



## 31. Bundeswettbewerb Informatik 2012/2013

# Die Aufgaben der zweiten Runde

## Allgemeine Hinweise

Herzlichen Glückwunsch zum Erreichen der zweiten Runde! Hier sind die Aufgaben. Sie sind anspruchsvoll, und ihre Bearbeitung ist aufwändig. Aber die Mühe lohnt sich, denn durch Teilnahme an der zweiten Runde

- wirst du sicher sehr viel lernen;
- kannst du dich für die Endrunde qualifizieren;
- wirst du als Endrundenteilnehmer vielleicht zu weiteren Veranstaltungen eingeladen;
- kannst du mit einer guten Leistung einen Buchpreis des Verlags O'Reilly gewinnen;
- hast du am Ende eine Arbeit fertig gestellt, die du als so genannte „Besondere Lernleistung“ in die Abiturwertung einbringen kannst;
- kannst du dich (als jüngerer Teilnehmer) um die Teilnahme an einer Deutschen Schülerakademie bewerben;
- hast du die Chance auf eine Einladung zu den „Forschungstagen Informatik 2013“ des Max-Planck-Instituts für Informatik in Saarbrücken.

Wir wünschen also viel Spaß und viel Erfolg bei der Bearbeitung!

Wichtig: An dieser Runde dürfen nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben mindestens 12 Punkte erreicht oder einer Gruppe angehört haben, der dieses gelungen ist. Gruppenarbeit ist in der zweiten Runde nicht zulässig.

**Einsendeschluss ist der 15. April 2013;** es gilt das Datum des Poststempels. Bitte sende deine Lösungen wieder an den **Bundeswettbewerb Informatik, Wachsbleiche 7, 53111 Bonn.**

Es gibt drei Aufgaben. **Wichtig:** Eine Einsendung darf Bearbeitungen zu höchstens zwei Aufgaben enthalten, deren Bewertung dann das Gesamtergebnis ausmacht. Sollte eine Einsendung Bearbeitungen zu allen drei Aufgaben enthalten, werden wir zwei davon zufällig auswählen und nur diese bewerten.

Die Bearbeitung einer Aufgabe sollte zunächst eine einfache, nachvollziehbare und vollständige Lösung aller Teilaufgaben enthalten. **Pluspunkte** für eine höhere Bewertung kannst du erreichen, wenn du die Aufgabe dort, wo es möglich und sinnvoll ist, eigenständig weiterentwickelst. Sinnvoll sind inhaltliche Erweiterungen und Verbesserungen, etwa von Datenstrukturen und Algorithmen; uninteressant sind aufwändige Tricks, z. B. zur reinen Verschönerung der Benutzungsoberfläche. Begründe für jede Erweiterung, weshalb sie sinnvoll ist und ihre Realisierung eine eigene Schwierigkeit darstellt.

Grundsätzlich gelten die Gliederungs- und Dokumentationsrichtlinien der 1. Runde weiter. Zu jeder Aufgabe gehört also eine Dokumentation, die die *Lösungsideen* sowie die *Umsetzung* der Ideen in das dazugehörige Programm beschreibt. Dabei sind (halb-)formale Notationen besser als Programmausschnitte. In die ausgedruckte Dokumentation gehören auch *Beispiele* (Programmeingaben und dazugehörige -ausgaben oder *Zwischenschritte*) aus denen ersichtlich wird, wie das Programm sich in unterschiedlichen Situationen verhält. Komplettiert wird die Dokumentation durch den *Quelltext*, wobei unwichtige und automatisch generierte Teile nicht ausgedruckt werden sollen.

Denke bitte daran, dass zur Bewertung möglicherweise nur die Dokumentationen herangezogen werden können. Diese sollten also einen lückenlosen und nachvollziehbaren Nachweis des Leistungsumfangs und der Funktionstüchtigkeit der Programme geben. Der Umfang der Einsendung soll sich in Grenzen halten; eine gute Dokumentation vermittelt kurz und präzise alles Nötige, insbesondere die wesentlichen Lösungsideen. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die die Aufgabenstellung kennen, wissen müssen, um die Lösungsidee zu verstehen und die Realisierung dieser Idee nachzuvollziehen. Generell sind zwar gute und originelle Ideen entscheidend, aber die Dokumentation hat schon oft den Ausschlag für oder gegen das Weiterkommen gegeben.

