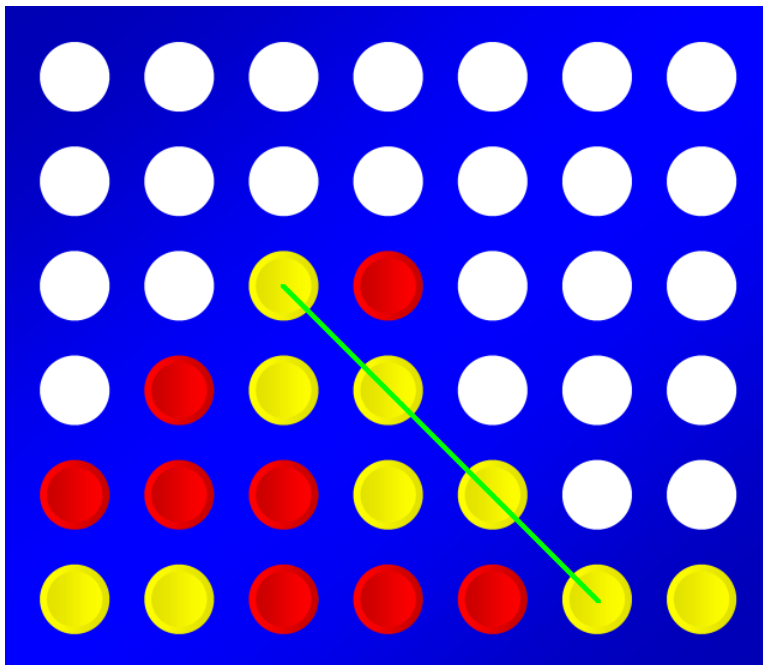


Projekt-README

Dang Quynh Tram, Nguyen
Matrikelnummer: 5311561

Vier-Gewinnt

Das Spiel Vier-Gewinnt ist ein Strategiespiel für zwei Personen. Dafür gibt es ein Gitter 7x6 (7 Spalten und 6 Reihen) und zwei-färbige Spielersteine. Die Spieler lassen abwechselnd sein Stein in eine freie Spalte fallen. Die freie Spalte ist die Spalte, die nicht voll mit 6 Steinen ist. Das Spiel beendet, wenn eine Person vier Steine in eine Linie (diagonal, vertikal oder horizontal) kriegt oder alle Spalten voll sind. Die Spielzüge laufen durch die Methode *play(Move... columns)*.



Der beste Zug gibt die Spalte an, mit der der Spieler höchstwahrscheinlich gewinnen kann. Durch die **Negamax**- und **Monte-Carlo-Methode** werden das Score für alle sieben Spalten bewertet. Welche, die das maximale Score hat, wird als den besten Zug gewählt. Dieser Schritt wird durch die Methoden *bestMove()*, *negamax(int depth)* und *monteCarlo_score()* durchgeführt.

Anschließend gibt es *undo()*-Methode, um die letzte Position in der *history*-Liste zurückzunehmen, und noch ein paar Hilfemethoden wie:

- (static) *newGame()*: Erzeugt ein neues Spiel mit einer leeren *history*-Liste.
- *nextMoveScore()*: Speichert die Scores der 7 Spalten, um das beste Score in der *bestScore()*-Methode zu suchen.
- *cango(int column)*: Prüft die Spalte, es voll ist. Wenn ja, gibt es falsch-Wert zurück. Die gewählte Spalte ist nicht akzeptiert.
- *getGrid()*: Erzeugt ein Gitter aus dem Spielverlauf (*history*-Liste).
- *getHistory()*: Gibt den Spielverlauf zurück.

Quelle: [Vier-Gewinnt, Wikipedia](#)

Beschreibung des Algorithmus'

Im Projekt werden **Negamax-** und **Monte-Carlo-Algorithmus** genutzt, um den besten Zug zu finden.

Monte-Carlo-Algorithmus hilft bei der Evaluierung des Scores eines Zuges. Mit dem angegebenen Spielverlauf (*history*-Liste) berechnet die Methode die Wahrscheinlichkeit, dass der Spieler, zu dem der letzte Zug in der *history*-Liste gehört, gewinnen kann. Dabei läuft sie 100 Experimente und zählt die Gewinne. Der Score ist 10-mals den Anteil der Gewinne in 100 Experimente. Bei den Game-Over-Fällen bekommt der Score die besondere Werte. Das Spiel beendet mit dem Sieg, nimmt es 20. Das Spiel ist unentschieden, nimmt es 10.

