

Nous avons tout d'abord étudié les différents layer du jeu genshin impact, un monde ouvert comportant également de nombreux biomes différents avant de nous pencher sur la carte élémentaire de league of legends qui subit des changements environnementaux et sonores au cours d'une partie.



Analyse sonore :

Genshin Impact :

Analyse des sons environnementaux dans les différents lieux du jeu et à différentes échelles.

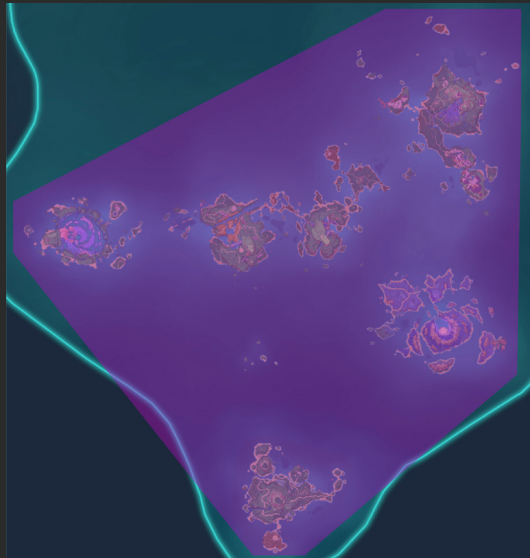
Genshin Impact est un monde ouvert où le joueur peut aller librement où bon lui semble. Le son joue alors un rôle important pour ajouter de la diversification à ce que entend le joueur pour renforcer son immersion.

1er Layer : Environnement musicale

Ce premier layer principalement utilisé pour identifier le thème de chaque region du jeu.

Genshin Impact utilisant des cultures actuelle comme référence chaque zone de couleur représente une culture spécifique.

Pour cette analyse nous allons nous intéresser à la zone orange et voir en nous rapprochant du point de vue du joueur, comment sont composées les différentes bandes sons du jeu.

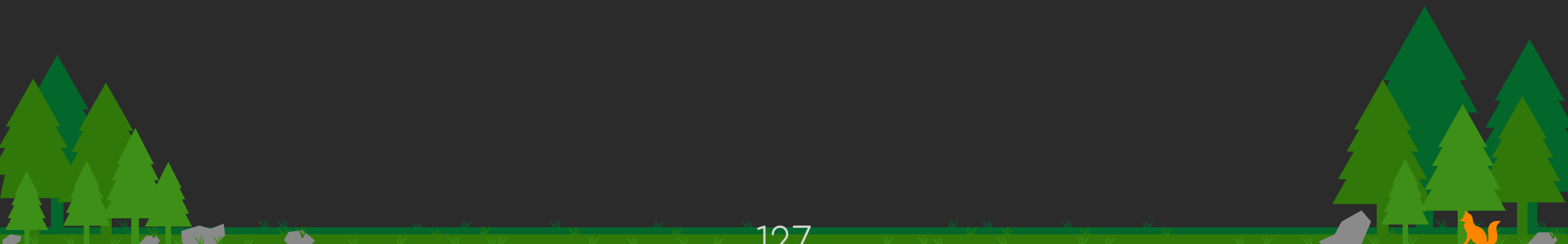




2ème Loyer : Musique environnementale

Le deuxième layer que nous pouvons identifier correspond à la bande musicale du jeu. En nous focalisant sur une seule région du jeu on voit que chaque zones ainsi colorisées sur l'image à coté représente une zone du jeu qui contient pour elle seule plusieurs musique en fonction des situation dans lesquelles le joueur peut se retrouver.

Les transitions de ses musiques se font rapidement, souvent par des event commun à tout le jeu pour symboliser au joueur qu'il passe d'une phase de gameplay à une autre. Une fois cette phase passée, le son retourne à sont état d'origine et la musique propre à la zone dans laquelle le joueur se trouve recommence à se jouer.



