



Un jeu de memory pour enfant

Table des matières

| | |
|--|------------------|
| <i>Présentation</i> | <i>3</i> |
| <i>Objectifs</i> | <i>3</i> |
| <i>Fonctionnement de l'application</i> | <i>4</i> |
| <i>Difficultés rencontrées</i> | <i>6</i> |
| Compteur de clics..... | 6 |
| Récupérer le nom de l'image d'une carte spécifique..... | 6 |
| La musique..... | 7 |
| Les conflits de clics sur les cartes..... | 7 |
| <i>Partage des tâches entre les membres du binôme</i> | <i>7</i> |
| <i>Perspectives d'améliorations.....</i> | <i>7</i> |
| <i>Sources.....</i> | <i>10</i> |

Présentation



Figure 1 - Dos de carte MemoJungle



Figure 2 - Carte du tigre



Figure 3 - Chronomètre tigre

Nous avons conçu un jeu de Memory avec pour thème les animaux, offrant un design coloré et ludique. Ce jeu se nomme MemoJungle, il est accompagné d'une musique entraînante pour divertir les joueurs. Les niveaux sont relativement simples, adaptés dès l'âge de 3 ans, et les animaux sont représentés de manière amicale. Nous avons même mis en place une page avec les règles du jeu, mais en poésie, ajoutant une petite touche créative. Le poème a été généré par ChatGPT. Notre mascotte est le tigre, roi de la jungle. Son visage se trouve sur le chronomètre. Nous l'avons conçu avec soin sur Photoshop.

À chaque découverte d'une paire de cartes identiques, une sonnerie de cloche résonne et les cartes scintillent en arc-en-ciel, ajoutant une dimension sensorielle amusante. De plus, il y a un effet sonore à chaque fin de partie, que l'on gagne ou que l'on perde, procurant une expérience relaxante et apaisante, dénuée de toute compétition.

Objectifs

Le but du jeu est de retrouver des paires de cartes dans un temps imparti. L'application nous présente une grille de 12 cartes d'animaux ou d'insectes. Il y a en tout une variété de 12 types d'animaux et d'insectes. La disposition et les cartes choisies sont générées aléatoirement à chaque début de partie. Les joueurs peuvent choisir parmi plusieurs niveaux de difficulté, allant de Facile à Expert, avec des durées de jeu variant de 4 minutes à 30 secondes. Nous avons pris en compte l'âge des utilisateurs, ainsi, même le niveau Expert reste accessible.

Pour compléter notre projet, nous avons sollicité l'aide de l'IA ChatGPT pour les aspects que nous n'avions pas encore abordés en cours, comme l'animation ou le son. Nous l'avons également utilisé pour obtenir des explications sur le fonctionnement d'un minuteur ou sur la mise en place d'une image de fond en ligne de commande. Nous ne parvenions pas à poser d'image d'arrière-plan en drag-and-drop.

Dans l'ensemble, nous considérons que nos objectifs initiaux ont été atteints. Nous avons développé un jeu de Memory fonctionnel, accompagné d'une page explicative des règles du jeu et proposant différents niveaux de difficulté en termes de temps.

Cependant, nous rencontrons actuellement un problème avec la fonction d'arrêt du son en fin de partie. Bien que cette fonction ne soit pas encore opérationnelle, la musique d'ambiance continue de jouer, jusqu'à ce que ce problème soit résolu.

Fonctionnement de l'application

Nous avons réussi à installer et à exécuter notre application sur nos téléphones respectifs, un iPhone 13 et un iPhone 14 Pro Max. Nous avons décidé de nous concentrer sur une seule taille d'écran pour cette version. Pour illustrer le fonctionnement de notre application, nous avons inclus une vidéo démonstrative en annexe.

Dans le développement de notre application, nous avons utilisé des bibliothèques pour différents aspects. Notamment, la bibliothèque UIKit a été employée pour gérer les animations, et la bibliothèque AVFoundation a été utilisée pour la gestion du son.

Pour expliquer plus en détails comment marche notre application, comme dit précédemment, l'objectif principal est de retrouver les 6 paires de cartes dans un laps de temps défini en fonction du niveau choisi. Au début de la partie, toutes les cartes sont disposées face cachée, et le minuteur démarre dès que le niveau est sélectionné. Lorsqu'un joueur clique sur une carte, une animation de retournement se déclenche. Lorsqu'une deuxième carte est retournée, qu'elle soit identique ou non à la première, les autres cartes deviennent temporairement indisponibles pendant un cool down de seconde.

