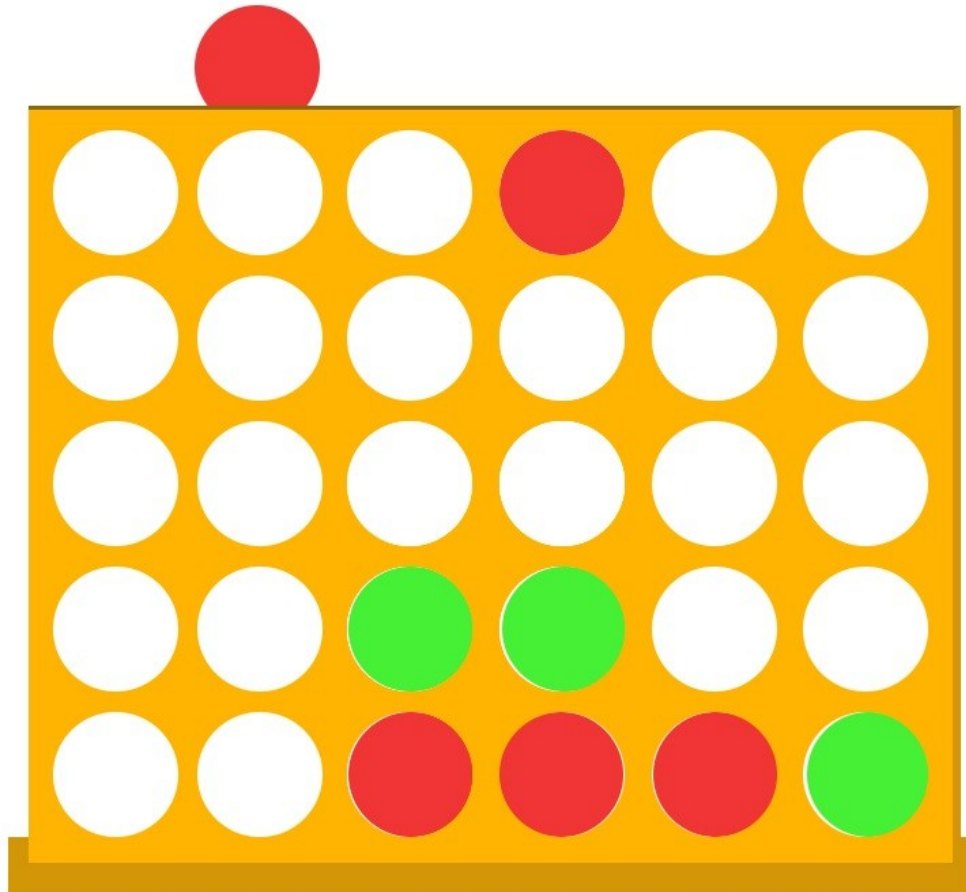


Projekt JavaScript | Vier gewinnt!

Cimdata-Dozent: Klaus Domass

Teilnehmer: Martin Langkabel & Lisa Müller

Modulzeitraum: Januar – Februar 2018



Im Rahmen des Projekts soll ein Vier-gewinnt-Spiel entwickelt und programmiert werden, in dem der PC-User gegen einen Computergegner (*non-player character*, NPC) antritt.

Das Spielprinzip:

Beim „Vier gewinnt!“ treten zwei Spieler gegeneinander an, jeder erhält Chips in einer Farbe. Die Spieler sind abwechselnd an der Reihe und können pro Spielzug einen ihrer Chips in eine Spalte des Spielgerüsts fallen lassen, dann ist der Gegner am Zug. Ziel ist es, eine ununterbrochene Reihe von vier Chips der eigenen Farbe in horizontaler,

vertikaler oder diagonalen Richtung zu erzielen und gleichzeitig zu verhindern, dass der Gegner eine solche Gewinnkonstellation erzielt.

Das Spiel endet entweder mit Sieg bzw. Niederlage, sobald es einem Spieler gelingt, eine Viererkette zu bilden – oder aber mit Unentschieden, wenn alle Slots gefüllt sind.

Geplante Umsetzung:

Das Spielfeld / Gerüst wird aus 30 Slots (in 6 Spalten á 5 Slots) bestehen. Die Spielzeit wird (unsichtbar) vom Fallenlassen des ersten Steins bis zum Spielende gemessen; die Ausgabe der Spielzeit erfolgt am Ende.

Der User (menschl. Spieler) ist als Erster am Zug, der NPC zieht nach – danach wechseln sich beide so lange ab, bis entweder ein Spieler gewonnen hat oder ein Unentschieden eintritt. Die Spielzüge des NPCs (d.h. die Spalte, in der dieser seinen Spielstein jeweils fallen lässt) werden zufällig generiert.

