Semesterprojekt “MAES 2017”
— *Kollisionsfreie Mobilität* —

Martin Fränzle

18. Dez. 2017

Da die Vermeidung von Kollisionen zwischen Verkehrsteilnehmern ein allgemeines Ziel der Fahrzeugführung ist, ergibt sich als eine immer wiederkehrende Teilaufgabe in der automatischen Fahrzeugführung die Berechnung kollisionsfreier Pfade zum Ziel oder Etappenziel. Diese Aufgabenstellung findet sich in zwei Spielarten: In der häufiger betrachteten, technisch weitaus einfacheren Form als reine Planungsaufgabe, in der zwischen statischen Hindernissen oder ggf. auch dynamischen Hindernissen mit vorhersehbarer Trajektorie lediglich ein kollisionsfreier Pfad gefunden werden muss, sowie in der anspruchsvolleren Variante, in der eine reaktive Strategie der kollisionsfreien Fortbewegung in einer unkooperativen Umgebung mit sich nur bedingt vorhersehbar bewegenden Fremdobjekten konstruiert werden muss. In diesem Semesterprojekt wollen wir uns der letzteren Variante zuwenden, welche im Gegensatz zur einfacheren Pfadplanungsvariante auch für autonomes Fahren in nicht-geschützten, also der Öffentlichkeit frei zugänglichen Umgebungen geeignet ist. Ziel der Konstruktion ist hierbei die Bereitstellung so genannter reaktiver Strategien, welche jeden Zug der unkooperativen Umgebung mit einer geeigneten Reaktion parieren können. Offensichtlich können — und wollen — wir für die Konstruktion entsprechender Strategien Variationen der in der Vorlesung vorgestellten spieltheoretischen Methoden verwenden.

1 Problemstellung

Ihr Problem besteht darin, einen mit einer (eigenartigen und frei definierbaren) Kinematik versehenen Serviceroboter in einem Umfeld einzusetzen, in dem Sie mit spielenden Kindern zu rechnen haben. Da es sich um wahre Spielkinder handelt, bewegen sich diese nicht völlig frei, sondern nach den von ihnen selbst definierten Regeln eines Hinkespiels. Während Ihr Roboter sich also in dem geometrisch abgeschlossenen und mit statischen Hindernissen durchsetzten Bewegungsraum in Richtung seines Missionsteilziels fortbewegt, muss er ständig auf gemäß diesen Regeln, ansonsten aber unverhersehbar vorbeihüpfenden Kindern rechnen und diesen gegebenenfalls geeignet ausweichen. Er darf hierbei allerdings das Missionsziel nicht aus den Augen verlieren, sondern muss gleichzeitig dessen Erfüllung sicherstellen, welche im endlos iterierten aufeinanderfolgenden Besuch einer Reihe von Missionspunkten besteht. Naturgemäß hat der Roboter nur einen beschränkten Energievorrat an Bord, so dass zusätzlich der rechtzeitige zwischenzeitliche Besuch einer elektrischen Ladestation notwendig werden kann. Abb. 1 zeigt einen entsprechenden Arbeitsraum. Wir nehmen hier und im Folgenden stets an, dass der Arbeitsraum diskret ist und genau ein Kind enthält. Ihre Aufgabe besteht darin, automatisch eine garantiert kollisionsfreie und im Sinne der genannten Missionsziele zielführende Strategie für Ihren Roboter zu konstruieren, sofern eine solche existiert, bzw. anderenfalls deren Nichtexistenz zu detektieren und mitzuteilen.

Eine Kollision liegt hierbei genau dann vor, wenn der Roboter bei seiner eigenen Bewegung die durch das Kind belegte Zelle oder eine unmittelbare Nachbarzelle durchquert oder umgekehrt das Kind bei seiner eigenen Bewegung die durch den Roboter belegte Zelle oder eine unmittelbare Nachbarzelle durchquert, wobei Kinder aufgrund baulicher Maßnahmen die Zellen und unmittelbaren Nachbarzellen von Ladestationen nicht betreten können und mithin während des Ladens keine Kollisionen auftreten können. Die Bewegungen von Roboter und Kind geschehen der Einfachheit halber strikt alternierend, wobei der Roboter den ersten Zug hat. Es versteht sich von selbst, dass die Außenmauern des Bewegungsraums sowie die Hindernisse sowohl für den Roboter als auch für das Kind undurchdringlich sind.