Negamax-Methode gibt den Score an einem Spalte zurück. Es hängt von der Tiefe des rekursiven Aufrufs ab. Das Score ist von der Monte-Carlo-Methode berechnet. Nach jeder Tiefe wird das negative Maximum des Scores der besuchten Spalten mit einem maximalen Wert (*value*) verglichen. Wenn es größer ist, speichert *value* dieses Maximum für den nächsten Vergleich. Die *history*-Liste wird nach jedem Aufruf geändert, weil sie einen neuen Wert (*Position* genannt) speichern muss, um das für **Monte-Carlo-Methode** benötigte Gitter zu erzeugen. Nach der Rückgabe des Scores wird diese Position gelöscht (mithilfe der *undo*(-)Methode).

Erklärung des Logging-Protokolls

Erklären Sie anhand eines Szenarios, inwiefern das beispielhaft beigefügte Logging-Protokoll die korrekte Umsetzung des Berechnungsalgorithmus' belegt.

Um das Laufen der Methoden bzw. Algorithmen bei des `bestMove()`-Methode zu verfolgen, kommt das Logging-Protokoll. Hier ist ein Beispiel:

Ein Spielfall:

VierGewinnt:

Spalte 1[2, 1]

Spalte 2[1, 1, 2, 2, 2, 1]

Spalte 3[1, 2, 1, 1, 2]

Spalte 4[1, 2, 1, 2, 1, 1]

Spalte 5[2, 1, 2, 1, 2, 2]

Spalte 6[2]

Spalte 7[]

history = [3, 4, 11, 10, 17, 18, 25, 24, 31, 32, 38, 39, 2, 9, 1, 0, 8, 15, 16, 22, 23, 29, 36, 30, 7, 5]

Logging-Protokoll nach dem Aufruf des `bestMove()`-Methode:

Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt nextMoveScore

INFORMATION: Negamax Methode mit Scores, die durch Monte-Carlo Methode berechnet werden. (1) Signal des Anfangs des Negamax-Algorithmus, der den Score einer Spalte berechnet.

Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt nextMoveScore

INFORMATION: Der Score der Spalte 1 : (2) Anfang mit der ersten Spalte, Tiefe = 2

Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax

INFORMATION: In der Spalte 1 (3) Die nächste erste Spalte, Tiefe = 1

Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: In der Spalte 1 (4) Tiefe = 0
 Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Die letzte Tiefe: Monte-Carlo-Score =3 (5) Wegen der letzten Tiefe gibt es der Score zurück.
 Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: In der Spalte 3 (6) Der 2. Spalte wird verpasst, weil sie voll ist.
 Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Die letzte Tiefe: Monte-Carlo-Score =5
 Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: In der Spalte 6
 Juli 14, 2022 10:55:22 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Die letzte Tiefe: Monte-Carlo-Score =9
 Juli 14, 2022 10:55:23 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: In der Spalte 7
 Juli 14, 2022 10:55:23 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Die letzte Tiefe: Monte-Carlo-Score =4
 Juli 14, 2022 10:55:23 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Tiefe 1 Maximum = -3 (7) Das negative Maximum der Scores in Tiefe 0 wird in Maximum-Wert der Tiefe 1 speichert. Wenn es weiterläuft, kann ein besserer Score in der Tiefe 1 kommen und diesen Wert ersetzen.
 ...
 INFORMATION: Der Score der Spalte 6 : (8) Das Spalte passt für den besten Zug.
 Juli 14, 2022 10:55:23 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: In der Spalte 1
 Juli 14, 2022 10:55:23 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: In der Spalte 1
 ...
 Juli 14, 2022 11:12:25 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Tiefe 1 ,Maximum = -9
 Juli 14, 2022 11:12:25 PM VierGewinnt.VierGewinnt negamax
 INFORMATION: Tiefe 2 ,Maximum = 9 (9) Der Score des besten Zuges ist hoch und ist der Höchste.
 ...
 Juli 14, 2022 10:55:25 PM VierGewinnt.VierGewinnt nextMoveScore
 INFORMATION: Score Array [4, -10, 4, -10, -10, 9, 4] (10) Score Array für alle Spalten. Seine Stelle (Index) entspricht auch die Spalte. Der Score -10 zeigt, dass diese Spalte voll ist.
 Juli 14, 2022 10:55:25 PM VierGewinnt.VierGewinnt bestMove
 INFORMATION: Der Spieler hat den besten Zug in der Spalte 6 gewählt. (11) Die Spalte des Maximums im Score-Array wird als den besten Zug gewählt.

Anmerkungen

Der Schalter des Logging-Protokoll liegt in der *bestMove()*-Methode an der ersten Zeile. Zum An-/Abschalten kann man einfach `//` vor dem Satz `logger.setLevel(Level.OFF);`