spe_ed

Computerspiele sind in der Informatik sogenannte „Technologietreiber“[1]. So gibt es eine Vielzahl von Technologien, die entweder auf Grund von Computerspielen entwickelt wurden oder zumindest durch sie eine weite Verbreitung gefunden haben. Darunter fallen unter anderem:

- Computerleistung. Das Streben nach immer besserer Grafikleistung und komplexeren Spielwelten hat sowohl die Entwicklung von Grafikkarten als auch die von CPUs befeuert.
- Eingabemethoden wie Mäuse, Joysticks oder Controller, die für präzise Eingaben bei Computerspielen notwendig sind.
- Speichermedien (und Laufwerke für diese), da für immer komplexere und realistischere Spiele immer mehr Speicherplatz benötigt wurde.
- Virtual Reality, die aktuell durch die Spieleindustrie sehr stark gefördert wird.
- Netzwerkkommunikation, damit Spieler gemeinsam an verschiedenen Orten zusammen spielen können (teilweise mehrere Millionen Spieler gleichzeitig).
- Künstliche Intelligenz[2]. Durch den Versuch, Spieler mittels künstlicher Intelligenz zu schlagen, wird die Forschung in dieser Richtung vorangebracht.

Computerspiele sind für Informatiker also nicht nur durch das eigentliche Spielen ein interessanter Zeitvertreib, sondern auch als Subjekt an sich (zum Beispiel in der Forschung). Daher wollen wir mit Ihnen zusammen dieses Jahr im InformatiCup das Computerspiel **spe_ed** spielen.

Spielbeschreibung

Die Aufgabe des diesjährigen InformatiCup ist, ein Programm zu schreiben, das selbstständig (also ohne menschliche Unterstützung) das Spiel **spe_ed** spielen kann.

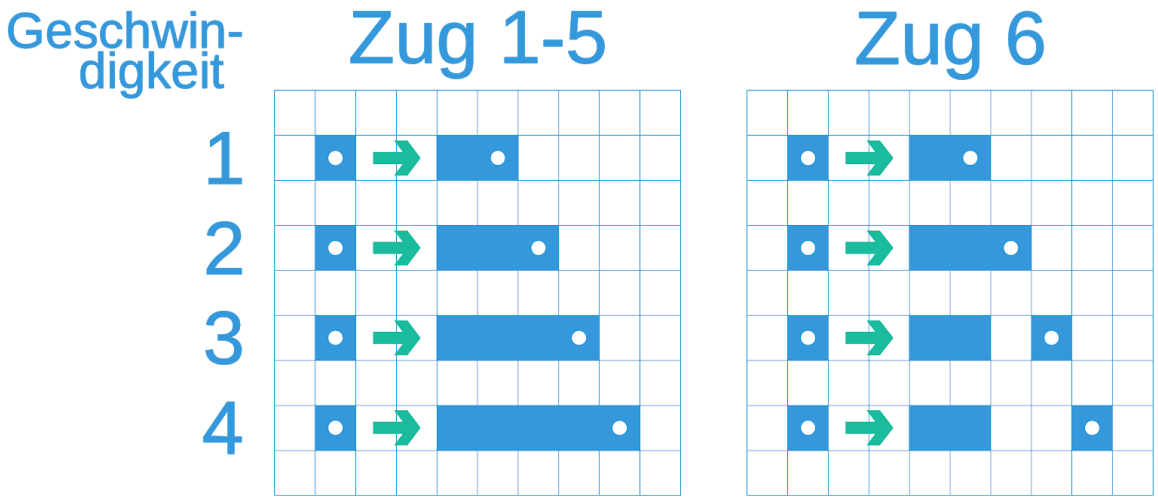
In **spe_ed** kontrollieren Sie einen Spieler, der sich über ein Spielbrett bewegt. Dabei hinterlässt dieser Spieler bei jedem Schritt eine Spur. Das Ziel des Spieles ist es, als letzter von maximal 6 Spielern auf dem Brett zu bleiben. Dabei scheidet ein Spieler aus, sobald er einen anderen Spieler oder eine bereits vorhandene Spur berührt oder den Rand des Spielfeldes überschreitet. Spuren ausgeschiedener Spieler bleiben auf dem Spielfeld.