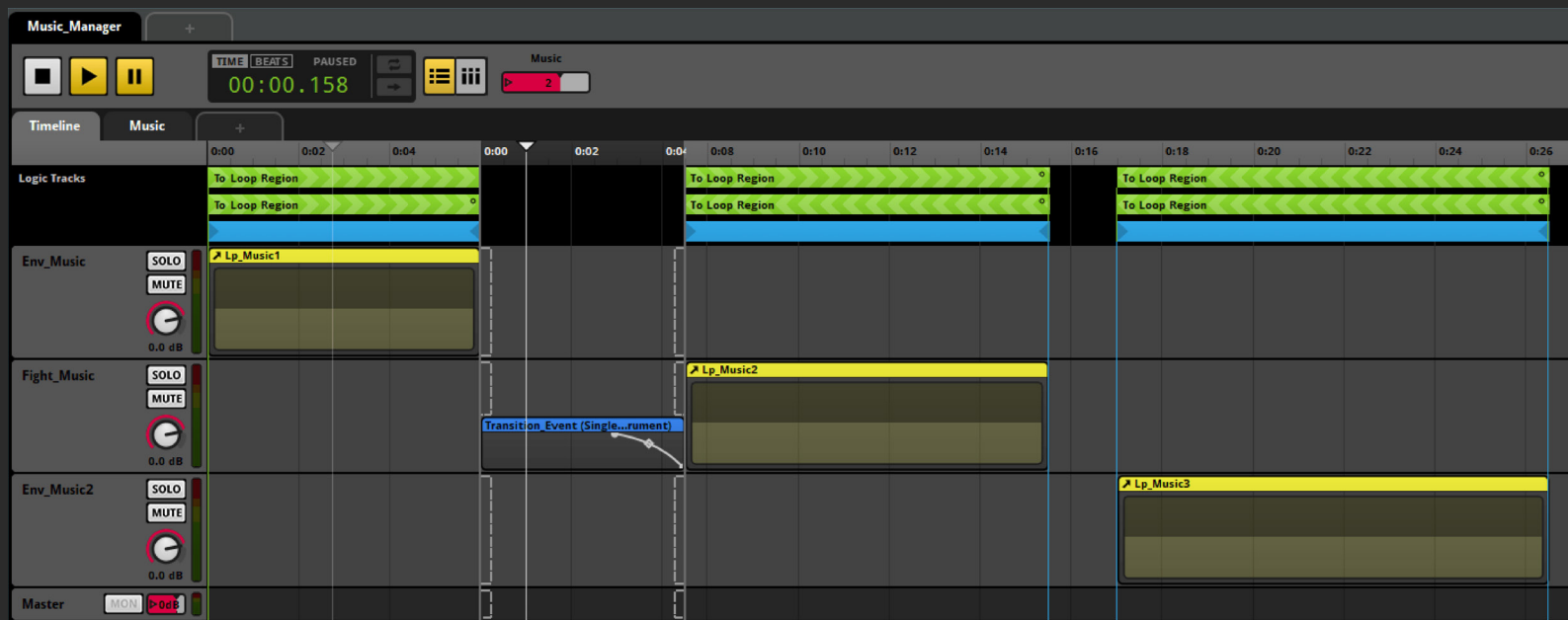
Réprésentation sur Fmod d'un event qui reprend les caractéristiques du 2ème layer de genshin impact

On retrouve 3 layers qui symbolisent chacun une musique environnementale.

Chacune de ses musiques se joue en boucle et est soumise à un paramètre (nommé «Music») qui permet lorsqu'il est changé de faire basculer l'audio sur une autre bande son.

Pour l'exemple illustré par le screen-shot, le son subit acutellement un transition par un Transition_event.

Il passe de la boucle «Lp_Music1» à «Lp_Music2» par une transition de 4 seconde dans laquelle se trouve un event spécifique. Cependant on peut également imaginer des transitions avec des fade-in et fade-out que l'on place directement sur les boucles entrantes et sortantes selon le besoin





3ème Layer : Musique environnementale

Le troisième layer que nous pouvons identifier se concentre sur les loop environnementale qui sont jouées dans des zones respective comme désigné par les couleurs.

Chaque zones possèdent ses propres loop environnementale construite avec différent layer qui lui sont propre. On peut trouver à l'écoute de ces loops une piste principale qui va caractériser l'ambiance générale (Ville avec des bruits de populations, montagne avec des vents très sifflants et fort, forêt avec le son des feuilles d'arbre qui bougent au vent,...).

Cependant on peut également entendre des events plus précis à l'interieur qui semble se déclencher indépendamment de la loop principale.

De cette façon la loop donne au joueur l'impression de ne pas se répéter grâce à ses events qui viennent la modifier en permanence.

