

Dokumentation

4-Gewinnt

Konsolenspiel über RS232

Vordiplomarbeit
TSU Uster

Erstellt durch:
Marco Di Menco
Patric Lengacher

Erstellt am:
3. März 2003



Inhaltsverzeichnis

1. Aufgabenstellung	4
1.1 Antrag an die Schule.....	4
1.1.1 Ausgangslage	4
1.1.2 Anforderung	4
1.1.3 Motivation zur Arbeit	4
1.2 Definitive Aufgabenstellung	5
2. Pflichtenheft.....	6
2.1 Serielle Kommunikation (RS232).....	6
2.2 Bedienoberfläche (Menu)	6
2.3 Spiel	6
3. Zeitplan.....	7
4. Analyse.....	8
4.1 User Interface.....	8
4.1.1 Hauptmenu	8
4.1.2 COM-Port Konfigurationsmenu	9
4.1.3 Spiel Konfigurationsmenu	9
4.1.4 Spieloberfläche	10
4.2 Kommunikation.....	10
4.2.1 Definition.....	10
4.2.2 Protokoll.....	11
4.3 Speicherung der Konfiguration	11
4.4 Ablauf	12
4.5 Kassendefinition	13
4.6 Verwendete Software	13
4.7 Systemanordnung	13
5. Softwaredesign.....	14
5.1 Klassendiagramm.....	14
5.2 Klassenbeschreibung	15
5.2.1 Console.....	15
5.2.2 Coin	15
5.2.3 Config	15
5.2.4 COMPort.....	15
5.2.5 Game	15
5.2.6 GamePlayer	15
5.2.7 Keyboard	15
5.2.8 Menu.....	16
5.2.9 MenuText.....	16
5.2.10 Status.....	16
5.2.11 Protocol.....	16



6. Implementation	17
6.1 Ausgabe von Boxen	17
6.2 Serielle Kommunikation	17
6.2.1 Checksumme für die Serielle Kommunikation	17
6.3 Implementation der Spielsteine und Felder.....	17
6.4 Prüfalgorithmus	18
7. Test	19
7.1 Allgemeine Funktionen	19
7.2 Menusteuerung	19
7.3 Spiel	20
7.4 Spiel mit serieller Kommunikation.....	21
7.5 Konfigurationsfile.....	21
8. Schlusswort	22
9. Unterschriften	23
9.1 Betreuer	23
9.2 Studenten.....	23
10. Literaturverzeichnis	24
10.1 Bücher.....	24
10.2 Internetlinkes	24
11. Anhang	25
11.1 CD.....	25
11.1 Benutzerhandbuch	25
11.2 Sourcecode	25



1. Aufgabenstellung

1.1 Antrag an die Schule

Es wurde uns von der Schule ein Word-Vorlage zugesandt in der wir beschreiben sollten, was die Ausgangslage, Anforderung usw. an unser Vordiplomprojekt sind. Dieser Antrag mussten wir bis zum 9. September 2002 an den Fachvorstand Informatik der Technikerschule Uster senden.

In den nachfolgenden drei Unterkapiteln sind die wesentlichen Teile unseres Antrages an die Schule aufgelistet.

1.1.1 Ausgangslage

Unser Ziel ist es eine Vordiplomarbeit zu schreiben, die möglichst viele Komponenten beinhaltet die wir bis jetzt in der Schule schon behandelt haben. Eine weitere Aufgabe die wir uns stellen, ist es der Miteinbezug von PC-Hardware und Schnittstellen. Dies aus dem einfachen Grund, weil wir schon jetzt und vermehrt auch in der Zukunft, damit konfrontiert werden.

Unsere Idee ist es das Spiel „Vier gewinnt“ (für zwei Personen) zu programmieren. Die Applikation soll über eine serielle Schnittstelle auf zwei verschiedenen Rechnern zur Ausführung kommen.

Grundlagen für dieses Projekt bildet das Gelernte in den Fächern C++ (Applikation und Ansteuerung des COM-Port`s) und Elektronik (Kommunikation der seriellen Schnittstelle).

1.1.2 Anforderung

Unsere Arbeit gliedert sich in drei Teile:

Gestalten einer grafischen Oberfläche mittels ASCII-Zeichensatz und Konsolenfunktionen zu realisieren (Setzen der Farbe und Positionierung).

Ermitteln eines Algorithmus für das Spiel „Vier gewinnt“ und Aufbau einer seriellen Kommunikation zwischen zwei PCs.

- Grafische Oberfläche für den User
- Applikation (Spiel)
- Kommunikation über serielle Schnittstelle

Sollte am Ende noch Zeit übrig bleiben, wollen wir die Applikation um ein weiteres Spiel erweitern.

1.1.3 Motivation zur Arbeit

Unser Ziel ist es unsere Kenntnisse in der Ansteuerung der PC-Hardware unter Verwendung der Programmiersprache C++ zu vertiefen. Weiter ist es für uns eine

Herausforderung mit einfachen Mitteln, sprich mit Hilfe der Konsole (MS-DOS) eine grafische Oberfläche für den Benutzer zu gestalten

1.2 Definitive Aufgabenstellung

Am 11. Oktober 2002 wurde uns per E-Mail die definitive Aufgabenstellung von Herr Venosta (Betreuer) zugestellt.

Herr
Di Menco Marco
Bahnhofwiese 2
8712 Stäfa

Herr
Lengacher Patric
Götze 7
8197 Rafz

Wald, 1.Oktober 2002

Vordiplomarbeit 2002 (Informatik)

Sehr geehrter Herr Di Menco und Herr Lengacher

Sie erhalten folgende Vordiplomarbeit zugeteilt, die Sie selbständig lösen möchten.

Konsolenspiel über RS232

Es soll ein Vier-Gewinnt programmiert werden, welches über zwei Rechner gespielt werden kann. Die Kommunikation der zwei Rechner erfolgt über die serielle Schnittstelle RS232.

Die Aufgabe umfasst folgende Punkte:

- Projektplanung
- Einarbeitung in die serielle Schnittstelle (Com)
- Entwurf der Analyse und des Designs
- Erstellen einer Bedienoberfläche
- Implementieren der Problemdomain
- Implementieren der Kommunikation über die serielle Schnittstelle
- Dokumentation aller Arbeitsschritte

Eingesetzte Mittel:

- C++ Entwicklungsumgebung

Ausgabe der Aufgabenstellung: 4. Oktober 2002
Abgabe der 2 Dokumentationen: 3. März 2003

Mit freundlichen Grüßen



D. Venosta

Abb.1 Definitive Aufgabenstellung



2. Pflichtenheft

Es soll ein Vier-Gewinnt programmiert werden, welches zum einen über die serielle Schnittstelle (RS232) auf zwei Rechnern gespielt werden kann. Zum Zweiten, muss es auch möglich sein das Programm auf einem PC laufen zu lassen, da wir davon ausgehen, dass nicht jeder Enduser über zwei PCs und ein Nullmodemkabel verfügt.

