Vision de l’alpha

Les champs avec la mention « Ajustable » signifie des champs qui sont accessible directement à partir des objets sur la scène. Il suffit de mettre les variables correspondants publics.

L’on vise à avoir toutes les fonctionnalités implémentées et être en mesure de jouer une partie complète (de A à Z mais d’un seul niveau) du jeu sans le visuel ni le polissage.

# Tâches à Éric

### Pool d’obstacles :

Une gestion de l’instanciation des obstacles est nécessaire. Il n’est pas obligatoire d’avoir plus d’une pool ou une pool variable templated mais ceci pourrait être pris en considération pour les implémentations future des « Nice To Have ».

La pool doit être en mesure de récupéré les obstacles qui ont heurté le joueur et/ou sorti de l’écran et de les « reseter ». Soit qu’elle gérera par elle-même l’instanciation des obstacles ou une seconde classe gérera son instanciation, à ta guise.

### Intégration du HUD :

En attente du placerholder de HUD avant d’implémenter.

### Transition visuelle :

En attente du placerholder de HUD avant d’implémenter.

### Camera :

La camera ne doit que défiler verticalement. Dans la demande actuelle du fonctionnement de la caméra, elle ne doit que suivre le joueur directement. Il n’y a donc pas question de mode élastique.

### Autres :

En ce qui concerne les obstacles qui tombent, valider à ce que ce soit des OnTrigger et non des OnCollision, l’on ne veut pas que le joueur se voit massacré par le poids de l’obstacle.

# Tâches à Mike

### Barre de fumée :

La fumée doit monter de façon constante, un champ ajustable au niveau de la vitesse et l’utilisation du booléen IsRaising pour dictée sa montée

### Trigger Son :

On s’attend à ce que toutes les actions des personnages aient une fonction style Fire & Forget (On le joue mais on n’en garde pas référence) avec un son ajustable. Il ne restera qu’au sound designer de glisser son dans les prefabs des objets correspondant.

Un trigger de son doit être présent pour :

* Le saut du joueur.
* Atterissage du joueur.
* Personnage heurté par un obstacle & lorsqu’il touche au nuage de fumée montante.
* Personnage récupère un collectable.

Un trigger de stinger (C’est un son plus long, cela ne change rien à son implémentation) :

* À la fin du niveau à la défaite (0 ou 1 collectible ramassé)
* À la fin du niveau à son succès (2 ou 3 collectible ramassé)

Cependant pour les stingers, il est possible que ceux-ci ait un impact sur le démarrage de la musique. Ce sera à confirmer avec le sound designer mais n’est pas une priorité importante encore.

Il faut bien faire attention à ce que les sons ne soient joués qu’une seule fois. EX : Lorsque le joueur tombe dans le nuage de fumée, l’on ne veut pas que le son soit appelé 3-4 fois. C’est une problématique qui ne sera pas présente et ne devras pas nécessairement être ajusté si le flow du gameplay est bien ajusté donc à ne pas prendre en considération importante pour l’instant.

### Trigger Musique :

On s’attend à entendre une musique de façon constante à travers le jeu.

* Musique de menus
* Musique de jeu

La musique de menu, comme elle s’indique doit toujours être joué lorsque l’on est dans un menu. La musique de jeu ne doit jouer que lorsque l’on est en jeu.

Une seule musique ne peut être activé à la fois et on ne s’attend pas à ce que la musique de menu recommence avec chaque changement de menu.

### Fin de niveau :

Lorsque l’on arrive à la porte de fin de niveau, il n’y a que deux possibilités.

Si l’on a 0 ou 1 collectibles -> Écran de défaite

Si l’on a 2 ou 3 collectibles -> Écran de victoire

Lorsque l’on meure par l’arrivée du nuage il n’y a qu’une possibilité -> Respawn du joueur, ce qui correspond au redémarrage du niveau.

### Intégrer la navigation :

En attente du workflow de navigation avant de préciser…