

Compte rendu Puissance 4

Félix Grisoni

Dans ce compte rendu je vais d'abord vous expliquer les Attributs majeur du programme puis vous détailler chaque fonction utilisées pour le Puissance 4.

Attributs :

- parent : Fenêtre principale à laquelle le damier est attaché.
- matrice : Liste représentant l'état actuel du jeu, avec des valeurs indiquant les emplacements des pions.
- couleur_pion : Liste des couleurs utilisées pour les pions.
- tour : Numéro du tour actuel.
- gagne : Indicateur de la fin du jeu et du gagnant.
- last_move : Liste enregistrant le dernier coup joué pour permettre l'annulation.

Méthodes :

- __init__(self, parent) :
 - **Description** : Constructeur initialisant les attributs et créant les widgets pour les boutons.
 - **Paramètres** : parent (Tkinter.Tk) - Fenêtre principale.
 - **Fonctionnement** : Initialise les attributs et crée les boutons pour les fonctionnalités telles que l'annulation de coup, la sauvegarde, le chargement, le redémarrage et la sortie du jeu. Affiche également le damier initial.
- afficher_message_victoire(self, joueur) :
 - **Description** : Affiche un message de victoire pour le joueur spécifié.
 - **Paramètres** : joueur (int) - Numéro du joueur gagnant.
 - **Fonctionnement** : Utilise la fonction showinfo() de la classe messagebox de Tkinter pour afficher une boîte de dialogue contenant un message de victoire approprié.
- afficher_message_egalite(self) :
 - **Description** : Affiche un message d'égalité si la grille est pleine sans gagnant.
 - **Fonctionnement** : Vérifie si la grille est pleine et s'il n'y a pas de gagnant. Si ces conditions sont remplies, affiche un message d'égalité à l'aide de la fonction showinfo().
- affiche_damier(self) :
 - **Description** : Dessine le damier et les pions en fonction de l'état actuel du jeu.
 - **Fonctionnement** : Dessine le damier et les pions en utilisant les méthodes create_rectangle() et create_oval() de Tkinter. Affiche également le numéro du tour.

- `click(self, event)` :
 - **Description** : Gère les clics de souris des joueurs pour placer les pions.
 - **Paramètres** : `event (Tkinter.Event)` - Événement de clic de souris.
 - **Fonctionnement** : Détermine la colonne cliquée en fonction des coordonnées de la souris, puis appelle la méthode `tombe_un_pion()` pour placer un pion dans cette colonne.

- `tombe_un_pion(self, colonne_cliquee, ligne_cliquee)` :
 - **Description** : Place un pion dans la colonne choisie par le joueur et vérifie les conditions de placement.
 - **Paramètres** :
 - `colonne_cliquee (int)` - Indice de la colonne choisie.
 - `ligne_cliquee (int)` - Indice de la ligne choisie.
 - **Fonctionnement** : Vérifie si la colonne cliquée est valide et si la case en bas de la colonne est vide. Ensuite, fait tomber le pion jusqu'à ce qu'il atteigne la première case vide en partant du bas de la colonne. Met à jour la matrice de jeu, le tour actuel et enregistre le dernier coup joué.

- `verifie_alignement(self)` :
 - **Description** : Vérifie s'il y a un alignement de quatre pions dans le damier.
 - **Fonctionnement** : Parcourt le damier pour tester les alignements verticaux, horizontaux et diagonaux. Si elle trouve un alignement, détermine le joueur gagnant et affiche un message de victoire.

- `redemarrer(self)` :
 - **Description** : Réinitialise le jeu pour une nouvelle partie.
 - **Fonctionnement** : Réinitialise le tour à 1, réinitialise le statut de victoire et la matrice pour un nouveau jeu. Met également à jour l'affichage du damier.

- `quitter(self)` :
 - **Description** : Ferme la fenêtre du jeu.
 - **Fonctionnement** : Utilise la méthode `destroy()` de Tkinter pour fermer la fenêtre principale.

- `undo(self)` :
 - **Description** : Annule le dernier coup joué.
 - **Fonctionnement** : Annule le dernier coup joué en retirant le dernier pion placé de la matrice, en réduisant le compteur de tour de 1, en réinitialisant le statut de victoire et en mettant à jour l'affichage du damier.
- `sauvegarde(self)` :
 - **Description** : Sauvegarde l'état actuel du jeu dans un fichier.
 - **Fonctionnement** : Utilise la bibliothèque pickle pour sérialiser les données de la matrice, du tour et du dernier coup joué dans un fichier.
- `charger_sauvegarde(self)` :
 - **Description** : Charge l'état précédemment sauvegardé du jeu depuis un fichier.
 - **Fonctionnement** : Lit les données sérialisées à partir du fichier à l'aide de la bibliothèque pickle et met à jour la matrice, le tour et le dernier coup joué en conséquence.