

Rapport du Projet

Jeu de Cartes Marocain en C++ et Qt



Encadrée par : Pr. Ikram Ben Abdel Ouahab

Réalisée par :

- Amal Cheg dali
- Oumaima El Khlifi



Université Abdelmalek Essaadi

Faculté des Sciences et Techniques de Tanger

Année académique 2023-2024

**Licence Ingénierie de développement d'application
informatique**

Module : Programmation Orientée Objet en C++

Project : Jeu de Cartes Marocain en C++ et Qt

****Rapport du Projet ****

LST-IDAI-S5

Encadré par : Pr. Ikram Ben Abdel Ouahab

Mr. Lotfi Aachak

Réalisée par :

Amal Chegda

Email: chegdali.amal@etu.uae.ac.ma

Oumaima El Khlifi

Email: elkhlifi.oumaima@etu.uae.ac.ma

Table de Matière

1

Introduction

2

Règles du jeu

3

Outil utilisés

4

Objectifs

5

Développement et test du jeu


6

Conclusion

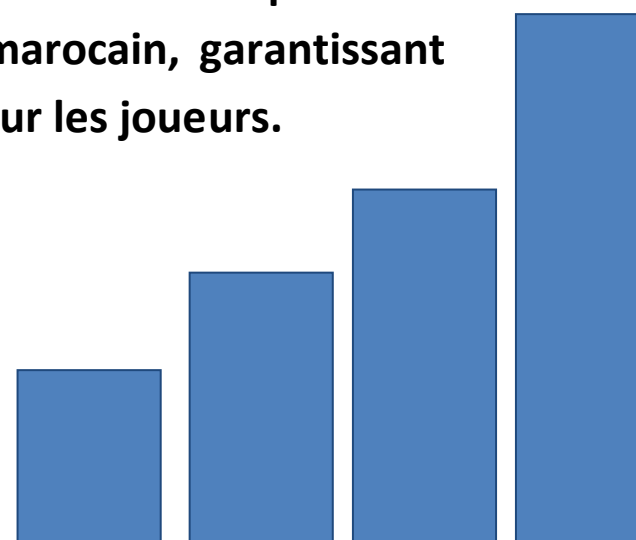


Introduction

Dans le cadre de ce projet ambitieux, l'objectif principal est de créer un jeu de cartes marocain en exploitant les principes fondamentaux de la programmation orientée objet (POO) avec le langage C++. Cette approche permettra de structurer le code de manière modulaire, favorisant ainsi la maintenance, la compréhension et l'extension du projet. L'ajout d'une interface graphique conviviale sera assuré grâce au Framework Qt, offrant une expérience utilisateur moderne et ergonomique.