2.1 Serielle Kommunikation (RS232)

- Es ist ein Protokoll zu definieren, das eine sichere und überwachte Schnittstelle zwischen den zwei Rechnern sicherstellt.
- Es soll mit C++ eine universelle Klasse erstellt werden, die, die Schnittstellen Aktivitäten koordiniert. Da die Applikation nicht zeitkritisch ist, genügt eine half-duplex Kommunikation.
- Die Schnittstelle wird nur während dem Spiel aktiviert.

2.2 Bedienoberfläche (Menu)

- Es soll eine Bedienoberfläche auf der C++ Konsole erstellt werden. Die Bedienoberfläche soll übersichtlich und einfach gestaltet werden. Sie soll ein Menu zum Starten, konfigurieren des Spiels, Hilfe und beenden der Applikation beinhalten.
- Die Befehle werden ausschliesslich mit der Tastatur eingegeben.

2.3 Spiel

- Jeder Benutzer identifiziert sich mit einem Spielernamen.
- Jeder Benutzer spielt abwechselnd ein Stein mit dem Ziel eine Viererreihe horizontal, vertikal oder diagonal zu bilden. Wer zuerst die Viererreihe komplettiert, hat gewonnen.
- Es soll in C++ ein Spielalgorithmus generiert werden der die Steine platziert, die Reihen auswertet und den Gewinner festlegt. Das Spielfeld besteht aus 6 Reihen und 7 Spalten.
- Während des Spiels, werden die Befehle zwischen den zwei PC ausgetauscht und quittiert.

3. Zeitplan

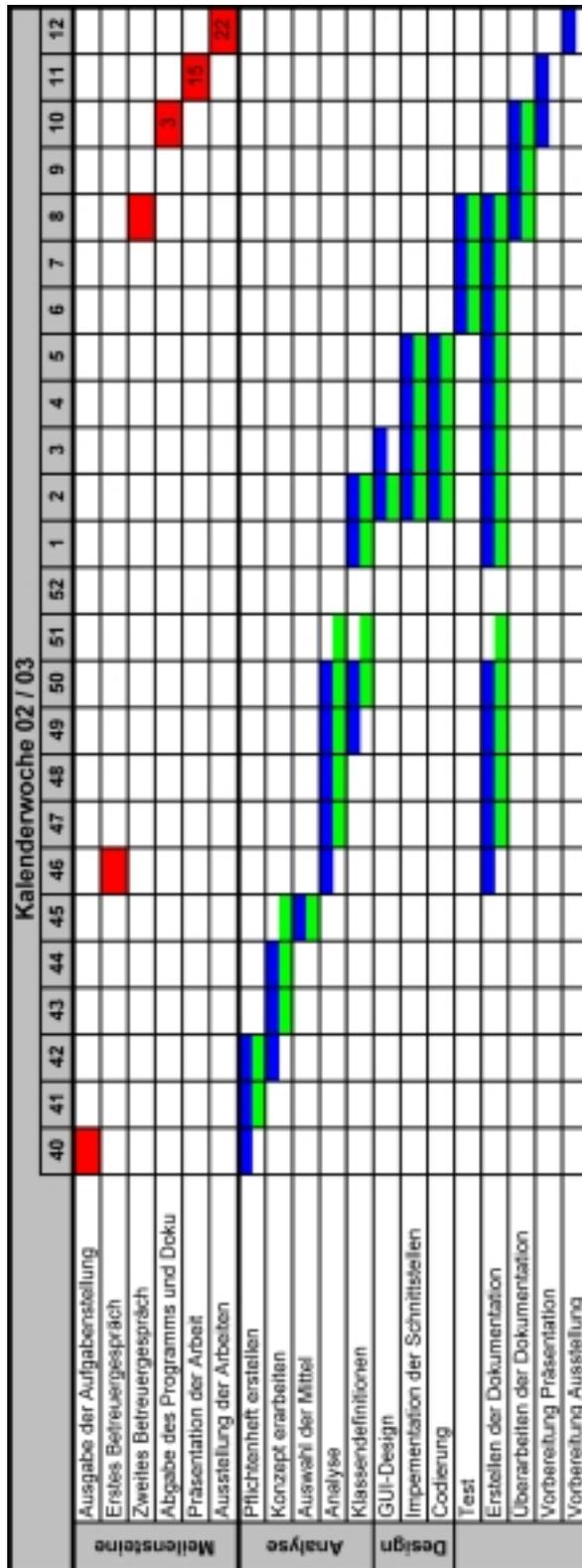


Abb.11 Zeitplan

Nach Abzug von der Schulzeit, Arbeitszeit und den privaten Verpflichtungen blieb für die Vordiplomarbeit eine Restzeit von rund 12-15 Stunden pro Woche übrig. Die Zeit die wir zur Verfügung hatten, haben wir in einzelne Projektschritte aufgeteilt.

Bis zu den Weihnachtsferien, die wir in unser Zeitplan einkalkuliert haben, hinkten wir eine Woche gegenüber dem Soll-Zeitplan hinterher. Dadurch das wir in der Halbzeit der Arbeit eine Bufferzeit „Ferien“ von zwei Wochen eingeplant haben, konnten wir diese Woche wieder aufarbeiten. Von dort an konnten wir den vorgegebenen Zeitplan mehr oder weniger einhalten.

Das Stundenpensum pro Person lag etwas über der Vorgabezeit von 120 Stunden, wir haben beide je rund 140 Stunden für das Projekt investiert.

- = Meilensteine
- = Soll-Zeitplan
- = Ist-Zeitplan

4. Analyse

4.1 User Interface

Da die Applikation auf einer Konsolenanwendung basiert, muss die ganze Steuerung ausschliesslich über die Tastatur erfolgen. Um das ganze Programm für den User einfach zu gestalten, soll die Steuerung des Programms über die Tasten „Escape“, „Enter“ und die vier Pfeiltasten erfolgen.

4.1.1 Hauptmenu

Das Hauptmenu dient dazu um in die verschiedenen Untermenüs zu gelangen. Befindet man sich durch bewegen der Pfeiltasten „Up“ und „Down“ auf dem entsprechenden Feld, so kann man mit Drücken der Entertaste in das ausgewählte Untermenü gelangen. Es besteht nur bei den Menüpunkten „COM-Port Konfiguration“ und „Spiel- Konfiguration“ ein solches Untermenü.

Vom Hauptmenu aus kann das Spiel gestartet oder die Applikation beendet werden.

Zu jedem Menüpunkt ist auf der rechten Seite in einer so genannte „Quick-Infobox“ eine Kurzübersicht über den jeweiligen Menüpunkt.

