Lastenheft Hexxagon

Version 0.5 vom 25.10.2019

Abstract

Dieses Lastenheft beschreibt die Anforderungen an das rundenbasierte Mehrspielerspiel Hexxagon, welches über einen Zeitraum von einem Semester entwickelt werden soll.

Motivation

Das diesjährige Softwaregrundprojekt steht ganz im Zeichen von James Bond (Agent 007), deshalb soll auch das Einzelprojekt einen gewissen James Bond Flavor bekommen.

Produkt und Einsatzszenarien

Im Wintersemester 2019 / 2020 soll das rundenbasierte Mehrspielerspiel Hexxagon entwickelt werden. In diesem Kapitel wird die Architektur des verteilten Systems beschrieben. Desweiteren werden die Komponenten des Systems beschrieben und technische Vorgaben gemacht.

Komponenten und Architektur

Das rundenbasierte Mehrspielerspiel Hexxagon ist ein verteiltes System mit Client-Server-Architektur. Die Kommunikation findet dabei ausschließlich zwischen Client und Server statt. Einzelne Clients kommunizieren nicht direkt miteinander.

Server

Der Server wird Ihnen zur Verfügung gestellt.

Client

Der Client wird vom Spieler bedient und kommuniziert mit einem Server.

Anwendungssprache, Implementierungssprache und Dokumentationssprache

Die Anwendungssprache ist deutsch oder englisch.
Die Implementierungssprache ist englisch.
Die Dokumentationssprache ist deutsch oder englisch.

Programmiersprachen und Technologien

Die zu verwendende Programmiersprache ist Java.
Als Framework zur Realisierung der graphischen Oberfläche soll libGDX verwendet werden.
Falls Sie eine andere Programmiersprache verwenden möchten, dann besprechen Sie dies mit Ihrem Tutor / Ihrer Tutorin.
Die Verwendung einer anderen Programmiersprache, eines anderen Frameworks etc. ist nur nach ausdrücklicher Erlaubnis Ihres Tutors / Ihrer Tutorin zulässig, da er/sie Ihre Abgabe korrigieren wird.

Plattformen

Ihre Anwendung sollte mindestens auf einer der nachfolgenden Plattformen lauffähig sein: Windows 10, Linux.

Netzwerkkommunikation und Nachrichtenprotokoll

Um das Spiel über das Netzwerk spielen zu können, müssen Nachrichten zwischen dem Client und dem Server ausgetauscht werden.
Zum Austausch der Nachrichten müssen JSON kodierte Strings über eine WebSocket Verbindung gesendet werden.
Das Nachrichtenformat wird in einem Netzwerk-Standard-Dokument spezifiert, welches Sie rechtzeitig erhalten werden.

Spielregeln von Hexxagon

Dieses Kapitel beschreibt die Regeln und den Spielablauf des Spiels Hexxagon.

Teilnehmeranzahl

Das Spiel Hexxagon kann von (exakt) **zwei Spielern** gespielt werden.