Durant cette petite seconde, toute tentative de clic sur une autre carte est inefficace, permettant aux joueurs de se concentrer sur les 2 cartes qu'il vient de retourner. De plus, ce laps de temps est consacré à l'animation de ces deux dernières. Si elles ne sont pas identiques, elles se retournent face cachée. Sinon, un effet sonore de sonnette retentit accompagné d'une animation de brillance avec un effet de zoom in. Cette animation de brillance est rendue possible par une succession de changements de couleur très rapide sur la carte.

Bien que cette mécanique puisse ralentir le joueur d'une seconde (étant donné que le jeu est chronométré) nous avons fait ce choix délibérément. Nous pensons qu'autoriser un enchaînement rapide des clics sur d'autres cartes aurait pu ôter le côté "prendre son temps" du jeu. Notre intention est de créer une expérience relaxante et décontractée, sans pression.

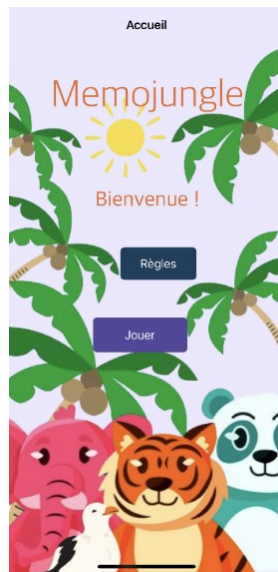


Figure 4 - Capture d'écran sur Iphone 13 de la page Accueil



Figure 5 - Capture d'écran sur Iphone 13 de la page Règles

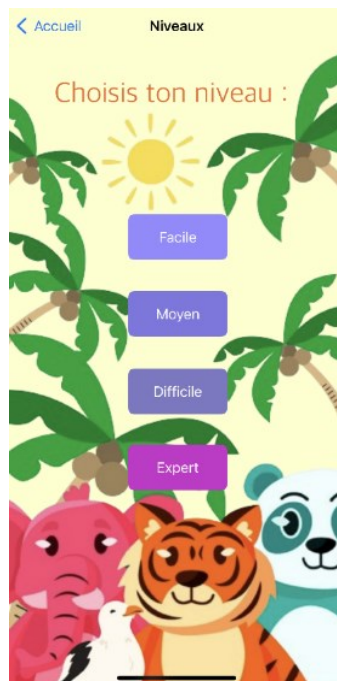


Figure 6 - Capture d'écran sur Iphone 13 de la page Niveaux



Figure 7 - Capture d'écran sur Iphone 13 des cartes cachées de la page Jeu



Figure 8 - Capture d'écran sur Iphone 13 d'une animation de cartes araignée sur la page Jeu



Figure 9 - Capture d'écran sur Iphone 13 de l'alerte de l'échec

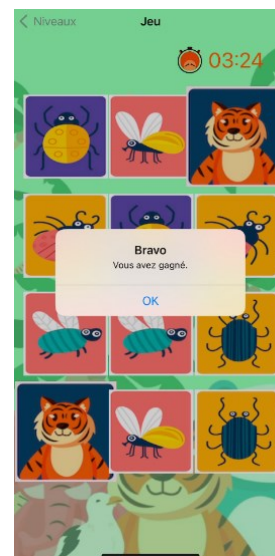


Figure 10 - Capture d'écran sur Iphone 13 de l'alerte de réussite

Difficultés rencontrées

Compteur de clics

Puisque c'est un memory, au démarrage, nous devons accorder une attention particulière au nombre de cartes qui sont à comparer. Ainsi, nous avons introduit une variable de comptage, fixée à 2. Toutes les deux cartes, on analyse les cartes. La partie difficile résidait dans la nécessité d'organiser correctement les conditions dans la fonction de clic, sans se perdre.

Récupérer le nom de l'image d'une carte spécifique

Initialement, nous avons disposé nos cartes de la même manière que nous l'avions fait pour les boutons du clavier dans le jeu du 'Pendu'. Au départ, nous avons envisagé de récupérer le nom de l'image du bouton en supposant que nous pourrions utiliser l'attribut par défaut dans les propriétés.

