Level Design - Unreal

1 - Énoncé de l'exercice

Sujet: Créer un niveau avec l'éditeur Unreal. Le niveau doit être décomposé en 10 parties distinctes (qu'on appellera « stages » dans la suite de ce document). La difficulté doit être progressive, et se baser sur un changement limité de paramètres (deux ou trois). Aucun travail n'est à effectuer sur la partie graphique, c'est un exercice technique uniquement.

Objectifs:

- Entraînement au level design et à la notion de courbe de difficulté.
- Compréhension de l'outil Unreal Editor.

Outil: Unreal Editor 4 (fonctionnant avec l'Unreal Engine 3).

J'ai choisi cette version pour 2 raisons :

- C'est la version la plus communément utilisée dans l'industrie actuellement, l'Unreal Editor 3 devenant assez âgé et étant de moins en moins utilisé. Il est donc plus utile de s'entraîner au maniement de cette version.
- La présence de l'éditeur Kismet est particulièrement adaptée à l'exécution de ma vision pour le niveau.

2 - Présentation du rendu

Temps de conception et de production : 30 heures.

Principe : Le joueur doit tirer sur des lampes pour les allumer. Une lampe reste allumée pendant un temps limité, puis s'éteint. Il faut que toutes les lampes soient allumées pour que le téléporteur vers la partie suivante du niveau devienne accessible.

Les paramètres sur lesquels on va jouer sont :

- **Nombre de lampes** (une lampe est ajoutée à chaque nouvelle partie, pour un total de 10 dans la dernière) et donc longueur et complexité du parcours
- **Temps pendant lequel les lampes restent allumées**: Au début, le joueur a un certain droit à l'erreur. Vers la fin, il doit boucler le stage rapidement pour allumer les dernières lampes pendant que les premières sont encore allumées.

Ce principe n'est pas follement original, mais il permet une liberté et une créativité dans les niveaux intéressante.

On notera aussi que le joueur est invincible, et qu'il a les munitions infinies, afin que ces éléments n'entrent pas dans les paramètres de difficulté.

Augmentation de la difficulté et diversité du level design

L'augmentation en difficulté, sujet de base de l'exercice, est alliée à une certaine variation dans le level design : l'objectif est d'exploiter le core concept de manières différentes. On a ainsi, après un début simple servant de tutorial, des stages axés plate-forme, d'autres axées orientation, d'autres axées sur le tir...

L'augmentation en difficulté se trouve aussi au sein d'un même stage : ceux-ci comportant des principes différents, on fait un apprentissage rapide au joueur du principe du stage au début, avant de continuer le stage sur le même principe de plus en plus difficile. La progression de la difficulté se fait donc à deux niveaux : entre tous les stages, et individuellement à l'intérieur d'un même stage.

Exploiter le concept : Speedrun

Le niveau implique une notion de course contre la montre, puisqu'il faut allumer toutes les lampes le plus rapidement possible avant qu'elles s'éteignent. Pour exploiter ce concept, on a donc introduit la notion de speedrun. Le speedrun est une « discipline » consistant à terminer tout ou partie d'un jeu dans le temps le plus court possible. Les pratiquants vont généralement tenter d'exploiter les bugs et glitchs du jeu afin d'optimiser leur performance. Un certain nombre de petites astuces et de passages secrets dans cet esprit-là a donc été implémenté dans les niveaux (téléporteurs permettant de gagner du temps, etc), afin de permettre au joueur tentant d'optimiser son temps de les exploiter. On encourage ainsi d'autant plus une exploration détaillée de la map.

Ceci introduit aussi une certaine replay value : au fur et à mesure qu'il connaît mieux les maps, le joueur peut tenter de battre son propre record. Le niveau ne s'arrête donc pas quand on l'a terminé une première fois, mais incite le joueur à y revenir.

Éléments de navigation

Plusieurs éléments sont importants pour aider le joueur à la navigation dans le niveau.

Position du téléporteur: Le joueur a besoin de savoir dès le début du stage où va apparaître le téléporteur à la fin : cela lui donne un objectif clair et concret, et lui évite la frustration de devoir parcourir le stage à nouveau pour trouver comment en sortir alors qu'il a déjà allumé toutes les lampes et l'a donc, dans son esprit, déjà terminé. Pour ceci, un cercle de murs, bien visible, est installé autour de l'endroit où va apparaître le téléporteur. Quand les lampes sont allumées, ces murs s'élèvent, révélant le téléporteur.

Ordre d'allumage des lampes: Le joueur a besoin d'indices l'informant sur l'ordre dans lequel il doit allumer les lampes. Pour ceci, elles s'allument selon un ordre de couleurs précis : du violet au rouge, selon le spectre lumineux. C'est un ordre que tout le monde connaît et qui fonctionnera donc de manière universelle. Il est à noter que la dernière lampe, dans chaque niveau, est blanche.



De plus, un message de confirmation s'affiche pour chaque lampe allumée : « Lamp 1 OK », etc. Dans ce message, le numéro de la lampe est indiqué, ce qui donne clairement l'indication au joueur.

Confirmation de réussite: Le joueur a besoin d'être sûr, après un tir, que la lampe soit bien allumée. Comme certaines lampes sont loin de tout mur, leur lumière ne se reflète pas et leur état n'est donc pas immédiatement évident. En plus de l'allumage de la lampe, on a donc un second feedback : le message de confirmation déjà cité plus haut s'affiche : « Lamp 1 OK ».