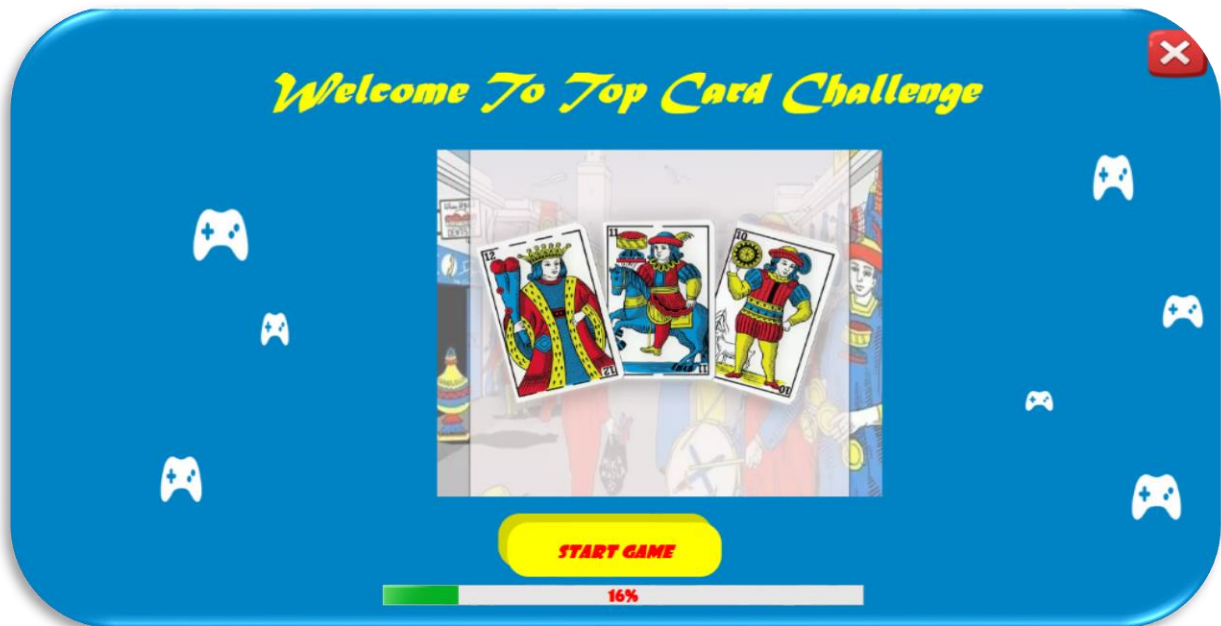


Le jeu sera conçu pour accommoder au moins deux joueurs, offrant ainsi des affrontements captivants entre un joueur humain, représenté par l'utilisateur, et un adversaire automatisé ou aléatoire incarné par l'ordinateur. Cette dualité promet une expérience de jeu riche, où les compétences stratégiques de l'utilisateur seront mises à l'épreuve face à un adversaire virtuel doté de comportements automatisés ou aléatoires. L'implémentation du jeu sera minutieusement adaptée aux règles spécifiques du jeu de cartes marocain, garantissant une expérience fidèle et immersive pour les joueurs.



Welcome To Top Card Challenge

Bienvenue dans l'univers de notre jeu de cartes marocain



Vision

Une expérience ludique qui célèbre la riche culture et la vivacité du Maroc. Chaque carte de ce jeu est imprégnée de l'esprit dynamique du Royaume.

Mission

Préparez-vous à plonger dans un monde de stratégie, d'intrigue et de convivialité, où chaque distribution de cartes devient une célébration de la richesse culturelle du Maroc.

Règle du jeu



Un jeu de cartes de 40 cartes est généralement

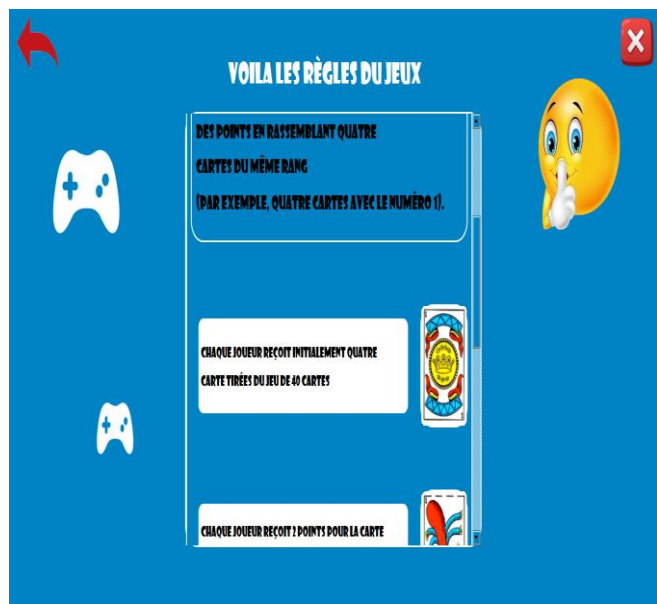
utilisé pour le Ronda, avec les cartes numérotées de (1,2,3,4,5,6,7,10,11,12) et trois figures (sota, caballo, rey).

Les joueurs pourraient marquer des points en rassemblant Quatre cartes du même rang (par exemple, quatre cartes avec le numéro 1).

Chaque joueur reçoit initialement Quatre cartes tirées du jeu de 40 cartes

Chaque joueur reçoit 2 points pour la carte de sa main ayant la plus grande valeur parmi les quatre

À la fin de chaque manche, les joueurs comparent leurs scores totaux, et le joueur ayant le score total le plus élevé est déclaré vainqueur.