Schicke uns das Papiermaterial in lesbarer Form, Schriftgröße mindestens 10 Punkt, bei Quelltext mindestens 8 Punkt. Verwende bitte lose, gelochte Blätter im Format DIN A4 (Hüllen mit Lochrand nur bei ausreichender Stabilität verwenden; keine Heftstreifen oder Mappen) und gib auf jedem Blatt Verwaltungsnummer, Vorname, Name und Seitennummer an. Die Verwaltungsnummer steht auf der Teilnahmebescheinigung der ersten Runde. Bitte gliedere deine Einsendung in (a) Allgemeines (z. B. Angaben zu technischen Voraussetzungen, persönliche Anmerkungen etc.), (b) Unterlagen zur ersten bearbeiteten Aufgabe und (c) Unterlagen zur zweiten bearbeiteten Aufgabe.

Außerdem sende uns bitte die lauffähigen Programme und ihre kompletten Quelltexte auf einer CD, DVD oder SD-Karte. Bei der Bewertung können Programme unter Windows (XP / Vista / 7), Linux und Mac OS X (10.6) ausgeführt werden.

Fragen zu den Aufgaben können per E-Mail an [bwinf@bwinf.de](mailto:bwinf@bwinf.de) oder telefonisch unter 0228-378646 (zu üblichen Arbeitszeiten) gestellt werden. Die Antwort auf E-Mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf unseren Webseiten ([www.bundeswettbewerb-informatik.de](http://www.bundeswettbewerb-informatik.de)). Auf [einstieg-informatik.de](http://einstieg-informatik.de) werden in der Community sicher wieder viele Teilnehmer über die Aufgaben diskutieren – ohne Lösungsideen auszutauschen.

Allen Teilnehmern der zweiten Runde wird bis Anfang Juni 2013 die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die im September 2013 in Kaiserslautern ausgerichtet werden wird, und zwar gemeinsam von Fraunhofer IESE, TU Kaiserslautern, Deutsches Forschungszentrum für Künstliche Intelligenz und Max-Planck-Institut für Softwaresysteme. Dort werden die Bundessieger und Preisträger ermittelt und ausgezeichnet. Bundessieger werden in der Regel ohne weiteres Auswahlverfahren in die Förderung der Studienstiftung des deutschen Volkes aufgenommen. Außerdem werden Geld- und Sachpreise vergeben. Der Rechtsweg ist ausgeschlossen.

Zum Schluss noch einmal: Viel Spaß und viel Erfolg!

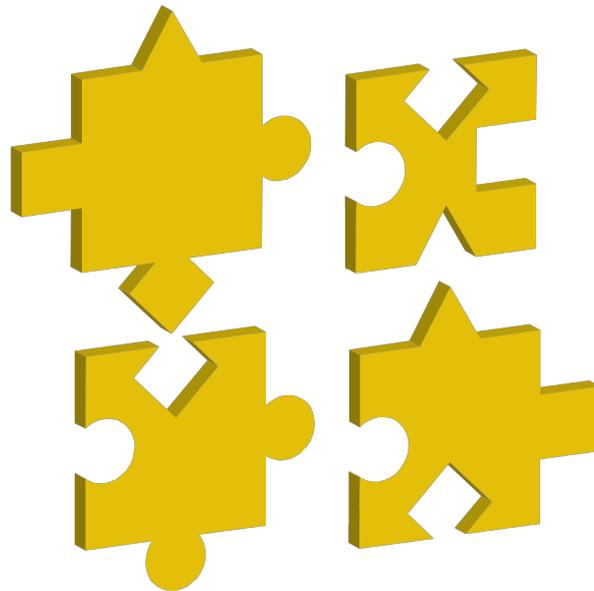
## Aufgabe 1: Komplementär

Meike liebt es, Puzzleteile zusammenzusetzen. Sie hat bereits Puzzles mit 1000 und mehr Teilen gelöst. Am schwierigsten sind dabei immer die Flächen, bei denen man vom Bildausschnitt auf einem bestimmten Teil nicht auf dessen Position im Gesamtpuzzle schließen kann, etwa bei einem gleichmäßig blauen Himmel oder einer großen grünen Wiese.

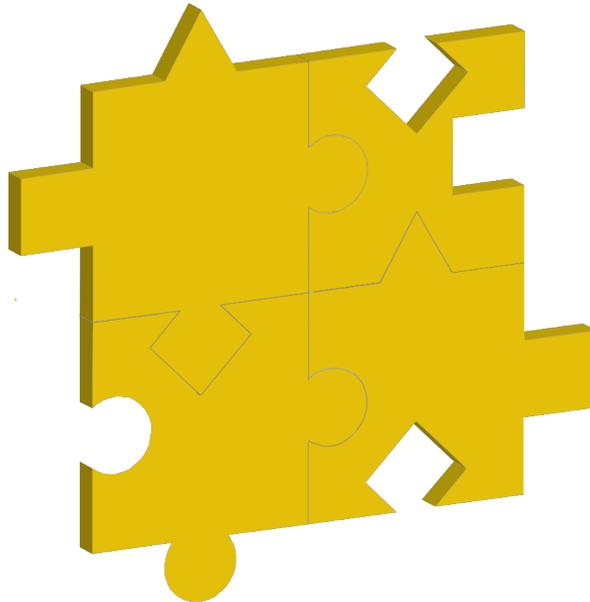
Meike überlegt, wie es wäre, wenn auf ihren Puzzles überhaupt kein Bild, sondern nur eine einheitliche Farbe aufgedruckt wäre. Ob sie wohl die Teile dann auch anders zusammensetzen könnte? Sie schneidet sich eine Menge von Puzzleteilen zurecht und versucht, diese zu einer Fläche ohne Löcher zusammenzusetzen. Sie stellt fest, dass es – bis auf Drehung – mehrere, genau eine oder gar keine Möglichkeit gibt, dies zu tun, abhängig von der Form der ausgeschnittenen Teile. Sie beschließt, dies mit dem Computer genauer zu untersuchen.

