**Répartition des tâches**

**Mégane** ayant des difficultés en code, c’est **le zoom** qui lui a été confiée.

**Thomas** s’est chargé de coder **le gyroscope** étant donné qu’il a déjà fait ça pendant son stage.

**Flore** voulait coder la **récupération de la couleur moyenne** de l’environnement parce que ça avait l’air « fun » et qu’elle voulait un peu de challenge. Elle s’est aussi occupée de la **division des canaux d’une couleur** et du script du **drag and drop**.

**Marine** savait coder un **double clic**, un **changement de niveau** et pouvait se charger de **vérifier la concordance des couleurs**, c’est donc ces tâches qui lui ont été confiés.

**Mathéa** ne sachant pas coder, c’est elle qui s’est chargée de **la mise en place des niveaux**.

Enfin, **Flore** et **Marine** étant les plus compétentes en code, ce sont elles qui **relurent et corrigèrent les scripts** ainsi que **l’harmonisation du code**.

**Explication des scripts :**

**PinchZoom :** On configure la puissance du zoom et on vérifie à chaque frame si deux doigts on touchés l’écran, si c’est le cas, on active le zoom. Pour cela, il faut stocker le nombre de touches, dans ce cas, il y en a deux. On vérifie ensuite la direction et la distance du pinch puis on l’applique en changeant le FOV.

**ChangeLevel:** Quand on trigger l’arche, on accède au niveau suivant.

**Arch :** Attribue une couleur aléatoire de l’arche gère son trigger. Quand on bouge, la caméra arrière du téléphone va récupérer la couleur moyenne de l’environnement de GetCamColor et modifier le material des cubes présents dans la scène. Si c’est le cas, le trigger de l’arche est activé et on peut passer à travers pour accéder au niveau suivant. Si non, le trigger reste désactivé.

**DoubleTapAdvance :** Permet à la caméra d’avancer quand on tape deux fois sur l’écran. On commence par configurer le cooldown, la vitesse de déplacement et la distance de déplacement. On va ensuite compter le nombre de « tap » qu’il y a eu sur l’écran. Le cooldown est lancé juste après le 1er tap, on considère qu’il y a eu un double tap s’il y a eu un autre input avant la fin du cooldown. S’il y a eu double tap, la caméra peut avancer. Pour ce faire, on utilise une coroutine qui va s’exécuter tant qu’une distance donnée n’a pas été parcourue.

**Gyroscope :** Va faire tourner la caméra en fonction du gyroscope du smartphone.A chaque frame, on udpate la rotation de la caméra basé sur l’orientation du gyroscope. On s’assure que la caméra fonctionne, que l’on soit en format paysage ou en format portrait.

**CheckChannel :** On commence par sélectionner de manière aléatoire une couleur (couleur qui n'est pas le noir à cause du fait que la caméra débute en étant noir). De cette couleur est alors récupéré un des 3 canaux RGB. Ce canal est déterminé par une énumération sélectionnable dans l’éditeur puis GetCamColor se charge de retourner une couleur ne contenant que ce canal. Le score est incrémenté. La suite du code vérifie la collision du cube sur lequel est appliqué le script avec un cube du joueur, si les 2 cubes sont de couleur proche, celui du joueur est détruit.

**DragCube :** On commence par initialiser la position du cube. On récupère ensuite la couleur moyenne de la caméra arrière du téléphone via GetCamColor, puis on applique au cube la couleur ne contenant qu’un des canaux de la couleur moyenne. Le choix du canal est faisable dans l’éditeur. On vérifie également si on glisse son doigt sur l'écran. Si c'est le cas, le cube se déplace sur l’axe y, en suivant le doigt, il ne bouge pas dans les autres directions. Lorsque le doigt est retiré, le cube retourne sa position initiale.

**Game Manager :** GameManager permet de n’avoir qu’un seul GetCamColor dans la scène. Les scripts peuvent appeler le GameManager pour récupérer les propriétés de GetCamColor grâce au mot-clef « static ». Dans ce script, on crée aussi 2 variables, l’une permettant de compter les points du joueur et l’autre d’initialiser le nombre de point requis pour passer le niveau. Une fonction permet d’ajouter des points. Si les points du joueur atteignent le nombre de point requis, alors le script va afficher le texte de victoire.

**GetCamColor :** Ce script récupère toutes les couleurs des pixels à l’écran. Il fait ensuite une moyenne de ces couleurs pour stocker le résultat dans la variable MainColor. Il refait le calcul à chaque fois qu’il est appelé. On s’assure que la caméra est bien lancée et que le tableau soit à la bonne taille. GetCamColor ne dépend pas de monobehaviour, ce qui permet de l’appeler de n’importe quel script. Il permet également de ne récupérer qu’un canal pour une couleur donnée.

Le code a été organisé en fonction des tâches assignées (un script par tâche). Il a été décidé de ne pas faire dépendre GetCamColor de Monobehaviour afin de faciliter sa réutilisation dans les 2 scènes.