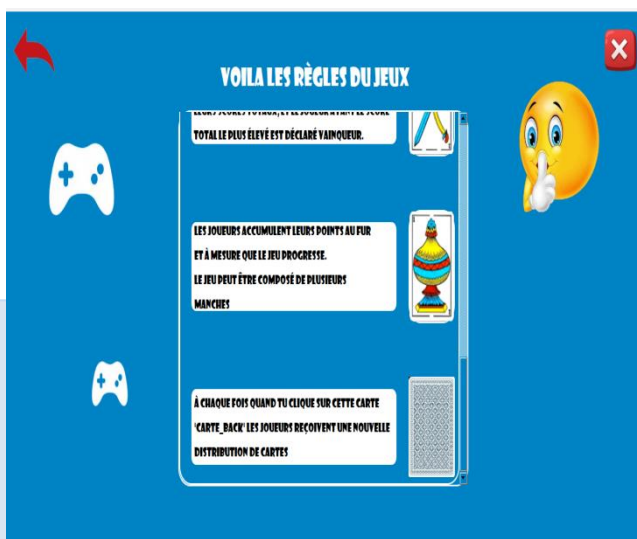


Les joueurs accumulent leurs points au fur et à mesure que le jeu progresse.

Le jeu peut être composé de plusieurs manches.

À chaque fois quand tu cliques sur la carte 'carte_back'

Les joueurs reçoivent une nouvelle distribution de cartes



Conception et Implémentation d'un Jeu de Cartes Marocain en C++ avec Qt

Une Exploration Pratique du

Développement d'Applications de Jeux

Outil et Technique utilisé



Le jeu doit disposer d'une interface graphique conviviale permettant à l'utilisateur de visualiser les cartes en main, les cartes jouées, et l'état du jeu.

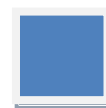


Les actions de jeu (prendre, jouer une carte, annoncer) doivent être facilement accessibles



Utilisation de structures de données appropriées pour représenter les cartes et leur état (en main, jouées, prises).

Utilisation du langage C++ pour implémenter la logique du jeu.



Utilisation de la programmation orientée objet (POO) pour concevoir une architecture modulaire et extensible du jeu.

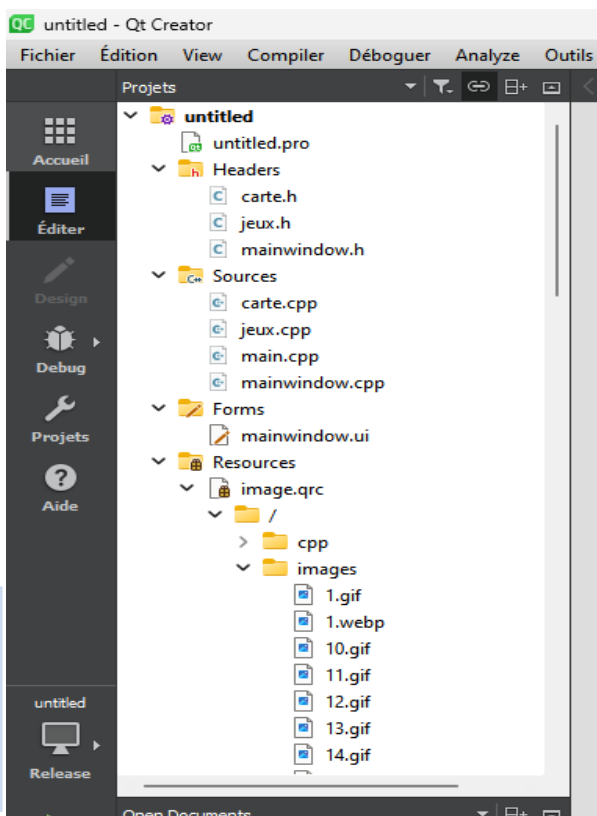
■ Objectif

Notre projet consiste à réaliser un jeu de cartes marocain. Nous créons une interface qui comprend plusieurs pages d'affichage, telles que la page permettant de choisir le niveau de jeu, la page de connexion, et la page de démarrage du jeu. En fin de partie, nous affichons le gagnant et lui offrons la possibilité de rejouer.



