

TSOPTIHSIARVEL

‘VIKING VILLAGE’

PIERRE DAUNIS - MATTÉO SISCARO - ADRIEN DAUTEL - LOGAN MAJSTOROVIC
BAPTISTE DENOUX - PABLO GOMEZ - VINH-AN BUIS

« Peut-on y mettre un dragon en hors champ? »

« Tant que vous pouvez le sonoriser, vous faites ce que vous voulez. »

La nuit tombe tel un voile maléfique sur la caverne. Les habitants de Tsoptihsiarvel, las d'une dure journée de labeur, se retrouvent en famille, ou à la taverne pour passer un bon moment.

Le petit père des peuples semble veiller sur la tranquille bourgade, mais les eaux sombres du lac Laogai ont déposé sur ses berges un présage qui pourrait bien mettre fin à la paisible vie des villageois.

Hagards et enivrés, ces derniers ne savent pas que de sombres desseins se tramont aux portes mêmes de leur logis...

I - Le commencement

Suite à un deep dive de Pierre dans le projet Unity qui a été retourné de fond en comble durant des sessions nocturnes de Level Design. Nous avons donc décidé d'opter pour une direction artistique de type Œufs de Pâques de la propreté; c'est-à-dire d'allier plaisanteries et qualité.

Notre objectif était donc d'intégrer des éléments humoristiques dans le village de Tsoptihsiarvel sans qu'ils ne cassent l'immersion du joueur. Il fallait que ces éléments soient pleinement intégrés dans l'ambiance, le contexte et le cadre du village. Les œufs de Pâques devaient rester cohérents avec le reste des éléments visuels et sonores.

Or cette mascarade a lieu dans un cadre bien défini : un village scandinave au centre d'un lac dans une sinistre grotte. Le but est d'avoir une ambiance assez sombre, nocturne et froide. Typiquement le genre de village que l'on pourrait trouver dans des jeux/films d'épouvante.

III - Production

1. Musique

Les sessions de productions musicales ont été réalisées sur Reaper 6, avec des VSTs communs à tout les membres de l'équipe afin de pouvoir plus facilement échanger les sessions et avoir un rendu cohérent sur l'intégralité du jeu.

IV - Intégration Unity

Chaque membre du groupe dispose de la scène « remaniée » du Village Viking afin de faire des test individuels, celle-ci n'est cependant versionnée que très rarement pour suivre des ajouts importants de systèmes audio ou de game design.

La mise en commun s'effectue sur une scène finale où les différentes sources sont recréées et reliées au Wwise, qui lui est commun et versionné.

Pour cela nous utilisons beaucoup de **Prefab** et de modules natifs à Unity ainsi que des **Scripts** pour les systèmes un peu plus complexes.

La majorité des scripts ont été réalisés pour répondre à des problématiques, par nécessité, ou pour expérimenter des techniques d'optimisation de performances ou de workflow.

//Dépendances

OnTriggerEnter/Stay/Exit

Ensemble de méthodes en C# nous permettant de définir quand un objet, entre, reste ou quitte un autre objet, afin de lancer des events, des calculs ou changer des valeurs.

Raycasting

Le Raycasting est une méthode en C# présente sur plusieurs **scripts(.cs)** qui nous permet de récupérer des informations en lançant un rayon depuis un objet, dans une direction précise.

//Systèmes

Player (FirstPersonController.cs)

Features concernées : Footsteps Material, Occlusion, Underwater State,

Scripts Toolbox

SphereDistance.cs

Est un script qui calcule la position d'un **GameObject** au sein d'une sphère sous forme de pourcentage, entre son centre et son rayon (100 = centre ; 0 = extrémités). Cette valeur est récupérable sous forme de **RTPC**.

Si plusieurs sphères se superposent, c'est la valeur la plus élevée qui est conservée.

SphereOnTriggerStayTimer.cs

SphereDistance.cs

Est un script qui calcule la position d'un **GameObject** au sein d'une sphère sous forme de pourcentage, entre son centre et son rayon (100 = centre ; 0 = extrémités). Cette valeur est récupérable sous forme de **RTPC**.

Si plusieurs sphères se superposent, c'est la valeur la plus élevée qui est conservée.

SphereOnTriggerStayTimer.cs

Est un script qui lance un décompte lorsque le joueur est à l'intérieur d'une sphère, le décompte se freeze si le joueur sors, une fois le décompte achevé, on lance une autre **méthode** ou un **event audio**.

SplineFollower.cs

Est un script qui permet de placer un **GameObject** sur une spline, un « chemin ». Ce **gameObject** suivra la position du joueur (au plus proche) en restant sur la spline.

1. Footsteps/Lands (Système de Raycasting)
2. Occlusion (Système de Raycasting)
3. Whispers/Shawn (Système de Timer OnTriggerStay)

Raycasting

Intégration Unity réalisée en utilisant **FMOD Studio** en version 2.01.04.

Le son est intégré de 3 manières différentes depuis FMOD Studio.

- En passant par les scripts natifs de **FMOD** sous Unity, pour ce qui concerne les emitters, bed d'ambiances et fonctions simples.
- En utilisant les scripts déjà existant des **GDP** pour différentes éléments de gameplay et en s'entretenant avec eux pour voir les meilleures possibilités d'intégration et debug.
- En créant des scripts supplémentaires pour des fonctions uniquement liées à l'audio, la plupart du temps pour récupérer des paramètres (**RTPC**) importants

V - Calendrier

