**Protokoll Informatik 3 – Praktikum**

Praktikumsgruppe:MEC – A

Mitglieder:Ben Lukas Boberg (Matr.-Nr.: 1173401)

Fabian Gretenkort (Matr.-Nr.: 1173280)

Jannik Störmer (Matr.-Nr.: 1172410)

Betriebssystem: Ubuntu 16.04 - Linux

Ubuntu 19.04 - Linux

Entwicklungsumgebung**:** Visual Studio Code 1.41.1

Programmiersprache:C/C++

Projektpfad/-name**:** https://www.github.com/bboberg/inf3project

Githubnamen:bboberg, fgretenkort, jstoermer24

Praktikumsaufgabe**:** B) Automatische Abarbeitung von Strategien für das Spiel „Einbahn Schiffe versenken“

Erstellen Sie ein Programm mit dem Sie eine Strategie für das Spiel „Einbahn Schiffe versenken“ bewerten können. Dafür sollen Sie einen Server implementieren, auf dem eine Spielfläche erzeugt wird, der Server Koordinaten entgegen nimmt und entsprechend bzgl. Treffer und Nicht-Treffer zurückgibt. Die Funktionalität des Spiels ist gegeben. Implementiert werden muss der Server. Ihr Client soll ohne Benutzer-Interaktion die Anfragen an den Server senden, bis das Spiel beendet ist. D.h., Sie müssen eine Strategie vollautomatisch programmieren. Implementieren Sie verschiedene Strategien und vergleichen Sie diese bzgl. der mittleren Anzahl von benötigten Spielzügen.

**Praktikumstermin 1 am 16.10.2019**

- Starten und einrichten von Ubuntu 16.04 - Linux auf den FH Computern

- Account- sowie Repositoryerstellung auf github.com -> bboberg/inf3project

- Hinzufügen der Collaborators für das Repository

- Linux Konsolenbefehle erläutert -> Grundkommandos erlernt

- Herunterladen des Repository https://www.github.com/amlsh/INF3\_PRAK

- Lokal Daten von dem vorgegebenen Repository für Eigenes gespeichert

- Datensatz auf www.github.com hochgeladen mittels Terminal -> git push

- Entwicklungsumgebung Eclipse eingerichtet und Demoprogramm aus Repository gestartet

- via Konsolenbefehle den StandardImageServer (StdImgSrv) ausgeführt

- experimentelles Einstellen für Kameraports, Kameraeinstellung (Graustufen etc. ) und Aufnahmen über Konsolenbefehle zum Server

- Server-Client-Kommunikation erläutert

**Zeitraum zwischen den Praktikumsterminen 1 und 2:**

- Virtual Machine Oracle auf Privatsystemen installiert

- Ubuntu 16.04 installiert und eingerichtet, teils auf Ubuntu 19.04 upgedated

- Entwicklungumgebung Eclipse Oxygen 4.7.3 versucht einzurichten (alle nötigen Pakete unter Linux installiert, Eclipse Oxygen 4.7.3 installiert, Fehlerbehebung)

- Entwicklungsumgebung Visual Studios Code 1.41.1 eingerichtet, aufgrund von Kompatibilitätsproblemen mit Eclipse Oxygen 4.7.3

- Herunterladen des Repositorys unter https://www.github.com/bboberg/inf3project

- Entfernen der nicht relevanten TASKS: 1, 2, 4, 5, 6

- Fehlerbehebung in der Doxygenfile ->

- erste Strategieüberlegungen für die Ausführung der Praktikumsaufgabe

- 1. Konzept: 2 ineinander verschachtelte for-Schleifen, die erst die x-Reihe abschießen und dann die y-Reihe, um Schiffe zu finden

- 2. Konzept: beide for-Schleifen lassen immer eine Stelle aus und suchen nur alle zwei Stellen nach Schiffen um Rechenaufwand und -zeit gering zu halten

**Praktikumstermin 2: 04.12.2019**

- Implementierung von „MyTCPserver“, von „TCPserver“ abgeleitet

- Überschreiben der Methode MyResponse in der Klasse MyTCPserver

- Kommunikation über Command telnet zwischen Client und Server getestet

- ShootResult mittels Koordinaten hinzugefügt

- Darstellung der „Spielwelt“ auf MyTCPserver über Einbindung von Task3 in „myFile.C“

