



# AR avec Unity (Final)



### La Citation du jour :

“Il y existe deux manières de concevoir un logiciel. La première, c’est de le faire si simple qu’il est évident qu’il ne présente aucun problème. La seconde, c’est de le faire si compliqué qu’il ne présente aucun problème évident. La première méthode est de loin la plus complexe.”

— C.A.R. Hoare



**Et pour commencer :**

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



### **Programme de la journée:**

- **Finir le MASTERMIND !**
- **Pour les plus rapides : test de la reconnaissance de plans**



## MASTERMIND en AR :

### Objectif :

Pouvoir jouer au mastermind en AR.

### Fonctionnalités attendues en AR :

- Trouver une image prédéterminée
- Faire apparaître le plateau de jeu mastermind
- Si l'image est perdue, faire disparaître le plateau de jeu
- Si l'image est retrouvée, faire réapparaître le plateau de jeu dans le même état que lorsqu'il a disparu.



# MASTERMIND en AR :

### Éléments de réponse :

Nous dans le cas de l'exercice 1, donc il suffit juste de le faire apparaître au bon endroit.

Attention au sens d'apparition, le joueur doit être en face de la première ligne de jeu.

Cette étape a du être déjà traitée dans la précédente journée.



## MASTERMIND en AR :

### Fonctionnalités du jeu de base :

- Le jeu doit créer une solution qui contient une combinaison de 4 éléments parmi 4 couleurs.
- Le joueur doit appliquer une des 4 couleurs à chacun des 4 éléments de sa ligne de proposition
- Le joueur doit avoir une interface pour demander la vérification de sa ligne
- Le jeu doit indiquer par des pions de couleur pour chaque éléments :
  - s'il est de la bonne couleur et à la bonne place,
  - s'il est de la bonne couleur mais pas à la bonne place
  - s'il n'est pas de la bonne couleur et pas la bonne place
- Si la combinaison est la bonne le jeu doit indiquer que le joueur a gagné !
- Si au bout de 12 essais le joueur ne trouve pas la bonne réponse, il a perdu et le jeu doit lui dire
- Quand la partie est finie (par une victoire ou une défaite, le jeu propose de recommencer



# MASTERMIND en AR :

### Éléments de réponse par fonctionnalités :

- Le jeu doit créer une solution qui contient une combinaison de 4 éléments parmi 4 couleurs.

Cette étape n'a pas de raison de changer. L'utilisation en AR ne change rien.

Vous avez besoin de disposer d'un tableau contenant la réponse attendu avec des couleurs choisies au hasard.





## MASTERMIND en AR :

Éléments de réponse par fonctionnalités :

- Le joueur doit appliquer une des 4 couleurs à chacun des 4 éléments de sa ligne de proposition

Attention, nous avons besoin ici de choisir un gameplay fonctionnel. Voici des propositions par ordre de difficulté à coder :

- **4 boutons UI** qui correspondent aux 4 sphères de la ligne, en appuyant sur un bouton on change la couleur de la sphère associée. Les couleurs changent de la première jusqu'à la quatrième puis recommencent à la première. pas de difficulté particulière.
- **systeme de pick up de couleur en UI.** on dispose en UI de 4 boutons avec chacune des couleurs. On clic sur une couleur puis on clic sur une sphère et ça applique la couleur à la sphère. Cette solution nécessite un raycast à partir de l'endroit où on touche pour cliquer sur la sphère.
- **systeme de pick up de couleur en 3D.** on dispose sur le plateau de jeu de 4 boutons avec chacune des couleurs. On clic sur une couleur puis on clic sur une sphère et ça applique la couleur à la sphère. Cette solution nécessite un raycast à partir de l'endroit où on touche pour cliquer soit sur une couleur soit sur la sphère.
- **DRAG'N'DROP** de sphères de couleur sur les emplacements des sphères. Cette solution implique de faire des raycast vers les sphères, de faire un drag (lier la sphère au raycast) et de prendre en compte la couleur si on la lâche sur un emplacement.

La dernière proposition est celle qui se rapproche le plus de la version physique du jeu.



## MASTERMIND en AR :

### Éléments de réponse par fonctionnalités :

- Le joueur doit avoir une interface pour demander la vérification de sa ligne

Comme précédemment, voici des propositions par ordre de difficulté à coder :

- **1 boutons UI** qui demande à faire la vérification de la ligne en cours. pas de difficulté particulière.
- **1 bouton 3D**. Ici, l'idée est de toucher un bouton 3D visible sur le plateau de jeu. En actionnant ce bouton, on demande la vérification. Cette solution nécessite un raycast à partir de l'endroit où on touche pour cliquer sur le bouton.

Selon le choix fait à la fonctionnalité précédente on va choisir la solution adaptée. L'ui pour continuer en UI ou le raycast pour continuer sur le plateau 3D. La demande de vérification ne fera la vérification QUE si toute la ligne a été préparée, s'il manque une ou plusieurs la vérification ne se fera pas et indiquera l'erreur !



## MASTERMIND en AR :

### Éléments de réponse par fonctionnalités :

- Le jeu doit indiquer par des pions de couleur pour chaque éléments :
  - s'il est de la bonne couleur et à la bonne place,
  - s'il est de la bonne couleur mais pas à la bonne place
  - s'il n'est pas de la bonne couleur et pas la bonne place

Nous sommes dans une autre partie qui ne change pas par rapport à une version 3D. La vérification a lieu et teste la même chose. On affiche les pions à côté de la ligne comme dans le jeu réel.

L'AR ne change rien.



### MASTERMIND en AR :

#### Éléments de réponse par fonctionnalités :

- Si la combinaison est la bonne le jeu doit indiquer que le joueur a gagné !
- Si au bout de 12 essais le joueur ne trouve pas la bonne réponse, il a perdu et le jeu doit lui dire
- Quand la partie est finie (par une victoire ou une défaite, le jeu propose de recommencer

Comme pour l'étape précédente, l'AR ne change rien. On affiche un message en UI sur le canva en overlay de la caméra avec un bouton restart.



## MASTERMIND en AR :

### Fonctionnalités du jeu avancée :

- Ecran de démarrage avec la possibilité de choisir :
  - le nombre de sphère par ligne
  - le nombre de couleur possible
  - Le nombre d'essais maximum
- Créer un système où l'on peut continuer de jouer s'il reste des essais dans une nouvelle configuration
- Créer un système de high score avec le nom de la personne ayant réussi en le minimum d'essais



## MASTERMIND en AR :

### Éléments de réponse :

Ici, on rend la création du plateau de jeu modulaire. On choisit des valeurs initiales, et on s'en sert pour définir le plateau de jeu. Les différents systèmes de jeu doivent tenir compte de la modification.

Comme pour la suite, il faut un système de persistance des données. On peut utiliser indépendamment les players prefs (peu de valeur à conserver et peu d'impact sur le jeu) ou un scriptable object.

On préférera le scriptable object, pour les settings In game ou les persistances entre les scène (première et deuxième fonctionnalités avancées) et les playerprefs pour sauvegarder le meilleur score (seulement deux valeurs, le nom et le score, et simplicité de la sauvegarde).



**MASTERMIND en AR :**

**BRAVO !**

**VOUS AVEZ UN PROJET SYMPA À PRÉSENTER !**

**IL NE VOUS MANQUE PLUS QU'UN TÉLÉPHONE QUI  
ACCEPTE L'APPLICATION !**