

Programmierübungen

Wintersemester 2006/2007

6. Übungsblatt

29. Juni 2007

Abgabe bis Montag, 16. Juli.

Auf den Vortragsfolien der Programmierübungen oder im Skript zur Einführung in die Informatik abgedruckte Quelltexte können verwendet werden, müssen aber der Programmierrichtlinie entsprechend formatiert und kommentiert werden.

Bitte bearbeiten Sie die Übungsaufgaben in Kleingruppen mit bis zu 3 Teilnehmern. Alle Teilnehmer einer Kleingruppe müssen in der selben Übungsgruppe eingetragen sein. Ihre Abgabe muss eine Datei `contributions.txt` enthalten, in der aufgelistet ist welche Anteile der Bearbeitung von jedem einzelnen Teilnehmer erstellt wurden. Jeder Teilnehmer muss die Funktionsweise aller Teile der Bearbeitung erläutern können. Beachten Sie die Programmierrichtlinie und kommentieren Sie Ihren Quelltext. Dokumentieren Sie unbedingt Ihre Lösungsidee in den Quelltext-Kommentaren.

<http://www.iste.uni-stuttgart.de/ps/Lehre/SS2007/inf-prokurs>

Aufgabe 6.1: Othello

(20 Punkte)

Sicher kennen Sie das Brettspiel Othello. Falls nicht, so sind hier die Regeln:

- Es wird auf einem Spielbrett mit 8x8 Feldern gespielt.
- Von einem Feld f ausgehend können gegnerische Steine *umschlossen* werden. Um die Menge der umschlossenen Steine zu bestimmen wird von dem Feld f aus in jede der Richtungen N, NO, O, SO, S, SW, W, NW eine Linie gezogen. Diese Linie durchquert eine Reihe Felder. Sie endet sobald ein Feld angetroffen wird, das unbesetzt ist oder mit einem Stein der eigenen Farbe besetzt ist. Endet die Linie auf einem Feld, das mit einem eigenen Stein besetzt ist, so sind die zuvor durchquerten Felder mit gegnerischen Steinen umschlossen (siehe Abbildung 2).
- Die Startaufstellung ist wie in Abbildung 1 angegeben.
- Zwei Spieler setzen abwechselnd einen Stein ihrer Farbe, Schwarz beginnt. Falls es möglich ist, so muss gesetzt werden.
- Jeder Spieler darf nur auf ein Feld setzen, falls er dadurch mindestens einen gegnerischen Stein umschließen kann, ist dies nicht möglich, so muss der Spieler passen.
- Nachdem ein Spieler einen Stein gesetzt hat, werden alle durch diesen Zug umschlossenen Steine zur Farbe dieses Spielers umgefärbt.
- Das Spiel endet, wenn keiner der Spieler mehr einen Stein setzen kann, Gewinner ist derjenige Spieler, der mehr Steine auf dem Spielbrett besitzt. Er erhält die Differenz von eigenen Steinen und gegnerischen Steinen als Punkte plus für jedes unbesetzte Feld je 1 Punkt. Bei gleicher Anzahl Steine für beide Spieler ist das Spiel unentschieden und beide Spieler erhalten je 32 Punkte.

Ein Programm, das Othello spielt, könnte so vorgehen:

- Eingabe: eine Spielsituation sowie die Farbe c , die am Zug ist (schwarz oder weiß) und eine Berechnungstiefe $n \geq 1$
- Gesucht: ein möglichst vorteilhafter Spielzug der Farbe c
- Vorgehen:
 - Ermitteln aller Felder auf die gesetzt werden kann
 - Für jedes dieser Felder: ziehe darauf und simuliere alle Zug-Sequenzen der Länge $n - 1$ Züge, die danach folgen können
 - Bewerte die Güte der resultierenden Spielsituationen
 - Wähle einen der Züge aus, die zu einer der am besten bewerteten Spielsituationen geführt haben
 - Finden Sie eine geeignete Strategie um eine Spielsituation zu bewerten. Beachten Sie, dass dies stets eine Heuristik ist und meist nur eine grobe Schätzung der tatsächlichen Güte darstellt. Ihre Bewertungsstrategie sollte berücksichtigen, dass es Felder geben kann, die bereits im sicheren Besitz einer Farbe sind und vom Gegner nicht mehr zurückgewonnen werden können – das betrifft oft die Rand- und insbesondere die Eckfelder.

Schreiben Sie ein Programm, mit dem zwei Spieler gegeneinander Othello spielen können. Lassen Sie das Programm nach jedem Zug das aktuelle Spielbrett ausgeben sowie am Spielende das Ergebnis. Auf dem Spielbrett sollen jeweils die Felder hervorgehoben werden, auf die im aktuellen Zug gesetzt werden darf.