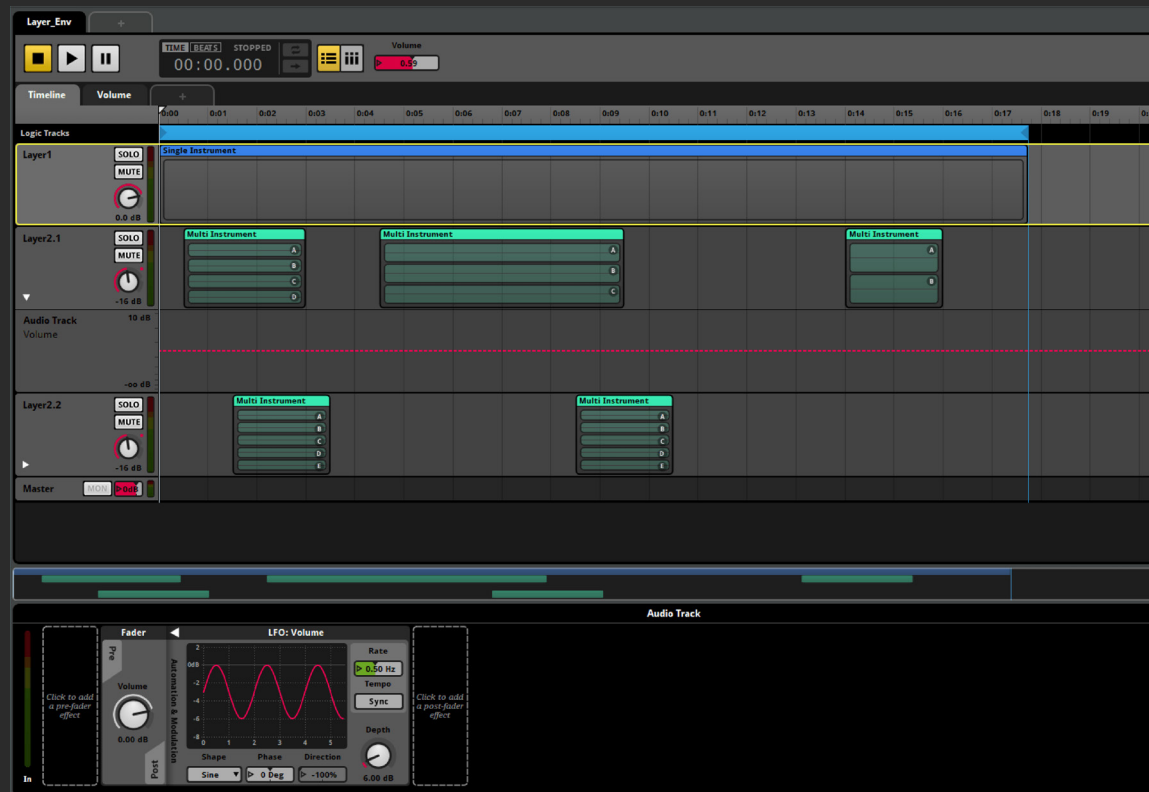


Représentation sur Fmod d'un event qui reprend les caractéristiques du 3ème layer de genshin impact

On y retrouve 3 audio track.

La première joue le rôle de loop principale et permet de caractériser l'environnement dans lequel le joueur évolue. Elle est soumise à un UFO qui fait varier ses decibels en suivant la courbe qu'on lui donne.

