

Dokumentation

Fabian Bruhns, Jan-Henrik Bruhn, Sven Mehlhop

21. Februar 2018

Inhaltsverzeichnis

1	Einleitung	4
2	Große Einleitung	5
2.1	Umgebungsvariablen	5
2.2	Herausforderungen	5
2.3	GUI	5
3	Programmübersicht	6
3.1	GUI	6
4	Algorithmen	7
4.1	Spielgraph	7
4.2	Enforce	7
4.3	Controller	7
5	Ende	8
5.1	Optimierung	8
5.2	Fehler	8
5.3	Ergebnisse	8
6	Arbeitsverteilung	9
7	Erklärung	10

Fabian:

- Einleitung
- Hauptteil Algorithmen subsection enforceGraph
- System Programmübersicht
- Ende Subsection Optimierung

Jan-Henrik:

- Hauptteil Algorithmen subsection Spielgraph
- Ende Ergebnisse
- Ende aufgetretene Fehler

Sven:

- Layout
- Hauptteil Algorithmen subsection Controller
- System Umgebungsvariablen
- System anforderungen

1 Einleitung

Kurze Projektbeschreibung.... Ziel des Projekts etc...

2 System

2.1 Umgebungsvariablen

Roboter vs Kind Bewegungsmuster Spielfeld um Begrenzung und Mauern erweitert

2.2 Anforderungen

1-Akzeptanz Buechi-Akzeptanz Batterie/Ladestation

2.3 GUI

Die Benutzeroberfläche zur Visualisierung des Problems und dem anschließenden Lösungsweg, wurde so gewählt das der Nutzer zu jedem Zeitpunkt die Schritte des Programmes nachvollziehen kann. Dazu wurde die GUI von der Logik getrennt und besteht als eigenständiger Teil im Projekt. Die GUI bezieht von einem Spielfeld wo sich die Objekte befinden. Dieses Spielfeld wird von der Prozesslogik verändert und von jener ausgelesen. Nach jeder Veränderung wird das Feld als Benutzeroberfläche neu gezeichnet.

Dem Benutzer ist es zu aller erst möglich das Spielfeld durch das Platzieren von Objekten zu gestalten. Dabei stehen dem Benutzer Wände, Roboter, Kind, Ladestationen und Zielfelder zur Verfügung. Der Roboter beschreibt das Feld auf dem der Roboter auf dem gekachelten Spielfeld startet. Dem entsprechend verhält sich das Kind zu seiner Figur. Die Batterien stellen Ladestationen für den Roboter da, welche er in einem Spielmodus besuchen muss, um seine Energie/Ladung wieder aufzufüllen. Die Zielfelder beschreiben die Felder die der Roboter in chronologischer Reihenfolge abarbeiten muss. Dabei wird das als nächstes angestrebte Feld grün markiert.

Ist das Spielfeld erstellt kann der Benutzer im folgenden die Bewegungsmöglichkeiten von Kind und Roboter frei wählen. Dabei sind auch Variationen möglich mit welchen der Roboter sein Ziel nicht erreichen kann, da die GUI mit Rückmeldungen an den Benutzer ausgestattet ist, welche dieses zur Laufzeit wiedergeben.

Um das Programm zu starten ist der Start-Button zu drücken. Dieser ermöglicht es dem Benutzer nach dem Laden der Logik das Programm zu Beschleunigen oder gänzlich zu stoppen. Für das Beschleunigen steht ein Regler zur Verfügung, welcher die Dauer zwischen den Schritten beschränkt oder verlängert. Das Stoppen wird durch einen gleichnamigen Button realisiert, welcher das Programm wieder in den Bearbeitungszustand setzt.

In den Menüreitern können weitere Einstellungen getroffen werden, wie Sprache, oder ob die Tankfüllung beachtet werden soll. Im Programm werden dann dynamisch die Sprache angepasst und die Tankfüllung angezeigt (oben rechts im Feld).

Sollten Fragen zum Spiel aufkommen können diese jederzeit mit dem Hilfefenster versucht geklärt zu werden. Weiterhin ist es noch möglich die Spielfeldgröße zu verändern.