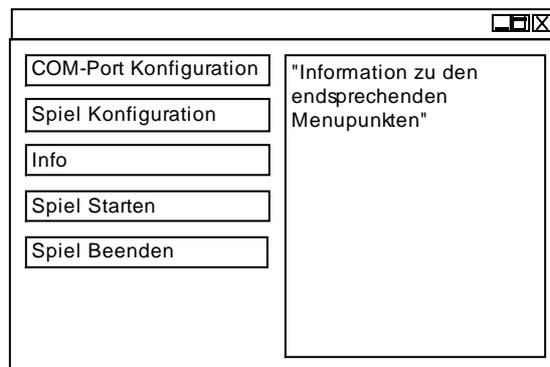


Abb.2 Endwurf Hauptmenu

4.1.2 COM-Port Konfigurationsmenu

Da ein PC über mehrere COM-Ports verfügen kann und das Spiel, wie im Pflichtenheft schon beschreiben, auch auf einen Rechner gespielt werden kann, so muss im Menu die Möglichkeit bestehen den COM-Port umzustellen.

Es soll zwischen Port 1-3 oder keinem Port umgestellt werden können. Wird „kein COM-Port“ ausgewählt so bedeutet dies, dass das Spiel auf einem Rechner zur Ausführung kommt.

Es soll weiter die Möglichkeit bestehen, dass man die Baudrate der seriellen Kommunikation umstellen kann.

Um einen der vorgegebenen Werte auszuwählen, kann mittels der Pfeiltasten „Up“ und „Down“ auf den gewünschten Wert gewechselt werden. Wird nun das Menu mit der Taste „Enter“ verlassen, so wird der ausgewählte Wert des Users übernommen, wird das Untermenu mit der Taste „Esc“ verlassen, so wird der alte Wert beibehalten.

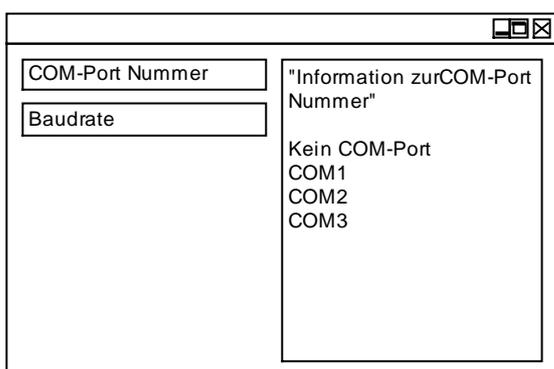


Abb.3 Entwurf Menu „COM-Port Konfig.“ COM-Nr.

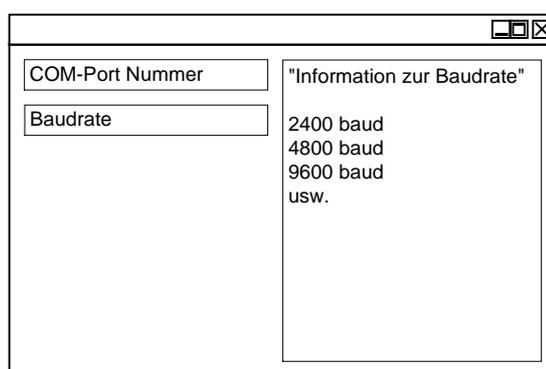


Abb.4 Entwurf Menu „COM-Port Konfig.“ Baudrate

4.1.3 Spiel Konfigurationsmenu

Im Untermenu Spielkonfiguration gibt es im wesentlichen zwei Punkte. Dies ist zum einen die Eingabemöglichkeit der Spielernamen und zum zweiten das Setzen eines Timeouts.

Nach Drücken der „Enter“ Taste wird das Feld zum editieren des Spielernamens aktiv. Für die Eingabe des Spielernamens stehen nur die Zeichen 1-9, A-Z und das Leerzeichen zur Verfügung. Der Name wird übernommen, wenn die Eingabe mit „Enter“ bestätigt wird. Wird die Eingabe mit „Esc“ verlassen, so wird der alte Spielernamen beibehalten.

Die Auswahl eines Timeouts ist analog zur Auswahl eines COM-Port's. Mit den Pfeiltasten „Up“ und „Down“ kann eine gewünschte Zeit ausgewählt werden. Mit der Taste „Enter“ wird der selektierte Wert übernommen und mit „Esc“ wird dieser verworfen.

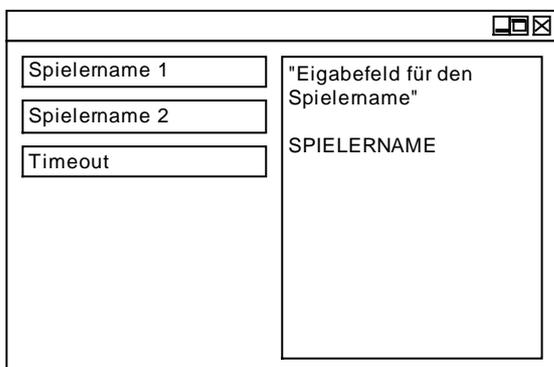


Abb.5 Entwurf Menu „Spiel Konfig.“ Spielname

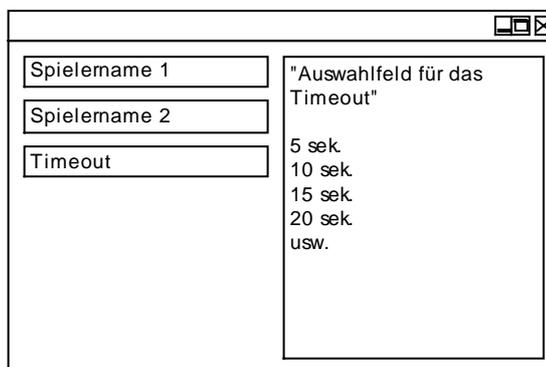


Abb.6 Entwurf Menu „Spiel Konfig.“ Timeout

4.1.4 Spieloberfläche

Die Spieloberfläche beinhaltet im wesentlichen ein „Vier Gewinn“ Spielfeld. Weiter wird eine Spalte mit den beiden Spielernamen und dem Timeout dargestellt.

Das Timeout ist ein Zähler, welcher von der eingestellten Zeit nach Null zählt. Ist die Laufzeit bei Null angekommen, so setzt die Applikation den Stein an der aktuellen Stelle automatisch und der Gegenspieler ist wieder am Zug.

Folgende Statusmeldungen werden vom Programm in die Statusleiste geschrieben:

- „Spielername X“ hat gewonnen
- „Spielername X“ Stein setzen
- Spielfeld voll → Ende
- Kommunikationsfehler z.B. COM-Port konnte nicht geöffnet werden.