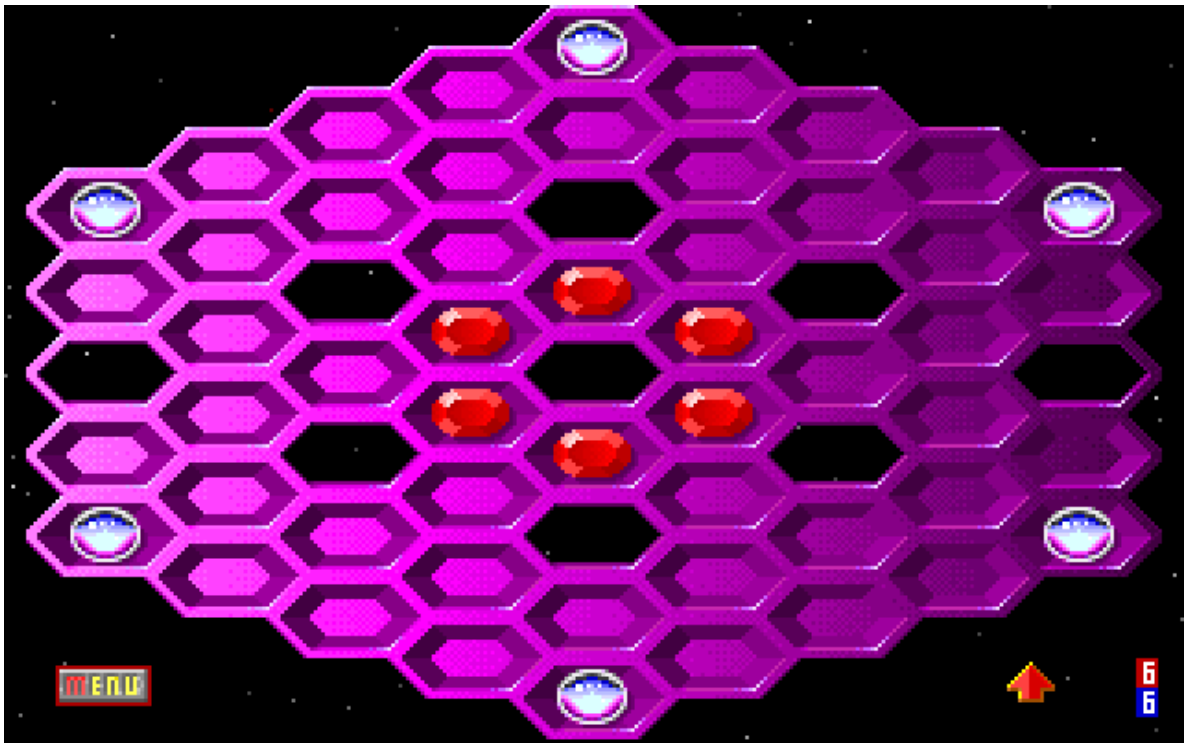
Spielfeld

Das Spielfeld besteht aus 61 hexagonalen Kacheln, und ist wie folgt aufgebaut:

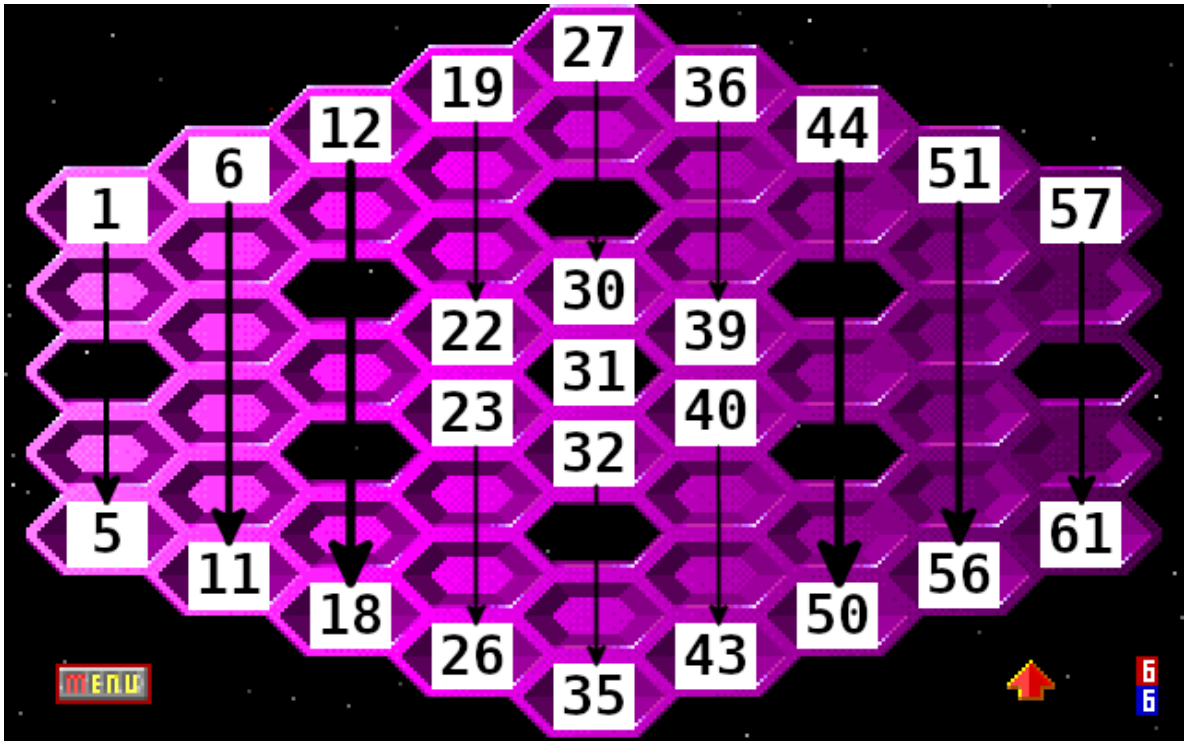
1. Spalte: 5 Kacheln
2. Spalte: 6 Kacheln
3. Spalte: 7 Kacheln
4. Spalte: 8 Kacheln
5. Spalte: 9 Kacheln
6. Spalte: 8 Kacheln
7. Spalte: 7 Kacheln
8. Spalte: 6 Kacheln
9. Spalte: 5 Kacheln