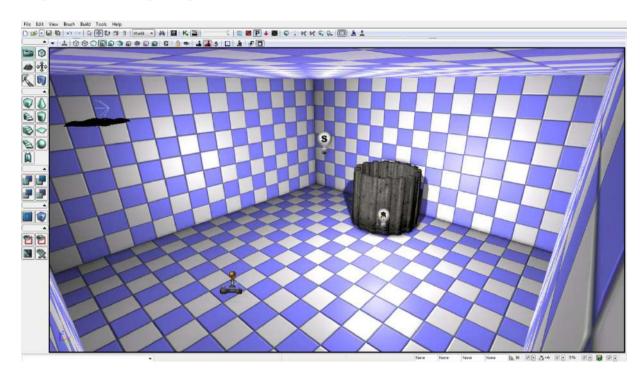
De plus, quand toutes les lampes d'un niveau sont allumées, et donc que le téléporteur apparaît, un message en informe le joueur : « STAGE [numero_du_stage] CLEAR !!! ».

Position des lampes: Enfin, les lampes sont très petites. Il faut néanmoins qu'elles soient suffisamment visibles. Pour ceci, elles émettent constamment une faible lumière, même quand on ne leur a pas tiré dessus. Le joueur peut voir cette lumière reflétée sur les murs de loin et voir ainsi d'un coup d'œil où se trouvent les lampes. De plus, cette lumière étant de la couleur de l'ordre de la lampe, ceci fournit d'autant plus tôt au joueur l'information quant à l'ordre dans lequel allumer les lampes.

Checkpoints: Il a déjà été expliqué que le joueur ne peut pas mourir. Mais il faut aussi faire attention à lui pardonner s'il commet des erreurs: en effet, s'il rate un saut en plein milieu du niveau, il ne va pas avoir envie de retourner au début pour rallumer toutes les lampes. Des téléporteurs ont donc été placés dans tous les endroits où le joueur peut potentiellement tomber, et le ramènent alors juste avant le saut où il a échoué. Il peut ainsi continuer immédiatement sans ressentir de frustration.

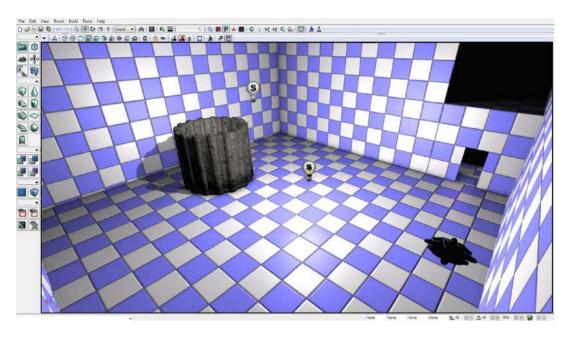
3 - Stage par stage : intention et construction

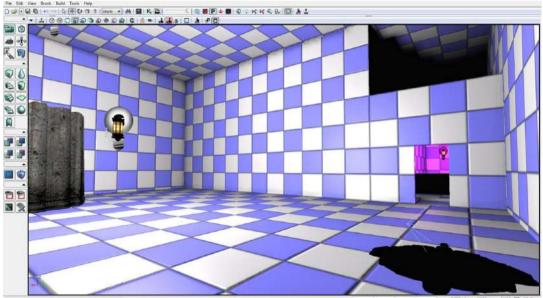
Stage 1 - Tutorial : principe de base



Stage très court servant d'introduction au principe de base. Un message de tutorial apparaît dès le début : "Stage 1 >>> The lamp is down on power - shoot it to light it again and make the teleporter appear". La lampe est immédiatement visible, le joueur n'a qu'à tirer dessus, et le mur s'élève, laissant apparaître le téléporteur.

Stage 2 - Tutorial: Ordre des lampes

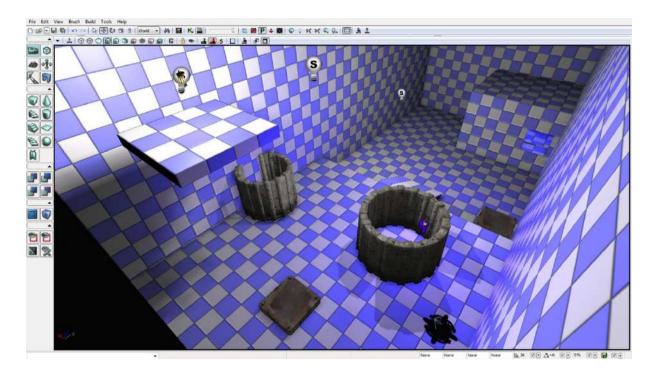




Ce stage introduit la notion d'allumage des lampes dans un ordre défini, l'augmentation progressive du nombre de lampes, ainsi que le fait que les lampes s'éteignent après un certain temps. A l'arrivée dans le stage, le message suivant s'affiche : « Stage 2 >>> Careful, lamps shut down after some time... and every lamp needs to be lit for the teleporter to appear. Each stage will add one more lamp than the previous one! ». Le joueur a donc une idée de ce qui va se passer. Pour illustrer celle-ci, une lampe est placée juste devant l'arrivée du stage : le joueur va reproduire le schéma appris au stage précédent et tirer dessus. Cependant, le message suivant s'affiche alors : « Lamp 2 OK - Light the lamps in the order of the light specter, from purple to red. The last lamp is always white! ». Comme cette lampe est blanche, il comprend que c'est censée être la dernière allumée. De plus, elle s'éteint au bout de 3 secondes, un temps insuffisant pour lui pour aller allumer l'autre lampe.

Il doit donc aller allumer l'autre lampe du stage, facilement accessible dans une pièce annexe à la pièce principale, qui va elle s'allumer en violet (indiquant donc que c'est la première), et s'éteindre au bout de beaucoup plus de temps, lui laissant le temps d'aller tirer à nouveau sur la lampe blanche pour l'allumer, et ainsi faire apparaître le téléporteur.