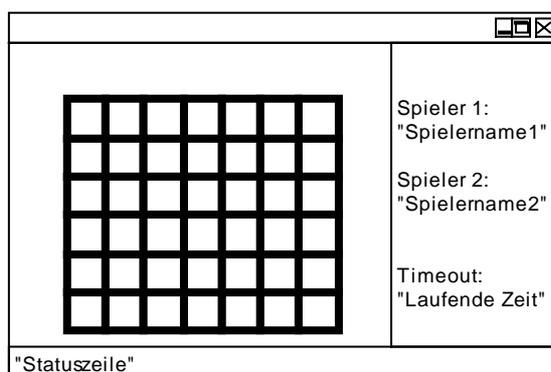


Abb.7 Entwurf des Spielfeldes

4.2 Kommunikation

4.2.1 Definition

Wird das Spiel auf zwei Rechnern gespielt, muss eine Kommunikation über die serielle Schnittstelle zwischen den beiden PCs stattfinden.

Da eine Reihe von Fehlern auftreten können, sei dies nur ein Unterbruch der Leitung oder ein Sendefehler, gilt es diese zu erkennen und richtig zu Verarbeiten.

Damit eventuell auftretende Fehler keine Auswirkungen auf das Programm haben, wird jede Kommunikation über die serielle Schnittstelle vom Rechner, der diese Information erhält, mit einer Meldung bestätigt. Das heisst, wenn z.B. ein Stein gesetzt wird, so wird die Meldung „Stein gesetzt“ gesendet. Danach sendet der Rechner auf der Gegenseite zurück, dass er diese Meldung erhalten hat und zeigt den Stein auf dem Monitor an. Der Rechner, der die Meldung gesendet hat, zeigt den Stein erst an, wenn dieser die Bestätigungsmeldung des anderen PCs erhalten hat.

Als zweite Sicherheit, soll die Meldung die versendet wird auch richtig interpretiert werden. So wird zu jeder Meldung eine Checksumme mitgesendet wird. Somit sollte auch ein Fehler in der Übertragung erkannt werden.

Tritt ein Fehler in der Kommunikation zwischen den beiden PCs auf, so wird das Spiel beendet und man kehrt ins Hauptmenu zurück. Von da aus kann man ein neues Spiel starten.



4.2.2 Protokoll

Definition des Sendeframes

Befehl	Anzahl Bytes	Daten ...		Checksumme (0xFF)
0x20	0x03	0xAA	0x55	0xDF

Befehl	Anzahl der Zeichen	Daten ...		Checksumme des Frames = 0xFF - Byte 0 - Byte 1 - ... = Checksumme (8bit LSB) (Ergänzung auf 0xFF)
0x10 Spielername setzen				
0x20 Stein setzen	0x01	Spaltennummer (char)	---	
0x21 Spiel übergeben	0x00	---		
0x50 Spiel starten	0x00	---		
0x51 Spiel abbrechen	0x00	---		
0x52 Spiel gewonnen / ende	0x00	---		
0x55 Verbindung aufbauen	0x00	---		
0xFF Reserviert für AW	---	---		

Abb.8 Definition Sendeframes

Definition des Antwortframes (Quittierung)

Antwort (0xFF)	Befehl	Checksumme (0xFF)
0xFF	0x20	0xDF

Antwort (0xFF)	Befehl	Checksumme (0xFF)
0xFF Antwort	0x10 Spielername setzen	Checksumme des Frames = FFh - Byte 0 - Byte 1 = Checksumme (8bit LSB) (Ergänzung auf FFh)
	0x20 Stein setzen	
	0x21 Spiel übergeben	
	0x50 Spiel starten	
	0x51 Spiel abbrechen	
	0x52 Spiel gewonnen / ende	
	0x55 Verbindung aufbauen	
0xFF Fehler		

Abb.8 Definition Antwortframe

4.3 Speicherung der Konfiguration

Da der User nicht jedes mal beim Starten des Spieles den Namen, die COM-Port Nummer usw. wieder neu eingeben muss, werden alle Daten, die der User editieren kann in einer Datei gespeichert.

Wird das Programm gestartet, so werden alle Werte aus der Datei gelesen und die Defaultwerte überschrieben. Ist keine solche Datei vorhanden (Sie befindet sich im gleichen Verzeichnis wie die Applikation), so wird eine Datei erstellt, in welche die Defaultwerte geschrieben werden.

Verändert der User im Menu einer der Werte, so wird der neue Wert in die Datei geschrieben. Wird das Spiel nun beendet, so werden beim Neustart die neuen Werte, die der User editiert hat, gleich übernommen und er kann gleich starten und mit dem Spielen beginnen.

4.4 Ablauf

Im Flussdiagramm soll die Hauptlogik des Programms dargestellt werden. Es ist ersichtlich, was aus dem COM-Port ausgelesen wird und wann etwas über den Port gesendet wird.

