



AR avec Unity (Suite)



La Citation du jour :

“La refactorisation c’est comme faire la vaisselle après la cuisine.”

—Toby Parkins



Et pour commencer :

- **Rappel des épisodes précédents**
- **Et les projets alors ?**
- **Et aujourd'hui on fait quoi ?**



Programme de la journée:

- **Des exercices**
- **Un projet à finir : MasterMind**



Exercices 1 :

Objectif :

Reproduire le fonctionnement de l'exemple de l'image tracker basic.

Fonction attendues :

- reconnaissance d'une image et une seule
- apparition d'un objet présélectionné
- si on perd de vue l'image, l'objet disparaît



Exercices 1 :

Éléments de réponse :

Nous allons utiliser l'Event produit par le script ARTrackerImageManager de l'AR Foundation pour faire apparaître ou disparaître l'objet.

Ce script contient une action (action : autre syntaxe pour delegate + event) :

```
public event Action<ARTrackedImagesChangedEventArgs> trackedImagesChanged;
```

On peut donc s'inscrire à cet event en récupérant comme arguments ARTrackedImagesChangedEventArgs. Cette structure contient 3 List<ARTrackedImage> nommées added, updated et removed.

A chaque fois que l'événement est envoyé, les trois listes sont donc envoyées aussi. La liste added contient les images qui ont été trouvées durant cette frame, updated les images qui sont toujours présentes durant la frame et removed les images qui ont été perdues durant la frame.



Exercices 2 :

Objectif :

Reconnaître une image, faire apparaître un objet et pouvoir changer d'objet grâce à un bouton.

Fonction attendues :

- reconnaissance d'une image et d'une seule
- apparition d'un objet présélectionné
- remplacement de l'objet par un autre lorsque on appui sur un bouton
- si on perd de vue l'image l'objet disparaît
- si on revient vers l'image c'est le dernier objet qui réapparaît



Exercices 2 :

Éléments de réponse :

Nous allons ici utiliser un bouton pour changer d'objet.

Il nous faut un tableau contenant les différents objets à instancier.

Lorsque l'on trouve la première image on instancie le premier.

Si l'on appuie sur le bouton, on détruit l'objet actuel et on instancie le suivant dans le tableau.

Si l'on perd l'image puis qu'on la retrouve on se sert du dernier objet instancié

On remarque que l'on peut faire un fonction d'apparition d'objet en lui donnant juste le numéro de l'objet dans le tableau, et on fait évoluer ce numéro en fonction des clics sur le bouton.

Attention il faut penser à rendre le bouton inutilisable quand il n'y a pas d'objet visible.



Exercices 3 :

Objectif :

Reconnaître une image, faire apparaître le plateau de mastermind. Il n'est pas encore nécessaire de pouvoir y jouer.

Fonction attendues :

- reconnaissance d'une image et d'une seule
- apparition du mastermind
- si on perd de vue l'image le mastermind disparaît
- si on revient vers l'image le mastermind réapparaît dans le même état d'avancement



Exercices 3 :

Éléments de réponse :

Ce n'est rien de plus que la première version, sans le bouton de changement d'objet.

Par contre il faut faire apparaître le plateau de jeu complet, donc en faire un prefab.

Il faudrait réfléchir de suite à la logique de sauvegarde de l'état du mastermind :

- Si l'on désactive juste le plateau, il faudra faire attention à ne pas pouvoir déclencher d'actions alors qu'il est invisible.
- Si on le détruit, il faut un système de sauvegarde pour le faire réapparaître en l'état.

La deuxième solution est plus propre et plus compliquée.



Projet à finir : Le master mind en AR

Objectif :

Pouvoir jouer au mastermind en réalité augmentée !

Fonction attendues :

- reconnaissance d'une image et d'une seule
- apparition du mastermind
- pouvoir y jouer, gagner ET perdre !
- si on perd de vue l'image le mastermind disparaît
- si on revient vers l'image le mastermind réapparaît dans le même état d'avancement



Projet à finir : Le master mind en AR

Éléments de réponse :

Nous avons déjà le système pour le faire apparaître en AR.

Il faut maintenant décider du gameplay en AR, voici des idées mais d'autres peuvent être trouvées :

- Systeme de changement de couleur avec des boutons en UI (un bouton par sphère, avec switch automatique à chaque clic)
- Systeme de drag'n'drop à partir d'une liste de couleur
- Systeme de pick de couleur : une liste de couleur, on touche une couleur puis on touche la sphère que l'on veut colorer

Il faudra évidemment un système de validation qui va déclencher la vérification, et un système d'affichage des résultats.

La validation peut ce faire par un bouton en UI, ou un élément 3D clairement identifié.

Pour l'affichage des résultats, le systeme classique des 4 pions de deux couleurs différentes sont à privilégier car c'est le systeme d'origine, mais du texte en UI peut fonctionner aussi. Attention les résultats doivent être toujours et clairement disponibles pour chaque essai.