---

[1] Siehe z. B.:
http://www.spielenutzen.de/?page_id=19
https://www.welt.de/print/wams/wissen/article131531220/Spieleindustrie-als-Technologietreiber.html
[2] https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii

Das Spiel wird in Runden gespielt. In jeder Spielrunde wird zuerst der nächste Spielzug geplant, danach führen alle Spieler gleichzeitig den geplanten Spielzug aus. Alle Spieler bewegen sich entweder horizontal oder vertikal.

Bei hohen Geschwindigkeiten ist die Spur Ihres Spielers nicht mehr konsistent. Jede sechste Spielrunde entsteht ein Loch in der Spur aller Spieler mit Geschwindigkeit größer gleich 3 auf allen Feldern außer dem ersten und dem letzten Feld der Bewegung (als ein Loch der Größe „Geschwindigkeit - 2“).



Tipp: Nutzen Sie die Löcher zu Ihren Vorteil. Sie können sowohl als Loch zum Durchgehen genutzt werden, als auch um über bereits bestehende Spuren zu springen!

Mögliche Spielzüge

Folgende Befehle stehen Ihnen jede Spielrunde zur Verfügung:

- `turn_left`: Dreht Ihren Spieler um 90° nach links (also z. B. von `up` nach `left`).
- `turn_right`: Dreht Ihren Spieler um 90° nach rechts (also z. B. von `up` nach `right`).
- `slow_down`: Verringert die Geschwindigkeit Ihres Spielers um 1. Sie scheiden aus, falls Sie die Minimalgeschwindigkeit von 1 unterschreiten.
- `speed_up`: Erhöht die Geschwindigkeit Ihres Spielers um 1. Sie scheiden aus, falls Sie die Maximalgeschwindigkeit von 10 überschreiten.
- `change_nothing`: Ändert diese Runde weder die Geschwindigkeit noch die Ausrichtung Ihres Spielers.

API-Schlüssel

Für die Teilnahme an den Spielen wird ein API-Schlüssel benötigt. Diesen erhalten Sie mit der Anmeldung Ihres Teams per E-Mail.

Der API-Schlüssel ermöglicht Ihnen die Teilnahme, allerdings kann jeder API-Schlüssel nur in einem Spiel gleichzeitig verwendet werden. Sorgen Sie deshalb bitte unbedingt dafür, dass Ihr API-Schlüssel geheim bleibt.

Jedes Team erhält ein Pseudonym, welches sich regelmäßig ändert (variabler Zeitraum). Dadurch ist es möglich, einzelne Teams über einen kurzen Zeitraum zu beobachten.

Online-Version zum Testen

Eine Online-Version zum Testen des Spiels finden Sie unter https://phinau.de/spe_ed/. Hier können Sie das Spiel ausprobieren und ein Gefühl dafür bekommen, bevor Sie mit der Implementation Ihrer Software beginnen. Hierbei handelt es sich noch nicht um eine vollständige Lösung, sondern nur um eine Benutzeroberfläche, mit der Sie **spe_ed** spielen können.

Eingabe / Ausgabe

Die Kommunikation mit dem Spielserver findet über Websockets (verschlüsselt) statt. Benutzen Sie dafür den Endpoint mit der URL `wss://msoll.de/spe_ed?key=KEY` (siehe API-Schlüssel). Jeder Schlüssel kann zeitgleich nur an genau einem Spiel teilnehmen. Nehmen Sie bereits an einem Spiel teil, so wird der Status-Code `429 Too Many Requests` zurückgegeben.

Alle Spieler landen in einer Lobby. Sobald genug Spieler zusammen gekommen sind, spätestens aber nach 5 Minuten, beginnt ein Spiel. Bis zur ersten Spielrunde findet keine Kommunikation über den Websocket statt.