Stage 3 - Tutorial : Déclenchement de mécanismes



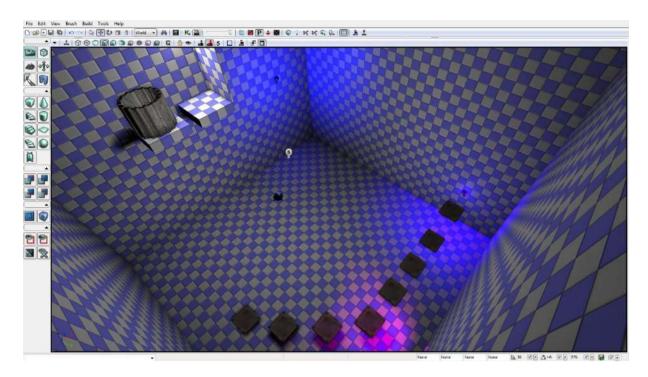
Ce dernier stage de tutorial met en scène le fait que tirer sur une lampe, outre l'allumer, peut aussi déclencher d'autres mécanismes. Le message d'introduction au stage est « Stage 3 >>> Shooting lamps can also trigger other events... ». Au début, la seule lampe accessible se trouve dans une pièce annexe, accessible via un saut facile sur une plate-forme mouvante (servant à introduire très doucement le challenge plate-forme qui va suivre).

Quand le joueur revient à la salle principale, il voit qu'un ensemble de murs circulaire, qui barrait précédemment l'accès à ce qui se trouvait à l'intérieur, est en train de se lever (notons qu'au cas où la lampe s'éteigne et donc que les murs se baissent avant que le joueur se soit éloigné, et qu'il se retrouve donc piégé à l'intérieur, une plate-forme dans cet espace va s'élever s'il monte dessus, lui permettant de ne pas être bloqué). Il révèle une deuxième lampe.

En tirant sur celle-ci, une plate-forme proche se met en marche : elle monte et descend vers une plate-forme élevée, donnant l'accès à la troisième et dernière lampe du niveau.

Le joueur a maintenant toutes les clés en main pour réussir, on va pouvoir commencer à mettre en place des niveaux un peu plus sérieux !

Stage 4 - Plate-forme facile



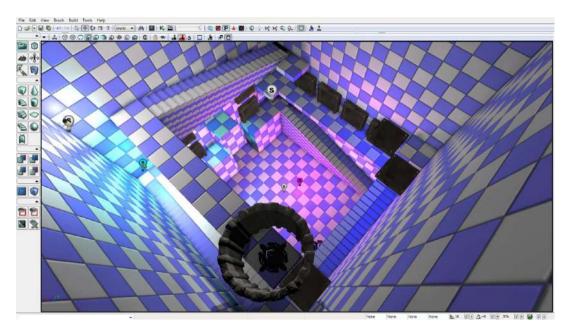
Le message d'entrée dans le niveau est « Stage 4 >>> Let's practice a bit of soft platform jumping ». Le ton est donné, on démarre la plate-forme en douceur. Une série de sauts permet au joueur d'accéder à la première lampe.

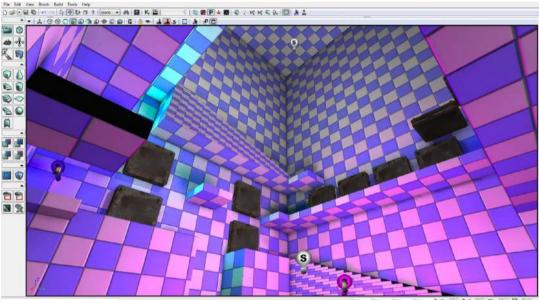
En tirant sur celle-ci, les plate-formes suivantes s'élèvent, permettant d'atteindre une plate-forme avec la seconde lampe.

Celle-ci, à nouveau en tirant dessus, active la plate-forme sur laquelle se trouve le joueur, qui s'élève et effectue un parcours faisant passer le joueur devant la troisième lampe (sur laquelle il doit tirer alors qu'il se trouve sur une plate-forme en mouvement — nouvelle augmentation de la difficulté, restant pour le moment facile car la plate-forme va lentement, mais introduisant une utilisation de ce même mécanisme plus difficile par la suite) puis lui donnant accès à une partie en hauteur du niveau.

Arrivé sur celle-ci, le joueur doit regarder un peu autour de lui pour trouver la dernière lampe qui se trouve juste au-dessus de sa tête (introduction au fait que la position des lampes n'est pas toujours évidente et qu'il faut parfois les chercher), et fait apparaître le téléporteur juste à côté en tirant dessus.

Stage 5 - Plate-forme movenne



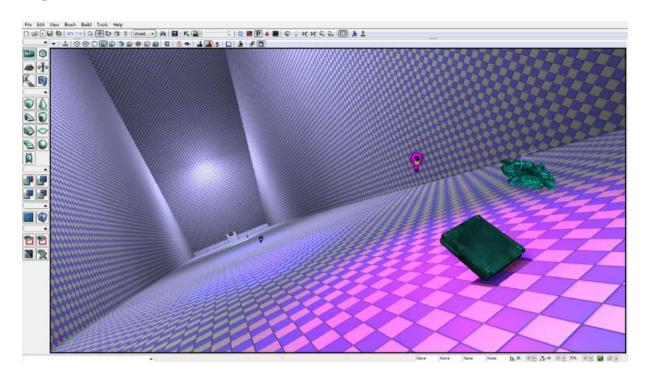


Le joueur arrive dans un niveau à la structure verticale : une première lampe se trouve devant lui, et des escaliers le long des murs de la tour lui permettent d'arriver tout en haut, suivant un parcours de plate-forme le long duquel se trouvent les lampes. On lui fait comprendre que les choses sérieuses commencent, avec ce message d'introduction : « Stage 5 >>> OK, more serious platforming - practice Long Jump by pressing Forward twice! ». Ceci donne aussi un indice pour terminer les niveaux plus facilement : l'utilisation du Long Jump, pas indispensable mais très utile. Au joueur de déterminer le style de jeu qu'il préfèrera.

