

## Aufgabe 6-1 (10+4 Punkte)

Implementieren Sie einen Computerspieler für das Brettspiel Mühle. Sie sollen dabei eine vorgegebene textbasierte Schnittstelle realisieren.

Berücksichtigen Sie, dass Ihr Programm beide Farben spielen können muss. Dem Programm steht pro Zug nur eine begrenzte Bedenkzeit zur Verfügung. Diese soll ein Parameter des Programms sein. Der Aufruf des Programms für ein Zeitlimit von 30 Sekunden soll damit wie folgt aussehen:

```
programmname -t 30
```

Auf der Internetseite der Vorlesung steht ein Java-Programm zur Verfügung, mit dem Sie mittels einer grafischen Oberfläche gegen Ihr Programm spielen können. Alternativ können zwei Computerprogramme gegeneinander antreten.

Zur Demonstration steht auch ein Zufallsspieler zum Download bereit, der mit dem Programm eingesetzt werden kann.

Alle Computer-Programme werden an einem Wettbewerb teilnehmen und der Sieger wird 4 extra Punkte bekommen.

**Beschreiben Sie Ihr Programm hinreichend in Bezug auf verwendete KI-Techniken, zum Beispiel Such-Algorithmen, Heuristiken etc.**

### Spielregeln

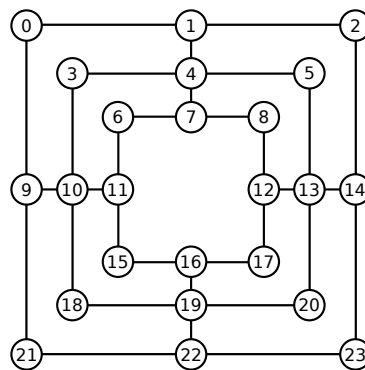
Das Spiel wird mit jeweils neun weißen und schwarzen Steinen auf dem abgebildeten Brett gespielt. Dabei werden die Steine auf den mit Kreisen markierten Stellen abgesetzt. Der Spieler mit den weißen Steinen beginnt. Das Spiel gliedert sich in drei Phasen:

**Setzphase** Zu Beginn des Spiels setzen die Spieler abwechselnd Steine auf beliebige freie Stellen des Bretts.

**Zugphase** Sobald der letzte Stein auf das Brett gesetzt wurde beginnt die Zugphase. Dabei ziehen die Spieler abwechselnd jeweils einen eigenen Stein um genau ein Feld weiter auf ein benachbartes freies Feld. Benachbarte Felder sind auf dem Spielbrett durch Linien verbunden. Kann ein Spieler in dieser Phase keinen Zug machen, so hat er das Spiel verloren.

**Flugphase** Sobald ein Spieler nur noch drei Steine auf dem Brett übrig hat beginnt für diesen Spieler die Flugphase. Er darf pro Zug einen seiner Steine auf ein beliebiges freies Feld setzen. Besitzen beide Spieler nur noch jeweils drei Steine, so befinden sich beide in der Flugphase.

**Schlagen** Während aller drei Spielphasen können gegnerische Steine geschlagen werden. Dies geschieht, wenn ein Spieler durch seinen Zug eine Mühle bildet. Eine Mühle



sind drei eigene Steine, die alle durch eine gerade Linie verbundenen Felder besetzen — es gibt also 16 mögliche Positionen für eine Mühle. Geschlagen werden darf ein beliebiger gegnerischer Stein, der nicht teil einer gegnerischen Mühle ist. Dieser wird aus dem Spiel entfernt. Falls beim Bilden einer Mühle kein solcher gegnerischer Stein vorhanden ist, verfällt die Möglichkeit zum Schlagen.

**Spielende** Das Spiel endet sobald ein Spieler nur noch zwei Steine besitzt. Dieser Spieler hat verloren. Das Spiel endet außerdem wenn ein Spieler in der Zugphase keinen Zug ausführen kann, da seine Steine eingesperrt sind. Dieser Spieler hat auch verloren.

## Spielprotokoll

Um die entwickelten Programme gegeneinander antreten zu lassen müssen alle Programme das gleiche Spielprotokoll unterstützen. Dieses wird im Folgenden beschrieben. Die Kommunikation erfolgt via Standard I/O (in Java System.out, System.in und in C printf(), scanf(), etc.).

1. Server sendet die Farbe des Spielers, “0” für Weiß und “1” für Schwarz, gefolgt von einem Newline-Zeichen
2. Programm sendet seinen Namen, gefolgt von einem Newline-Zeichen
3. Server sendet den Spielstand wie folgt, gefolgt von einem Newline-Zeichen

**Brettzustand RestSteineWeiß RestSteineSchwarz**

**Brettzustand** 24 Zeichen aus (W (weißer Stein), B (schwarzer Stein), E (leeres Feld)), die den Zustand des Spielfeldes (platzierte Steine) beschreiben; die Reihenfolge entnehmen Sie der Zeichnung

**RestSteineWeiß** eine Zahl, die die noch zu setzenden weißen Steine darstellt

**RestSteineSchwarz** eine Zahl, die die noch zu setzenden schwarzen Steine darstellt

Ein Beispiel ohne abschließendes Newline-Zeichen, bei dem Weiß einen Stein links oben und Schwarz einen Stein rechts unten platziert haben, wäre:

WEEEEEEEEEEEEEEEEEEEEEB 8 8

4. Programm sendet seinen Zug in folgendem Format. Dabei werden Felder wie in der Zeichnung angegeben mit ganzen Zahlen im ASCII-Format (nicht binär!) adressiert.

**Quellfeld Zielfeld Entfernen**

**Quellfeld** Gibt an, welcher Stein bewegt werden soll; übergeben Sie hier -1 in der Setzphase

**Zielfeld** Das freie Feld auf den der Stein gesetzt/bewegt werden soll

**Entfernen** Gibt an, welcher gegnerische Stein entfernt werden soll; übergeben Sie hier -1 falls kein Stein geschlagen wird

Ein Beispielzug ohne abschließendes Newline-Zeichen, bei dem ein neuer Stein in die rechte obere Ecke gesetzt und kein Stein geschlagen wird, wäre:

-1 2 -1

Die Schritte 3 und 4 des Protokolls wiederholen sich bis das Spiel endet. Die Spielprogramme werden dann automatisch beendet. Eine Verletzung des Protokolls oder die Übermittlung eines illegalen Zugs hat den Verlust der Partie zur Folge. Sie können den Quellcode des Zufallsspielers als Beispiel für die Implementierung des Protokolls benutzen.

### Austausch per Rubikon

Sie dürfen sich zu der Aufgabe gerne über das Rubikon austauschen und ggf. ihre Programme bereits vor der Abgabe gegeneinander antreten lassen.