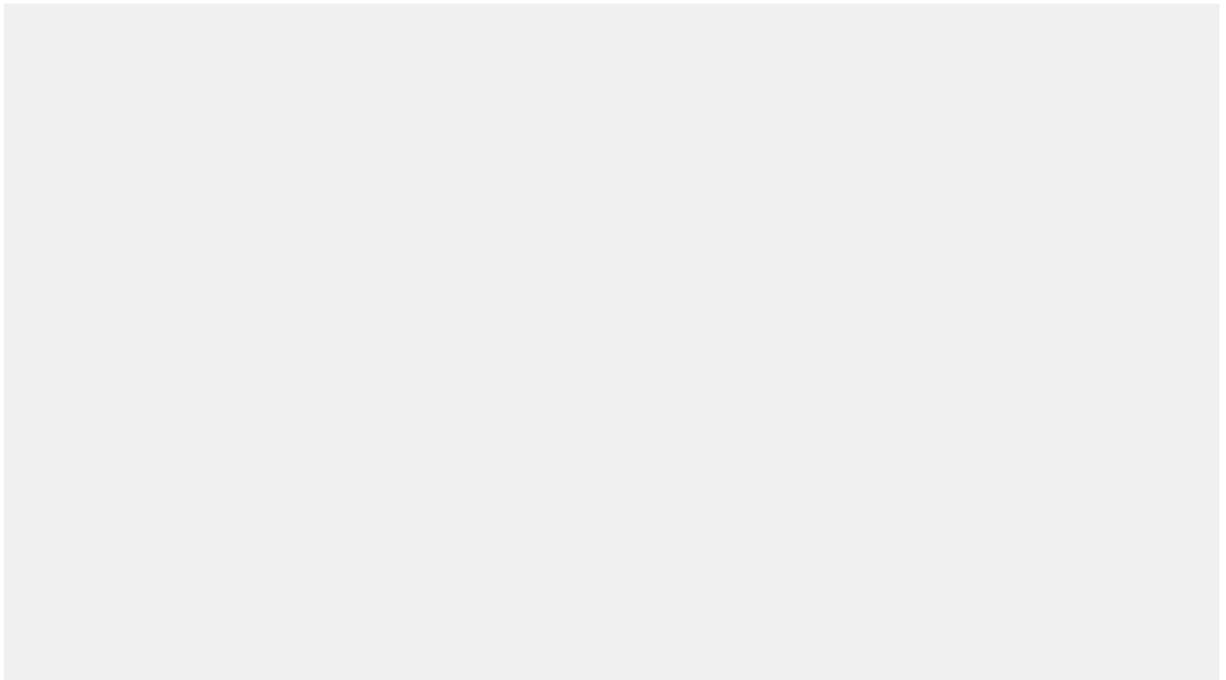




# Streichholz - KI



– Implementierung der KI–



Wir können bisher gegen unsere KI spielen, sie lernt jedoch noch nicht. Dies wollen wir im letzten Schritt ändern. Dazu sind jeweils eine fehlende Funktion in `Ai` und `Schachtel` zu ergänzen. Wir fangen diesmal mit `entfernePerLe(zug)` aus `Schachtel` an. Diese Funktion nimmt den zu entfernenden Zug entgegen und entfernt ihn aus `perLen`.

## AUFGABE 8

Implementiere den Code für `entfernePerLe()`.

## DIE KI LERNT

Alles ist vorbereitet. Jetzt kommt der letzte Schritt mit dem die KI tatsächlich „intelligent“ wird. Dazu erstellen wir jetzt den Code für `Lerne()`. Wir fangen dazu erst einmal damit an, den jeweils letzten Zug des Spiels zu entfernen. Aus der Praxis wissen wir bereits, dass es auch vorkommen kann, dass bereits ein Zug zuvor der Fehler gemacht wurde, aber darum kümmern wir uns im Anschluss.

Bevor man hier wild anfängt los zu programmieren, sollte man sich erst einmal darüber Gedanken machen, welche Informationen man benötigt. Verdeutlichen wir uns das an einem Beispiel in Abbildung 1:



Abbildung 1

In dieser Stellung hat die KI einen Fehler gemacht in dem sie an dem mittleren Bauer vorbeigezogen hat. Der letzte Zug der gespielt wurde, ist der falsche Zug der KI. Jedoch ist das nicht die Stellung die die KI in den Schachteln gespeichert hat, sondern die in Abbildung 2. Wir müssen also verschiedene Spielzustände betrachten,

wenn wir den falschen Zug oder die Stellung, in der der falsche Zug gemacht wurde, erhalten möchten.



Abbildung 2

## AUFGABE 9

Erstelle den Code für *Lerne()*. Die Funktion wird aufgerufen, wenn die KI verloren hat und soll den zuletzt gespielten Zug aus der entsprechenden Schachtel entfernen.

Unsere KI ist fast fertig. Inzwischen lernt sie aus ihren Fehlern und kann einfache Fehler vermeiden. Jetzt müssen wir nur noch dafür sorgen, dass sie im Lernprozess mehr in die Tiefe geht und nicht nur den Zug aus der letzten Schachtel betrachtet. Wenn Weiß beispielsweise mit seinem dritten Zug gewinnt, hat die KI zwei Züge gemacht. Nun kann der Fehler schon im ersten Zug der KI stecken und mit dem zweiten Zug würde sie immer verlieren.

## AUFGABE

Erweitere *Lerne()* so, dass die KI nicht nur die zuletzt ausgewählte Schachtel betrachtet, sondern auch die davor, insofern die Anzahl der verfügbaren Züge der letzten Schachtel eins ist.

Damit ist die Schach – KI fertig! Du kannst jetzt deine KI auch digital trainieren. Wenn du die Limitationen der KI erfahren möchtest, erweitere doch einfach auch die Anzahl der Bauern. Du musst dann aber auch die Funktion *checkState()* anpassen, welche die Siegbedingungen überprüft. Du findest sie in *script.js*.