En montant les escaliers selon un parcours linéaire, il va se trouver devant diverses situations impliquant des obstacles et de la plate-forme, où il devra faire preuve d'observation puis de timing pour passer sans retomber plus bas. Le temps pendant lequel les lampes restent allumées reste assez large pour lui laisser une marge d'erreur relative.

En arrivant en haut, s'il a bien touché toutes les lampes, il accède au téléporteur pour passer au stage suivant.

Stage 6 - Tir sur cibles mouvantes



Le joueur se retrouve ici dans une situation bien différente de celles auxquelles il a déjà été confronté : il tombe immédiatement sur une plate-forme, avec une lampe devant lui. Cette plate-forme se trouve tout en haut d'une très longue pente à 45°. En tirant sur la première lampe, la plate-forme se met en mouvement, et le joueur, voyant de loin que des lampes bougent tout le long de la lampe, va alors comprendre que la plate-forme sur laquelle il se trouve va suivre le trajet de la pente, et qu'il va devoir tirer sur les lampes sans bouger de celle-ci. Il est aidé dans cette compréhension par le message d'introduction du stage : « Stage 6 >>> Ride to the depth ».

La deuxième lampe n'est pas trop dure à atteindre : elle bouge lentement selon un aller-retour horizontal passant juste devant le joueur.

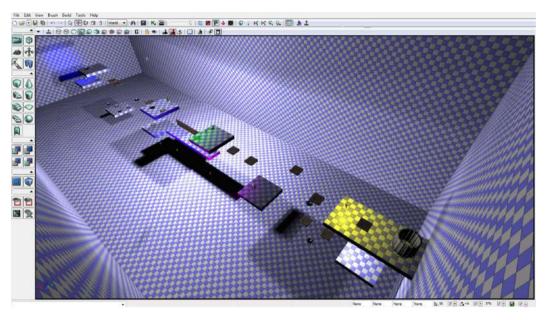
La troisième est plus difficile : légèrement décalée sur la gauche par rapport au joueur, elle bouge de haut en bas, un peu plus rapidement.

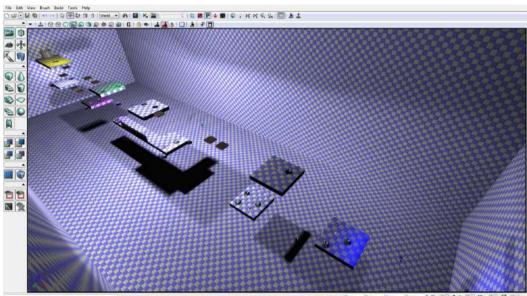
Si le joueur l'atteint, on profite du fait qu'il est en train de regarder vers la gauche pour faire apparaître la quatrième lampe : celle-ci va arriver du haut, et suivre un court moment le trajet de la plate-forme en décrivant trois cercles, puis s'en aller. Il faut arriver à la tirer au vol, alors qu'elle n'est pas pile devant le joueur et qu'elle décrit un trajet plus complexe.

Enfin, pour changer encore un peu, on va le placer devant les deux dernières lampes d'un coup : visibles de relativement loin, elles finissent par s'approcher de la plate-forme, et la suivent en restant au-dessus. Seulement, cette fois, c'est la plate-forme elle-même qui se met à décrire un trajet chaotique en zigzag ! Il faut donc parvenir à atteindre les lampes alors que le joueur suit un trajet relativement imprévu.

En arrivant tout en bas du parcours, le joueur est déposé devant le téléporteur s'il a touché toutes les lampes. Dans le cas contraire, il peut se jeter dans le vide, ce qui le téléportera à nouveau au début du niveau pour retenter sa chance.

Stage 7 - Plate-forme difficile





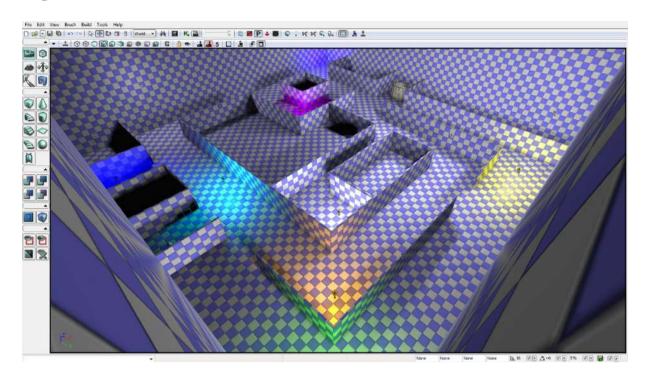
Le message d'introduction fait comprendre que le parcours va être assez varié niveau mouvement : « Stage 7 >>> Back and forth, up and down ».