Die Bewegungsmuster des Roboters mit seiner höchst eigenartigen Kinematik und des Kindes mit den selbstgesetzten Regeln seines Hinkespiels werden dabei jeweils durch kommaseparierte Listen von Bewegungsmustern der Form $(\texttt{l} \mid \texttt{r} \mid \texttt{u} \mid \texttt{d})^*$ angegeben. Ein Bewegungsmuster `lur` bedeutet dabei, dass das entsprechende Objekt innerhalb eines Zuges eine Bewegung ausführen kann, welche zunächst eine Zelle nach links, dann eine nach oben, dann wieder eine Zelle nach rechts fortschreitet. Diese Bewegung ist in

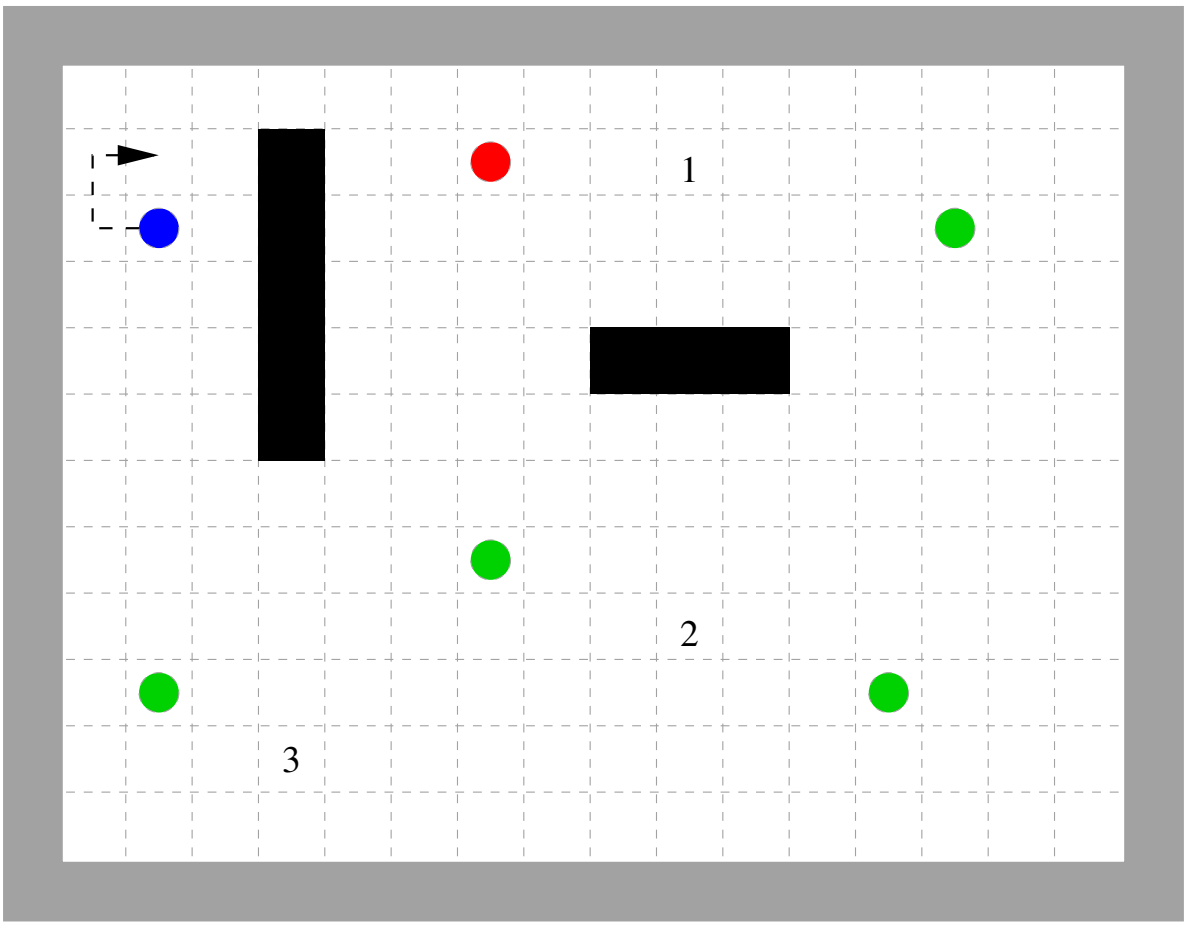

Abbildung 1: Beispiel eines Arbeitsraums: Der Serviceroboter (blau) bewegt sich zwischen den Außenwänden des Bewegungsraums (grau) und Hindernissen im Bewegungsraum (schwarz), muss dabei aber dem spielenden Kind (rot) ausweichen. Er muss hierbei die Zielpunkte 1 bis 3 immer wieder in ihrer natürlichen Reihenfolge $1, 2, 3, 1, 2, 3, \ldots$ anfahren und kann zwischendurch bei Bedarf an den Ladestationen (grün) nachtanken. Der gestrichelte Pfeil visualisiert das Bewegungsmuster `lur`.

