

Rapport SAE41_2022

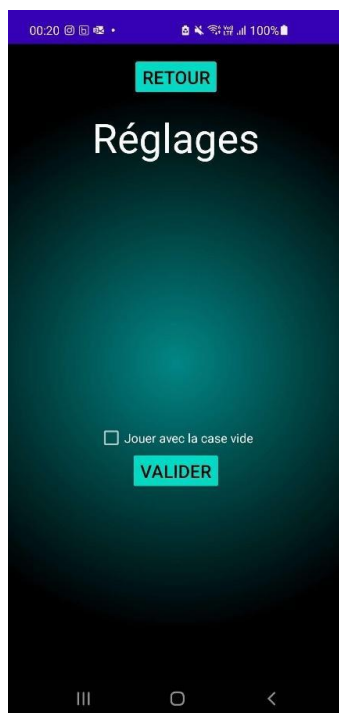
Lors de cette SAE nous avons eu pour objectif de concevoir une application ou plutôt un jeu pour les appareils Android à l'aide du langage de programmation JAVA. Le jeu se nomme Mastermind, rappelons les règles du jeu, il y'a d'un coté un défenseur qui doit choisir une série de 4 couleurs qui forme une combinaison secrète, d'un autre coté il y a un attaquant qui doit trouver la combinaison et pour cela il vas choisir une série de couleur que l'attaquant vas ensuite vérifier, lorsqu'une couleur ne figure pas dans la combinaison l'attaquant ne fait rien en revanche si une couleur figure sur la combinaison mais qu'elle est mal placé alors il ajoute un pion blanc dans la série d'indice et si elle est bien placé il ajoute un pion noir. L'ordre de cette série d'indices n'est pas représentatif de la combinaison que le défenseur a essayée il doit trouver l'ordre par lui-même. Notez que les meilleurs joueurs n'ont jamais besoin de plus de 6 coups pour gagner.

Fonctionnalités du jeu :

Menu principale



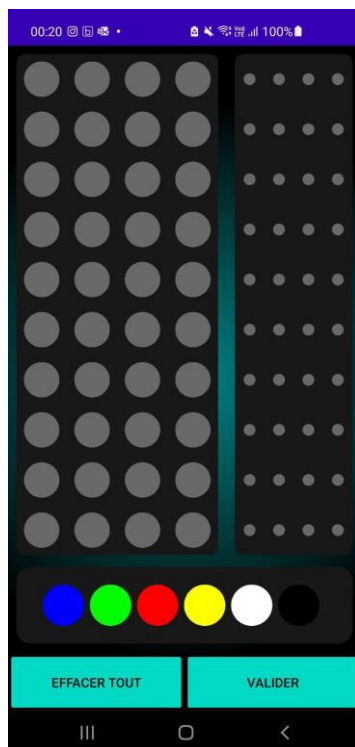
Option de réglage qui permet de faire le choix d'activer ou non les cases vides



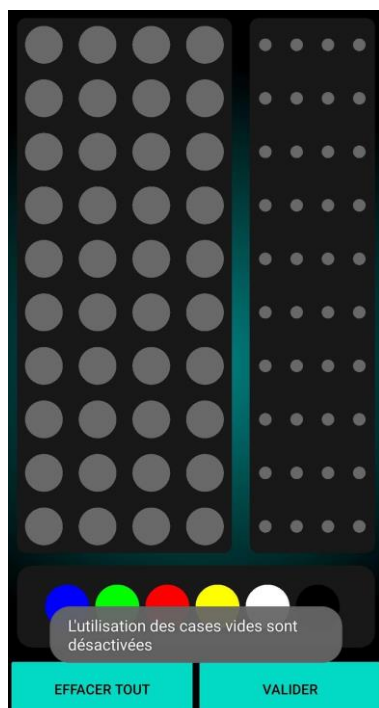
Indication de l'état de l'option pour jouer avec des case vide



Affichage du jeu avec la possibilité de valider (la validation n'est pas automatique pour éviter les fautes de frappe) ainsi que la possibilité d'effacer lorsque l'attaquant veut repartir sur de bonne base



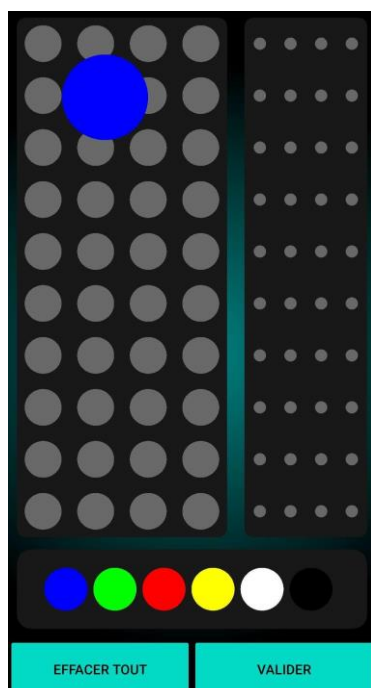
Message d'erreur lorsque des cases sont laissé vide et que l'option est désactivée