In dieser Aufgabe geht es darum, gegebene Puzzles automatisch zu lösen und neue eindeutig lösbare Puzzles zu erfinden. Die Puzzles genügen den folgenden Bedingungen:

Jedes Puzzleteil hat eine quadratische Grundform; jede der vier Seiten ist aber entweder eingebuchtet oder ausgebuchtet. Zu jedem Typ von Einbuchtung gibt es genau einen passenden Ausbuchtungstyp. Für die Aufgabe gibt es vier verschiedene Einbuchtungen und entsprechend vier verschiedene passende Ausbuchtungen. Die folgende Abbildung zeigt vier mögliche Puzzleteile.



Ein  $(m \times n)$ -Puzzle besteht aus  $m \cdot n$  Teilen, die zu einem  $(m \times n)$ -Feld zusammengesetzt sind. Dabei spielen Ein- und Ausbuchtungen am Rand keine Rolle. Alle Puzzleteile sind einmalig, d.h. es gibt keine zwei Teile, die genau gleich sind. Die Puzzleteile sind auf der Oberseite bedruckt und können daher zum Zusammensetzen zwar beliebig gedreht, aber nicht gewendet werden. Das folgende Bild zeigt eine Lösung des oben abgebildeten  $(2 \times 2)$ -Puzzles.



## Aufgabe

1. Schreibe ein Programm, das für ein gegebenes  $(m \times n)$ -Puzzle herausfindet, ob es lösbar ist, und ggf. eine Lösung ausgibt. Zum Test des Programms stehen auf dem BwInf-Server mehrere eindeutig lösbare Puzzles bereit. Das Programm soll diese einlesen und als Ausgabe berechnen, in welcher Reihenfolge und Ausrichtung die Teile zusammengesetzt werden können. Das Programm sollte zumindest das  $(2 \times 2)$ -, das  $(3 \times 3)$ - und das  $(4 \times 4)$ -Puzzle lösen können.
2. Erweitere das Programm aus Teil 1 so, dass es zufällige, eindeutig lösbare Puzzles erfindet. Die Ausgabe soll dabei im selben Format erfolgen, wie es als Eingabe für Teil 1 vorgegeben ist.

## Aufgabe 2: Marsch auf Mars

Zum Bau einer Forschungsstation werden 100 identische autonome und mobile Roboter aus einem Raumshuttle auf die Marsoberfläche abgeworfen. Das Shuttle kann die Roboter nur einzeln und in unregelmäßigen Zeitabständen abwerfen, so dass die Roboter nach dem Abwurf zufällig über der Abwurfebene verstreut sind. Dabei wird aber sichergestellt, dass jeder Roboter (außer dem ersten) nur dann abgeworfen wird, wenn sich im Umkreis von 200 m seines Aufschlagpunktes bereits ein anderer Roboter befindet. Die Roboter müssen sich nun an einem beliebigen Ort zusammenfinden.

Jeder Roboter kann sich mit einer Maximalgeschwindigkeit von 1 m/s in eine frei wählbare Richtung bewegen. Er kann auch andere Roboter in seiner Nähe erkennen und relativ zu seiner eigenen Position und Ausrichtung orten, aber dieser Mechanismus ist auf eine Reichweite von 200 m beschränkt. Die Roboter besitzen keine weiteren Fähigkeiten zur Kommunikation. Da keine Karten zur Verfügung stehen, kann auch kein Treffpunkt von vornherein vereinbart werden.

Überlege dir ein Verfahren, das die Roboter verwenden können, um sich an einem gemeinsamen Ort zu treffen. Erstelle eine Simulation, die die Bewegungen der Roboter mit deinem Verfahren von ihrer zufälligen Abwurfposition bis zum Zusammenfinden graphisch darstellt. Beschreibe und diskutiere dein Verfahren anhand kommentierter Augenblicksbilder der Simulation.

