

# Développement du Jeu Puissance 4 avec IA et Interface Graphique

- Présenté par: Achraf Bouchtita & Lina Diani
- Encadrant: Stéphane Ayache

# Introduction

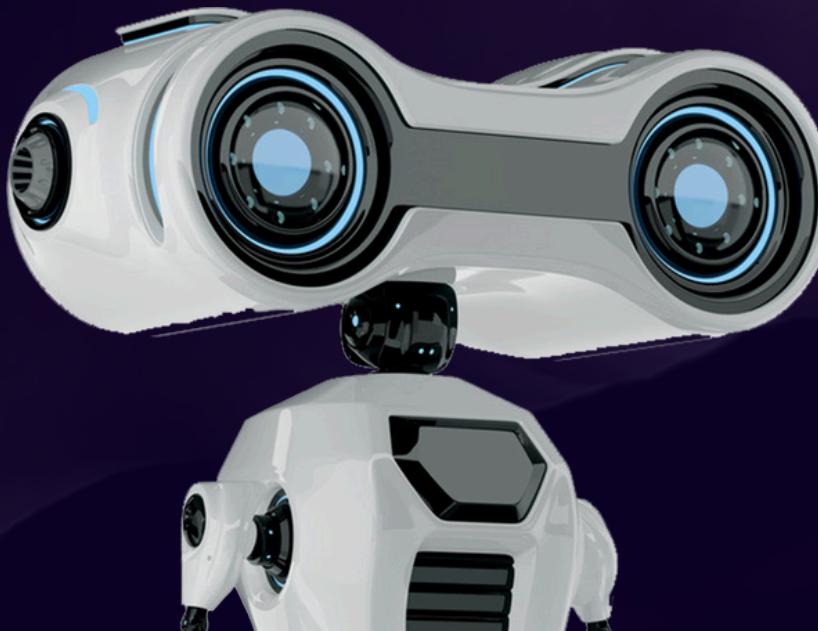
Plan de la présentation :

- Description du jeu et de ses règles.
- Les fonctionnalités que nous avons implémentées.
- Conception et architecture générale.
- Les algorithmes de l'IA utilisés.
- Démonstration de l'interface graphique.

# Description du jeu et Règles

05

- Description du Jeu
  - Jeu de société classique joué sur une grille de 6 lignes et 7 colonnes.
- Règles du Jeu
  - Objectif : aligner quatre jetons de sa couleur.
  - Alignement peut être horizontal, vertical ou diagonal.
  - Partie nulle si la grille est pleine sans alignment.



03

# Fonctionnalités Implémentées

- Modes de Jeu
  - Mode Console pour une interaction basique.
  - Mode Graphique utilisant SDL2 pour une expérience visuelle améliorée.
  - Mode Deux Joueurs pour jouer avec un ami.
  - Mode Joueur contre IA avec les algorithmes DFS et Minimax.

# Architecture Générale

board.c et board.h : Gestion de la grille de jeu.

list.c et list.h : Gestion des structures de données nécessaires.

puissance4.c et puissance4.h : Logique principale du jeu et implémentation de l'IA.

mainconsole.c : Point d'entrée pour le jeu en mode console.

gui.c et gui.h : Gestion de l'interface graphique avec SDL2.

maingui.c : Point d'entrée pour le jeu en mode graphique.

# Algorithmes de l'IA

## Algorithme du DFS

- Description de l'Algorithme DFS
  - Exploration exhaustive: Le DFS explore toutes les possibilités du jeu, en partant de l'état initial jusqu'à une profondeur maximale définie. Il parcourt chaque branche de l'arbre de jeu jusqu'à atteindre la profondeur limite ou une condition terminale.
  - Fonction heuristique: À chaque nœud de l'arbre, une fonction heuristique évalue l'état du jeu pour déterminer la qualité de cette position. Cette fonction attribue un score basé sur des critères tels que le nombre de jetons alignés ou les positions stratégiques sur la grille.