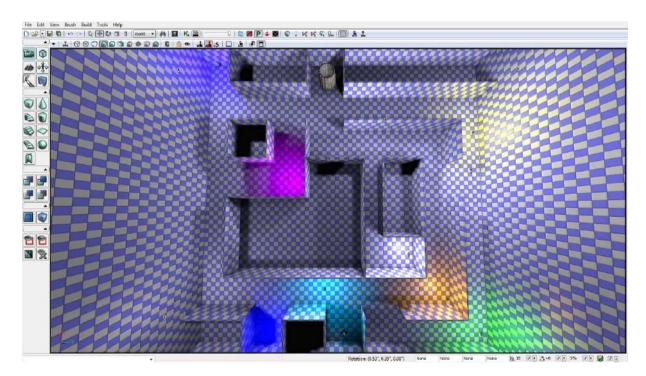
Dès le début, le joueur se retrouve face à une lampe. Mais en tirant dessus, il se révèle qu'elle est blanche, et donc que c'est la dernière! En effet, la fin du niveau est située juste au-dessus du début, et il sera possible de tirer sur cette lampe depuis une trappe dans le plafond.

Pour le moment, après ce faux espoir, le joueur se retrouve devant des plate-formes tournantes pour arriver au « vrai » début du stage. Ceci passé, il va avoir une suite d'épreuves de plate-formes bien plus difficiles que précédemment, et de sauts. En effet, on introduit dans ce niveau les Jumpers, des éléments qui font accomplir un saut prédéterminé au joueur, sur lequel il n'a pas d'influence. Des lampes sont positionnées sur le trajet de ces sauts, et il appartient au joueur d'arriver à les tirer au vol. Ceci introduit encore une nouvelle mécanique et un nouveau challenge utilisant toujours le même système de lampes à tirer.

A la fin du niveau, située au-dessus du début, le joueur tire sur l'avant-dernière lampe, et une trappe s'ouvre alors dans le sol, permettant de tirer sur la dernière lampe. Le téléporteur apparaît alors derrière le joueur.

Stage 8 - Orientation

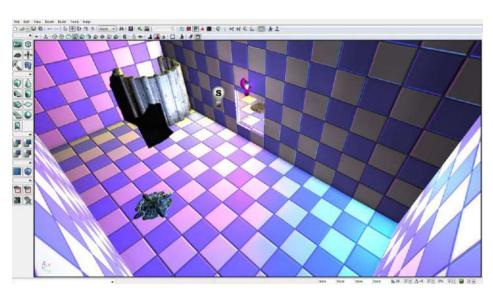


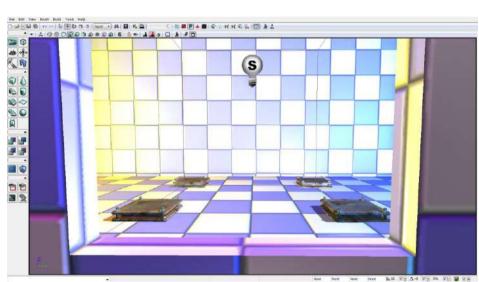


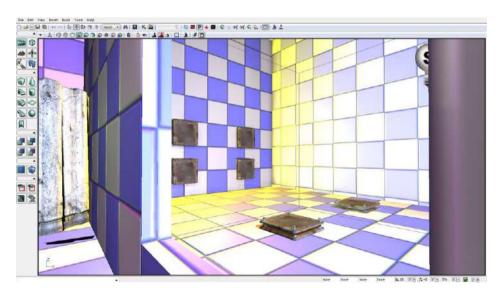
On annonce immédiatement la couleur avec le message d'introduction : « Stage 8 >>> Labyrinth ». L'idée est ici complètement différente de ce que le joueur a vu jusqu'alors : plus de rapidité, ni de précision de tir, ni de plate-forme, le joueur est dans un labyrinthe qu'il va devoir commencer par explorer pour en comprendre la disposition. C'est une épreuve d'orientation et de mémoire : en regardant les couleurs des lampes, il va devoir déduire le trajet optimal à suivre pour les allumer dans l'ordre, et se retrouver dans le labyrinthe en suivant le bon trajet dans le temps imparti. Le téléporteur apparaît au fond du labyrinthe, dans un endroit que le joueur ne peut pas manquer en

revenant du cul-de-sac où se trouve la dernière lampe.

Stage 9 - Stand de tir







On change encore complètement la manière d'appréhender le concept de base : « Stage 9 >>> Welcome to the Shooting Range! ». Le joueur se retrouve dans une salle assez petite, où une ouverture à sa hauteur le laisse voir dans une autre salle, mais il ne peut pas la traverser. Cette salle est un stand de tir ! En tirant sur la première lampe, il va déclencher une séquence d'apparition des lampes suivantes, qui vont passer rapidement devant lui selon des trajets divers, et il va devoir leur tirer dessus pendant le court laps de temps où elles sont visibles.

Un intérêt supplémentaire au stage est que le pattern d'apparition des lampes n'est jamais le même : un random est inclus dans leur position d'apparition ou leur trajet ! Ainsi, la difficulté est augmentée, et il n'y a pas de sentiment de répétition.

Le schéma, augmentant en difficulté au fur et à mesure, est le suivant :

Lampes 2 à 4:

Une trappe au sol parmi 4 (disposées en carré), déterminée aléatoirement, s'ouvre, et la lampe en sort. Elle reste en l'air un temps limité puis redescend et la trappe se referme. Le temps durant lequel la lampe reste en l'air se réduit graduellement de la 2 à la 4.

Lampes 5 à 7:

Une trappe dans le mur de gauche parmi 4 (disposées en carré), déterminée aléatoirement, s'ouvre, et la lampe en sort. Elle traverse la salle de gauche à droite, et le joueur doit la toucher avant qu'elle atteigne le mur de droite. La vitesse de déplacement de la lampe augmente graduellement de la 5 à la 7.

Lampe 8:

La lampe sort du mur de gauche comme les trois précédentes. Elle peut suivre deux patterns différents, déterminés aléatoirement.

