

Rechnernetze Dokumentation

Sheraz Azad und Sven Marquardt

Wintersemester 2015/16

Inhaltsverzeichnis

Bitfolge	Bedeutung
001	Reihe voll
110	Gewonnen
101	Unentschieden
011	Weiter
100	Fertig
111	Nochmal bei Fehlübertragung

1 Schichtenmodell

1.1 Überlegungen für das Spiel Vier gewinnt

Die Kommunikation untereinander findet mittels einer Münze statt, welche hochgehalten wird, wobei wir den Binärcode verwenden. Zeigt die Münze Kopf stellt diese die 0 dar und zeigt die Münze Zahl stellt sie die 1 dar. Damit die beiden Positionen unterscheidbar sind, wird die Münze pro Position, also entweder Kopf (0) oder Zahl (1), jeweils für 2 Sekunden hochgehalten erst danach findet ein Positionswechsel statt.

Damit eine gregelte und sinnvolle Kommunikation zwischen den Kommunikationspartnern stattfinden kann, wurden Kommunikationsregeln festgelegt. Das Spiel Vier gewinnt hat 7 Reihen mit jeweils 6 Feldern, da wir bei diesem Spiel nur die Spaltenangabe brauchen um unseren Spielzug zu machen, wurden 3 Bits verwendet von 001 (1) bis 111 (7) welche die einzelnen Spalten darstellen.

**HIER EIN BILD VON EINEM VIER GEWINNT SPIELFELD
HIER FEHLT NOCH WIE MAN ENTSCHIEDET WER ALS ERSTES DRAN IST**

Weitere Bitcodierungen für die Kommunikation sind:

**HIER EIN BEISPIEL FÜR DEN SPIELFLUSS DEN ENTWEDER
ICH IN LÜBECK ODER SVEN HAT**

Tabelle 1: Hybrides Modell

5. Anwendungsschicht	Das Spiel "Vier gewinnt"
4. Transportschicht	Wird nicht verwendet
3. Vermittlungsschicht	Wird nicht verwendet
2. Sicherungsschicht	Übersetzen der Kommunikation in Bits und Fehlererkennung
1. Bitübertragungsschicht	Übertragung von 0 und 1 durch Medium Münze

1.2 Vor- und Nachteile der Realisierung

Zweierteam mit dem die Analyse der Spielrealisierung gemacht wurde bestand aus Malte Grebe und Niklas Klatt.

Vorteile:

Aufgrund der Kommunikationsregeln ist das Spiel leicht zu verstehen und zu bedienen. Durch die ständige Überprüfung wird dafür gesorgt, dass keine Fehler bei der Übertragung auftreten. Dadurch dass ein Weiter (011) erwartet wird, gibt es Spielpausen und man kann in Ruhe sein Spielfeld aktualisieren.

Nachteile:

Ein Zug dauert ca. eine Minute, da jede Position zwei Sekunden gehalten wird. Spieler 1 oder 2 fängt zu früh mit der Übertragung vom nächsten Spielzug an, dadurch gibt es eine Fehlerübertragung die wiederholt werden muss.

Verbesserte Spielrealisierung:

Einführung einer Spielfeldsynchronisierung um sicherzustellen, dass keine Fehler beim Eintragen der Positionen eingetreten sind.

1.3 Verbesserte Kommunikation durch Stifte

Es gibt zwei Varianten die Kommunikation durch Stifte zu verbessern.

Die **erste Variante** ist, dass man einen waagerechten Stift als 0 und einen senkrechten Stift als 1 interpretiert. Dadurch lassen sich die drei Kommunikationsbit leicht, schnell und eindeutig darstellen.

HIER EINE ZEICHNUNG MIT HALBWINKEL UND DIE BITS

ZU DEN WINKELN

Die **zweite Variante** ist, dass man die Stifte in bestimmten Winkel hinlegt. Hier können wir zum Beispiel sagen das wenn der Stift in einem 90 Grad Winkel liegt, dieser die Bitfolge 001 für Reihe voll darstellt. So können wir die verschiedenen Bitfolgen angeben und bräuchten mit dieser Variante sogar nur einen Stift statt drei.

1.4 Kommunikation durch Klatschen

Problem

Dadurch das alle Teams zeitgleich angefangen haben zu klatschen, konnte man nicht unterscheiden ob das Klatschgeräusch vom gegenüber sitzenden Kommunikationspartner kam, oder von einem Kommilitonen aus einer anderen Gruppe. Aufgrund dieser Tatsache sind bei allen Teams Fehler bei der Kommunikation entstanden.

Lösung

Auch hier gibt es zwei Lösungsansätze,

2 Zuverlässige Datenübertragung

2.1 Messung der Häufigkeit von Rahmenverlusten

Jedes mal wenn der Client einen Rahmen sendet erhöht der Server einen Counter für einen empfangenen Rahmen um eins. Nachdem die Übertragung stattgefunden hat, kann der Server nun überprüfen ob es einen Rahmenverlust gab, indem er die Datei aufruft und die Differenz zwischen der Länge der Datei und dem Counter berechnet.

$\text{Rahmen insgesamt} - \text{Empfangene Rahmen} = \text{Verlorene Rahmen}.$

Wenn die Differenz null beträgt, dann ist kein Rahmenverlust aufgetreten.

2.2 Messung der Bitfehlerrate

3 Anwendungsschicht und Tools

3.1 ns lookup tool

4 Switch

5 Router

6 Transportschicht