Abb. 1 dargestellt. Die Liste `ld,,u,r` erlaubt mithin die vier Bewegungsmöglichkeiten links und dann abwärts, Stillstand, aufwärts, sowie rechts. Der Roboter kann ohne zwischenzeitliche Aufladung höchstens N Elementarschritte der Form `l, r, u` oder `d` durchführen. Eine Aufladung erfolgt per Schnellladung und erfordert lediglich einen Aufenthalt an der Ladestation für einen Zeitschritt.

2 Aufgabenstellung

Die gesamte Aufgabenstellung gliedert sich in die folgenden aufeinander aufbauenden Teilaufgaben:

1. Definieren Sie eine Eingabesprache zur Definition von rechteckigen Arbeitsräumen mit rechteckigen Hindernissen und mit Ladestationen und Zielpunkten wie in Abb. 1 oder entwickeln Sie alternativ einen entsprechenden graphischen Editor. Die Größe der möglichen Arbeitsbereiche darf dabei auf 64×64 beschränkt werden.

   Entwickeln Sie hierauf aufbauend ein Programm zur Definition von Arbeitsräumen, Bewegungsmustern für Roboter und Kind, Eingabe der Ladekapazität N des Roboters, sowie Definition der Initialpositionen von Roboter und Kind.

2. Entwickeln Sie ein Programm, welches für ein gemäß (1.) definiertes Problem eine Strategie zum nur einmaligen kollisionsfreien Besuch der Zielpunkte $1, 2, \ldots, k$ in ihrer natürlichen Reihenfolge berechnet oder die Nichtexistenz einer solchen Strategie erkennt. Ladekapazitäten sollen hierbei

zunächst einmal keinerlei Rolle spielen, d.h. der Roboter kann sich unbeschränkt im Wechsel mit dem Kind bewegen.

Für die Strategiekonstruktion sollen die spieltheoretischen Methoden aus der Vorlesung angepasst werden. Erläutern Sie den resultierenden Algorithmus bitte ausführlich in Ihrem Bericht.

3. Entwickeln Sie ein Programm, welches die berechnete Strategie animiert, indem es den Arbeitsraum graphisch darstellt und hierauf die Bewegungen eines der Strategie folgenden Roboters in Reaktion auf ein randomisiert seinen Hüpfregeln folgendes Kind anzeigt.

4. Erweitern Sie Ihre Animation dahingehend, dass die verfügbare Ladekapazität N des Roboters sowie die Wirkung von Aufladevorgängen Berücksichtigung finden, indem die jeweils verfügbare momentane Restkapazität angezeigt wird und ein entladener Roboter unrettbar im Feld stehen bleibt.

5. Erweitern Sie die Berechnung der Gewinnstrategie aus (2.) dahingehend, dass die verfügbare Ladekapazität N des Roboters berücksichtigt wird und bedarfsgerechte Nachladung in die Strategie einbezogen wird. Hierzu soll der Spielgraph nicht verändert (also insbesondere nicht um eine die Restkapazität darstellende Dimension angereichert) werden, sondern das Problem durch geeignete Strategiekonstruktion auf dem Spielgraphen aus (2.) gelöst werden.

6. Erweitern Sie die Berechnung der Gewinnstrategie aus (5.) dahingehend, dass der unendliche häufige Wiederbesuch der Zielpunkte $1, 2, \ldots, k$ in ihrer natürlichen Reihenfolge erzielt wird.

3 Bericht über das Semesterprojekt

Für eine erfolgreiche Bearbeitung des Semesterprojektes sind die folgenden Auflagen einzuhalten:

- Schreiben Sie einen gut gegliederten und leicht lesbaren Bericht, in dem Sie Ihre Ideen und Entwurfsentscheidungen darstellen. Schrecken Sie dabei nicht davor zurück, auch „Sackgassen“ und Fehlschläge zu beschreiben, insbesondere dann, wenn es sich um Grenzen der verwendeten Werkzeuge oder Methoden handelt, an die Sie während der Bearbeitung der Aufgaben gestoßen sind.

- Kommentieren Sie Ihr Vorgehen, aufgetretene Probleme und Ihre Lösungsansätze. Bewerten Sie Ihre Entwurfsentscheidungen und interpretieren Sie Ihre Ergebnisse.

- Alle Programme, Testfälle und Programmausgaben sind als zip oder tar.gz Archiv in elektronischer Form an die unten angegebenen E-Mail Adressen zu schicken.