## Aufgabe 3: Tourality

Tourality ist ein Strategiespiel und wird zu zweit auf einem Spielplan mit  $20 \times 20$  Feldern gespielt. Jeder Spieler hat eine Spielfigur; außerdem gibt es Chips und Hindernisse. Auf jedem Feld kann sich entweder eine Spielfigur, ein Hindernis oder ein Chip befinden, oder es ist frei. Ein Standard-Spielplan hat 40 Chips und 40 Hindernisse. Die Hindernisse sind beliebig verteilt, aber so, dass sie keine Bereiche komplett isolieren.

Die Spieler ziehen abwechselnd. In einem Spielzug kann die Spielfigur genau um ein Feld nach oben, unten, links oder rechts bewegt werden oder stehen bleiben. Die Spielfigur darf sich nur auf Felder bewegen, die frei oder durch einen Chip belegt sind. Bewegt ein Spieler seine Spielfigur auf ein Feld mit Chip, wird der Chip entfernt und der Spieler bekommt einen Punkt gutgeschrieben.

Ziel des Spiels ist, möglichst viele Punkte zu sammeln, also mehr Chips als der Gegenspieler zu ergattern.

Zu Beginn des Spiels befinden sich die Spielfiguren auf den Startpositionen  $s_1$  und  $s_2$  des Spielfelds. Der Spieler, dessen Figur auf  $s_1$  steht, beginnt. Jedes Spiel hat zwei Runden, mit vertauschten Startpositionen, so dass jeder Spieler einmal beginnen darf.

### Aufgabe

Schreibe eine „KI“ für Tourality: ein Programm, das einen Tourality-Spieler simuliert und im Spiel gegen andere Tourality-KIs möglichst erfolgreich abschneidet. Deine KI wird schließlich in einem Turnier gegen die KIs anderer Teilnehmer antreten.

Die virtuellen Tourality-Spiele werden auf unserem Turnier-Server durchgeführt, der unter [turnier.bundeswettbewerb-informatik.de](http://turnier.bundeswettbewerb-informatik.de) erreichbar ist und ab 2.1.2013 offiziell zur Verfügung steht. Dort findest du ausführliche technische Anleitungen. Insbesondere gelten folgende Regeln:

- Dein Programm kann das gesamte Spielfeld „überblicken“, kennt also zu jeder Zeit die Positionen der Hindernisse, der Chips und beider Spielfiguren.
- Dein Programm teilt seine Spielzüge dem Turnier-Server einzeln mit.
- Dein Programm darf nur gültige Züge machen. Ein ungültiger Spielzug wird vom Turnier-Server ignoriert, wird also wie ein Zug ohne Bewegung behandelt.
- Dein Programm darf nicht zu lange „nachdenken“ und hat pro Zug nur eine bestimmte Anzahl von Zyklen zur Verfügung. Auch die insgesamt zur Verfügung stehende Rechenzeit ist begrenzt; sie wird in der Spielbeschreibung auf dem Server veröffentlicht. Ihr könnt davon ausgehen, dass sie sehr großzügig bemessen ist! Nach Ablauf dieser Zeit bleibt die Spielfigur deines Programms stehen, und das gegnerische Programm kann – im Rahmen seiner eigenen Rechenzeit – alle übrigen Chips abräumen.

## **Teste deine – Künstliche – Intelligenz!**

Damit deine Bearbeitung dieser Aufgabe in der 2. Runde gewertet wird, musst du deine KI auf den Turnier-Server hochladen. Wenn deine KI die anschließend durchgeführte grundlegende Prüfung besteht, wird sie für den Wettbewerb freigegeben.

Damit du einschätzen kannst, wie gut deine KI ist, kannst du sie nun gegen andere, bereits freigegebene KIs antreten lassen. Fordere dazu über den Turnier-Server die KIs anderer Mitspieler heraus. Die Ergebnisse dieser Spiele fließen nicht in die Bewertung ein. Du kannst nach und nach weitere, verbesserte KIs hochladen und auch sie ausprobieren.

## **Einsendung und Bewertung**

Bis zum Einsendeschluss der zweiten Runde musst du eine der von dir hochgeladenen KIs als „eingereicht“ markieren. Diese KI wird – zusammen mit deiner Dokumentation zu dieser Aufgabe – von uns nach den allgemeinen Kriterien des Bundeswettbewerbs Informatik bewertet. Außerdem nehmen alle eingereichten KIs an einem abschließenden Turnier teil, dessen Ausgang in die Bewertung eingeht. *Wichtig:* Nenne in deiner Dokumentation unbedingt deinen Benutzernamen und die Mail-Adresse, mit denen du beim Turnier-Server registriert bist, damit deiner Einsendung die richtige KI zugeordnet werden kann.