■ Editeur

Pour réaliser ce projet, nous avons créé trois classes : la classe Carte, Deck et MainWindow. La classe Carte contient trois attributs : nom, valeur et image de la carte, ainsi que des méthodes getters et setters pour récupérer le numéro et le nom de la carte, ainsi qu'un constructeur. La classe Deck est conçue pour stocker 40 cartes, les mélanger et tirer une carte



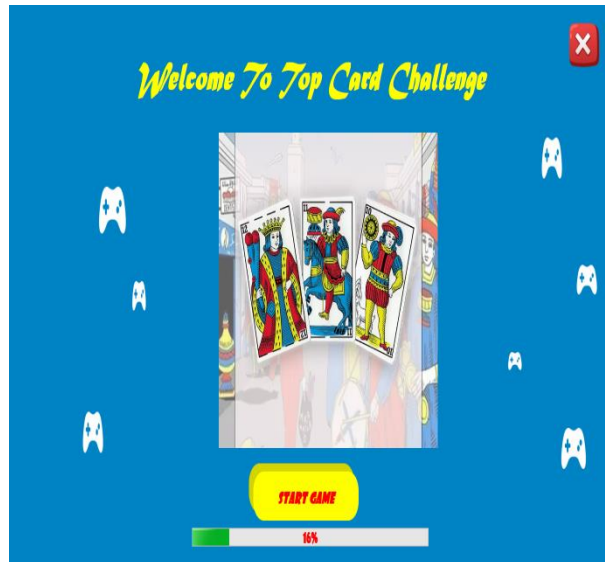
Développement et Test du jeu

Step1

La page de garde du jeu est constituée d'un titre de jeu «Top Card Challenge», d'un bouton et d'une image qui décrit qu'il s'agit d'un jeu marocain.

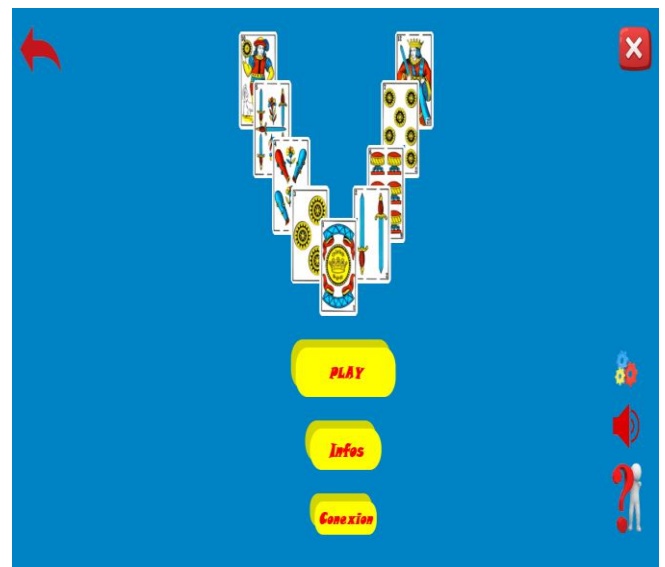
En cliquant sur le bouton 'Play', nous pouvons passer à la deuxième page qui est dédiée au lancement du jeu, aux paramètres, aux informations sur le jeu et à la connexion.

Pour effectuer ce passage, En utilisant le code `ui->stackedWidget->setCurrentIndex(0);` où `setCurrentIndex` nous permet de spécifier la page que nous souhaitons afficher.



Step2

Quand le joueur clique sur "Connexion", il peut saisir son prénom et son nom, qui s'affichent ensuite sur la page de distribution des cartes.



Step3

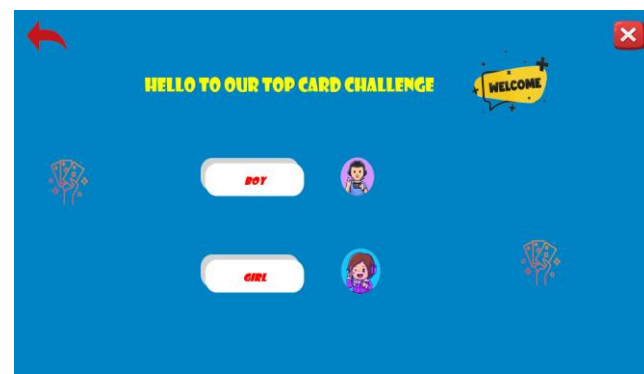
Quand nous cliquons sur 'Connexion', nous revenons à la page précédent.