- Dokumentieren Sie alle Ergebnisse und Probleme sowie evtl. notwendige Parameter, die für eine Nachstellung der beschriebenen Fälle notwendig wären, so dass sie leicht nachvollzogen werden können.

Der Abschlussbericht selbst sollte formal die folgenden Elemente beinhalten:

- Eine Titelseite, auf der die Namen aller Gruppenmitglieder genannt sind. Mit der jeweiligen Unterschrift ist außerdem zu bestätigen, dass die Aufgaben zu gleichen Teilen von den Gruppenmitgliedern bearbeitet wurden oder eine genaue Aufschlüsselung, wer welchen Teil beigetragen hat.

- Eine kurze Einleitung.

- Eine detailierte Beschreibung der entwickelten Algorithmen und der erstellten Porgramme gemäß den einzelnen Teilaufgaben.

- Eine Zusammenfassung der wichtigsten Ergebnisse des Semesterprojektes.

- Eine tabellarische Übersicht darüber, welches Gruppenmitglied an welcher Teilaufgabe die Hauptverantwortung übernommen hat und wieviel Zeit die einzelnen Gruppenmitglieder jeweils in diese Teilaufgabe investiert haben.

Der Abschlussbericht sollte (ohne einen möglichen Anhang, in dem zusätzliche Screenshots, Programmausgaben, Programmcode u.ä. angefügt werden können) zwischen 10 und 15 Seiten lang sein. Geben Sie den Abschlussbericht in ausgedruckter Form bis **spätestens zum 28. Februar 2017 um 12:00 Uhr** bei Martin Fränzle (OFFIS, D 119/120) ab. Alternativ können Sie den Bericht auch im Sekretariat bei Frau Kathrin Kuper (OFFIS, D 121) abgeben. Zusammen mit den anderen elektronischen Abgabebestandteilen (Programme, Testfälle, usw. — s.o.) wird auch eine elektronische Version des Berichtes erwartet. Die elektronische Abgabe ist an alle unten genannten E-Mail-Adressen zu schicken.

> Bitte beachten Sie, dass später abgegebene Berichte nicht mehr akzeptiert werden können, es sei denn zuvor wurde eine explizite Erlaubnis eingeholt. Das Gleiche gilt für die elektronischen Versionen.

Generelles

Die Arbeit sollte in Gruppen mit höchstens drei TeilnehmerInnen bearbeitet werden. Es ist nicht vorgesehen, das Projekt alleine bzw. in größeren Gruppen durchzuführen. Ausnahmen sind in begründeten Fällen möglich, bedürfen aber der vorherigen Absprache!

Bitte beachten Sie:

- Die Bewertung wird wie angekündigt erfolgen.
- Sie werden die Ergebnisse, die Sie in Ihrem Bericht dargestellt haben, im Abschlusskolloquium verteidigen müssen.
- Dieses Kolloquium wird folgendermaßen strukturiert sein:
  - Sie präsentieren Ihre Bearbeitung der Semesteraufgabe in einem Vortrag von ca. 30 Minuten Dauer. Sorgen Sie am besten selbst dafür, die Präsentation gleichmäßig auf alle Gruppenmitglieder aufzuteilen. Sie können gerne eine elektronische Präsentation vorbereiten.
  - Ihr Vortrag kann dabei durch die Prüfenden durch Rückfragen unterbrochen werden, um Unklarheiten zu beseitigen oder tiefergehendes Verständnis zu ergründen.
  - Anschließend findet eine weitergehende Diskussion von ebenfalls ca. 30 Minuten Dauer statt. Diese kann sich auch auf weitere Vorlesungsinhalte erstrecken.
  - Die Prüfungstermine werden zwischen Veranstaltern und Ihnen vereinbart. Die Prüfung kann frühestens einige Tage nach der Abgabe des Semesterprojektes erfolgen.
- Eine Verschiebung der Abgabefrist kann nur in gut begründeten Ausnahmefällen gewährt werden. Hierzu bedarf es zwingend einer Absprache!

Falls noch Fragen oder Probleme bei der Bearbeitung der Aufgaben auftreten sollten, können Sie sich gerne per Mail an die folgenden E-Mail Adressen wenden, an die auch die elektronische Abgabe erfolgen soll:

- `peter.nazier.mosaad@uni-oldenburg.de`,
- `martin.fraenzle@informatik.uni-oldenburg.de`.

Bitte stellen Sie jedoch sicher, dass Sie zuvor einen Blick in das Stud.IP werfen, wo FAQs oder Erläuterungen verfügbar gemacht werden.