Jeder der Spieler sollte nach Auswahl des Benutzers entweder durch einen menschlichen Benutzer oder durch das Programm gesteuert werden können. Es sollen also Partien Mensch–Mensch, Mensch–Maschine, Maschine–Maschine möglich sein. Für jeden programmgesteuerten Spieler soll eine separate Berechnungstiefe vorgegeben werden können.

Hinweis 1: Schaffen Sie drei Komponenten:

- Spielsteuerung mit Entscheidung, ob ein Zug legal ist, Umfärben der eingeschlossenen Spielsteine nach einem Zug und Punktrechnung (**10 Punkte**)
- Ein Paket zur Kommunikation mit einem menschlichen Benutzer, also zum Einlesen des nächsten Spielzugs (**3 Punkte**)
- Ein Paket für den künstlichen Spieler (**5 Punkte**)

Hinweis 2: Die Spieler-Pakete (menschlich/rechnergesteuert) sollten gleiche Schnittstellen haben (**2 Punkte**). Zur Laufzeit haben sie die Aufgabe, der Spielsteuerung mitzuteilen, zu welchem Zug sich der jeweilige Spieler entscheidet. Verschiedene Implementierungen der Schnittstelle realisieren diese Funktionalität jedoch auf verschiedene Arten. Erwägen Sie den Einsatz von Objektorientierung (tagged types und Vererbung, siehe Skript zu „Einführung in die Informatik 1“, Kapitel 4.5, 4.6).

Hinweis 3: Tauschen Sie sich mit anderen Teams über die Gestaltung der Spieler-Schnittstelle aus. Lassen Sie bei Interesse nach Abgabeschluss Ihre Spielstrategie gegen die Ihrer Kommilitonen antreten. Tauschen Sie jedoch vor Abgabeschluss keine Implementierung aus!