Step4

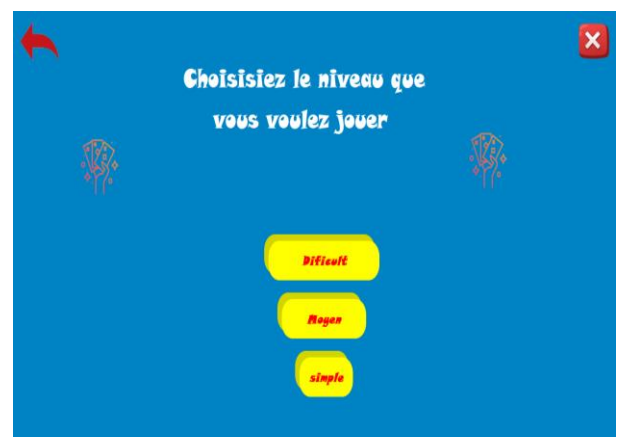
Ensuite, le joueur peut également choisir s'il veut jouer avec la fille ou le garçon.

Si, par exemple, il clique sur la fille, une image correspondante s'affichera sur la page de distribution des cartes.



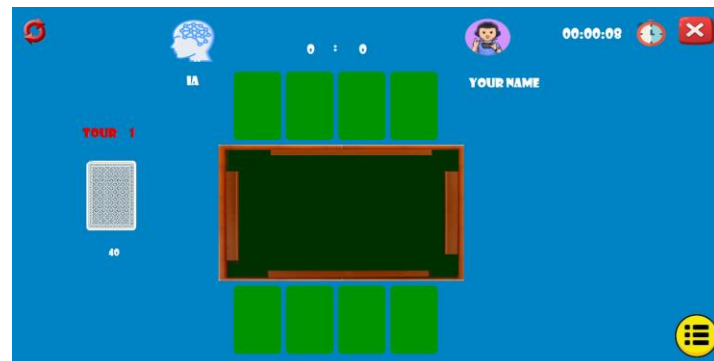
Step5

Le joueur va choisir le niveau de jeu il veut jouer, que ce soit facile, difficile ou moyen.



Step6

Dans la page de distribution des cartes, nous avons les noms des deux joueurs, à savoir l'être humain et l'IA, ainsi que le score de chaque joueur avec son image. Nous affichons également un chronomètre et les emplacements des cartes.



Step7

Lorsqu'on clique sur l'image, on distribue quatre cartes à chaque joueur, puis les joueurs commencent à jouer



Step8

Le premier joueur commence en jouant une carte. S'il a une carte avec un numéro supérieur à celle du joueur AI, il gagne deux points, et vice versa. En cas d'égalité de numéros de carte, les joueurs remportent chacun un point.

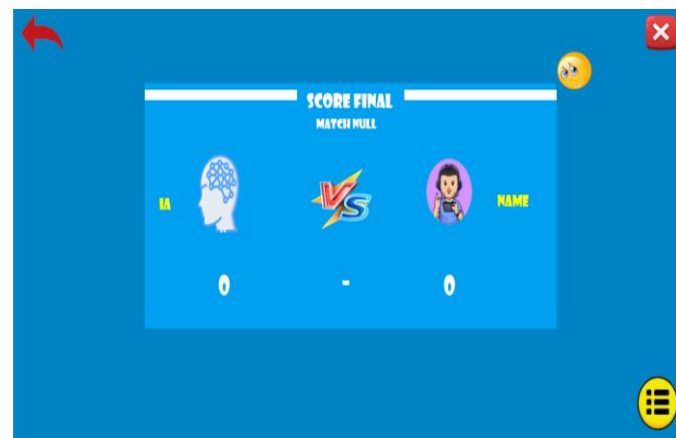


Match Null

Step9

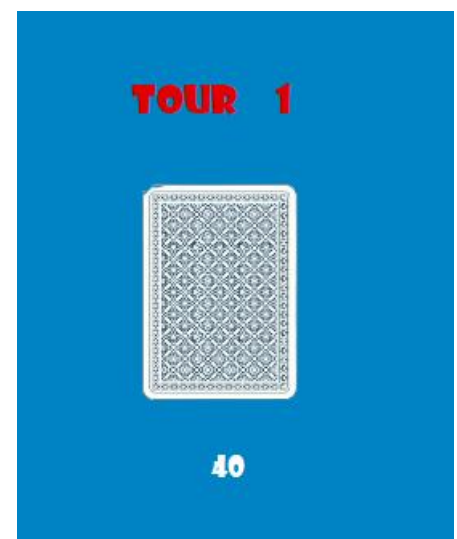
Si les deux joueurs ont le même score, on affiche une page qui contient les noms des joueurs et leurs scores

