

Mathematische Modellierung WS 16/17

Erstes Softwareprojekt

Spielanleitung

Zu Beginn des Spiels gibt es drei Haufen aus Kieselsteinen. Die Anzahl der Steine der einzelnen Haufen kann dabei beliebig gewählt werden. Wir legen aber fest, dass kein Haufen mehr als 7 Steine haben darf. Eine beliebte Startaufstellung ist: Der erste Haufen besteht aus 7 Steinen, der zweite aus 5 und der dritte aus 3 Steinen.

Die beiden Spieler ziehen abwechselnd. In einem Zug nimmt der Spieler von einem beliebigen Haufen beliebig viele Steine weg. Diese werden aus dem Spiel entfernt. Er muss mindestens einen Stein nehmen und kann nicht mehr Steine nehmen, als der Haufen enthält.

Der Spieler, der den letzten Stein aus dem Spiel nimmt (danach sind also alle drei Haufen leer), hat gewonnen.

Beispiel: Startaufstellung 7, 5, 3 wie oben. Spieler A nimmt 3 Steine von ersten Haufen weg. Jetzt besteht der erste Haufen aus 4 Steinen, der zweite Haufen aus 5 und der dritte Haufen aus 3 Steinen. Jetzt nimmt Spieler B 3 Steine vom dritten Haufen. Damit sind nur noch zwei Haufen übrig, der erste mit 4 Steinen und der zweite mit 5 Steinen. Spieler A nimmt 2 Steine vom ersten Haufen. Jetzt liegen auf dem ersten Haufen noch 2 Steine und auf dem zweiten Haufen noch 5. Spieler B nimmt 4 Steine vom zweiten Haufen. Ergebnis: 2 Steine auf dem ersten Haufen, 1 Stein auf dem zweiten Haufen. Spieler A nimmt 1 Stein vom zweiten Haufen. Ergebnis: nur noch der erste Haufen ist übrig, mit 2 Steinen. Spieler B nimmt die verbleibenden 2 Steine vom ersten Haufen. Spieler B hat gewonnen, weil er den letzten Stein genommen hat.

Programmieraufgabe

Ich habe dieses Spiel in Java implementiert. Laden Sie die Klassen `Spiel`, `SpielGui`, `Spielzug` und `Spielzustand` aus Ilias herunter und starten Sie die Klasse `Spiel` in Ihrer Entwicklungsumgebung.

Hier kann ein Spieler gegen den Computer spielen, indem er Buttons klickt, um Elemente von einem Haufen wegzunehmen. Der menschliche Spieler beginnt. Nach jedem Zug macht der Computer automatisch seinen Zug.

In meiner Implementierung macht der Computer jeweils einen zufälligen regelkonformen Zug. Ihre Aufgabe ist es nun, den Computer so zu programmieren, dass er – wann immer möglich – gewinnt.

Ändern Sie hierfür die Methode `Spiel.macheSpielzugComputerRandom(...)` oder rufen Sie statt dieser Methode in `SpielGui.macheSpielzugComputer()` eine selbst implementierte neue Methode auf.

Schriftliche Ausarbeitung

Bitte geben Sie zusätzlich zu Ihrer Implementierung eine schriftliche Erklärung ab, warum Ihre implementierte Spielstrategie optimal ist. Erklären Sie, warum Sie in einer Spielsituation einen bestimmten Zug machen (für jede mögliche Spielsituation). Dabei können Sie gerne unterschiedliche Spielsituationen zusammenfassen, wenn Sie erklären, warum diese gleich behandelt werden können.

Insgesamt kann die ganze Erklärung ungefähr so aussehen wie die Behandlung des Beispiels aus der ersten Vorlesung (gleiches Spiel, aber mit nur einem Haufen). Ich vermute, das ergibt ungefähr ein bis zwei Seiten Text plus Bilder (falls nötig oder hilfreich).

Verfahren

Laden Sie Ihre Implementierung in Ilias hoch oder schicken Sie sie mir per E-Mail. Ihre schriftliche Erklärung können Sie ebenfalls hochladen oder per E-Mail schicken, falls sie elektronisch vorliegt, oder einfach in der Vorlesung abgeben (gerne handschriftlich).

Abgabeschluss ist der 31. Oktober 2016.

Ich korrigiere und bewerte Ihre Abgaben anschließend. Sie bekommen darauf eine Note, die 20% der Gesamtnote für die Veranstaltung ausmacht. Die Gesamtnote für die Vorlesung setzt sich zusammen aus 50% Klausur und 50% Softwareprojekten. Diese Aufgabe macht 40% der Softwareprojekte aus. Im Laufe des Semesters folgen noch zwei Softwareprojekte, deren Bewertungen zusammen die restlichen 60% ausmachen.

Stellen Sie gerne jederzeit Fragen zur Implementierung, zur schriftlichen Ausarbeitung oder zur Abgabe. Das geht natürlich in der Vorlesung oder per E-Mail.

Update Verfahren

Bitte geben Sie Ihren Code und Ihre schriftliche Ausarbeitung bis zum 31.10.2016 ab. Wenn ich diese korrigiert habe, bekommen Sie Ihre Arbeit bewertet zurück. Sie haben dann die Gelegenheit, diese innerhalb von zwei Wochen zu verbessern oder zu ergänzen und die neue Version nochmals abzugeben, um mehr Punkte zu erreichen.