## Algorithme du Minimax

- Description de l'Algorithme Minimax
  - Évaluation des positions de jeu: Le Minimax évalue chaque position possible du jeu jusqu'à une profondeur déterminée, en anticipant les mouvements futurs de l'adversaire et du joueur lui-même. Cela permet de simuler les différentes issues possibles du jeu et de choisir la meilleure stratégie.
  - Élagage alpha-bêta: Pour améliorer l'efficacité, le Minimax utilise l'élagage alpha-bêta, qui élimine les branches de l'arbre de jeu qui ne peuvent pas influencer la décision finale. Cela réduit considérablement le nombre de nœuds à explorer, rendant l'algorithme plus performant sans sacrifier la précision.

# Interface Graphique:

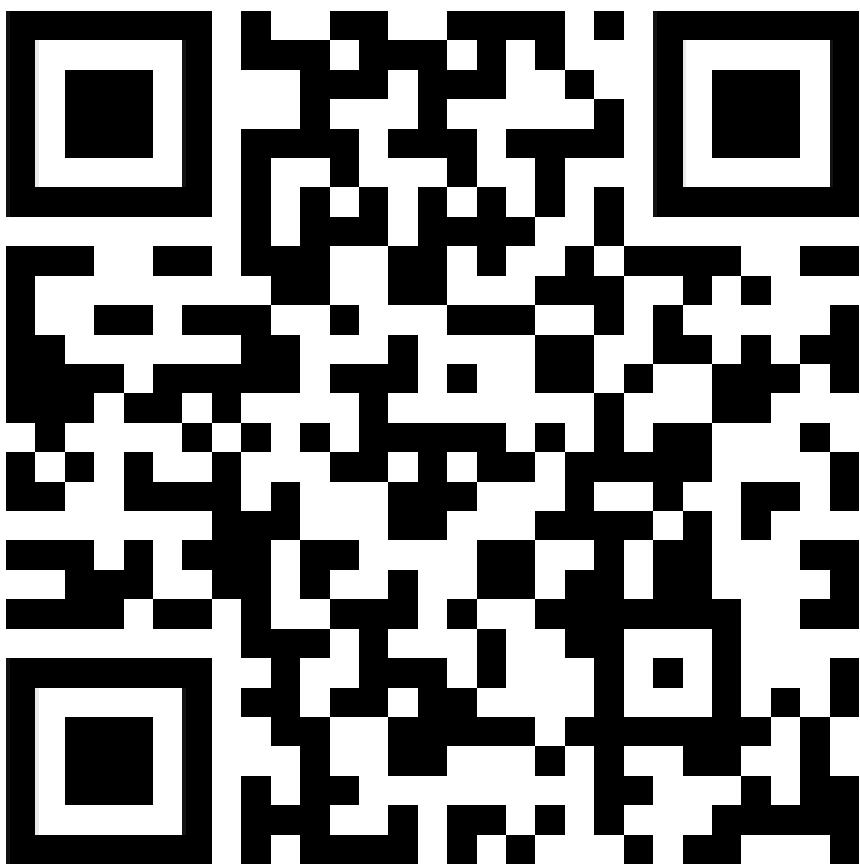
## Initialisation et Rendu Graphique

- Initialisation de SDL et SDL\_ttf
  - `SDL_Init`, `SDL_CreateWindow`, `SDL_CreateRenderer`,  
`TTF_Init`, `TTF_OpenFont`
- Rendu Graphique
  - Dessin de la Grille et des Jetons
  - Nettoyage de l'écran
  - Dessin des lignes de la grille
  - Dessin des jetons avec `SDL_RenderFillCircle`