Si les deux joueurs ont le même score, on affiche une page qui contient les noms des joueurs et leurs scores. Ensuite, s'ils veulent rejouer ou quitter le jeu, ils doivent cliquer sur le menu



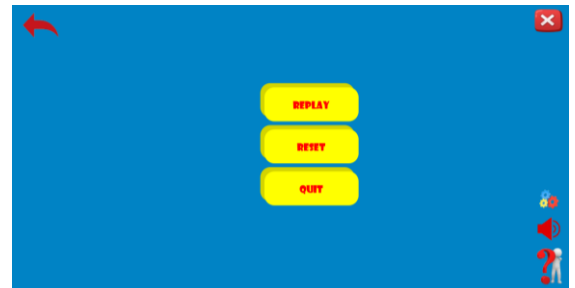
Remarque

Si l'on clique sur la carte chargée de distribuer les cartes, la distribution des nouvelles cartes ne débutera pas tant que l'animation de retour des cartes de la distribution précédente ne sera pas terminée.



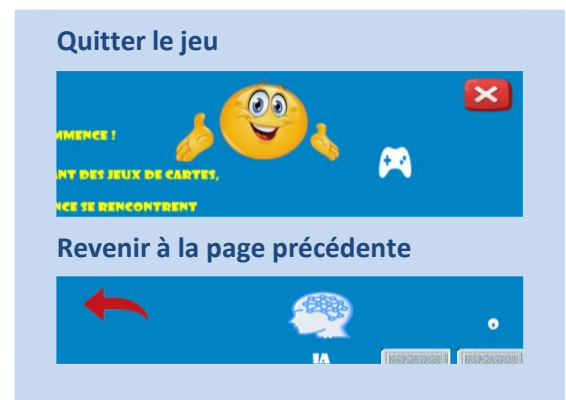
Step10

Sur la page du menu, trois options sont proposées au joueur : rejouer, quitter le jeu et recommencer la partie.



Step11

Sur chaque page de jeu, nous avons ajouté un bouton pour quitter le jeu et un autre bouton pour revenir à la page précédente.



Step12

Si un joueur a un score plus élevé que l'autre, il remporte la manche et a la possibilité de rejouer.

Pour la page du gagnant, on a mis le nom du gagnant, que ce soit l'IA ou l'être humain, ainsi que son image.



Fonctions du jeu

Le code de construction de l'interface utilisateur repose sur la bibliothèque Qt pour créer une interface graphique interactive. Les huit boutons (pushButton1 à pushButton8) représentent les cartes du joueur, tandis que des tableaux de boutons (boutonsCartes et boutonsCaches) sont utilisés pour symboliser les cartes du joueur et celles cachées de l'IA.

Les animations, déclenchées par les clics sur les boutons, sont gérées via la classe QPropertyAnimation, assurant des déplacements fluides des cartes vers de nouvelles positions. La distribution des cartes depuis le deck, orchestrée par on_btnAfficherCarte_clicked, est accompagnée d'animations révélant les cartes. La gestion des tours s'assure de la distribution complète des cartes et du nombre de tours joués. En cas de conditions remplies, la fonction rejouer réinitialise le jeu en remettant les cartes dans le deck, réinitialisant les scores, et réaffichant les boutons pour une nouvelle partie. Les fonctions des clics sur les boutons de l'IA simulent ses actions à travers des animations et des manipulations de cartes cachées. L'affichage du gagnant compare les scores des joueurs, tandis que l'utilisation de stackedWidget permet de basculer entre différentes du jeu et celui du gagnant.

1

Construction de l'interface utilisateur (UI):

Le code utilise la bibliothèque Qt pour construire l'interface graphique.

Il y a des boutons (pushButton1 à pushButton8) qui représentent les cartes du joueur.

2

Initialisation du jeu

Un tableau de boutons (boutonsCartes) est créé pour représenter les cartes du joueur.

Un tableau de boutons cachés (boutonsCaches) est créé pour représenter les cartes cachées de l'IA

3

Animations

Lorsqu'un bouton est cliqué (on_pushButton1_clicked, on_pushButton2_clicked, etc.), une animation de déplacement est déclenchée pour déplacer la carte vers une nouvelle position.