Der Spieler kann durch Maus- / Cursor-Bewegung die Spalte wählen, in der er seinen Stein fallen lässt; die jeweils ausgewählte Spalte wird durch den darüber befindlichen, aktuell zu spielenden Stein markiert. Die Mausbewegung des Users bewirkt einen Wechsel der aktuellen Spalte – der Spielstein gleitet allerdings nicht stufenlos entlang der vollen Breite des Spielfelds mit, sondern springt beim Überqueren festgelegter Grenzen (entlang der Breite des Gerüsts / x-Achse) zur jeweils rechts oder links benachbarten Spalte. Von Spalte 1 ist die Bewegung des Spielsteins ausschließlich nach rechts möglich, entsprechend bewegt sich der Spielstein von Spalte 6 aus nur nach links (kein direkter Sprung zwischen 1 und 6).



Das Fallenlassen des Spielsteins erfolgt per *keydown* der Leertaste; der Stein fällt in den untersten verfügbaren Slot (Schwerkraft). Sind bereits 5 Spielsteine in einer Spalte vorhanden, ist diese voll – folglich können keine weiteren Steine hier fallengelassen werden.

Anforderungen an die Software:

- Spielzüge des NPCs werden zufällig generiert => eine (noch nicht gefüllte) Spalte muss zufällig gezogen werden
- es muss zum Ende eines jeden Spielzugs abgefragt werden, ob der Spieler eine Slot / Felder-Kombination erzielt hat, die zu einem Gewinn führt. Falls ja, wird das Spiel beendet und es erfolgt eine entsprechende Informationsausgabe, andernfalls ist der andere Spieler am Zug
- tritt kein Gewinn / Niederlage-Fall ein, wird weitergespielt, bis alle 30 Slots belegt sind
- beim Fallenlassen der Spielsteine muss abgefragt werden, wie viele Slots in der jeweiligen Spalte noch frei sind, um zu ermitteln, wie „tief“ der aktuelle Spielstein „fallen“ muss / darf

Benutzeroberfläche:

Das Spielfeld / Gerüst befindet sich zentriert auf der Webseite, unterhalb befindet sich ein Neustart-Button, so dass der Nutzer nicht gezwungen ist, ein angefangenes Spiel zu beenden. Ebenfalls finden sich dort eine kurze Erläuterung der Bedienung, die Information über die Spielsteinfarben von User und NPC sowie Links zu einer Seite mit einem Regelüberblick. Endet das Spiel regelkonform (durch Gewinn / Niederlage bzw. Unentschieden), erfolgt eine Ausgabe mit Information über die Art des Spieldausgangs in Form eines Banners – ähnlich einer Werbebanner-Einblendung – die ebenfalls die Möglichkeit des Spiel-Neustarts anbietet.

Zeitmanagement:

Aufgaben	kalkulierter Zeitaufwand
Grafische Gestaltung Spielfeld und Implementierung in HTML und CSS	1 Std.
Heraussuchen geeigneter Übungen aus dem Kurs sowie Recherche zu unbekannten Elementen	3 Std.
JS-Programmierung	2 Tage
Debugging und Testen	1 Tag
Grafisches Aufpolieren (Feinheiten)	3 Std.
Präsentation	0,25 Std.

Mögliche Umbauten / Erweiterungen der Software:

- Sounds:
 - "Klack" beim Fallenlassen des Steins
 - anderes "Klack" beim Switchen / Wählen der Spalte
 - Sound bei Gewinn (menschl. Spieler)
 - Sound bei Verlieren (menschl. Spieler)
 - Sound bei Unentschieden
- menschl. Spieler kann Chip-Farbe mit Hilfe dreier Schieberegler wählen, je einer für die Werte für R , G und B . Der NPC hat die festgelegte Farbe Schwarz ($\text{rgb}(0,0,0)$). Um eine sichere farbliche Unterscheidung der Spielsteine zu gewährleisten, beginnt die Skala der vom User zu wählenden Werte für R , G und B erst bei 50 statt 0. Alternativ / zusätzlich: Eingabemöglichkeit von Zahlenwerten (50-255) in drei Felder, um händisch Farbwerte zu wählen.
- NPC bzw. der Zufallsalgorithmus könnte intelligenter gemacht werden, so dass dieser die aktuelle Spielsituation interpretiert und dementsprechend handelt. So würde der Schwierigkeitsgrad für den User steigen.
- Zu Beginn Abfrage des Spielernamens, der später für eine Bestenliste / Ranking mit den kürzesten gespielten Zeiten genutzt wird.