In jede dieser Kacheln kann ein Spielstein gesetzt werden, außer die Kachel ist blockiert (fehlt).

Die originale Version des Spiels Hexxagon lässt es zu, selbst Spielfelder zu konfigurieren. Der Einfachheit halber, soll es in dieser Version des Spiels nur ein Spielfeld geben, die Startkonfiguration sieht wie folgt aus:



Um eine Kachel eindeutig zu identifizieren, wird folgende Nummerierung der Kacheln festgelegt:



Spielsteine

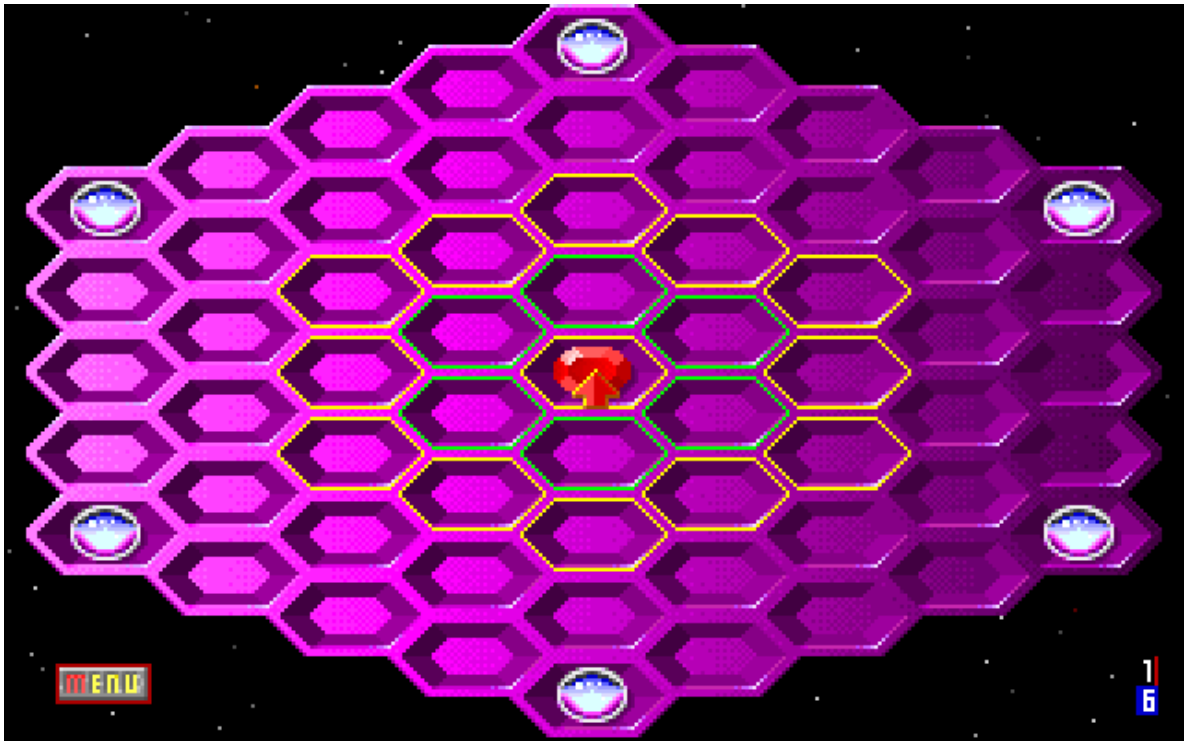
Der erste Spieler bekommt die roten Spielsteine, der zweite Spieler bekommt die weißen Spielsteine.

