



Hendrik Krauß

Finden oder verlieren

Ganz unterschiedliche Ansätze bewährten sich im c't-Sim-Wettbewerb

Beim Programmierwettbewerb zum c't-Bot-Projekt schickten die Tüftler ihre Roboter durch unbekannte Labyrinth. Pragmatische Ansätze, die stracks zum Ziel führten, gab es im Teilnehmerfeld ebenso wie verschwurbelte Implementierungen, die sich heillos verzettelten oder gleich abstürzten.

Seit dem Beginn des c't-Bot-Projekts im Frühjahr puseln Robotikinteressierte inner- und außerhalb der Redaktion an ihren c't-Bots herum. Vom 18. bis 20. Oktober stellten sich ihre Kreationen der Herausforderung, den schnellsten Weg durch ein unbekanntes Labyrinth zu finden.

Die von den Teilnehmern eingesandte, in C programmierte Steuerungssoftware würde normalerweise auf dem Mikroprozessor der c't-Bot-Hardware laufen, über Sensormesswerte nachsinnen und auf dieser Basis den Wichtel durch seine Umgebung lotsen. Im Wettbewerb waren aber nicht reale Sensoren, sondern der in Java geschriebene Simulator c't-Sim die Quelle der Messwerte. Er machte der Steuerungssoftware weis, sie würde einen Hardware-Bot durch ein reales Labyrinth chauffieren.

Martin Henschke, dessen rollender Dreikäsehoch den Namen n++ trägt, gewann einen goldfarbenen eloxierten c't-Bot für den



Die drei Bestplatzierten gewannen c't-Bot-Bausätze in Gold, Silber und Bronze.

Turniersieg. Seine Grundidee, beharrlich dem Wandverlauf zu folgen, war von der in c't vorgestellten Referenzimplementierung geliehen. Da nach den Spielregeln Start und Ziel an der Außenwand des Labyrinths lagen, führt der Weg an ihr entlang mit Sicherheit zum Ausgang. n++ stellte sich jedoch gewitzter an als seine Inspiration: So fuhr er etwa immer in einer leichten Schlangenlinie an der Wand entlang. Die meisten Sensoren des c't-Bot sind nach vorne gerichtet. Wegen seines Tunnelblicks dreht die Referenzimplementierung den Bot alle Nase lang um 90 Grad, um nach der Wand zu gucken – doch die ständige Anhalterei zum Tänzchen kostet Zeit. Der Schaukelstil von n++ kombiniert das notwendige Übel der Seitenblicke mit Prestissimo beim Vorankommen.

Die teilnehmenden Bots traten paarweise gegeneinander an. Beide dampften von der Südwand des Labyrinths los; der Startpunkt des einen Bots lag in der Westhälfte, der des Konkurrenten symmetrisch in der Osthälfte. Das Ziel befand sich mittig am nördlichen Rand. Fand es niemand innerhalb der Spielzeit, gewann die Runde, wer näher am Ziel stand. Das kam nicht selten vor: Neben Programmierfehlern wie Rohrbrüchen in der Pointer-Klempnerei verschluckte sich auch manche Steuerungssoftware aufgrund konzeptioneller Probleme. Schon die langgezogene Grube quer vor dem Ziel der Vorrunde löste bei einem Teilnehmer verheerende Irritation aus.

Mit jeder Runde wurden die Labyrinth kniffliger. Das Turnier

forderte, dass die Steuerungssoftware unterschiedliche Situationen meistert – kann im K.-o.-System ein einziger Lapsus doch gleich den Rauswurf bedeuten. Es spricht also etwas für die garantierte ins Ziel findenden Wandfolger. Freilich erkaufen sie sich ihre Robustheit mit Abstrichen bei der Effizienz: Besonders in Labyrinth, die viel breiter sind als lang oder die viele von der Außenwand abzweigende Schnörkel aufweisen, weicht der risikolose Weg enorm von der kürzesten Strecke ab.

Wandlos glücklich

Der von Josef Schützenberger aus Österreich ertüftelte Austro-Bot orientiert sich daher nicht spießbürgerlich an der Wand; stattdessen versucht er zunächst herauszubekommen, wo Norden ist. Erfasst er auf nur einer Seite des Startfelds ein Hindernis, so muss das aufgrund der Spielregeln die südliche Außenwand des Labyrinths sein. Andernfalls fährt er mit einer Schätzung los und beobachtet die Lichtsensoren, bis er eine der Lampen erkennt, die gemäß den Regeln die Ecken des Parcours markieren. Hat er sich orientiert, fuhrwerk er schlankweg nach Norden, Hindernissen natürlich ausweichend. Austro-Bot macht sich das Leben einfacher, indem er nach und nach eine Karte aufbaut, damit er sich an schon besuchten Stellen nicht mehr langwierig den Überblick verschaffen muss.

Seine Strategie erlaubte ihm einen Durchmarsch bis ins Finale, sorgte dort jedoch für eine Überraschung. Es zeigte sich, dass die Integration der beiden Ideen – das Orientieren an den Lichtern und die Karte – an einer entscheidenden Stelle hakte. Austro-Bot startete nach Norden guckend, unterlag jedoch zu-

nächst dem Irrglauben, er sei nach Westen ausgerichtet. Somit trug er die freien Felder im Osten auf seiner Karte im Norden ein. Nach kurzer Fahrt stellte er seine wirkliche Orientierung fest, aktualisierte aber seine Karte nicht – sie zeigte ihm den nördlichen Bereich weiterhin als frei. Er traute in dieser Situation seinem Gedächtnis zu viel, seinen Augen zu wenig und fiel in Verwirrungsstarre. Ein Fauxpas, der ihn jedoch nicht mehr um den silbernen Bot brachte.

Ralph Wetzels CreyBot, der den dritten Platz belegte, verfolgte wiederum einen anderen Ansatz: Er suchte das Labyrinth mit dem A*-Algorithmus systematisch ab, beginnend mit Abzweigungen, die nah am Ziel liegen. Dieses vermutete er zwanglos in Richtung seiner Startorientierung. Auch wenn sich diese Meinung später als falsch erweist, hält er sie tapfer bis zum Ende durch. Diese Haltung macht ihn nicht nur als US-Präsidenschaftskandidaten interessant, sondern vermeidet auch zusätzliche Umstände fürs Drehen oder Zurücksetzen halbfertiger Karten – was Austro-Bot zum Verhängnis geworden war. Andererseits neigte CreyBot deshalb zum Exzentrischen: Im Spiel um den dritten Platz startete er nach Westen guckend und wählte das Ziel in dieser Richtung. Er suchte vergeblich die gesamte westliche und südliche Außenwand nach einem Durchschlupf ab und erreichte das wahre Ziel eher nebenbei, als er keinen anderen Weg nach Westen finden konnte.

Die Echtzeitübertragung aller Spiele per Web ermöglichte es den Interessierten, das Turnier live mitzuverfolgen. Aufzeichnungen aller Spiele lassen sich auf www.ct-bot.de auch in Zeitlupe oder Zeitraffer angucken. (pek)



Grüner und blauer Bot fahren um die Wette durchs Finale. Der Simulator liefert ihnen Sensorwerte wie in einem realen Labyrinth.