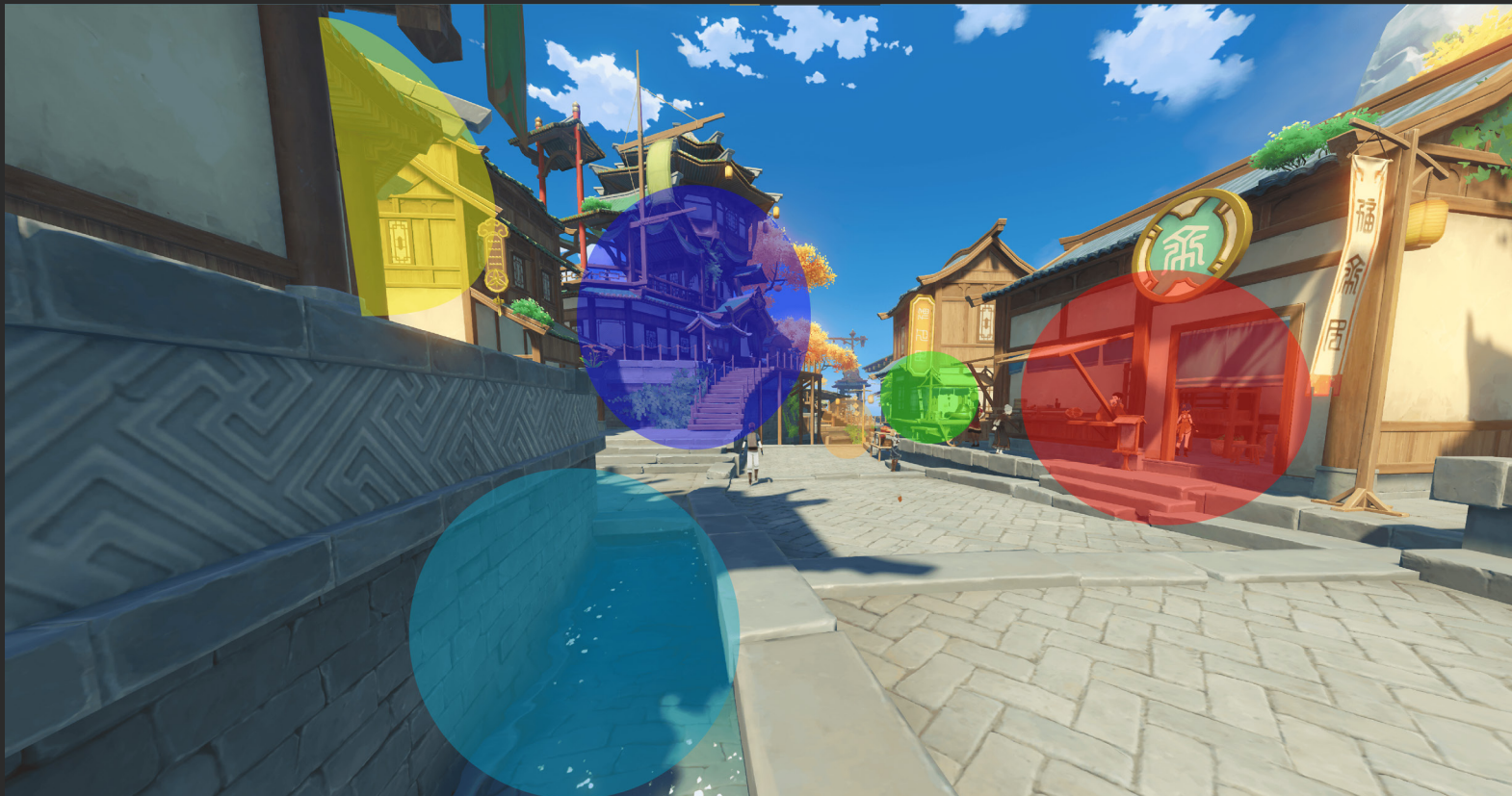
Enfin on peut ajouter une ou deux autres pistes composées de multi instruments pour permettre de varier des éléments pour donner plus de profondeur à l'ensemble de la loop. On peut également ajouter un paramètre sur les pistes contenant des multi instruments pour faire varier leur intensités.



4ème Loyer : Event 3D

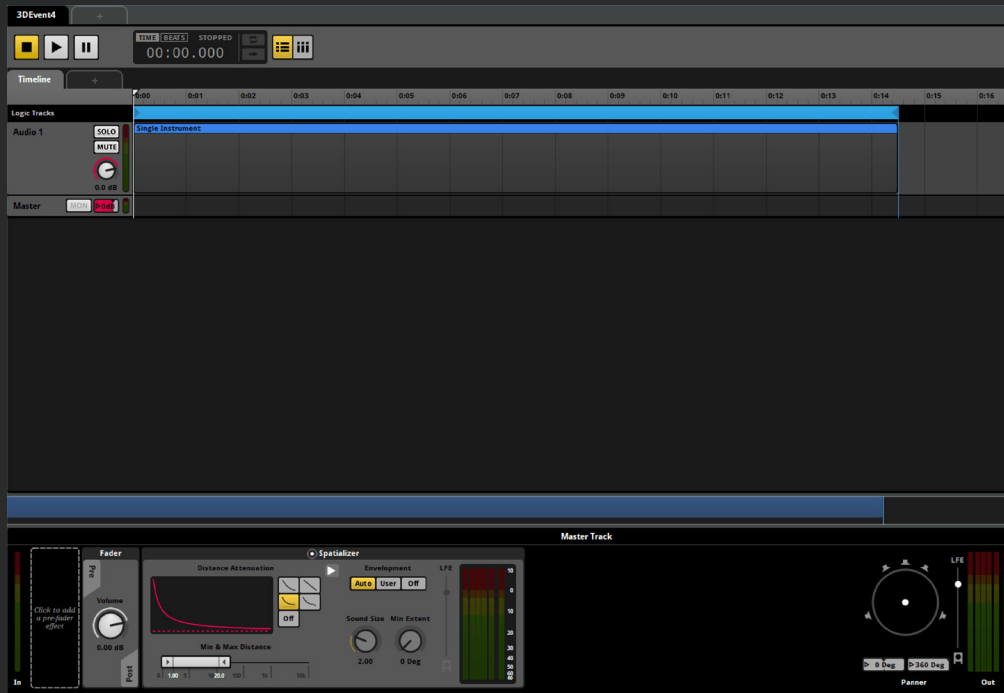
Enfin, au plus proche de joueur se trouve un ensemble d'event 3D qui ajoute à l'environnement une dernière couche de réalisme.

