

Proposal zur Übung KI – 5. Aufgabenset

Bei dem Problem, das implementiert werden soll, handelt es sich um ein (derzeit noch namenloses) Spiel für zwei oder mehr Spieler, das typischerweise auf kariertem Papier gespielt wird. Dabei ziehen die Spieler reihum jeweils eine Linie, die horizontal, vertikal oder diagonal über 3 Gitterknoten führt:

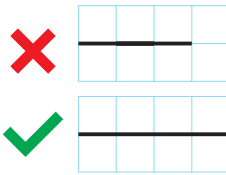


Dabei gelten folgende Regeln:

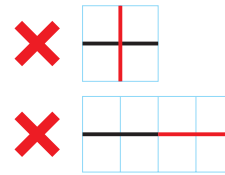
Eigene Linien dürfen sowohl berührt als auch geschnitten werden:



Es dürfen keine Linien ergänzt werden, nur komplett neu ansetzen ist möglich:



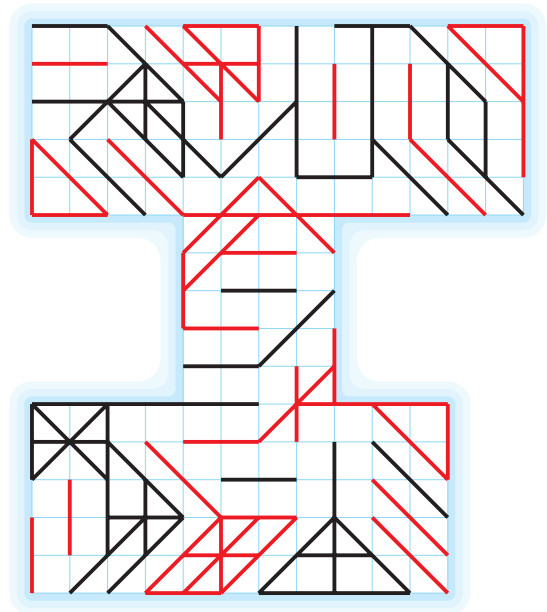
Gegnerische Linien dürfen weder berührt noch geschnitten werden:



Verloren hat derjenige Spieler, der als erstes keine Linie mehr setzen kann.

Strategien, um das zu erreichen, zielen vor allem auf zwei Dinge ab: Zum einen die Konstruktion möglichst großer Bereiche, die für die Gegner nicht mehr erreichbar sind; zum anderen ist der Gegner daran zu hindern, selbst solche Bereiche aufzubauen.

Die nebenstehende Grafik zeigt ein vollständig gespieltes Spiel.



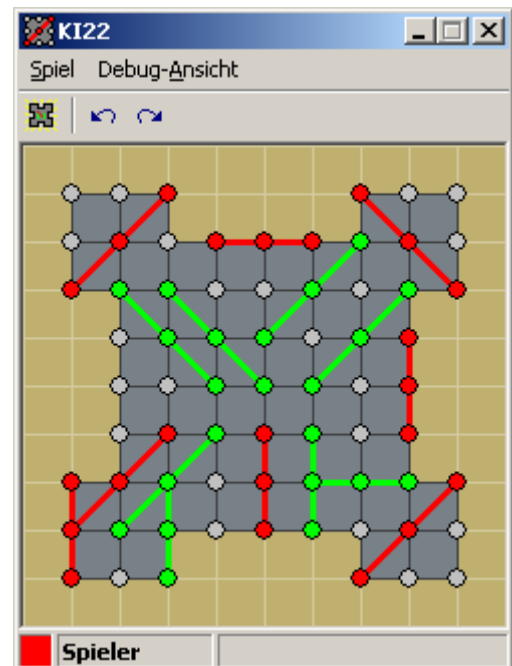
Kurzdokumentation

Das im Proposal beschriebene Spiel wurde in der Programmiersprache Delphi ausprogrammiert. Es wurde ein weitestgehend objektorientierter Ansatz zur Implementation gewählt. In der vorliegenden Fassung sind die folgenden Features realisiert:

- sechs KI-Stufen
- beliebige Zwei-Personen-Spiele: Mensch/Mensch, Mensch/KI oder KI/KI-Spiele (letztere können pausiert werden)
- Undo und Redo über das ganze Spiel hinweg
- Zeichnen eigener Spielfelder und Möglichkeit, diese abzuspeichern

Bedienung

Das Programm sollte weitestgehend selbsterklärend sein: Im Menü »Spiel« kann das Dialogfeld »Neues Spiel« aufgerufen werden, in dem man alle Parameter des Spiels festlegen kann. Das Menü »Debug-Ansicht« ermöglicht Einblick in interne Strukturen des Spiels: Die RemainingField-Einträge zeigen für den jeweiligen Spieler, welche Züge dieser nicht(!) mehr durchführen kann. PossibleOwners zeigt auch solche Kreuzungspunkte farbig an, die zwar noch nicht von einer Linie berührt werden, jedoch eindeutig einem Spieler zugeordnet werden können. Die Buttonbar enthält je nach Situation Buttons zum Starten eines neuen Spieles (mit denselben Parametern wie das bereits laufende), Stoppen eines KI-Zuges, Anhalten und Fortsetzen eines KI-gegen-KI-Spiels und die Undo-/Redo-Funktion.



Die KI-Levels

Bei den sechs KI-Stufen handelt es sich eigentlich um mehr oder weniger verschiedene Algorithmen:

0. **zufällige Züge** — Es wird irgend ein beliebiger gültiger Zug ausgewählt.
1. **optimale Züge defensiv** — Es wird der Zug ausgewählt, der die höchste Bewertung besitzt. Als Kriterium dient die Anzahl Kreuzungspunkte, die sich der KI-Spieler durch den Zug reservieren kann. Besitzen mehrere Züge die maximale Bewertung, so wird einer von diesen zufällig ausgewählt.
2. **optimale Züge offensiv** — Ebenso, jedoch mit einem anderen Bewertungskriterium: Entscheidend ist die Anzahl der gegnerischen Züge, die durch den Zug verhindert werden.
3. **optimale Züge im Mittel** — Eine Mischung aus Stufe 1 und 2: Als Bewertungsfunktion dient ein gewichteter Mittelwert aus der Anzahl reservierter Kreuzungspunkte und der Anzahl verhinderter Züge. (Die Gewichtung ist empirisch bestimmt, so daß beide Anteile zahlenmäßig etwa gleich wiegen.)
4. **MiniMax-Strategie** — Eine MiniMax-Spielbaumsuche der Tiefe 5. Da ein vollständiger Spielbaum dieses Spiels eine Ordnung von 100 bis sogar 1000 haben kann, werden für jeden Knoten nicht alle möglichen Züge untersucht, sondern nur die 5 laut der offensiven Strategie am höchsten bewerteten. (Dies ist durchaus ausreichend, denn bereits die offensive Strategie ist äußerst spielstark.) Dennoch kann die Berechnung eines Zuges mehrere Sekunden dauern. Daher gibt es in der Toolbar einen Button, der die Zugberechnung abbricht und den bisher bestbewerteten Zug durchführt.
5. **MiniMax-Strategie mit Alpha-Beta-Pruning** — Spielerisch identisch mit der MiniMax-Strategie, dank Alpha-Beta-Optimierung jedoch ein wenig schneller.