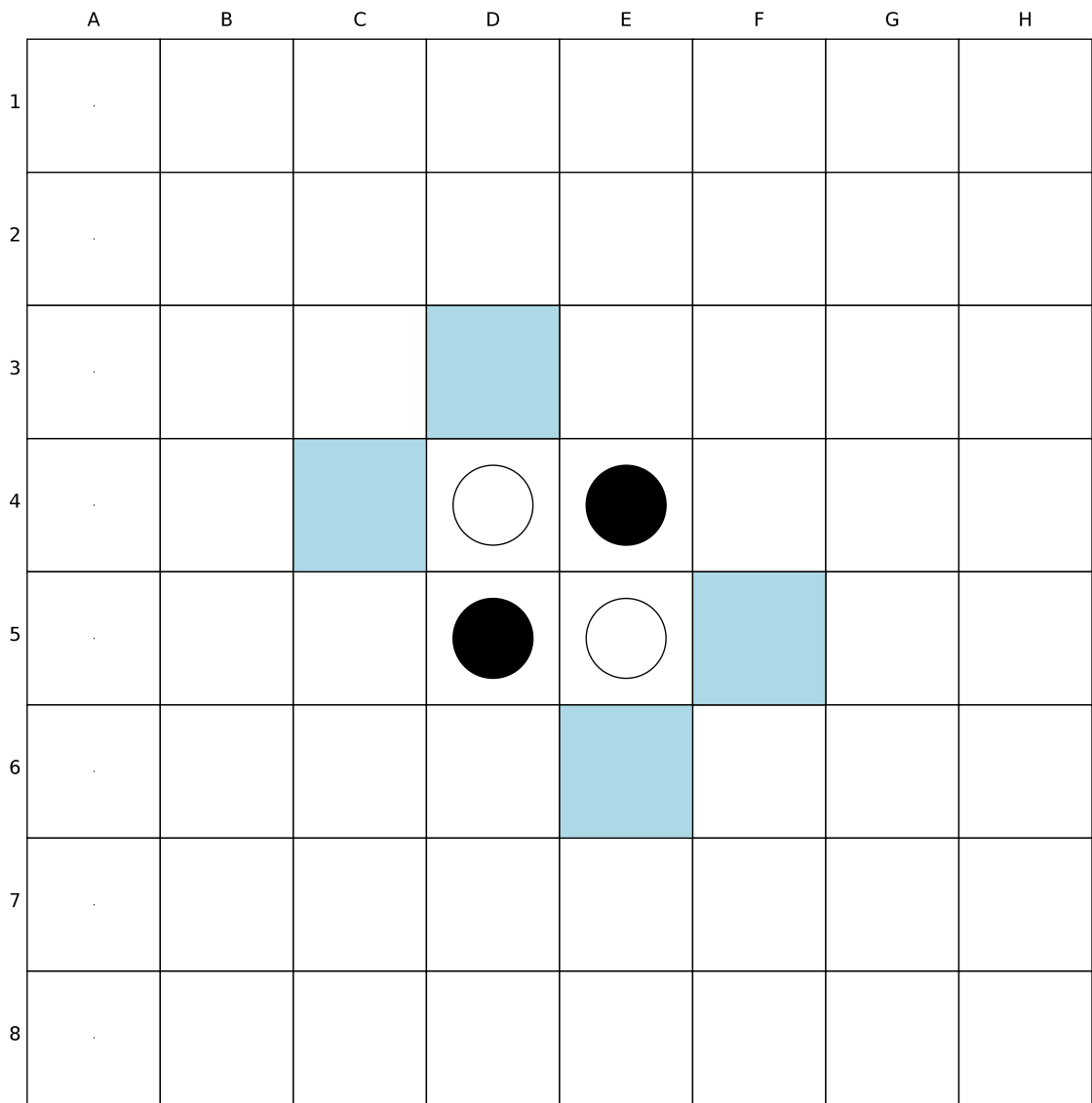


Abbildung 1: Startkonfiguration

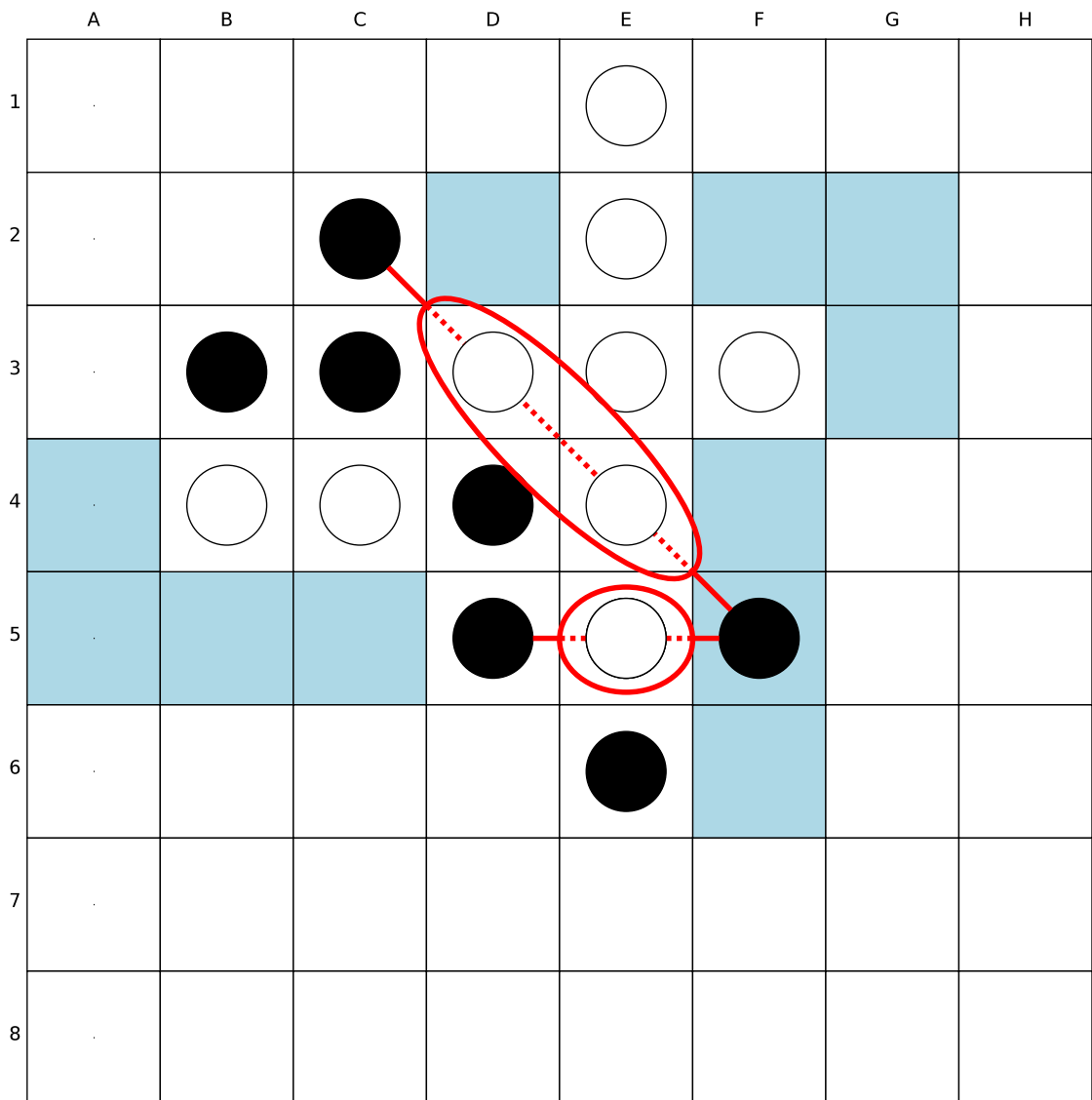


Abbildung 2: Zug von Schwarz auf F5 umschließt D3, E4 und E5