Spielzüge

Es gibt zwei Möglichkeiten, um mit einem Spielstein zu interagieren.

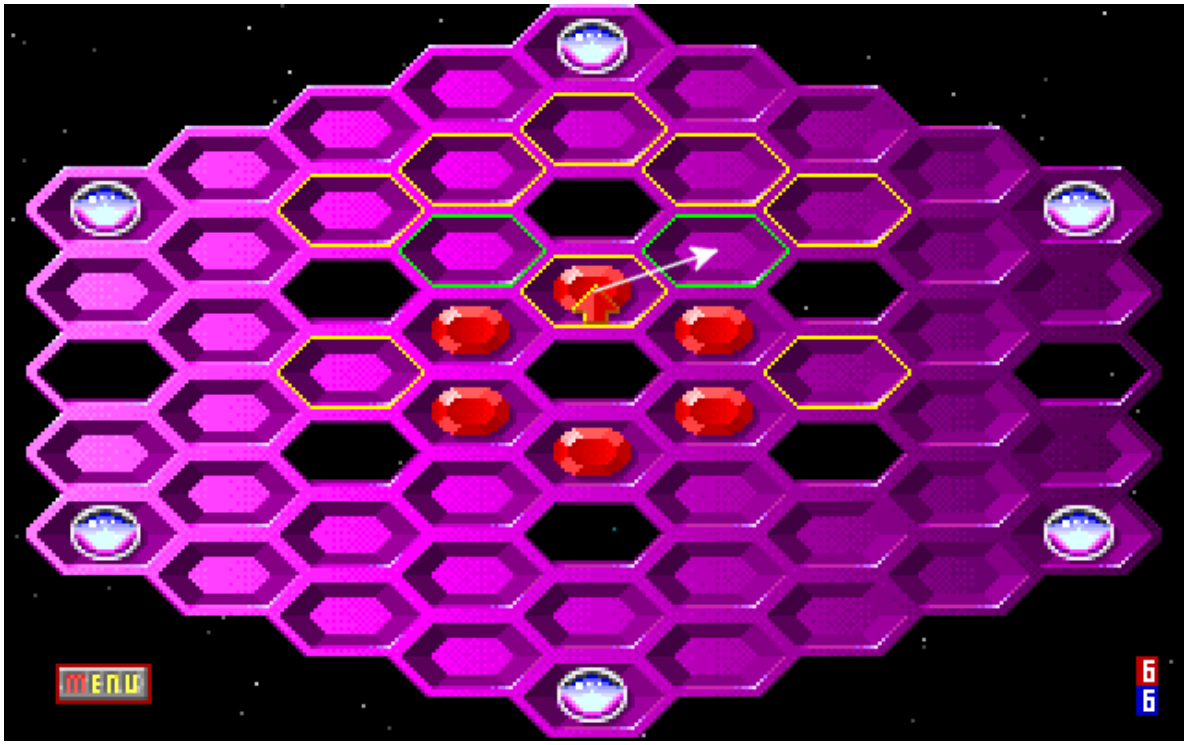
1. Man bewegt den Spielstein auf eine direkt benachbarte Kachel (Kacheln mit grüner Umrandung, siehe Abbildung). Wird der Spielstein auf eine der direkt benachbarte Kacheln bewegt, so erzeugt man dadurch einen neuen Spielstein in der eigenen Farbe.
2. Man bewegt den Spielstein auf eine benachbarte Kachel einer direkt benachbarten Kachel (man überspringt also eine Kachel, Kacheln mit gelber Umrandung, siehe Abbildung). Überspringt man mit dem Spielstein eine Kacheln, so wird kein neuer Spielstein erzeugt, der eigene Spielstein wird lediglich auf die gewählte Kachel gesetzt.