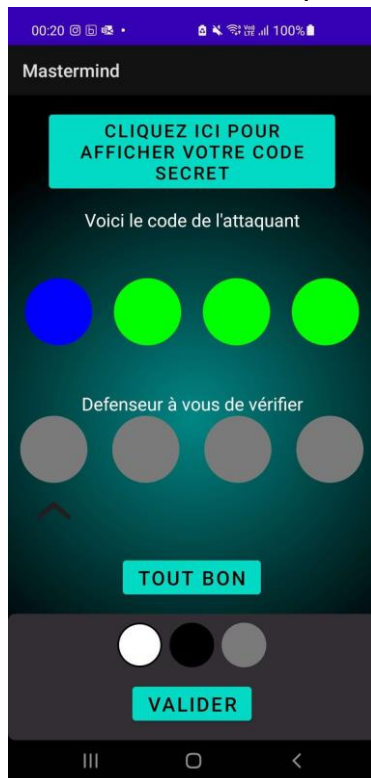
Choix de la combinaison du défenseur durant le mode hot seat (2 joueurs) avec une flèche d'affiché pour indiquer l'emplacement du choix



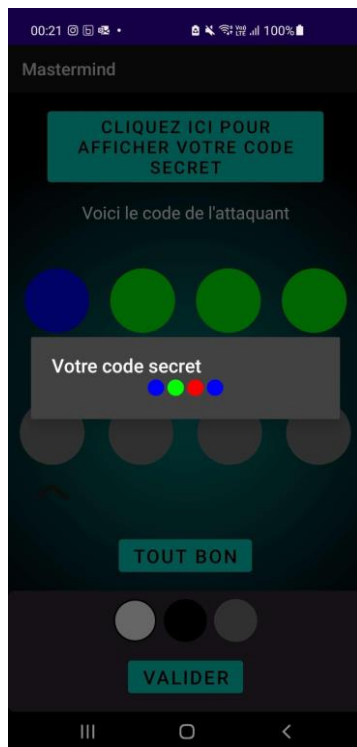
Option drag and drop des pions pour jouer



Côté défenseur possibilité de choisir si tout est bon afin de finir le jeu ou alors de sélectionner les pions qui serviront d'indice à l'attaquant



toujours côté défenseur possibilité d'afficher dans une pop-up la combinaison choisie en cas d'oubli



Nous avons fait le choix d'utiliser 2 méthodes de sélection des couleurs lors du choix de la combinaison 1 seule clique suffit et il y a une sorte de navigation entre les pions, durant la partie nous avons utilisé le drag and drop, et pour le défenseur lorsqu'il doit choisir les indices nous avons repris l'idée du choix de la combinaison. Nous avons décidé cela avant de savoir que ce n'est pas ce que vous attendiez vraiment de notre part nous pensions bien faire en utilisant le plus de chose possible que nous avons pu voir en cours

Structure du programme :

Explication d'un algorithme :

L'algorithme de correction fonctionne en comparant chaque élément de la supposition avec chaque élément de la réponse pour déterminer si la couleur de chaque case est présente dans la réponse, un pion blanc est placé si la couleur est présente dans la réponse, et un pion noir si la couleur est à la bonne place dans la supposition. Afin d'éviter qu'une même case du bon code ne puisse donner l'information à deux cases de la supposition que la couleur est ailleurs, j'ai créé un tableau pour enregistrer les cases déjà utilisées et indiquer que la couleur est ailleurs.

Cependant, il peut arriver qu'une case qui a été utilisée pour indiquer que la couleur est ailleurs se trouve en face de la même couleur dans la correction. Dans ce cas, on utilise la valeur stockée dans la table qui enregistre les utilisations des cases du bon code, afin de renvoyer au numéro de la case de la correction à nullifier.

Enfin, pour éviter de donner des informations au joueur sur l'emplacement des possibles bonnes cases, j'ai organisé la correction dans l'ordre suivant : noirs, blancs, nulls. Cela signifie que les pions noirs sont affichés en premier, suivis des pions blancs, puis des cases qui n'ont pas été utilisées. Ce processus permet d'assurer que la correction ne révèle pas l'emplacement des bonnes cases au joueur.

Conclusion personnelle :

Victor :

Lors de ce projet, je me suis principalement occupé de la partie Back-End, notamment de l'algorithme permettant d'évaluer une combinaison candidate, mais aussi de la page pour donner les codes de défense au début de la partie. Je ne me souviens pas d'avoir eu de grosses difficultés, et ce projet est mon préféré de cette année. Tout d'abord, car pour moi, créer un jeu est plus plaisant que de créer un site internet, et ensuite, parce que j'ai pu travailler en groupe.

Yanis :

Curieux de savoir comment était conçu les Applications/Jeux Android j'ai toujours voulu apprendre à en développer un/une et ce projet a été ma première expérience dans ce domaine que j'ai finalement trouvé passionnant. Le fait d'y avoir travaillé en équipe je me suis occupé du choix du design ainsi que la correction de bug et quelques modifications sur certaines fonctionnalités. Ayant touché au développement d'application dans d'anciens projets j'ai pour la première fois été touché à un projet Android et cela a renforcé mon idée de poursuivre mes études dans ce domaine.

Rayan :

Lors de ce projet je me suis attelé à la partie ergonomie du jeu, notamment à l'option drag and drop permettant d'interagir avec le plateau. Ce projet m'a permis de mieux comprendre le cycle de vie d'une activité et les échanges de données entre elle. J'ai aussi pu découvrir le logiciel Figma qui m'a fait gagner énormément de temps sur la partie front-end de l'application. Cela nous a permis de nous mettre d'accord sur le choix d'emplacement des certains boutons ou d'autres choix ergonomiques.

Je peux donc conclure que ce projet a été très agréable à réaliser pour cette fin d'année.

Diagramme de classe

