Vision du prototype

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 tous les outils nécessaires pour effectuer un « gym ».

# Tâches à Éric

### Player :

* Vitesse de déplacement ajustable
* Force de saut ajustable
* Effet de toxicité ajustable (Pourcentage) (Combien de malus de toxicité est obtenu à chaque fois qu'on est touché par un obstacle)
* Limite de l'effet de toxicité ajustable (Pourcentage) (Jusqu'à combien de toxicité le joueur peut se rendre)
* Nombre de collectable actuel
* Malus de Toxicité actuel

Jump() -> Fait sauter le joueur :

Doit pouvoir changer le state du player au début et à la fin du saut (pour triggerer des sons & animations)

HandleInput() -> Gère les contrôles du joueur et appelles les actions appropriées :

Contrôle gauche/droite

Saut

OnCollisionEnter2D()-> Gère les collisions, possibilité de triggerer un son à chaque collision

Avec Door-> Appel LaunchScreen(*Nombre de collectable*).

Avec Obstacle-> Incrémente la toxicité du joueur.

Appel DisplayToxicity(*toxicity*) du UI pour updater l'affichage de celui-ci.

Appel Vanish() de l’obstacle

Avec Smoke-> Respawn du joueur.

Avec Collectable-> Augmente le nombre de collectables du joueur.

Appel DisplayCollectable(nbCollectable) du UI pour updater l'affichage de celui-ci.

Appel Vanish() du collectable

Respawn()-> Remet le joueur au début du niveau, la solution facile est de simplement recharger le niveau actuel.

# Tâches à Mike

### Platform:

* Vitesse de déplacement ajustable
* Maximum de hauteur ajustable (*La platforme est placé à son point le plus bas et ne peux parcourir que la distance verticale spécifié*)

On s'attend à ce que la plateforme monte/descend continuellement mais qu'il y ait une interpolation pour le changement de direction. C'est à dire que la plateforme ralentira avant de changer de vitesse et ne reprendra pas sa vitesse maximale avant un court délai.

Le Prefab de plateforme doit être fait pour supporter une texture qui tile, permettant ainsi au level designer de pouvoir scaler la plateforme (horizontallement) aisément sans que la plateforme se distingue des autres au lieu de créer différente plateformes pour chaque tailles.

Un prefab de Door, Obstacle, Smoke & Collectable devra être créé.

Chacun ayant un Placeholder visuel (se limiter à boite/cercle), un collider

## Obstacle :

Les obstacles doivent tomber constamment (ils ignorent le terrain).

Doivent avoir une vitesse de tombé statique ajustable.

Spawn() -> Place l’objet à un endroit au hasard dans le haut de l’écran.

Pour prévenir des tombés successives aux mêmes endroits, le système devra être fait pour que l’écran soit séparé en plusieurs parties (EX : séparé en 3 tiers horizontallement (À considérer ajustable?)) et initié un aléatoire à l’intérieur de cette séparation.

Vanish()-> Obstacle & Collectable devront implémenter une interface qui possède cette fonction qui désactivera le visuel de l’objet ainsi que toute détection de collision envers celui-ci.

Une fois que l’obstacle est hors de l’écran, il devra spawner à nouveau, inversant les effets de la fonction Vanish si jamais il avait heurté le joueur.

L’on peut apporter des changements à chacun de ces points. Il est important de garder en note que le but est de livrer quelque chose de fonctionnel, les raccourcis (sans répercussion négative) seront acceptés.