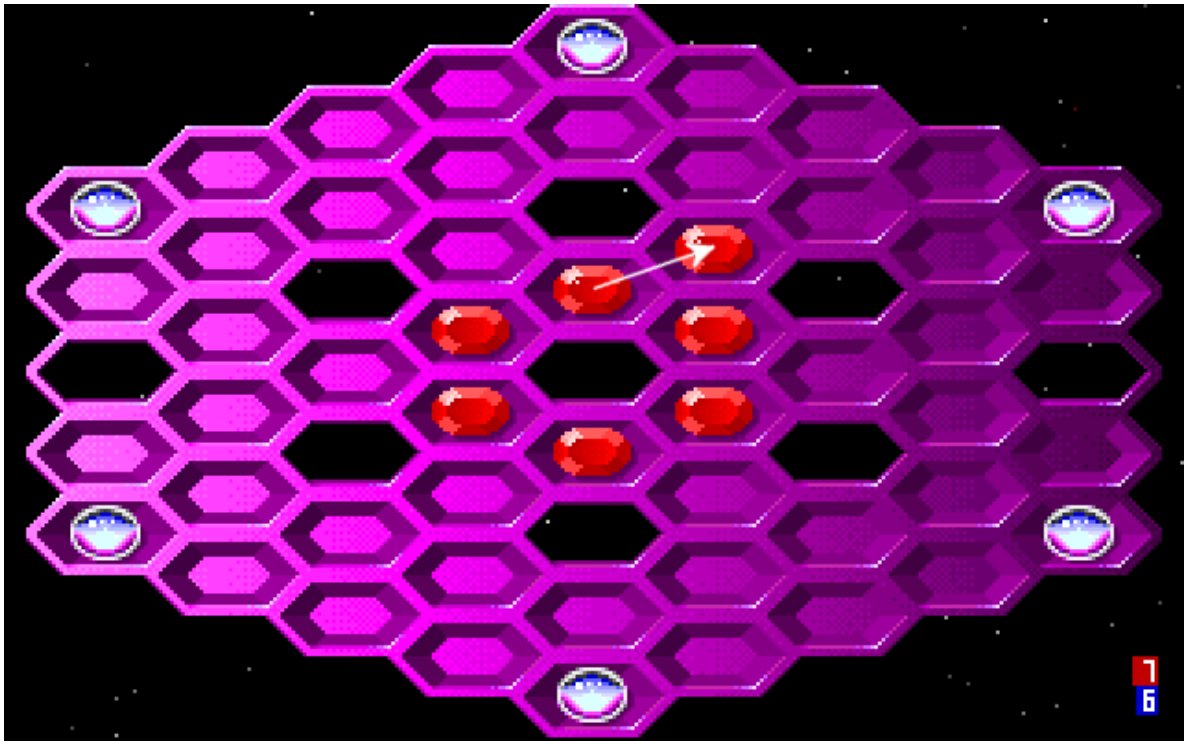
Anmerkung: Ein Spielstein kann **nicht** auf blockierte (leere) Kacheln platziert werden. Veranschaulichende Abbildungen zu dieser Besonderheit sind im Abschnitt *Beispielhafte Spielzüge* zu finden.