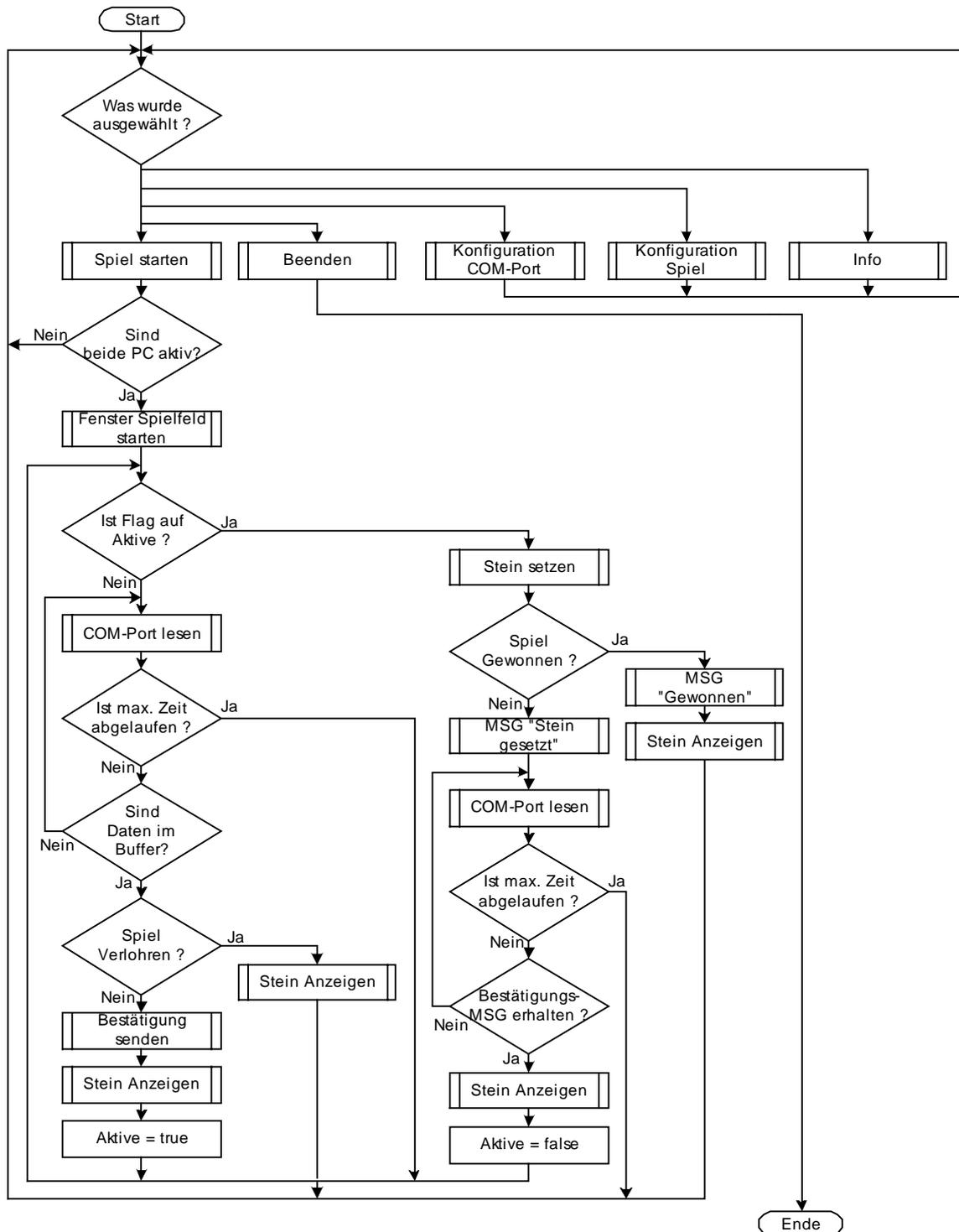


Abb.9 Hauptlogik

4.5 Klassendefinition

Nach der Analyse wie und was das Programm können muss oder wie das Ganze aussehen muss, galt es ein Konzept zu entwickeln, wie die einzelnen Klassen voneinander abhängig sind.

Das Projekt enthält drei Hauptklassen, Game für das Spiel, Menu für das Menu und Config für die Konfiguration.

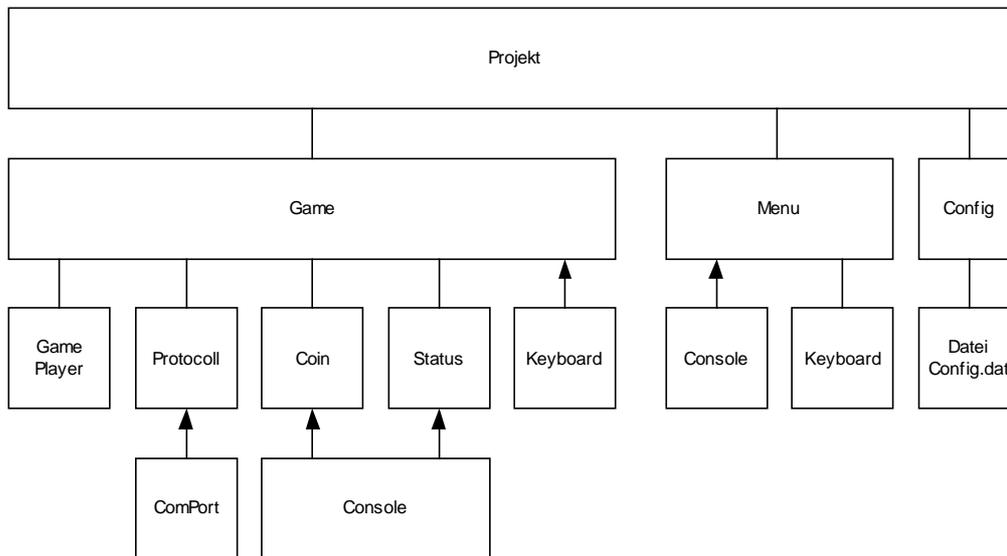


Abb.10 Klassendefinition

4.6 Verwendete Software

Als Entwicklungsumgebung haben wir das Visual Studio 6.0 von Microsoft verwendet. Während der ersten Tests der seriellen Schnittstelle haben wir das Terminal, das bei Win2000 schon beim Systemzubehör dabei ist, verwendet.

4.7 Systemanforderung

Betriebssystem : Win2000
 Arbeitsspeicher : 128 MB
 Bildschirmauflösung : 600 x 800
 Bildschirmfarben : 16 Color

Um die Applikation auf zwei Rechnern laufen zu lassen, muss jeder dieser Rechner über eine serielle Schnittstelle (COM-Port) verfügen. Die beiden PCs werden mittels Null-Modemkabel miteinander verbunden.

Die Systemanforderungen kommen daher, da wir zum Testen nur Rechner mit der oben genannten Hard- und Software zur Verfügung hatten.

Es ist jedoch nicht ausgeschlossen, dass die Applikation auch auf einem anderen OS oder einem PC mit geringerer Hardware lauffähig ist.



