

---

Softwarepraktikum SS 2018  
**Assignment 5**

---

Group - 3

Ramil Sabirov	369500	ramil.sabirov@rwth-aachen.de
Joel Choi	345575	joel.choi@rwth-aachen.de
Eric Remigius	366895	eric.remigius@rwth-aachen.de

---

# Task 1

An dieser Stelle wollen wir auf das Assignment 3 verweisen. Dort wurde bereits unsere Zeitschätzung für die nächste Tiefe erläutert.

Kurzgefasst: Wir wählen sehr konservativ den maximalen Verzweigungsgrad des Variantenbaumes als Faktor für die nächste Tiefe. Damit gehen wir nahezu allen möglichen Timeouts aus dem Weg. Jedoch beenden Spiele wir auch des öfteren mit einem riesigen Zeitkonto, welches dann verschwendet ist.

## Task 2

Wir verwenden die Suchfenster innerhalb des Deepeners (Iterative Deepening Calculator). Nach der ersten Suche mit Tiefe 1 **ohne** Fenster, also mit Grenzen  $+\infty$  und  $-\infty$  haben wir einen Richtwert für den Stellungswert  $val_1$ . Für weitere Tiefen  $t$ , zentrieren wir ein Suchfenster mit Breite  $b$  um den zuvor gefundenen Stellungswert herum und suchen den neuen Stellungswert in dem Intervall  $[val_{t-1} - b, val_{t-1} + b]$  in der Hoffnung, dass der gesuchte Wert tatsächlich in dem gegebenen Intervall liegt und von Anfang an mehr Teilbäume abgeschnitten werden können. Leider kann es passieren, dass dies nicht der Fall ist und die Einschränkung zu stark war. Der Alpha-Beta Algorithmus liefert in dem Fall entweder einen Wert kleiner-gleich der unteren Schranke  $val_t \leq val_{t-1} - b$  oder einen Wert größer-gleich der oberen Schranke  $val_t \geq val_{t-1} + b$ . Jedoch ist der berechnete Stellungswert nicht aussagekräftig, da der optimale Pfad beschnitten worden sein könnte. Also ist eine Neusuche notwendig:

- **Fall 1: untere Schranke wird unterschritten.**

Dann liegt der tatsächliche Stellungswert unterhalb der unteren Schranke. Wenn also das Intervall  $[\alpha, \beta]$  war, dann erfolgt die Neusuche im Intervall  $[-\infty, \alpha]$ .

- **Fall 2: obere Schranke wird überschritten.**

Analog zum ersten Fall muss oberhalb der oberen Schranke neu gesucht werden. Also im Intervall  $[\beta, +\infty]$ .

In der Implementierung wird das jeweilige Fenster dem Calculator über einen Parameter gegeben von der Klasse *CalculatorConditions*. Diese enthält Vorgaben, welche der jeweilige Calculator zu erfüllen hat. Zum Beispiel gehört eine maximale Tiefe und ein Zeitlimit auch dazu.

## Task 3