Pattern 1: La lampe suit un trajet vers la droite, en décrivant des zigzags de haut en bas.

Pattern 2 : La lampe suit un trajet vers la droite, en décrivant des cercles.

Lampe 9:

Les quatre trappes au sol s'ouvrent, et la lampe sort de celle située le plus près du joueur, sur sa droite. Elle peut suivre deux patterns différents, déterminés aléatoirement.

Pattern 1 : La lampe saute de trappe en trappe, dans le sens des aiguilles d'une montre.

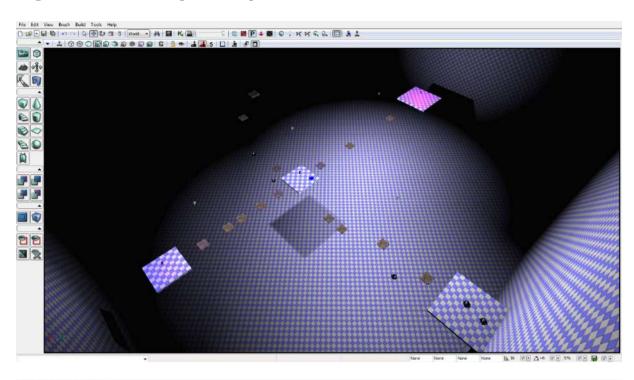
Pattern 2 : La lampe saute de trappe en trappe, dans le sens contraire des aiguilles d'une montre.

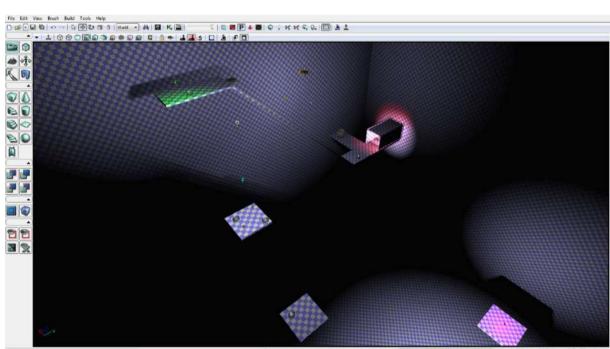
Une fois que la lampe est revenue à sa trappe d'origine, les quatre trappes se referment aussitôt.

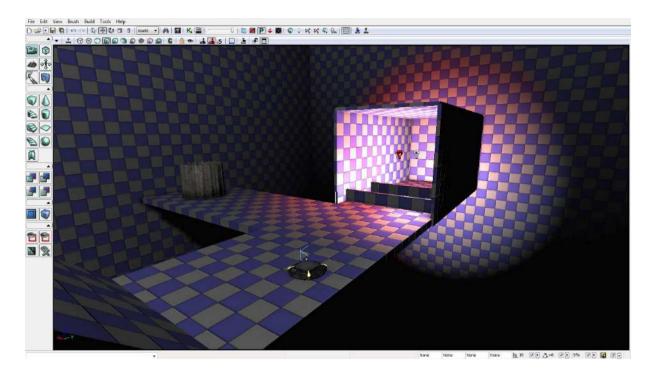
Si le joueur a échoué à allumer toutes les lampes (et il est improbable qu'il y arrivera du premier coup), un tir dans la première lampe au-dessus de la fenêtre du stand de tir relance la séquence.

S'il a touché toutes les lampes, le téléporteur apparaît à côté de lui.

Stage 10 - Somme des expériences précédentes







Ce dernier niveau a pour but de « résumer » ce que le joueur a fait jusque là, et de le placer dans toutes les situations qu'il a vu précédemment, comme une dernière épreuve lui demandant de se rappeler de tout ce qu'il a appris. Le titre, « Stage 10 >>> The last trial... », lui laisse clairement entendre qu'il est proche de la fin.

Le niveau commence donc par de la plate-forme assez difficile. Le timing est plus complexe que jamais, et demande encore plus de précision. Après une introduction sans lampe, un embranchement se présente au joueur : il est sur une plate-forme stable, avec une lampe. Deux chemins de plate-forme se présentent, un de chaque côté, menant à des lampes éloignées. Le joueur doit regarder les couleurs des trois lampes pour suivre le bon ordre : il doit d'abord prendre le chemin de droite, puis celui de gauche, et enfin, revenu à sa plate-forme de départ, tirer sur la lampe qui s'y trouvait, qui est en fait la troisième.

On passe ensuite à une séquence de sauts avec des Jumpers, plus hauts que jamais, demandant de toucher deux lampes lors de deux sauts consécutifs.

A l'issue de ce passage assez court, une version miniature de la pente du niveau 6 se présente : à nouveau, deux lampes se trouvent le long de cette pente, décrivant des trajets rappelant ceux du niveau 6, mais légèrement différents.

Enfin, le joueur arrive face à un stand de tir comme dans le niveau 9 : une première lampe suit un trajet aléatoire de haut en bas ou de gauche à droite. S'il la touche, il fait apparaître une autre lampe qui fait deux sauts rapides puis revient se cacher derrière le stand. S'il la touche aussi, la dernière apparaît : elle part du bas et effectue, de manière déterminée aléatoirement, soit un rebond sur les murs et le plafond avant de revenir se cacher, soit un trajet ascendant, deux boucles, puis un trajet descendant pour revenir se cacher.

Si le joueur finit par toucher toutes les lampes dans les temps, le dernier téléporteur apparaît au bout d'un court chemin partant du stand de tir. Un message apparaît pour le féliciter : « STAGE 10 CLEAR!!! GOOD JOB, AND THANKS FOR PLAYING! ».

... Mais où mène ce dernier téléporteur... ?

