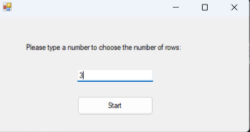
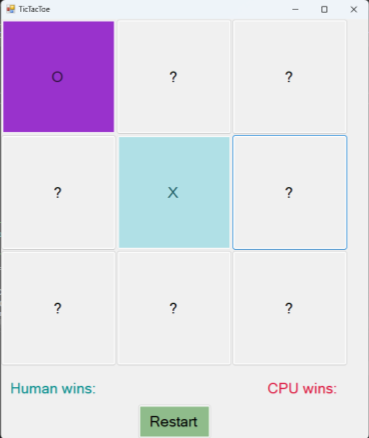
**Dokumentace pro hru Piškvorky (Tic-Tac-Toe) napsanou v jazyce C#.**

Tato dokumentace poskytuje přehled o implementaci hry Piškvorky v programovacím jazyce C# s využitím grafiky Windows Forms, hra zahrnuje protihráče řízeného umělou inteligencí pomocí algoritmu Minimax. Zde jsou některé klíčové prvky **implementace:**

* Hrací pole má nastavitelný rozměr, který lze změnit na začátku hry.

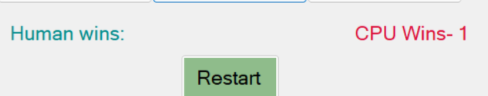


* Hra umožňuje střídání mezi hráči X a O, kteří se snaží vytvořit výherní kombinace.
* Každé políčko na hracím poli je reprezentováno tlačítkem, na které hráči klikají, aby umístili svůj symbol.



* Po každém tahu je provedena kontrola výhry, která zjistí, zda hráč vytvořil výherní kombinaci.
* Pokud někdo z hráčů zvítezil, počítadlo na poli ukaže statistiky o počtu výher pro hráče X a O.





* Hra obsahuje jednoduchou AI, která se snaží najít nejlepší tah pro hráče O pomocí algoritmu MiniMax.
* Hra má jednoduché grafické uživatelské rozhraní pomocí Windows Forms, které zahrnuje hrací pole, tlačítko pro restart a zobrazení počtu výher pro oba hráče.

**Struktura kódu**

Soubor Form1 představuje hlavní okno aplikace a obsahuje veškerou logiku hry. Zahrnuje následující prvky:

* Enum Player: Obsahuje hodnoty X, O a None pro reprezentaci hráčů a prázdných polí.
* Proměnné pro uchování aktuálního hráče, instance třídy Random pro generování náhodných čísel, počet výher hráčů, seznam tlačítek na hrací ploše a další.
* Metoda PlayerClickButton(): Obsluhuje kliknutí hráče na tlačítko na hrací ploše.
* RestartGame(): Restartuje hru po skončení.
* Metody pro vybírání tahů CPU (GetRandomMove a GetBestMove) na základě algoritmu MiniMax.
* MiniMax(): Implementuje algoritmus MiniMax pro vyhodnocení nejlepšího tahu pro CPU.
* BoardToString: Pomocná metoda pro převod herního pole do řetězce pro cachování výsledků.
* CPUmove: Obsluhuje tah CPU.
* CheckGame(): Kontroluje stav hry a vyhlašuje vítěze nebo remízu.
* AdjustLayout(): Přizpůsobuje rozložení oken a tlačítek na hrací ploše.
* CheckWin(): Kontroluje, zda hráč nebo CPU vyhráli hru.
* RestartGame(): Restartuje hru a vynulovává herní pole.

Kód obsahuje také další pomocné metody pro správné vykreslování hrací plochy, vybírání tahů a kontrolu vítězství.

**Popis hry**

Hra Piskvorky je implementována v okně aplikace Form1. Po spuštění aplikace hráči se hraje na hrací ploše o velikosti určené při vytvoření instance třídy Form1. Hráči se střídají v umisťování svých symbolů na hrací plochu klikáním na tlačítka.

* Hráč "X" začíná hru.
* Hra sleduje stav hry a detekuje, kdy hra skončila.
* Po skončení hry je možné hru restartovat pomocí tlačítka "Restart".

**Memoizace**

Pro urychlení výpočtů je použita memoizace. Výsledky výpočtů algoritmu jsou ukládány do cache, aby se opakovaně neprováděly stejné výpočty pro stejný stav hry.

**Algoritmus MiniMax**

Pro tahy CPU je použit algoritmus MiniMax s alpha-beta prořezáváním. Algoritmus MiniMax slouží k nalezení nejlepšího tahu, který CPU může provést, a to tak, aby maximalizovalo svou výhodu a minimalizovalo výhodu hráče "X".

**Závěr**

Toto je dokumentace k implementaci hry Piškvorky v jazyce C# s použitím Windows Forms. Hra umožňuje hráčům střídat se v umisťování svých symbolů na hracím poli a kontroluje výhru a remízu. Implementace zahrnuje také jednoduchou umělou inteligenci pro hru proti počítači. Hru lze snadno rozšiřovat a upravovat podle vlastních potřeb a preferencí.