Comme illustré dans l'image, plusieurs endroit du jeu emettent leurs propre son à différentes échelles et intensité. Chacun de ses events reprend les comportements et sons qui serait attendu (Conversation pour des personnes en terrasse, bruit de cuisine pour le restaurant, eau qui coule pour les gouttières,...).



Représentation sur Fmod d'un event qui reprend les caractéristiques du 4ème layer de genshin impact

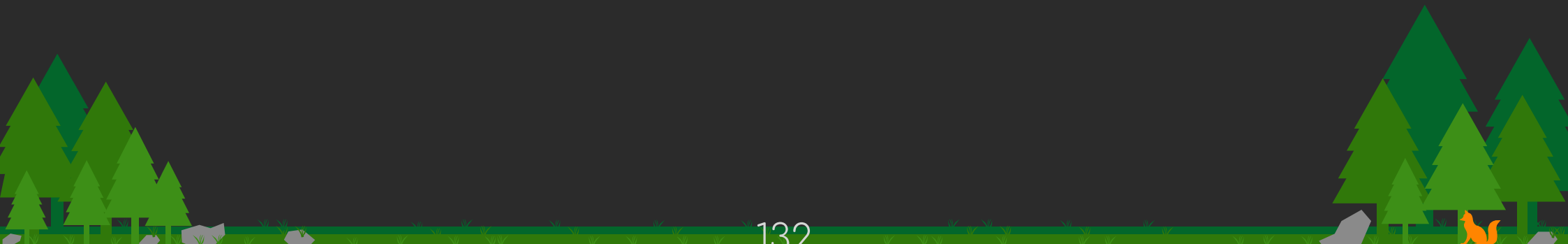
Cet event se compose simplement d'une loop qui contient un single instrument qui est spécialisé pour donner plus ou moins d'intensité en fonction du placement du joueur.



Suite à cette analyse et par rapport à l'axe de la direction sonore que nous voudrions prendre, il serait bénéfique pour la narration sonore de prendre exemple sur la formation du son dans genshin impact.

Les transitions sonores n'utilisent que peu d'event transitif pour changer de loop et passe majoritairement par de plus long fade-in et fade-out pour faire évoluer l'ambiance sonore.

De plus, l'utilisation d'un layer d'event 3D unique à chaque endroit du jeu, permet de bien renforcer l'immersion et l'identité d'une zone.



League of Legends :

But de cette recherche : Comprendre la formation sonore des transitions.

Analyse principale effectuée dans différents endroits de la jungle

Carte de base :

La carte de base possède un ou plusieurs events loop qui vont se rejouer sans coupure ainsi que des multi instruments qui se déclenchent aléatoirement dans leurs zones lorsque le joueur s'y trouve.



Son et loop de la rivière :

- Vent évoluant avec un pitch différent et un volume progressif en fonction de l'endroit où se trouve le joueur. Le son augmente particulièrement lorsque le joueur avance vers le top.
- Son de rivière continue qui s'entend dans la zone du sentinelle bleue et qui s'intensifie lorsque le joueur avance dans la rivière.
- Son de cascade lorsque le joueur se déplace vers la botlane.

Son et loop du sentinelle bleue :

- Vent qui change en volume et en intensité constamment.
- Event de son de grillon, event de tonnerre qui arrive à intervalle régulier en changeant d'ordre. Pluie contre le sol avec une reverb qui change parfois pour être plus présente ainsi que des events de bulles d'eau ou de boue pour symboliser le côté marais.
- Events d'insectes volants avec différentes tonalités et des grenouilles pour renforcer la narration sonore du marais.

Son et loop du roncier rouge :

- Loop de vent plus léger, atténuée par une loop qui reprend des bruits environnementaux de forêt (feuille qui bouge, bois qui craque, bruit d'animaux lointains).
- Event d'un hiboux et différents cris d'oiseaux en multi instruments. Event de hyène, d'animaux qui hurlent et aboient pour certains avec une reverb mais un niveau sonore assez bas pour donner un effet "lointain". On entend aussi quelques insectes qui bourdonnent et le chant de grillons.

