Wykonał: Wojciech Rojek, 02.02.2021r

Algorytm min-max w zastosowaniu gry kółko i krzyżyk

Rozdział 1: Wprowadzenie

Algorytm min-max jest metodą minimalizowania maksymalnych strat. Twierdzenie minimax zostało ustanowione przez Johna Von Neumanna. Polega ona na tym, że mając funkcję oceniającą wartość stanu gry w dowolnym momencie, obliczamy drzewo wszystkich możliwych stanów w grze do pewnej głębokości. W mojej implementacji gry algorytm min/max oblicza optymalne ruchy dla „komputera".

Rozdział 2: Zasady gry

Na początku gracz wybiera swoją figurę. Następnie rozpoczyna się rozgrywka. Rozpoczyna losowo wybrany gracz. Gra toczy się do czyjejś wygranej lub remisu. Po zakończeniu gry, gracz ma ponownie możliwość wybrania figury i rozpoczęcia kolejnej gry.

Rozdział 3: Opis interfejsu

Cały program opiera się na interfejsie klasy „Game". Poszczególne funkcje tej klasy realizują możliwość gry w kółko i krzyżyk, z „komputerem", którego ruchy wyznaczane są za pomocą algorytmu min i max.

Opis funkcji:

def __init__(self,root) - Konstruktor, jako argument przyjmuje „korzeń" – główną warstwę aplikacji okienkowej w tkinter. Konstruktor ten inicjuje początkowy stan programu, zmienne przechowujące odpowiednie obrazy oraz zmienne pomocnicze do określania stanu gry. Z poziomu konstruktora wywoływana jest metoda „_newGame()".

def _newGame(self) - Funkcja odpowiadająca za inicjacje listy Stringów, na której można sprawniej przeprowadzać operacje niż na graficznej planszy. Funkcja generuje dwa przyciski ustawia ich wygląd, kolor, rozmiar i przypisuje im odpowiednio grafikę kółka i krzyżyka. Przyciski te obsługuje funkcja „_initBoard()" opisana poniżej.

def _initBoard(self): - Funkcja obsługująca kliknięcie w przycisk „xButton” lub „oButton”.Jako argument przyjmuje string „X” lub „O”, w zależności od wciśniętego przycisku. Po wciśnięciu w ten przycisk figura gracza ustalona jest jako „X”, a komputera jako „O”. Po wciśnięciu odpowiedniego przycisku, figura gracza zostaje odpowiednio ustalona, a komputer przyjmuję niewybraną figurę . Z poziomu tej funkcji wykonywana jest metoda _generateBoard(). Gracz rozpoczynający rozgrywkę wybierany jest losowo. Jeżeli rozpoczyna komputer, pierwszy ruch wykonywany przez niego jest losowy.

def _generateBoard(self): - Funkcja generująca planszę 3x3 w formie 9 przycisków. Funkcja _click(button) obsługuje wygenerowane przyciski. Wciśnięcie w dany przycisk przez gracza oznacza postawienie tam danej figury.

def _click(self,button): - Funkcja obsługująca kliknięcie w pole planszy. W zależności od figury gracza, po wciśnięciu umieszcza grafikę kółka lub krzyżyka, a następnie aktualizuje i sprawdza stan rozgrywki. Po wykonaniu tych czynności wykonywana jest funkcja _best_move(), która oblicza ruch dla komputera.

def _checkIfEnd(self,figure): - Funkcja sprawdzająca czy gra nie została zakończona. Sprawdzany jest stan gry, czy któryś gracz wygrał lub czy jest remis. Funkcja generuje odpowiednie okna dialogowe w zależności od końcowego stanu gry. Po zakończeniu rozgrywki plansza jest czyszczona i następuje powrót do wyboru figury przez gracza.

def _updateBoard(self,button,figure): - Funkcja aktualizująca rozkład figur na planszy w formie listy 3x3. Operowanie na takiej planszy pozwala łatwo sprawdzać stan rozgrywki. Jako argumenty przyjmuje wciśnięty przycisk oraz figurę, która została umieszczona w poprzednim ruchu.

def _checkWinner(self): - Funkcja sprawdzająca sam stan gry. Nie przyjmuje parametrów. Zwraca None, ‘tie’ lub zwycięzką figurę w postaci stringa.

def _clearRoot(self): - Funkcja czyszcząca „roota” z wszystkich komponentów, przycisków i etykiet. Nie przyjmuje argumentów i nic nie zwraca.

def _putAI(self,button): - Funkcja obsługująca postawienie figury dla komputera. Generuje na odpowiednim polu grafikę figury postawionej przez komputer. Następnie sprawdzany jest stan rozgrywki i kolejno następuje ruch gracza.

def _best_move(self): - Funkcja obliczająca optymalny ruch przez komputer. Funkcja sprawdza wolne pola a następnie wywołuje funkcję minima(depth,isMaximazing). Po obliczeniu przez algorytm minimax, na odpowiednie pole zostaje położona figura gracza.

def _minimax(self,depth,isMaximizing): - Funkcja realizująca algorytm min/max. Funkcja jako argument przyjmuję głębokość „drzewa kombinacji ruchów” oraz czy sprawdzany ruch jest dla gracza maksymalizującego(komputer) czy dla gracza minimalizującego. Funkcja wykonywana jest rekurencyjnie. Funkcja zwraca najlepszy możliwy wynik dla najlepszej możliwej kombinacji ruchów, które dążą do zwycięstwa.