Cependant, il s'est avéré que cette méthode ne nous permettait pas de positionner les cartes de manière aléatoire. De plus, il était difficile d'extraire le nom de l'image. Heureusement, il était beaucoup plus facile de modifier directement l'image d'un bouton.

Finalement, nous avons opté pour un tableau de boutons nommé `Cartes`, et un tableau associatif `tabImagePosition` où la clé est le tag du bouton et la valeur est le nom de l'image. Ainsi, `tabImagePosition` nous a servi de repère pour le positionnement aléatoire des cartes.

La musique

Comme dit plus haut, nous avons rencontré des difficultés à interrompre la musique de fond une fois qu'une partie, qu'elle soit gagnée ou perdue, était terminée. Pour remédier à cela, nous avons rendu la variable de la musique de fond statique afin que les deux contrôleurs, à savoir celui de la vue de la première page et celui de la vue de la partie de jeu (respectivement *ViewController* et *MemoryViewController*), puissent y accéder et la stopper.

Bien que la musique se soit arrêtée, ce n'était pas au moment opportun, car elle s'arrêtait de jouer juste après avoir cliqué sur le bouton "OK" du pop-up qui s'affiche lorsque la partie est finie. Finalement, nous avons choisi de le laisser à l'état initial.

Les conflits de clics sur les cartes

Nous considérons que lorsque deux cartes étaient cliquées (et en animation pour se révéler) les autres cartes ne devaient pas être cliquables. Sinon, cela entraînerait un conflit avec les actions suivantes définies dans le code.

Pour illustrer, seulement l'analyse de deux cartes étaient nécessaires pour vérifier si c'est une paire. Nous avons donc dû modifier le paramètre "*enabled*" à *false*, des cartes qui sont face cachée, pour les empêcher d'être cliquables et donc retournables.

Un autre exemple est que lorsque l'on retourne deux cartes différentes et que l'animation pour les remettre en face cachée est déclenchée, il était aussi nécessaire d'installer un cool down d'une seconde pour suspendre toutes les autres actions en attente.

Partage des tâches entre les membres du binôme

Nous avons consacré un total de 17 heures à ce projet. En ce qui concerne notre méthode de travail, nous avons toujours collaboré dans les salles de travail, assises côte à côte. Cette proximité nous permettait de partager nos idées et d'organiser notre travail de manière fluide.

Pendant que l'une d'entre nous disposait les différents éléments de notre jeu, l'autre redimensionnait les images pour les adapter à la taille des boutons. Ou bien, pendant qu'une personne tapait du code, l'autre effectuait des recherches. Nous échangeons régulièrement nos rôles pour favoriser une répartition équilibrée des responsabilités.

Lorsque nous rencontrions des bugs, nous nous réunissions pour comprendre leur origine et travailler ensemble à les résoudre.

Perspectives d'améliorations

Si nous avions disposé de plus de temps, nous aurions pu intégrer plusieurs fonctionnalités. Tout d'abord, nous pourrions intégrer un système de record à battre basé sur le temps de jeu du joueur, offrant ainsi aux joueurs un défi supplémentaire et les incitant à améliorer leurs

performances, pour plus de challenge. De plus, plutôt que de définir les difficultés de niveau uniquement en fonction du minuteur, nous pourrions envisager une approche basée sur le nombre de cartes, permettant ainsi une progression plus cohérente et adaptable.

Aussi, il faut noter que nous avons fait une erreur d'inattention : dans le fichier *MemoryViewController.swift*, dans la variable *nomCartes*, qui comporte les noms des images de cartes disponibles, nous avons écrit deux fois *"coccinelle.jpg"*. Nous aurions dû écrire *"coccinelle2.jpg"*. Le fichier de l'image se trouve bien dans le dossier des assets mais nous avons juste oublié le '2'.

```
var nomsCarte : [String] = [  
    "coccinelle.jpg",  
    "elephant.jpg",  
    "mouche.jpg",  
    "panda.jpg",  
    "mouche2.jpg",  
    "mouche3.jpg",  
    "punaise.jpg",  
    "tigre.jpg",  
    "singe.jpg",  
    "scarabee.jpg",  
    "araignee.jpg",  
    "coccinelle.jpg",  
]
```

Capture d'écran du code avec le doublon *"coccinelle.jpg"*

Aussi, nous pourrions améliorer l'expérience sonore en faisant en sorte que la musique s'arrête correctement en fin de partie, offrant ainsi une conclusion plus fluide à l'expérience de jeu.