Son et loop de la carte lors de sa transformations avec l'élément de feu :

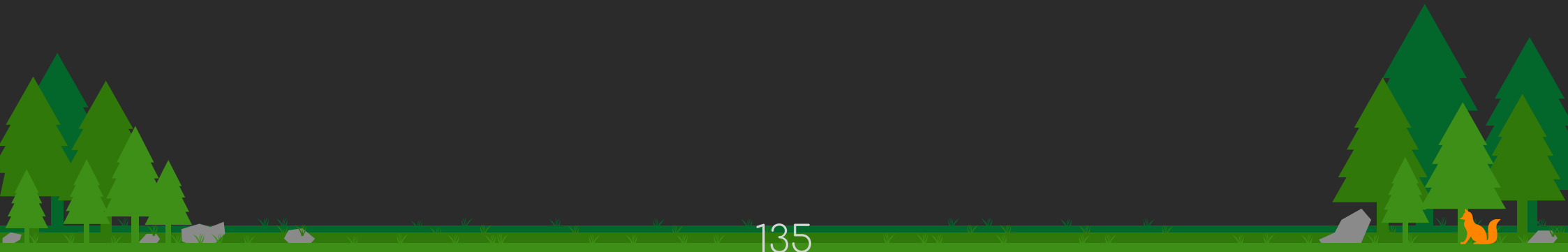
- Timeline qui comprend plusieurs événements.
 - Son d'une forte intensité et qui couvre tous les autres pendant un court instant. Son semblable à un tremblement de terre. Sons de craquement et fissure ainsi qu'un début d'incendie avec un son de feu puissant. Cette partie ne dure que quelques secondes avant de laisser place à une nouvelle loop qui prend le dessus sur l'ensemble des autres de la carte.
 - Ajout d'un event de flamme constant qui loop (fonctionnement qui reprend les loops de vent utilisé précédemment)
 - Event de feuille et bois qui brûle et craque à plusieurs reprises avec un event de son éteint par de l'eau.
 - Son de bulles plus grave, semblable à des bulles de boue ou de lave.
 - Les bruits d'animaux terrestres se font plus rares mais il y a toujours autant d'oiseaux



Intentions :

Suite à cette recherche sur les transitions entre les sons environnementaux, on peut remarquer que les sons environnementaux sont constitués d'une loop principale qui sert à définir le contexte de ce à quoi elle est assignée. On ajoute ensuite plusieurs sons sous forme de multi instruments à différents endroits de la loop principale pour casser la répétition. D'autres sons, plus importants, sont parfois ajoutés mais ils n'apparaissent pas à des moments réguliers.

Un point important de la transition entre ces deux loops est d'ajouter une coupure en lien avec la transition souhaitée pour noyer le son de la première loop avant de le faire diminuer et laisser place à la nouvelle loop. Cependant, cette méthode ne sert seulement lorsqu'un événement in-game sert de déclencheur. Pour des transitions entre deux ambiances sonores sans événement in-game remarquable, il faudra utiliser des transitions plus longues et douces.



Hiérarchie Fmod :

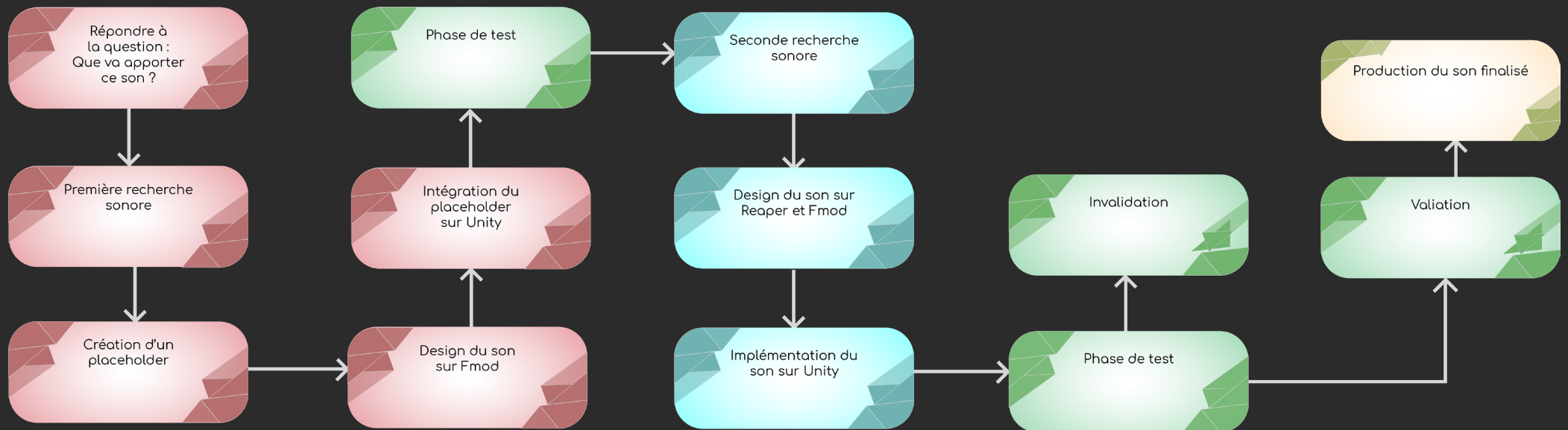
La construction d'une hiérarchie complète à été primordiale pour la gestion de la création et de l'implémentation du son par le designer sonore et le reste de l'équipe.