In jeder Spielrunde wird über den Websocket der Spielstatus als Nachricht in JSON (UTF-8, hier kommentiert und formatiert) verschickt:

```
{
  "width": 40, // Breite des Spielfeldes
  "height": 40, // Höhe des Spielfeldes
  // cells enthält die aktuelle Belegung des Spielfelds.
  // cells[0] enthält die Werte an den Koordinaten (0, 0) (linke
  // obere Ecke) bis (0, width - 1), cells[height - 1] die Werte
  // an den Koordinaten (height - 1, 0) bis (height - 1, width -
  // 1). Mögliche Werte sind:
  //   0      nicht belegt
  //   n > 0  belegt durch Spieler mit ID = n
  //   -1     gleichzeitig durch mehrere Spieler belegt
  "cells": [[0, 0, 1, 0, 0, 0, 2, 0, 1, ...], ...],
  "players": { // Spielerinformationen
    "1": { // ID des Spielers (String!)
      "x": 5, // x-Position des Spielers
      "y": 23, // y-Position des Spielers
      // Richtung: "left", "right", "up" oder "down"
      "direction": "up",
      "speed": 1, // Geschwindigkeit, 1-10
      "active": true // wenn false: ausgeschieden
      // Pseudonym, nur vorhanden wenn running = false
      "name": "Qualm-Quellcode-Vorteil"
    },
    "2": { "x": 4, "y": 6, ... },
    ...
  },
  "you": 2, // Spieler-ID, natürliche Zahl > 0
  "running": true, // wenn false: Spiel beendet
  // Deadline, innerhalb derer der Server den Spielzug
  // erhalten haben muss. Format: RFC 3339 mit Zeitzone UTC
  "deadline": "2020-10-01T12:00:00Z"
}
```

Sie müssen, solange Sie noch aktiv im Spiel sind, nach dem Erhalt der Spielinformationen innerhalb der Deadline exakt eine Nachricht in JSON (UTF-8) mit dem nächsten Befehl (siehe Spielbeschreibung) über den Websocket zurückschicken:

```
{"action": "turn_left"}
```

Diese beiden Schritte wiederholen sich, bis das Spiel beendet ist.

Jede Protokollverletzung (auch mehrere Befehle innerhalb einer Spielrunde zu senden) führt dazu, dass Sie aus dem aktuellen Spiel scheiden und es somit verlieren.

Serverzeit

Der Server stellt einen GET-Endpunkt mit der URL `https://msoll.de/spe_ed_time` bereit, der die aktuelle Serverzeit (UTC) in JSON (UTF-8) zurückgibt. Bitte nutzen Sie diesen, um Ihre Systemzeit mit der Serverzeit abzugleichen, da für die Einhaltung der Deadline die Serverzeit maßgeblich ist.

Bewertung

Erwartet wird eine Software, die das oben beschriebene Spiel **spe_ed** selbstständig spielen kann und dabei möglichst oft gegen andere Spieler gewinnt. Die Software muss dabei ohne Unterstützung von Personen laufen. Die reine Gewinnrate ist dabei aber nur ein Teil der Bewertung.

Die Jury muss Ihre Software erstellen und laufen lassen können. Bitte erstellen Sie ein Dockerfile[3], auf dessen Grundlage auf einem Referenzsystem (Windows 10, Ubuntu 20.04, Mac OS X) ein lauffähiges Docker-Image mit einem Aufruf an `docker build`[4] gebaut werden kann. Gerne können Sie zusätzlich ein bereits gebautes Docker Image einreichen. Beim Start werden dem Container Umgebungsvariablen[5] übergeben (`URL`: URL des Endpoints (siehe Eingabe / Ausgabe); `KEY`: Api-Schlüssel, TIME_URL: URL zur Serverzeit), die Ihre Software nutzen soll, um an genau einem Spiel teilzunehmen und sich anschließend zu beenden. Ein Beispiel finden Sie unter https://github.com/informatiCup/InformatiCup2021/tree/master/docker_example.

Zusätzliche Erweiterungen können auch gerne unabhängig vom Docker Image zur Verfügung gestellt werden, solange Ihre Lösung wie oben beschrieben lauffähig ist. Neben der Software erstellen Sie bitte eine Ausarbeitung, die Ihren theoretischen Lösungsansatz beschreibt. Ihre Einreichung wird ganzheitlich bewertet. Im Anhang finden Sie eine Checkliste der Bewertungskriterien. Nutzen Sie diese Liste, um die Vollständigkeit Ihrer Lösung zu überprüfen.

Außerdem

Die FAQs zum laufenden Wettbewerb sowie Beispieleingaben und -ausgaben finden Sie in Kürze online auf https://github.com/InformatiCup/InformatiCup2021.

---

[3] https://docs.docker.com/engine/reference/builder/
[4] https://docs.docker.com/engine/reference/commandline/build/
[5] https://docs.docker.com/engine/reference/run/#env-environment-variables

Checkliste der Bewertungskriterien

Theoretischer Ansatz

Der theoretische Ansatz muss in einer Ausarbeitung, die zusammen mit der Implementierung eingereicht wird, dargestellt werden. Bewertet werden sowohl der Inhalt als auch die Form.