5. Softwaredesign

5.1 Klassendiagramm

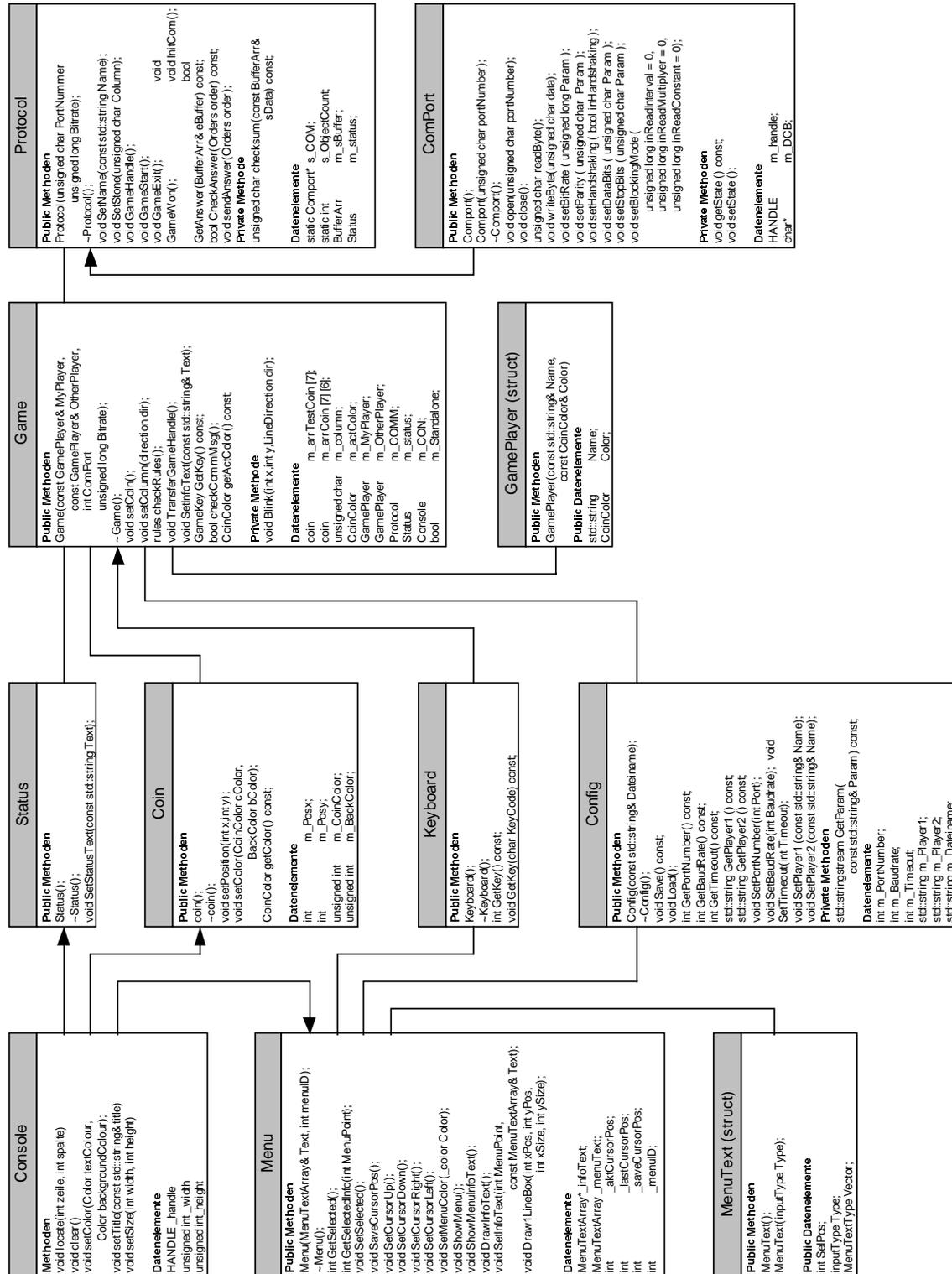


Abb.12 Klassendiagramm



5.2 Klassenbeschreibung

5.2.1 Console

Diese Klasse sollte es einem ermöglichen die Grösse des Konsolenfensters zu definieren. Weiter beinhaltet sie Funktionen für das Setzen des Fenstertitels, die Farbe von Hinter- und Vordergrund zu verändern, den Inhalt des Fensters zu löschen und das Setzen des Cursors auf eine bestimmte Position in der Konsole.

Die Klassen Menu, Coin und Status sind von der Basisklasse Console abgeleitet

Die Console-Klasse haben wir von unserem C++ Fachlehrer M.Burri im Unterricht erhalten.

5.2.2 Coin

Die Klasse Coin definiert den Spielstein, sie enthält die Farbe und Position. Jeder Spielstein kann auf der Console ausgegeben werden.

Die Klasse Coin ist abgeleitet von der Klasse Console. Von der Klasse Console werden im wesentliche die Funktionen, setzen der Farben und setzen der Position benötigt.

5.2.3 Config

Diese Klasse beinhaltet alle Funktionen, die benötigt werden, um das Konfigurationsfile zu erstellen.

Sie hat Funktionen zum Speichern und Lesen der Datei „Config.dat“. Weiter hat zu jedem Parameter, der im File vorkommt, ein Set und eine Get Methode.

5.2.4 COMPort

Diese Klasse steuert den Comport an. Sie bietet die Möglichkeit die nötigen Parameter wie Baudrate, Parity, Handshaking, DatenBits, Stoppbits einzustellen. Zusätzlich kann sie byteweise senden und empfangen.

5.2.5 Game

Diese Klasse Game steuert die Spieloperationen, überprüft die Regeln und ermittelt den Gewinner. Diese Klasse ist von der Basisklasse Keyboard abgeleitet, da das Keyboard je nach Spielstatus aktiv oder inaktiv(Gegenspieler am Zug) sein soll.

5.2.6 GamePlayer

GamePlayer ist eine Struktur die sich in der Datei Game.h befindet. Die Struktur kann Spielerdaten, wie Farbe des Steines und Name des Spielers, speichern.

5.2.7 Keyboard

Die Klasse Keyboard beinhaltet im Wesentlichen eine Funktion die den Keyboardbuffer ausliest und den Wert als Rückgabewert zurückgibt.



5.2.8 Menu

In der Menu-Klasse ist die ganze Menusteuerung abgehandelt. Alle Informationen die in dieser Klasse ermittelt werden, werden den entsprechenden Funktionen zB. Der Spielklasse übergeben.

5.2.9 MenuText

MenuText ist eine Struktur, die sich in der Klasse Menu befindet. Die Struktur beinhaltet die Texte die im Menu ausgegeben wird.

5.2.10 Status

Diese Klasse ist von der Klasse Console abgeleitet und generiert eine Statuszeile. Beispielsweise für das Anzeigen von Fehlermeldungen.

5.2.11 Protocol

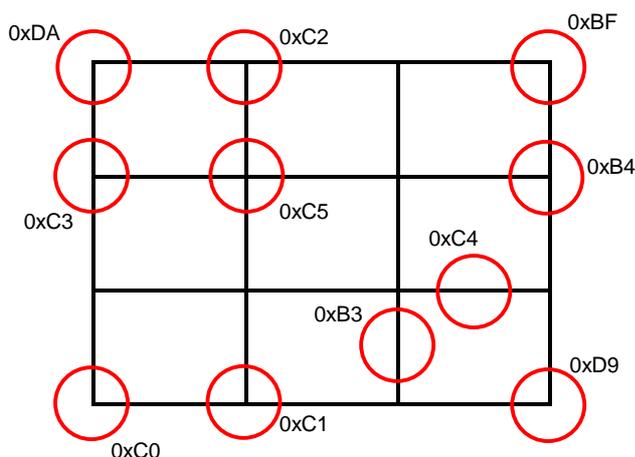
Diese Klasse enthält das Kommunikationsprotokoll. Das Protokoll besteht aus einem Sendeframe und aus einem Quittierungsframe. Der Sendeframe besteht aus einem 8-bit Befehl, 8-bit Anzahl Datenbytes, n-Bytes Daten und einer 8-Bit Checksumme. Der Quittierungsframe enthält 8-bit Quittierung-ID, den Befehl und eine Checksumme.

6. Implementation

6.1 Ausgabe von Boxen

Da es nicht möglich ist auf der Konsole Kreise oder Linien zu Zeichnen, haben wir versucht mittels ASCII-Zeichen eine annähernd grafische Oberfläche zu gestalten, die dem User das Bedienen der Applikation so einfach wie möglich macht.

Für die Ausgabe der Boxen werden folgende ASCII-Zeichen verwendet:



6.2 Serielle Kommunikation

Da wir keine zeitkritische Applikation haben, entschieden wir uns nur eine half-duplex Kommunikation zu tätigen. Das heisst auch, dass wir keine Multithread Applikation erstellen.