Liasan Timeline	2D/3D	Place Holders	Intégration	Polish	Familles	Sous famille	Event_Name	Description de la mécanique	Description de l'event	Description du son	Paramètres/Variables du son	Bugs dans les nouvelles map
K / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unit_Aly	Unit_Fox	Unit_Fox_Select	Se déclenche lorsque l'unité est sélectionnée	One Shot	Accrolement élastique		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Fox	Unit_Fox_Attack	Se déclenche lorsque l'unité attaque	One Shot	Brut élastique émettant à un glissement métallique		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Spider	Unit_Spider_Select	Se déclenche lorsque l'unité est sélectionnée	One Shot	Brut de croquer émettant qui se froisse avec de la réverbération		<input checked="" type="checkbox"/>
K / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Spider_Attack	Se déclenche lorsque l'unité attaque	One Shot	Brut de croquer et émission de venin avec peu de "fry" et plus de "wet"		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Snake	Unit_Snake_Select	Se déclenche lorsque l'unité est sélectionnée	One Shot	Très d'un serpent		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Snake	Unit_Snake_Attack	Se déclenche lorsque l'unité utilise son attaque spéciale	One Shot	Sort d'un couleuvre de venin d'un serpent avec reverb et plus d'augmenter		<input checked="" type="checkbox"/>
J / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Snake_Attack	Se déclenche lorsque l'unité attaque	One Shot	Mouvement d'un serpent avec impact reverb + pitch		<input checked="" type="checkbox"/>
K / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Globe	Unit_Globe_Select	Se déclenche lorsque l'unité spawn	One Shot	Son métallique qui résonne à un bruit d'un tube métallique très pointu un bruit sans résonance mais avec une reverb		<input checked="" type="checkbox"/>
K / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Globe_Death	Se déclenche lorsque l'unité meurt	One Shot	Son d'un vent égaré qui fade out. On entend une légère oscillation. Présence de basse.		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Globe_Death	Se déclenche lorsque l'unité meurt	One Shot	Niveau de musique résonante avec une augmentation du "fry"		<input checked="" type="checkbox"/>
K / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unit_Enemy	Unit_Fox	Unit_Fox_Select	Se déclenche lorsque l'unité est sélectionnée	Multi Instruments	One d'un reverb		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Fox_Attack	Se déclenche lorsque l'unité attaque	Multi Instruments	Mix avec angle qui génère un laps de temps		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Fox_Death	Se déclenche lorsque l'unité meurt	Multi Instruments	Accrolement soufflant d'un serpent		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Basilisk	Unit_Basilisk_Select	Se déclenche lorsque l'unité est sélectionnée	One Shot	One émettant égaré		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Basilisk_Attack	Se déclenche lorsque l'unité attaque	One Shot	Impact d'énergie avec reverb		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Basilisk_Death	Se déclenche lorsque l'unité meurt	One Shot	One émettant égaré		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Spider	Unit_Spider_Select	Se déclenche lorsque l'unité est sélectionnée	Multi Instruments	Brut de croquer émettant dans de la chair ou un texture vibratoire		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Spider_Attack	Se déclenche lorsque l'unité attaque	Multi Instruments	Brut de croquer émettant dans de la chair ou un texture vibratoire		<input checked="" type="checkbox"/>
M / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Unit_Spider_Death	Se déclenche lorsque l'unité meurt	Multi Instruments	One égaré de souffrance		<input checked="" type="checkbox"/>
L / Layer4	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unit_Globe	Unit_Globe_Select	Se déclenche lors de l'activation des espionnes ennemies	One Shot	Son de corde de brume égaré par des effets de BD		<input checked="" type="checkbox"/>
E / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Building	Build_Nexus	Build_Nexus_Select	Se déclenche lorsque l'on sélectionne le Nexus	Loop	Son d'un générateur électrique qui oscille		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Nexus_LevelUp	Se déclenche lorsque le niveau d'énergie du Nexus passe un niveau supérieur ou diminue à un niveau inférieur	Event avec reverb pour jouer les différents niveaux	Son émettant ou descendant selon le paramètre. Son avec une large reverb	Son qui se joue en one shot avec un effet reverb. Instrument liés avec un paramètre qui va de 0 à 10. Les nombres sont entiers. Le paramètre correspond à l'augmentation ou la réduction du niveau du Nexus.	<input checked="" type="checkbox"/>
E / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Nexus_Death	Se déclenche lorsque le Nexus meurt	One Shot	Spring d'un avec longue réverbération. Placeholder qui sera changé par un dingue plus évocateur de la condition d'arrêt du jeu et de la mort du Nexus.	Niveau 9 : Gain au paramètre 9. Perte au paramètre 10. Niveau 4 : Gain au paramètre 10. Perte au paramètre 9. Niveau 3 : Gain au paramètre 9. Perte au paramètre 10. Niveau 2 : Gain au paramètre 10. Perte au paramètre 9. Niveau 1 : Gain au paramètre 9. Perte au paramètre 10.	<input checked="" type="checkbox"/>
F / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Nexus_Ton	Se déclenche lorsque le Nexus émet un tick d'énergie	One Shot	Brut d'une vague d'énergie qui se charge avant d'être transformée en onde de choc. On perçoit un coup de tonnerre.	Utilisation d'un AttackPad avec réverbération.	<input checked="" type="checkbox"/>
G / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Nexus_Chillout	Se déclenche lorsque le Nexus module la possibilité de bouger	One Shot	Son d'un générateur avec des bruits égarés et avec des fréquences pour le côté égaré. Ajout d'un son de générateur qui résonne en fond.		<input checked="" type="checkbox"/>
F / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Nexus_Chillout	Se déclenche lorsque le Nexus module la possibilité de bouger	One Shot	Son d'un générateur coupé brusquement avec un résonnement qui se finit. On perçoit des bruits de réverbération pour le côté égaré.		<input checked="" type="checkbox"/>
F / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Nexus_Hit	Se déclenche lorsque le Nexus est touché	One Shot	Son de choc sur une surface en verre. Brut résonnant et élastique au fil.		<input checked="" type="checkbox"/>
H / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Build_Turret	Build_Turret_Rise	Se déclenche quand la tour est activée	One Shot	Brut de fonctionnement de la tour (carré) sur les mouvements des pièces avec un choc qui augmente en fond.		<input checked="" type="checkbox"/>
H / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Turret_Fall	Se déclenche quand la tour de désactive	One Shot	Brut de fonctionnement de la tour (carré) sur les mouvements des pièces avec un choc qui diminue en fond.		<input checked="" type="checkbox"/>
H / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Turret_Shoot	Se déclenche quand la tour tire un projectile	One Shot	Brut d'un tir laser élastique.		<input checked="" type="checkbox"/>
H / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_Turret_Impact	Se déclenche à l'impact d'un projectile	One Shot	Brut d'un tambour de batterie rendu plus grave avec de la reverb.		<input checked="" type="checkbox"/>
H / Layer3	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Build_VisionTower	Build_VisionTower_Rise	Se déclenche lorsque la tour de vision est activée	One Shot	Drone d'énergie qui se dissipe en fade out + résonnement de la tour.		<input checked="" type="checkbox"/>
	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Build_VisionTower_Fall	Se déclenche lorsque la tour de vision est désactivée	One Shot	Drone d'énergie qui résonne d'énergie en fade out + résonnement de la tour.		<input checked="" type="checkbox"/>

Lien vers la hiérarchie Fmod : <https://docs.google.com/spreadsheets/d/1ofVqalWLLILJwE8RNz5Plqq6dYukCyZis6wiqnhaytU/edit#gid=0>

Schématisation de l'implémentation du son :

L'implémentation du son suis une pipe line précise allant de sa création à son intégration finale. Pour cela nous avons fait un schéma pour formaliser les étapes d'itérations.



Time line sonore :

Pour nous aider à la représentation temporelle de nos sons nous avons réalisé une timeline. Ce document schématise le déclenchement de nos sons sur une partie de jeu typique. Cela nous permet de nous rendre compte de la fréquence à laquelle se déclenche certains sons ainsi que l'intensité sonore qui se dégage de chaque phases de jeu.