Les animations sont réalisées à l'aide de la classe QPropertyAnimation.

4

Activer le son et la musique

On a créé une fonction pour démarrer la musique lorsque les cartes sont distribuées.

5

Distribution de cartes :

La fonction `on_btnAfficherCarte_clicked` est utilisée pour distribuer les cartes.

Les cartes sont tirées du deck (deck) et associées aux boutons correspondants.

Des animations sont utilisées pour montrer les cartes.

La fonction `Tour` semble vérifier si toutes les cartes ont été distribuées et si le nombre de tours a été atteint. Si tel est le cas, elle appelle la fonction `rejouer` pour recommencer le jeu.

6

Comparaison de cartes :

La fonction `comparerNumerosCartes` compare les numéros de deux cartes et ajuste les scores des joueurs en conséquence

7

Gestion du score :

Les scores des joueurs (`scoreJoueur1` et `scoreAI`) sont maintenus et affichés à l'interface utilisateur.

8

Affichage du gagnant:

La fonction `afficherGagnant` compare les scores des joueurs et affiche le gagnant dans l'interface graphique.

9

Fonctions pour les clics sur les boutons de l'IA:

Les fonctions `on_pushButton5_clicked`, `on_pushButton6_clicked`, etc, simulent les actions de l'IA en effectuant des animations et en cachant les cartes. En conséquence.

10

Fonctions de rejouer:

La fonction `Tour` semble vérifier si toutes les cartes ont été distribuées et si le nombre de tours a été atteint. Si tel est le cas, elle appelle la fonction `rejouée` pour recommencer le jeu.

11

Fonctions de rejouer:

La fonction `rejouer` réinitialise le jeu, remet les cartes dans le deck, réinitialise les scores et réaffiche les boutons pour une nouvelle partie.

12

Basculement entre les écrans :

L'utilisation de `stackedWidget` permet de basculer entre différentes pages de l'interface, probablement pour afficher l'écran du jeu et l'écran du gagnant

Conclusion

Le jeu de cartes développé présente une mise en œuvre interactive et visuellement attrayante de la distribution et de la comparaison de cartes entre un joueur humain et une intelligence artificielle (IA). Voici les principaux points à retenir :

- 1. Mécanique de Jeu :** Le jeu offre une expérience de jeu basée sur la comparaison des valeurs de cartes, attribuant des points aux joueurs en fonction des résultats. La distribution aléatoire des cartes et l'utilisation d'animations contribuent à rendre le jeu plus captivant.
- 2. Animations Visuelles :** L'intégration d'animations pour déplacer les boutons de cartes ajoute un aspect visuel agréable, améliorant l'expérience utilisateur. Cela renforce l'immersion dans le jeu et rend les actions des joueurs plus dynamiques.
- 3. Gestion des Boutons :** La désactivation temporaire des boutons pendant les animations est une approche judicieuse pour éviter les clics répétitifs. Cependant, il est important de s'assurer que la réactivation des boutons est correctement gérée pour maintenir le flux du jeu.
- 4. Comparaison des Cartes :** La fonction de comparaison des cartes semble effectuer la logique de manière adéquate, attribuant des points en fonction des valeurs des cartes. Il est crucial de tester divers scénarios pour garantir la robustesse de cette fonctionnalité.

Ressources

<https://chat.openai.com/>

<https://www.youtube.com/watch?v=iowPI7VpWLA&list=PL8Thl0DA8FbUBpEihprYsoQJSbcLm40nZ>

<https://www.youtube.com/watch?v=E4KUri4UdGM&list=PL6ujmQXEKRmjb7qB4kZZTYdSmfd0MGJf9>

<https://www.bing.com/ck/a?!&&p=3fa00260ea083f74JmltdHM9MTcwMzExNjgwMCZpZ3VpZD0xNzFhNjI2Mi05OGFmLTU0NjktMDQwNy03MWI3OTk2ODY1ODMmaW5zaWQ9NTE2Ng&pptn=3&ver=2&hsh=3&fclid=171a6262-98af-6469-0407-71b799686583&psq=qt+documentation&u=a1aHR0cHM6Ly9kb2MuYXQuaW8v&ntb=1>



Merci

Fin !