Theoretische Ausarbeitung

Die theoretische Ausarbeitung soll die verwendete Theorie beschreiben.

- ❏ **Hintergrund**: Der Hintergrund der Lösung. Welche theoretischen Ansätze wurden verwendet? Warum wurden diese verwendet?
- ❏ **Auswertung**: Wie gut ist die Qualität der Lösung? Nach welchen (wissenschaftlichen) Kriterien wurde bewertet? Warum wurden diese Kriterien verwendet?
- ❏ **Diskussion**: Wie „gut“ (auch außerhalb der reinen Qualität) ist die Lösung? Was für Probleme gibt es noch? Wie lässt sich die Lösung praktisch einsetzen?
- ❏ **Quellen**: Wurden (wissenschaftliche) Quellen richtig und angemessen verwendet?

Formalien

Eine gute Form ist entscheidend für die Lesbarkeit einer Ausarbeitung. Beachten Sie deshalb neben dem reinen Inhalt Ihrer Ausarbeitung auch einige Formalien.

- ❏ **Rechtschreibung**: Rechtschreibung und Grammatik sind korrekt.
- ❏ **Lesefluss**: Es gibt einen Lesefluss in der Ausarbeitung.
- ❏ **Layout**: Das Dokument hat ein einheitliches Layout. Dieses kann frei gewählt werden, darf aber nicht den Lesefluss stören.
- ❏ **Zitate**: Es wird richtig und einheitlich zitiert.
- ❏ **Quellenangaben**: Quellen sind richtig und einheitlich angegeben.

Softwarearchitektur und -qualität

Da eine etablierte Softwarearchitektur nur mit hohem Aufwand zu ändern ist, sollte Sie besonders gründlich durchdacht und ausführlich begründet werden. Ausgewählte Aspekte der Softwarequalität sind für die Bewertung von besonderer Bedeutung. Gerne dürfen auch hier nicht genannte Aspekte aus den sehr weiten Feldern Softwarearchitektur und -qualität beleuchtet werden.

- ❏ **Architektur**: Beschreibung der Komponenten und deren Beziehungen
- ❏ **Software testing**: Begründetes Konzept, Umsetzung
- ❏ **Coding conventions**: Begründetes Konzept, Umsetzung
- ❏ **Wartbarkeit**: Mit welchem Aufwand kann das System angepasst werden?

Handbuch

Das Handbuch beschreibt den Installationsprozess der Lösung, insbesondere die Abhängigkeiten. Zudem wird die Benutzung der Software erläutert.

- ❏ **Bauanleitung**: Erwartet wird eine Dockerfile, mit dem ein lauffähiges Docker Image auf einem Referenzsystem (Windows 10, Ubuntu 20.04, Mac OS X) erstellt werden kann. Bitte stellen Sie sicher, dass das Docker Image allein mit Hilfe von `docker build` erstellt werden kann (siehe Bewertung). Für Erweiterungen, die nicht Teil des Docker Image sind, erstellen Sie bitte eine zusätzliche Bauanleitung für ein Referenzsystem.
- ❏ **Benutzeranleitung (Erweiterungen):** Eine kurze Anleitung, wie eventuelle Erweiterungen auszuführen und zu bedienen sind.

Qualität der Lösung

Die Qualität der Lösung wird anhand der gewonnenen Spiele aus einer Menge von Testspielen von der Jury ermittelt.

- ❏ **Punktzahl**: Wie viele Punkte sammelt Ihre Software?
- ❏ **Strategien**: Zeigt Ihre Lösung interessante Spielzüge / Strategien?

Erweiterungen

Erweitern Sie Ihre Lösung über die Anforderungen der Aufgabenstellung hinaus. Seien Sie kreativ!

Präsentation

Im InformatiCup-Finale werden die besten Lösungen vor einer Fachjury präsentiert.

- ❏ **Foliendesign**: Sind die Folien ansprechend? Lenken die Folien nicht vom Inhalt der Präsentation ab?
- ❏ **Vortragsstil**: Weckt der Vortragsstil Interesse an der Präsentation?
- ❏ **Verständlichkeit**: Ist der Vortrag verständlich? Wird der Hintergrund der Lösung in einem angemessenen Tempo erklärt? Wird nötiges Vorwissen geschaffen?
- ❏ **Reaktion auf Nachfragen**: Können Nachfragen beantwortet werden?