

# Contents

|          |                                  |          |
|----------|----------------------------------|----------|
| <b>1</b> | <b>Installationsbeschreibung</b> | <b>1</b> |
| <b>2</b> | <b>Spielziel</b>                 | <b>1</b> |
| <b>3</b> | <b>Strategie des Computers</b>   | <b>1</b> |

## Vier Gewinnt Projektbeschreibung

### 1 Installationsbeschreibung

Zum kompilieren des Programmes genügt es einen C-Compiler (z.B. den GCC) zu benutzen. Unter Linux öffnet man eine Konsole, navigiert in den Ordner in dem die Quelldatei (.c) liegt und gibt den Befehl

```
gcc viergewinnt.c -o viergewinnt
```

in die Konsole ein. Eine ausführbare Datei "viergewinnt" wird erstellt und das Programm kann nun mit dem Befehl "./viergewinnt" gestartet werden! Unter Windows lässt sich das Programm ebenfalls unter Cygwin kompilieren und starten! Desweiteren ist dem finalen Build ein Makefile beigelegt welches das Programm mit dem target viergewinnt in einer deutschen Version sowie mit dem default target in einer englischen Version kompiliert.

### 2 Spielziel

Das Ziel des Spiels besteht darin vier Spielsteine der gleichen Farben in eine diagonale, horizontale oder vertikale Reihe zu bekommen. Der Spieler welcher dieses als erstes erreicht gewinnt das Spiel. Falls das Spielbrett komplett gefüllt ist und kein Spieler vier Spielsteine der gleichen Farben vertikal, horizontal oder diagonal anordnen konnte endet das Spiel in einem Unentschieden.

### 3 Strategie des Computers

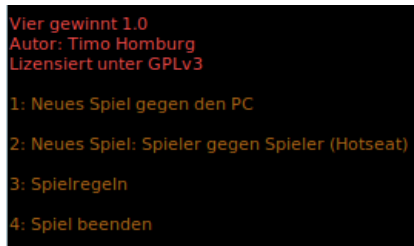
Die Strategie des Computers ist keinesfalls perfekt, basiert aber auf einem eigenen Programmieransatz der im 1. Semester meines Bachelorstudiums entwickelt wurde. Der Ansatz gewichtet jede Spalte des Spielbretts mit einer Punktzahl um den Computergegner damit zu steuern.

Der Algorithmus arbeitet wie folgt:

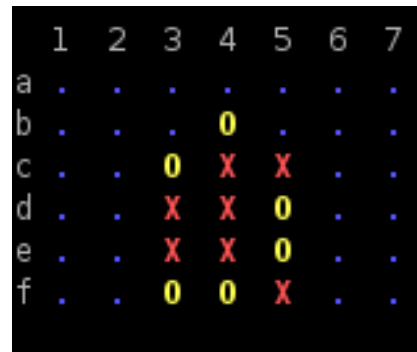
1. Prüfung in der ersten Spalte (von links) ob man noch einen Stein einwerfen kann.
2. Prüfung in der ersten Spalte (von links) ob um das unterste freie Feld herum 3 Steine in einer Reihe vertikal, horizontal oder diagonal liegen. Wenn ja so wird die Spalte bewertet (bei eigenen 3 Steinen mit 120450 und bei 3 Steinen des Gegners mit 17207)

3. Wenn keine 3 gleichen Steine gefunden wurden, so wird geprüft ob in der Spalte rechts oder links der zu prüfenden Spalte mehr Steine liegen. Ist dies der Fall so wird in der nächst höheren Zeile der zu prüfenden Spalte geprüft ob sich um diese herum drei gleiche Steine befinden. Wenn 3 gleiche Steine des Gegners oder eigene 3 gleiche Steine existieren wird die Bewertungsfunktion aus 2. erneut aufgerufen und die Bewertung von der ersten Bewertung abgezogen. Danach wird geprüft ob die 2. Bewertung gleich der höchsten Bewertung (120450) oder einem Vielfachen von 120450 war, also ob der PC eine Zeile über der einzuwerfenden Zeile gewinnen könnte. Ist dies der Fall wird 2 Zeilen über der einzuwerfenden Zeilen der zu prüfenden Spalte darauf geprüft ob der PC auch in dieser gewinnen kann. Ist dies der Fall so wird eine Bewertung unter der höchsten Bewertung des gegnerischen Spielers genommen (17206) Wenn der PC nicht in dieser Zeile gewinnen kann so wird die Bewertung auf -99000 gesetzt.
4. Im nächsten Schritt wird Schritt 2 für 2 Steine horizontal, vertikal und diagonal wiederholt. Die Bewertungen sind hierbei 2458 für den Gegner und 351 für den PC. Im Sonderfall dass 2 gegnerische Steine (oder eigene Steine) in einer Reihe mit 2 freien Plätzen um sie herum oder 2 gegnerische oder eigene Steine mit freien Plätzen um sie herum und zwischen ihnen existieren (sog. Zwickmühlen im Spiel) wird die Bewertung angepasst, da ansonsten ein Gewinn des Gegners in der übernächsten Runde unausweichlich ist oder die eigene Gewinnchance dadurch merklich gesteigert werden kann. Die Bewertungen sind deshalb für diese Fälle 17207 für den Gegner und 5735 für den PC.
5. Danach wird Schritt 2 für einen Stein horizontal, vertikal und diagonal wiederholt. Die Bewertungen sind hierbei 50 für den Gegner und 7 für den PC.
6. Der sich aus den Schritten 2-5 ergebene Wert wird im Array Bewertung gespeichert. Zusätzlich wird in einer Variablen bmax festgehalten welches der höchste Wert war der im Array gespeichert wurde und die Spaltennummer wird ebenfalls in einer Variablen l gespeichert.
7. Sollte die Bewertung einer Spalte kleiner als -99100 werden so wird die Bewertung der Übersichtlichkeit halber auf -99100 angepasst.
8. Die Schritte 1.-7. werden für die restlichen 6 Spalten wiederholt.
9. Wenn wie z.B. am Anfang des Spieles noch keine Bewertungen vorliegen können, so wird eine der Spalten 3,4 oder 5 über die Randomfunktion ermittelt, da diese die grössten Gewinnchancen versprechen.

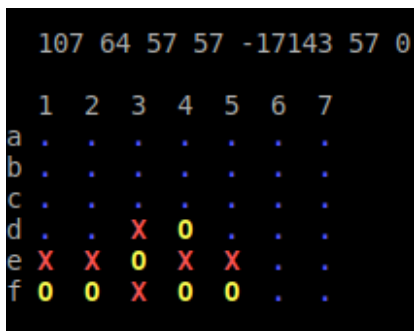
Im Programm gibt es übrigens einen Debugmodus der mit der Menüanwahl 5 im Hauptmenü aktiviert werden kann. In diesem Modus werden die Bewertungen die der Computer errechnet über den Spalten angezeigt.



(a) Spielmenü



(b) Spielbildschirm



(a) Debug Modus



(b) Der Computer hat gewonnen