6.2.1 Checksumme für die Serielle Kommunikation

Da die serielle Kommunikation über RS232 nur geringe Fehlerkorrekturmöglichkeiten bietet, haben wir uns entschieden eine Checksumme einzubauen.

Sie arbeitet mit einer Aufsummierung aller zuzesendenden Zeichen. Das nun erhaltene Resultat wird auf 8-bit gekürzt und von Hex FF subtrahiert.

So erhält man eine relativ hohe Sicherheit, dass der empfangene Frame korrekt ist.

6.3 Implementation der Spielsteine und Felder

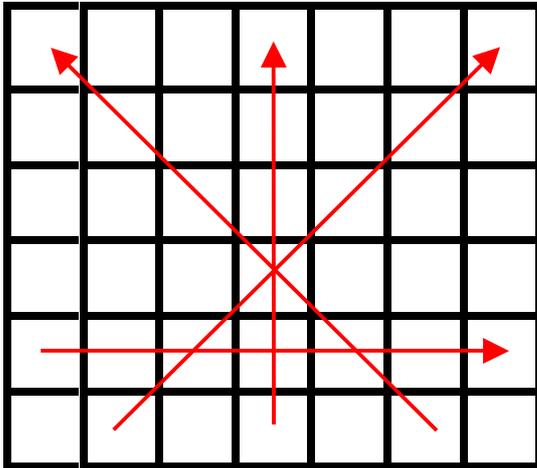
Das Spielfeld haben wir mit dem selben Prinzip wie das Menu aufgebaut. Das heisst wir haben ebenfalls ASCII-Zeichen verwendet um eine Oberfläche zu gestalten.

Es besteht aus 6 x 7 Spielstein Elementen die aneinander gereiht sind. Diese einzelnen Elemente können eine Hintergrund- und eine Steinfarbe aufnehmen.

6.4 Prüfalgorithmus

Nach jedem Setzen eines Steines wird geprüft, ob der Spieler das Spiel für sich entscheiden konnte oder das Spielfeld voll ist.

Da das Spielfeld aus einem zweidimensionalen Array von Spielsteinen besteht, muss nur die Farbe geprüft werden.



Es müssen nur die Richtungen horizontal, vertikal, diagonal nach oben links und diagonal nach oben rechts geprüft werden.



7. Test

Das wohl wichtigste am ganzen Projekt, neben einer guten Analyse, ist das Testen des Programms.

Das Testen des Programms ist in zwei Phasen abgelaufen. Die ersten Tests wurden gleich nach der Erstellung der einzelnen Klassen oder Programmteilen mit dem Debugger durchgeführt.

In der zweiten Phase des Tests wurde das Programm als ganzes getestet. Um einige Fälle zu testen die vorkommen, können haben wir eine Tabelle erstellt. Diese haben wir dann Punkt für Punkt abgearbeitet um sicherzustellen, ob das Programm auch funktioniert.

Zu den Punkten auf der Liste unterhalb, haben wir noch eine ganze Reihe von anderen Tests durchgeführt. Diese jedoch alle hier aufzuführen würde den Umfang dieser Dokumentation sprengen.

Die Liste die wir verwendet haben um das Programm zu Testen, ist in den Kapiteln 7.1 bis 7.5 dargestellt.

7.1 Allgemeine Funktionen

TestszENARIO	Resultat
Beim Starten der Applikation wird das Hauptmenu dargestellt und die erste Box „COM-Port Konfiguration,“ ist das aktive Feld.	✓
Das Programm soll über den Menüpunkt „Spiel beenden“ oder mit der Taste „Esc“ beendet werden.	✓
Das Programm kann nach umstellen der COM-Port Nummer auf einem oder mehreren Rechnern gespielt werden(Einzelplatz oder Mehrplatz).	✓

7.2 Menusteuerung

TestszENARIO	Resultat
Wird mit den Pfeiltasten „Up“ und „Down“ zwischen den einzelnen Menüpunkten hin und her gewechselt, so soll das aktive Feld auch wechseln. Das alte Feld soll auf „nicht aktiv“ gesetzt werden.	✓
Wird zwischen den einzelnen Menüpunkten gewechselt, so wird in der so genannten „Quickinfobox“ der entsprechende Infotext des ausgewählten Feldes angezeigt.	✓
Befindet man sich mit dem aktiven Feld auf den Menüpunkten „COM-Port Konfiguration“ oder „Spiel Konfiguration,“ so wird mittels drücken der Entertaste das entsprechende Untermenü angezeigt.	✓
Wird eines der beiden Untermenüs mit der Taste „Esc“ verlassen kehrt man wieder ins Hauptmenu zurück. Das Feld für das entsprechende Untermenü, dass verlassen wurde, soll im Hauptmenu aktiv sein.	✓
Wird bei den Menüpunkten „Spielername 1“ und „Spielername 2“ die Entertaste gedrückt, so wird das zu editierende Feld aktiv.	✓



Die Eingabe der Usernamen darf maximal 20 Zeichen lang sein. Ist das Maximum erreicht, so werden die Eingaben vom User unterdrückt.	✓
Dem User darf es nur möglich sein die Zeichen A-Z, 0-9 oder das „Blank“ einzugeben.	✓
Wird die Eingabe mit der Taste „Enter“ bestätigt oder mit „Esc“ verworfen so wird das editierbare Feld auf nicht aktiv gesetzt.	✓
Wird bei den Auswahlfeldern „COM-Port Nummer“, „Baudrate“ oder „Timeout“ die Entertaste gedrückt, so wird in der „Quickinfobox“ die Aktuelle Einstellung aktiv. Ist das Feld aktiv, so kann mit den Pfeiltasten „Up“ und „Down“ ein anderer Wert ausgewählt werden.	✓
Wird die Eingabe mit der Taste „Enter“ bestätigt oder mit „Esc“ verworfen so wird das editierbare Feld auf nicht aktiv gesetzt.	✓
Beim Starten des Spieles soll angezeigt werden, auf welchem COM-Port die serielle Kommunikation stattfindet. (Statuszeile)	✓