Du côté de l'interface graphique, plusieurs ajustements pourraient être apportés pour la rendre plus attrayante et personnalisable. Par exemple, nous pourrions introduire des thèmes d'images variés, tels que des animaux de la ferme, des animaux aquatiques, des chiffres ou des métiers. On peut même permettre aux joueurs d'adapter les couleurs selon leurs préférences.

En ce qui concerne les animations, nous pourrions explorer différentes façons de rendre les cartes plus dynamiques et captivantes. Par exemple, animer l'image des animaux sur les cartes, les faire faire une action : saluer, sourire, sauter, parler, aider le joueur à trouver une carte en pointant du doigt une autre carte, etc.

Ensuite, nous pourrions améliorer la visibilité des boutons de cartes en les mettant au premier plan lorsqu'ils se retournent. Pour l'instant, ils sont en arrière-plan. Pour adoucir l'apparence des cartes, on pourrait arrondir leurs coins pour un aspect plus esthétique.



Figure 11 - Capture d'écran de l'animation d'une paire de cartes identique

Enfin, pour garantir une expérience fluide sur tous les types d'appareils, nous pourrions adapter les images et la disposition pour prendre en compte les écrans de taille variable, assurant ainsi une jouabilité optimale. Ces perspectives d'améliorations nous permettraient de rendre notre jeu encore plus attractif et adapté aux attentes des joueurs.

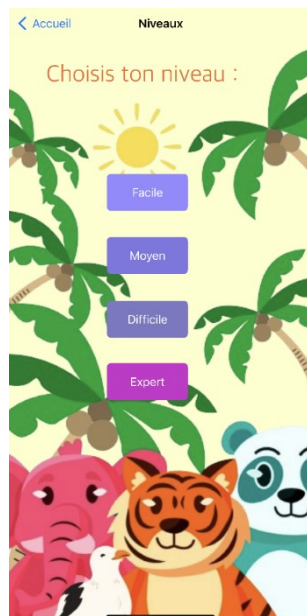


Figure 12 - Capture d'écran sur Iphone 14 Pro Max sur la page Niveaux

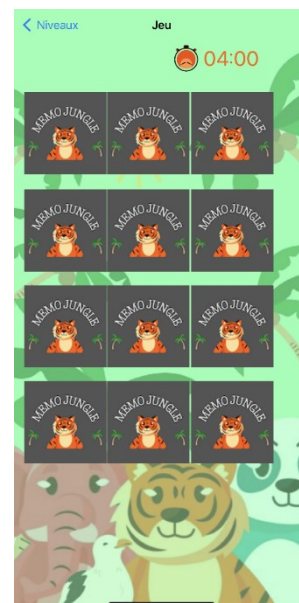


Figure 13 - Capture d'écran sur Iphone 14 Pro Max sur la page Jeux



Figure 14 - Capture d'écran sur Iphone 13 de la page d'accueil en mode Paysage

Sources

Images de jeu de cartes des animaux prises sur Freepik

https://fr.freepik.com/vecteurs-libre/carte-jeu-memoire-dessinee-main_37575459.htm#fromView=search&page=1&position=1&uuid=c2203f74-7f2a-4eda-b28b-c9f0debcd1f4

Images de jeu de cartes des insectes prises sur Freepik

https://fr.freepik.com/vecteurs-libre/cartes-jeu-memoire-dessinees-main_37451761.htm#fromView=search&page=1&position=9&uuid=c7cbb00a-36d2-4aeb-8a72-3a9d1747c033

Musique d'ambiance libre de droit : Ehrling - You And Me

https://www.youtube.com/watch?v=ZReJpek182U&t=1s&ab_channel=NoCopyrightBackgroundMusic

Musique de fin de partie réussite, d'échec et de paire trouvée

https://www.youtube.com/watch?v=GGoYBJvPEdY&ab_channel=MemeFX