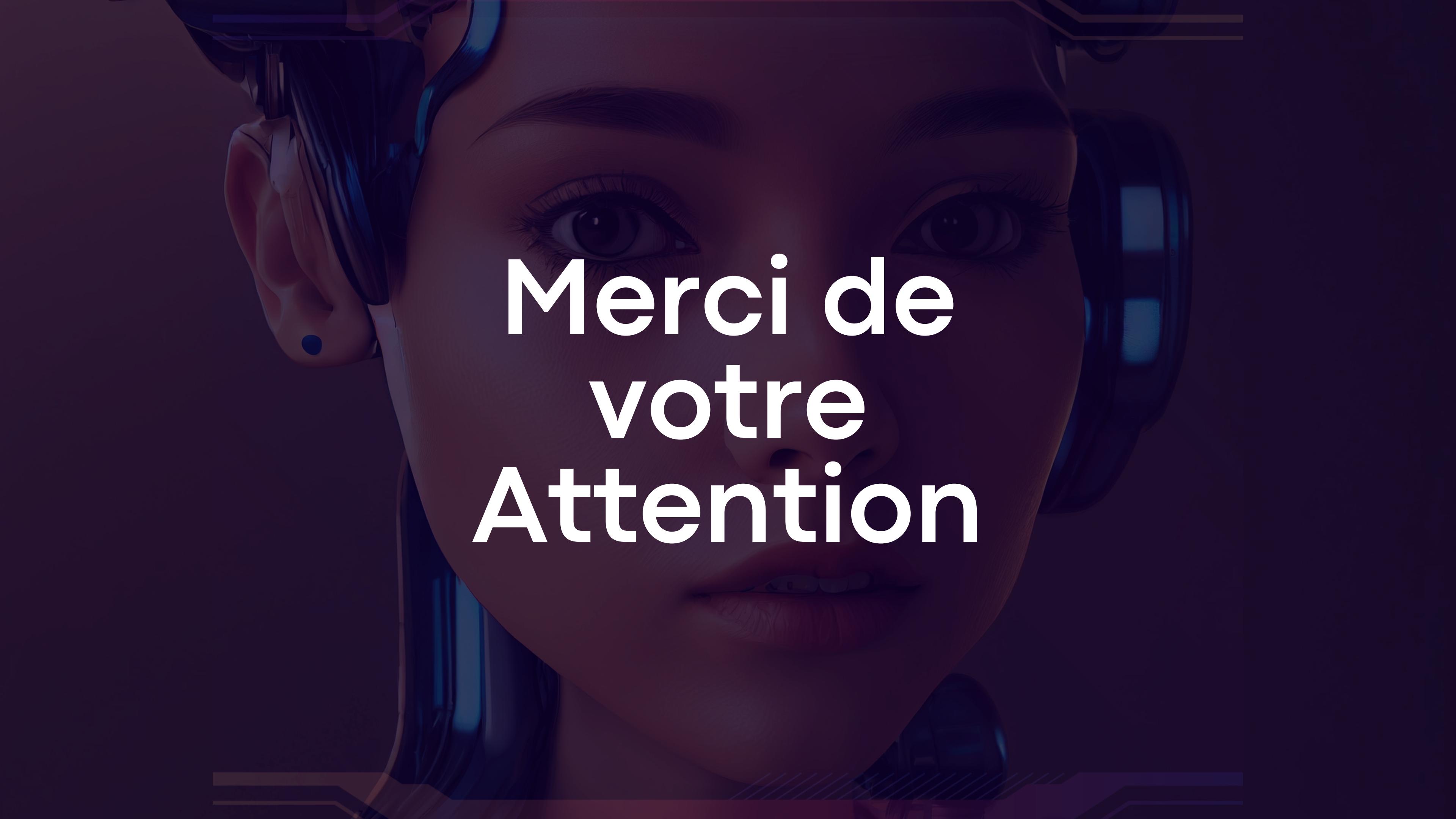
## Gestion des Événements et Boucles de Jeu

- Gestion des Événements
  - Capture et Traitement des Événements
  - Événements de fermeture (SDL\_QUIT)
  - Événements de clic de souris (SDL\_MOUSEBUTTONDOWN)
- Boucles de Jeu
  - Modes de Jeu
    - Joueur contre Joueur
    - Joueur contre IA (DFS ou Minimax)

## CONCLUSION



GIT QR Code



Merci de  
votre  
Attention