7.3 Spiel

TestszENARIO	Resultat
Wenn der User die Pfeiltasten „Left“ oder „Right“ bewegt, muss der Stein sich oberhalb des Spielfeldes nach rechts oder links bewegen.	✓
Befindet sich der Stein oberhalb des Spielfeldes auf der letzten Position und der User drückt nochmals die Pfeiltaste „Right“, so soll der Stein wieder auf dem ersten Feld angezeigt werden. Dies soll auch in die Gegenrichtung funktionieren.	✓
Drückt der Spieler die Pfeiltaste „Down“, so wird an der aktuellen Position der Stein ins Feld gesetzt.	✓
Hat ein Spieler erreicht, dass er vier oder mehrere Steine in einer Reihe hat, so soll dies mit blinken angezeigt werden. Weiter soll in der Statusleiste vermerkt werden, dass ein Spieler gewonnen hat. Danach soll das Hauptmenu wieder angezeigt werden.	✓
Hat es ein Spieler mit einem Stein gleich mehrere Reihen komplett so sollen alle Reinen, die zum Gewinn des Spieles geführt haben, durch blinken angezeigt werden. Weiter soll in der Statusleiste vermerkt werden, dass ein Spieler gewonnen hat.	✓
Ist eine Kolonne voll mit Steinen, so darf es für den User nicht mehr möglich sein, diese Reihe auszuwählen und dort einen Stein zu setzen.	✓
Wenn das Spielfeld voll ist und keiner der beiden Spieler das Spiel für sich entscheiden konnte, soll dies in der Statuszeile angezeigt werden und das Spiel beendet werden. Es soll das Hauptmenu wieder angezeigt werden.	✓



7.4 Spiel mit serieller Kommunikation

TestszENARIO	Resultat
Wird das Programm auf einem Rechner gespielt, so soll keine Kommunikation aufgebaut werden.	✓
Wählt der User im Menu eine COM-Port Nummer aus, so soll die serielle Kommunikation über diese Schnittstelle stattfinden.	✓
Wählt der Spieler einen Port aus der nicht vorhanden ist, so soll dies in der Statuszeile vermerkt werden. Das Spielfeld darf nicht angezeigt werden.	✓
Antwortet der andere PC nicht in der geforderten Zeit (COM-Timeout), so soll das Spiel beendet und das Hauptmenu angezeigt werden.	✓
Wenn die Kommunikation unterbrochen wird, (z.B. durch Kabelunterbruch) muss die Applikation beendet werden.	✓
Der Name des Gegenspielers muss am Anfang des Spieles übermittelt werden, der Spielername sollte am richtigen Ort stehen. (Name des Gegenspielers)	✓
Beim Setzen des Steines, sollte zuerst auf dem PC des Gegenspielers der Stein gesetzt werden.	✓
Hat ein Spieler gewonnen, so wird der Gegenspieler durch Blicken der Steine und über die Statuszeile informiert, dass er das Spiel verloren hat. Es soll das Hauptmenu wieder angezeigt werden.	✓
Wenn der aktive Spieler die Taste „Esc“ drückt, werden beide Applikationen beendet.	✓

7.5 Konfigurationsfile

TestszENARIO	Resultat
Ist kein Konfigurationsfile beim Starten der Applikation vorhanden so soll eines erstellt werden, in welche die Defaultwerte geschrieben werden.	✓
Ist ein Konfigurationsfile beim Starten der Applikation vorhanden so werden die Werte ausgelesen um die Defaultwerte zu überschreiben.	✓
Ist man in einem Untermenu in welchem einer der Werte aus dem File zum Zuge kommt so soll dieser als Aktiv ersichtlich sein. (z.B. Ist im File der COM-Port auf die Nummer zwei gesetzt so soll beim Wechsel in das Menu „COM-Port Nummer“ der Wert „COM 2“ aktiv sein.)	✓
Kann ein Wert aus dem File nicht interpretiert werden so sollen alle Werte aus dem File verworfen werden und ein neues mit den Defaultwerten erstellt werden.	✓
Wird eine Einstellung im Menu verändert, so wird der neue Wert wieder in das File geschrieben.	✓
Wird eine Änderung vorgenommen, so soll dieser erst übernommen werden, wenn der Menüpunkt mit der Entertaste verlassen wurde. Wird der Menüpunkt mit der Taste „Esc“ Taste verlassen, so soll der alte Wert beibehalten werden.	✓



8. Schlusswort

Geschafft! Nun ist es soweit, das Ende der Vordiplomarbeit ist erreicht.

Die Aufgabenstellung war für uns beide eine ziemliche Herausforderung, da wir eine reine Konsolenapplikation erstellen und diese möglichst übersichtlich gestaltet mussten. Des Weiteren, hatten wir einige Schwierigkeiten mit der Ansteuerung der seriellen Schnittstelle, dies hat nicht auf Anhieb so einfach funktioniert, wie wir uns das am Anfang unserer Arbeit vorgestellt haben.

Da wir nur eine geringe Erfahrung im Erstellen einer technischen Dokumentation hatten, mussten wir in diese eine Menge Zeit investieren.

Trotz all diesen Umständen sind wir der Meinung, dass wir mit unserer Arbeit ein gutes Projekt erarbeitet zu haben.

Als Fazit gilt zu sagen, dass wir trotz der strengen Zeit und der aufgetretenen Probleme, eine Menge von der Arbeit profitieren konnten. Zum einen haben wir unsere Kenntnisse in der Programmiersprache C++ vertiefen können, zum anderen haben wir gelernt wie wir die PC-Hardware (Serielle Schnittstelle) ansteuern können.

Wir wollen hier die Möglichkeit ergreifen unserem Betreuer Dominik Venosta und unserem C++ Fachlehrer für ihre tolle Unterstützung zu danken.



9. Unterschriften

9.1 Betreuer

Dominik Venosta
Unterpuntstrasse 10
8636 Wald

.....
Uster, 3. März 2003

9.2 Studenten

Marco Di Menco
Bahnhofwiese 2
8712 Stäfa

.....
Uster, 3. März 2003

Patric Lengacher
Götze 7
8197 Rafz

.....
Uster, 3. März 2003



10. Litaraturverzeichnis

10.1 Bücher

Programmierung C++

Autor : Peter Prinz, Ulla Kirch.Prinz
Buchtitel : Lernen und Professionell anwenden
Verlag : MITP
ISBN-Nr. : 3-8266-0423-7

Autor : Gerhard Willms
Buchtitel : C++ Das Grundlagenbuch
Verlag : DATA BECKER GmbH & Co. KG.
ISBN-Nr. : 3-8158-1437-5

Autor : Markus Burri
Unterlagen : Dokumentation die vom C++ Fachlehrer abgegeben wurde

Serielle Schnittstelle (COM-Port)

Autor : R. Derrer
Unterlagen : Dokumentation die vom Elektronik Fachlehrer abgegeben wurde

10.2 Internetlinkes

<http://www.devmentor.ch>

11. Anhang

11.1 CD

Auf der CD befinden sich folgende Daten:

- Ausführbares Programm „exe“
- Kompletter Sourcecode
- Dokumentation und Handbuch



11.1 Benutzerhandbuch

Im Anhang zu dieser Dokumentation ist ein kleines Benutzerhandbuch zu finden. In diesem Handbuch befinden sich alle notwendigen Informationen die der Enduser benötigt um Änderungen in den Einstellungen vorzunehmen oder das Spiel laufen zu lassen.

11.2 Sourcecode

Im Anhang ist der Komplette Sourcecode in gedruckter Form zu finden.