Épilogue - Un nouveau départ

En prenant le téléporteur, le joueur se retrouve... à l'entrée du premier niveau ! Un message teaser insinue que tout ceci n'était qu'un échauffement, et l'invite à rejouer pour améliorer son temps : « You completed the level! Now what about a speedrun for REAL gameplay? ».

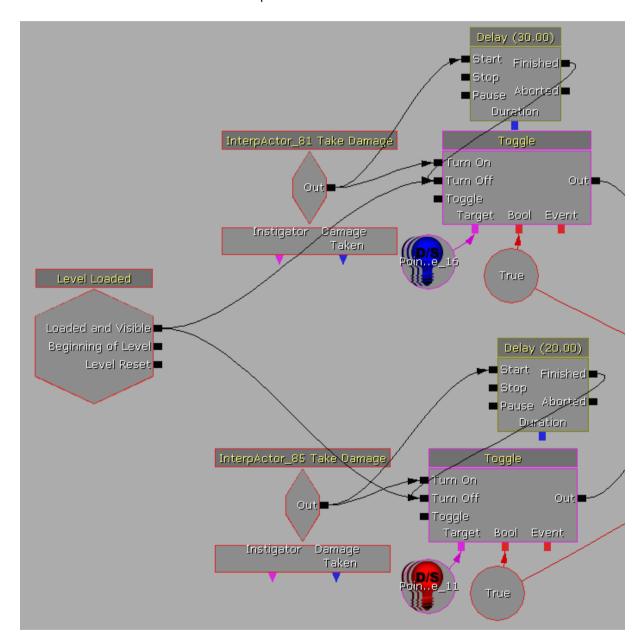
De plus, le temps mis par le joueur pour finir l'intégralité du niveau lui est indiqué. A lui de battre maintenant son propre record !

4 - Kismet : quelques éléments de script

Cette section va détailler quelques éléments centraux des scripts qui ont été mis en place pour le fonctionnement des niveaux.

Allumage, calcul du temps, et éteignage des lampes

Examinons le cas d'un niveau à deux lampes :



Les éléments utilisés sont les suivants :

- Level Loaded, pour effectuer des actions dès le lancement du niveau.
- Take Damage, pour détecter quand le joueur tire sur les lampes.
- Toggle, pour allumer ou éteindre les lampes
- Delay, pour déterminer le temps au bout duquel une lampe allumée s'éteindra.

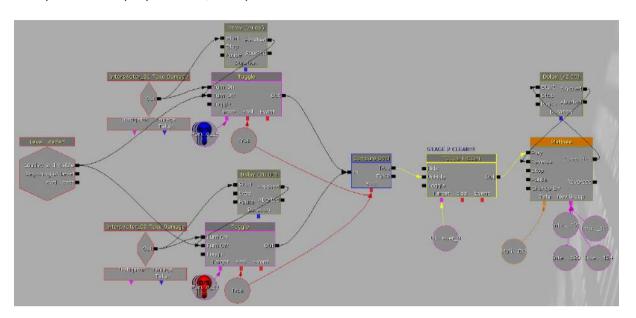
Les booléens visibles sur le screenshot seront utilisés pour un autre processus qui sera détaillé plus bas.

Le système fonctionne de la manière suivante : une lampe est composée d'un mesh, et d'une source lumineuse. Celle-ci est Off quand la lampe est éteinte, et On quand elle est allumée. Par défaut, toutes les lumières sont On. Il faut donc dès le lancement du niveau toutes les éteindre : pour ceci, Level Loaded est relié au Turn Off du bloc Toggle lié à chaque lumière.

Pour détecter le tir du joueur sur la lampe, un détecteur Take Damage sur le mesh de la lampe est mis en place : il est lié premièrement au Turn On du blog Toggle de la lumière assignée à ce mesh, ce qui allume donc la lumière dès que le mesh prend un tir. Mais il faut aussi prendre en compte le temps au bout duquel la lumière est censée s'éteindre : pour ceci, le Take Damage est aussi lié à un bloc Delay. Dès que le mesh de la lampe prend un tir, le bloc Delay lance un timer à la durée déterminée, et comme la sortie Finished du Delay est liée au Turn Off de la lumière, quand ce timer est fini, la lumière s'éteint à nouveau.

Vérification de l'allumage de toutes les lampes, et validation de l'accès à la suite

On reprend l'exemple précédent, mais plus étendu :



On ajoute ici au système précédent l'utilisation des éléments suivants :

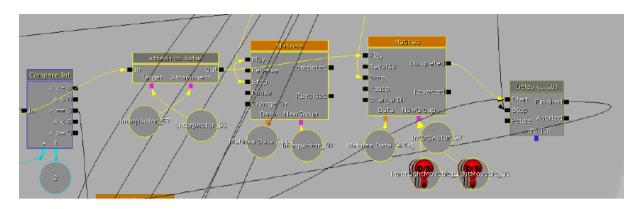
- Variables booléennes, pour déterminer si une lampe est allumée.
- Compare Bool, pour comparer ces variables et savoir si toutes les lampes sont allumées.
- Toggle Hidden, pour faire apparaître le téléporteur.
- Matinee, pour faire bouger les murs bloquant l'accès au téléporteur.

L'important est ici de savoir quand toutes les lampes sont On. Pour ceci, on passe par une variable booléenne intermédiaire : un booléen est lié au Toggle de chaque lampe. Il va donc changer d'état selon les actions effectuées par ce Toggle. Il est initialisé à True. Dès le lancement du niveau, comme le Toggle reçoit une impulsion dans son Turn Off (envoyée par le Level Loaded), le booléen change d'état et passe à False. Il changera d'état à True quand le Turn On du Toggle sera activé. Mais il deviendra à nouveau False quand le Turn Off sera activé à l'issue du temps assigné au timer. En résumé, ce système permet d'avoir le booléen à True quand la lampe est allumée, et à False quand elle est éteinte.