3 Programmübersicht

Wir trennen unser Programm in 3 eigenständige Teile: Graphentheorie, Controller und

3.1 GUI

Die Benutzeroberfläche zur Visualisierung des Problems und dem anschließenden Lösungsweg, wurde so gewählt das der Nutzer zu jedem Zeitpunkt die Schritte das Programmes nachvollziehen kann. Dazu wurde die GUI von der Logik getrennt und besteht als eigenständiger Teil im Projekt. Die GUI bezieht von einem Spielfeld wo sich die Objekte befinden. Dieses Spielfeld wird von der Prozesslogik verändert und von jener ausgelesen. Nach jeder Veränderung wird das Feld als Benutzeroberfläche neu gezeichnet.

Dem Benutzer ist es zu aller erst möglich das Spielfeld durch das Platzieren von Objekten zu gestalten. Dabei stehen dem Benutzer Wände, Roboter, Kind, Ladestationen und Zielfelder zur Verfügung. Der Roboter beschreibt das Feld auf dem der Roboter auf dem gekachelten Spielfeld startet. Dem entsprechend verhält sich das Kind zu seiner Figur. Die Batterien stellen Ladestationen für den Roboter da, welche er in einem Spielmodus besuchen muss, um seine Energie/Ladung wieder aufzufüllen. Die Zielfelder beschreiben die Felder die der Roboter in chronologischer Reihenfolge abarbeiten muss. Dabei wird das als nächstes angestrebte Feld grün markiert.

Ist das Spielfeld erstellt kann der Benutzer im folgenden die Bewegungsmöglichkeiten von Kind und Roboter frei wählen. Dabei sind auch Variationen möglich mit welchen der Roboter sein Ziel nicht erreichen kann, da die GUI mit Rückmeldungen an den Benutzer ausgestattet ist, welche dieses zur Laufzeit wiedergeben.

Um das Programm zu starten ist der Start-Button zu drücken. Dieser ermöglicht es dem Benutzer nach dem Laden der Logik das Programm zu Beschleunigen oder gänzlich zu stoppen. Für das Beschleunigen steht ein Regler zur Verfügung, welcher die Dauer zwischen den Schritten beschränkt oder verlängert. Das Stoppen wird durch einen gleichnamigen Button realisiert, welcher das Programm wieder in den Bearbeitungszustand setzt.

In den Menüeiterei können weitere Einstellungen getroffen werden, wie Sprache, oder ob die Tankfüllung beachtet werden soll. Im Programm werden dann dynamisch die Sprache angepasst und die Tankfüllung angezeigt (oben rechts im Feld).

Sollten Fragen zum Spiel aufkommen können diese jederzeit mit dem Hilfefenster versucht geklärt zu werden. Weiterhin ist es noch möglich die Spielfeldgröße zu verändern.

4 Algorithmen

probleme übersicht roboter probleme übersicht gruppe

4.1 Spielgraph

Enforce algo blabla

4.2 Enforce

4.3 Controller

Es folgen Notizen.

Um zu bestimmen ob ein bestimmtes Ziel (Zielfläche oder Batterie) erreicht werden kann, wird eine Metrik benötigt, die die Distanz angibt. Je nachdem, ob der Controller optimistisch oder pessimistisch handeln soll kann dies der kürzeste Pfad durch den Graphen sein (der Enforce-Wert wird mit jeder Transition so stark verringert wie möglich), oder der längste Pfad, bei dem der Enforce-Wert jedoch mit jeder Transition um mindestens 1 sinkt.

5 Ende

5.1 Optimierung

5.2 Fehler

Aufgetretene Fehler

5.3 Ergebnisse

Folgendes haben wir während des Projekts ermittelt.

6 Arbeitsverteilung

Teilaufgabe	Name	Anteil
1. Graphischer Editor mit Eingabe der Definitionen	Fabian	90%
	Jan-Henrik	5%
	Sven	5%

7 Erklärung

Hiermit versichern wir, dass wir diese Arbeit selbständig verfasst und keine anderen, als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Arbeit in der aufgeführten Arbeitsverteilung im Abschnitt 6 erledigt.


Sven Mehlhop