- Implementierung der for-Schleifen für die automatische Absuche der Reihen

**Zeitraum zwischen den Praktikumsterminen 2 und 3:**

- Implementierung der Methode Reihe für Reihe

- Datenaustausch zwischen Client und Server erstellt

- erste Fehler behoben, Finalisierung steht aus

**Praktikumstermin 3 am 04.12.2020**

- Vervollständigung und Formatierung des Protokolls

- Reparatur der Reihe für Reihe-Strategie

- Implementierung der Zufallsbeschuss-Strategie

- Implementierung der Alle-zwei-Felder-Strategie

**Bedienungshinweise:**

Für die Bedienung der Software wird ein unter Linux konfiguriertes Endgerät vorrausgesetzt. Außerdem ist die Entwicklungsumgebung Microsoft Visual Studio Code 1.41.1 zu verwenden um mögliche Kompatibilitätsprobleme der Software bzw. des Programms auszuschließen. Nur wenn diese Parameter erfüllt sind, wird empfohlen mit den folgenden Hinweisen fortzufahren.

* 1. Laden Sie sich das Programm von github.com unter folgender Terminaleingabe herunter:
  + git clone https://www.github.com/bboberg/inf3project
* 2. Übersetzen Sie das Programm mit der Terminaleingabe „make all“
* 3. Öffnen Sie ein zweites Terminal und starten Sie den Server mit der Terminaleingabe „./server 2022“
* 4. Starten Sie den Client in dem anderen Terminalfenster mit der Eingabe „./client“
* 5. Das Programm läuft nun automatisch durch
  + Bildschirmausgabe: „Socket created“

„localhost resolved to 127.0.0.1“

„connected“

* Desweiteren werden die Anzahl der Schüsse, um alle Schiffe zu versenken, für die „Reihe für Reihe-Strategie“ pro Durchlauf (10 Durchläufe) ausgegeben
* Anschließend werden die Anzahl der Schüsse, um alle Schiffe zu versenken, für die „Zufallsbeschuss-Strategie“ pro Durchlauf (10 Durchläufe) ausgegeben
* Abschließend werden die Anzahl der Schüsse, um alle Schiffe zu versenken, für die „Alle-zwei-Felder-Strategie“ pro Durchlauf (10 Durchläufe) ausgegeben. Dazu werden noch die minimale sowie maximale Anzahl der Schüsse und der Mittelwert für die Alle-zwei-Felder-Strategie ausgegeben
* 6. Wenn mehr Durchläufe des Programmes gewünscht sind, können diese in der client.C-Datei geändert werden. Die for-Schleifen für die „Reihe für Reihe-„ und die „Zufallsbeschuss-„Strategien sind durch den Kommentar „//Change number of process“ gekennzeichnet. Ändern Sie hier einfach die „10“ auf den gewünschten Wert. Für die „Alle-zwei-Felder-Strategie“ ist Die Zahl in der Klammer der Zuweisung „shtNmb= mittelwertshootAdvanced(&c,10);“ auf die gewünschten Prozesse zu ändern. Diese ist ebenfalls durch den Kommentar „//Change number of process“ gekennzeichnet.

**UML-Diagramme**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Statistik der benötigten Spielzüge für die Strategien**

Die Statistik bezieht sich auf 100 Durchläufe und wurde manuell mittels Standardabweichungsformel (Stichprobe) ausgerechnet.

-Statistik der „Reihe für Reihe-Strategie“:

n = 100 Durchläufe

µ = 96,6

s = 2,9

min = 88

max = 99

Summe = 9656

-Statistik der „Zufallsbeschuss-Strategie“:

n = 100 Durchläufe

µ = 414,5

s = 203,7

min = 162

max = 876

Summe = 41454

-Statistik der „Alle-zwei-Felder-Strategie“:

n = 100 Durchläufe

µ = 82

s = 3,3

min = 75

max = 89

Summe = 8198

Es ist klar zu erkennen, dass die „Zufallsbeschuss-Strategie“ definitiv die ineffizienteste Lösung der Aufgabenstellung ist und diese nicht weiter verfolgt wird. Nicht nur der Mindest- sowie Maximalwert, sondern auch der Mittelwert µ sind deutlich über den Werten der anderen Strategien.

Die „Alle-zwei-Felder-Strategie“ hingegen ist in diesem Testdurchlauf rechnerisch die effizientere Lösung gegenüber der „Reihe für Reihe-Strategie“. Wenn man die Mindest-, Maximal- und Mittelwerte sowie die Summen der beiden Strategien miteinander vergleicht, zeichnet sich klar ein Effizienzvorteil ab. Daher wird diese Strategie bevorzugt verwendet, da neben der Zeit auch Rechnerkapazitäten eingespart werden kann.