On lie ensuite l'ensemble des booléens à un bloc Compare Bool, dont l'entrée In est liée à tous les blocs Toggle, ce qui fait qu'il est activé à chaque Toggle : ainsi, chaque fois qu'une lampe change d'état, le système vérifie si elles sont toutes allumées ou pas. La sortie True du Compare Bool est liée aux évènements de fin de niveau, ce qui fait que cette séquence ne s'enclenche que quand tous les booléens sont à True, donc toutes les lampes allumées : on active d'abord l'entrée Unhide d'un bloc Toggle Hidden agissant sur le téléporteur de sortie (qui est Hidden par défaut) pour le faire apparaître, et par la suite une matinee faisant s'élever les murs autour de ce téléporteur pour en libérer l'accès.

Notons enfin que comme le jeu est fait pour être rejoué immédiatement une fois tous les stages terminés, il fallait que ces murs se baissent à nouveau au bout d'un temps, pour que le joueur ne les trouve pas levés en faisant le stage pour la deuxième fois. Un timer est donc lié à la matinee, et l'inverse, faisant se baisser les murs, à la fin du temps imparti.

Créer une trajectoire complexe en fusionnant deux trajectoires simples



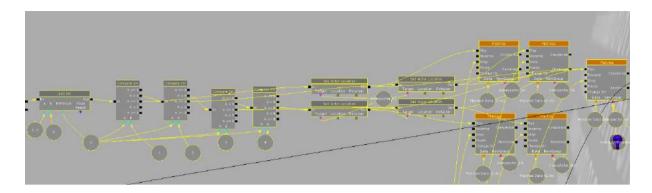
Pour créer par exemple le mouvement d'une lampe qui avance en décrivant des cercles en même temps, la méthode choisie n'a pas été de créer ce mouvement en un seul bloc, une seule trajectoire d'animation : cela rendait sa création plus difficile, plus longue, plus précise, et des changement de trajectoires subséquents assez compliqués. Il était beaucoup plus pratique de décomposer ce mouvement en deux parties plus simples à travailler individuellement : le mouvement qui avance d'une part, et les cercles d'autre part.

Ainsi, on commence par placer un mesh sur la map, qui sera mis en Hidden pour être invisible. On assigne à ce mesh une matinee avec le mouvement de déplacement horizontal. Il va en quelque sorte constituer l' « ancre » décrivant le trajet principal de la lampe, à laquelle celle-ci va être rattachée.

Ensuite, pour la lampe en elle-même, on lui assigne une matinee décrivant uniquement des cercles. Puis on attache la lampe au mesh invisible, pour qu'elle le suive tout le temps.

Ensuite, quand on veut lancer la séquence d'animation complète de la lampe, il n'y a qu'à lancer simultanément les deux matinees, celle de la lampe et de l'objet invisible : comme la lampe est attachée à ce dernier, elle va décrire son mouvement de cercles tout en suivant la progression horizontale du mesh. On a ainsi créé une trajectoire complexe mais facile à mettre en place et à modifier grâce à la fusion de deux trajectoires simples.

Randomizer d'évènements



Dans le stage 9, et dans une moindre mesure dans le 10, on a inclus des éléments aléatoires dans le lieu d'apparition et le pattern de déplacement des lampes. Le randomizer utilise les éléments suivants :

- Random Int, pour générer un entier de manière aléatoire.
- Add Int, pour stocker le résultat du random dans un entier.
- Variables entières, pour stocker le résultat du random.
- Compare Int, pour lancer l'action voulue selon le résultat du random.

On commence par générer un entier aléatoire dans les limites voulues. Pour stocker cet entier dans une variable qui ne va plus bouger pendant les tests subséquents sur sa valeur, il faut utiliser la méthode suivante : avec un Add Int, on additionne cet entier à 0, et on stocke le résultat (identique donc à l'entier aléatoire de départ) dans une variable entière. Pourquoi passer par cet étape et ne pas juste utiliser le Random Int ? Parce qu'à chaque fois qu'il est appelé, un nouveau random est généré. Comme on a plusieurs comparaisons dans la suite de l'opération, le Random Int serait donc appelé plusieurs fois et susceptible de fournir des résultats erronés. On est donc obligé de passer par là pour obtenir un entier aléatoire qui ne va pas bouger pendant toute la séquence.

Une suite de Compare Int va ensuite vérifier la valeur de l'entier, et lancer la séquence voulue en fonction : pour chaque Compare Int, si l'entier est égal à la valeur assignée au Compare Int, on lance la séquence. Sinon, on passe au Compare Int suivant.

Dans l'exemple du screenshot, on voit le système utilisé pour la deuxième lampe du stage 9 (qui suit toujours le même pattern de mouvement mais peut apparaître aléatoirement sous chacune des quatre trappes du sol). Selon le résultat du random, on déplace la lampe grâce à un Set Actor Location sous la trappe voulue, et on lance ensuite le matinee faisant s'ouvrir la trappe correspondante, et la matinee accomplissant le mouvement de la lampe.

Pour l'autre utilisation principale de ce randomizer, à savoir choisir l'un ou l'autre pattern de déplacement pour une lampe, on va simplement lancer l'une ou l'autre matinee liée à la lampe selon le résultat du random.

Intégralité du Kismet

Le screenshot ci-dessous montre l'intégralité du Kismet utilisé pour le niveau, classé stage par stage.