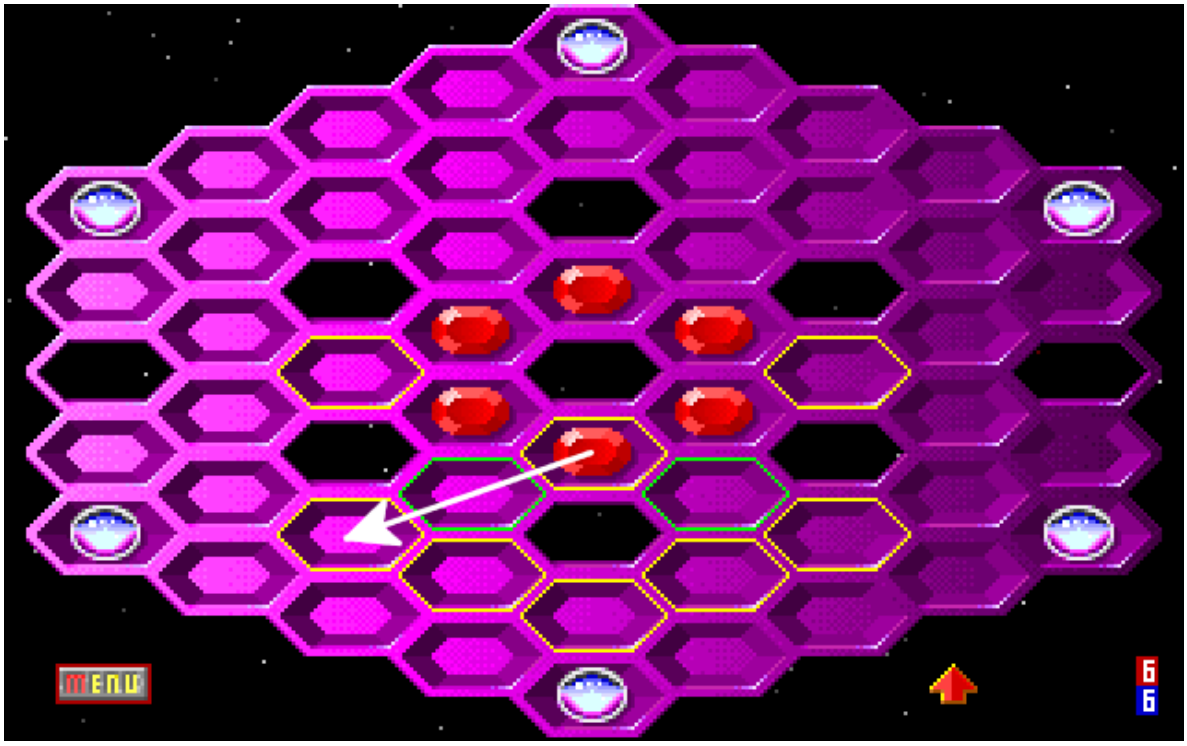
Beispielhafte Spielzüge

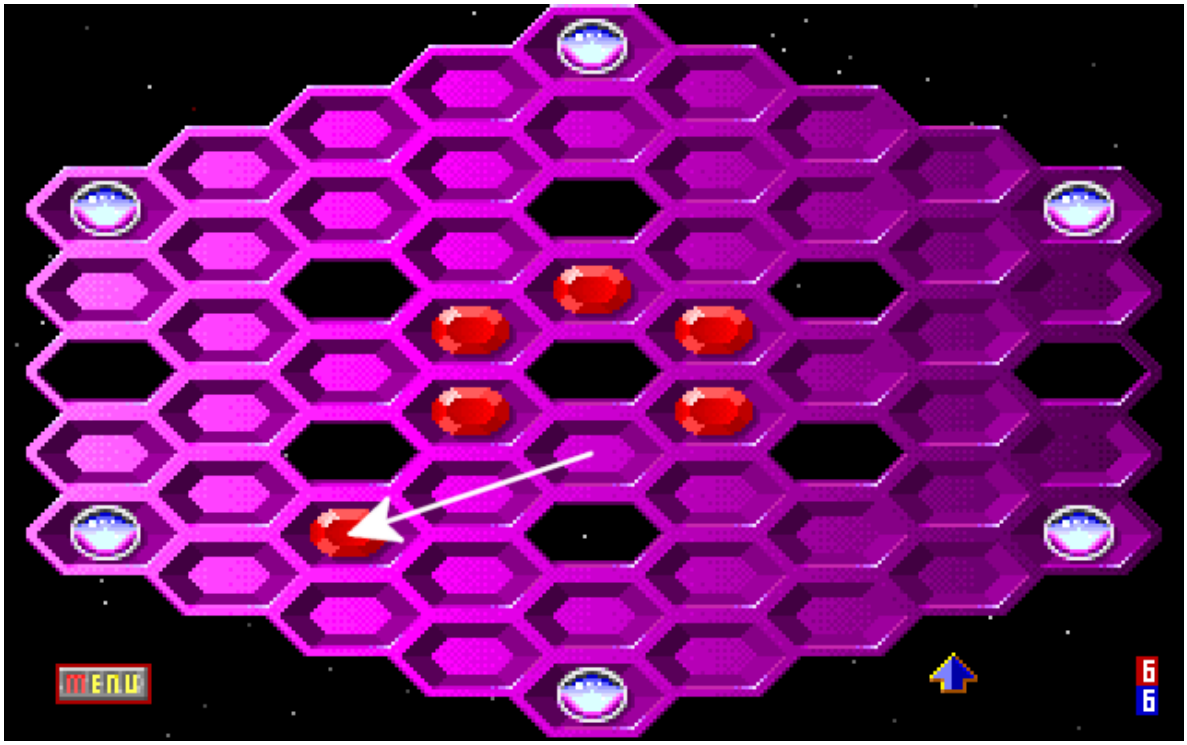
Spielzug ein Feld





Spielzug zwei Felder



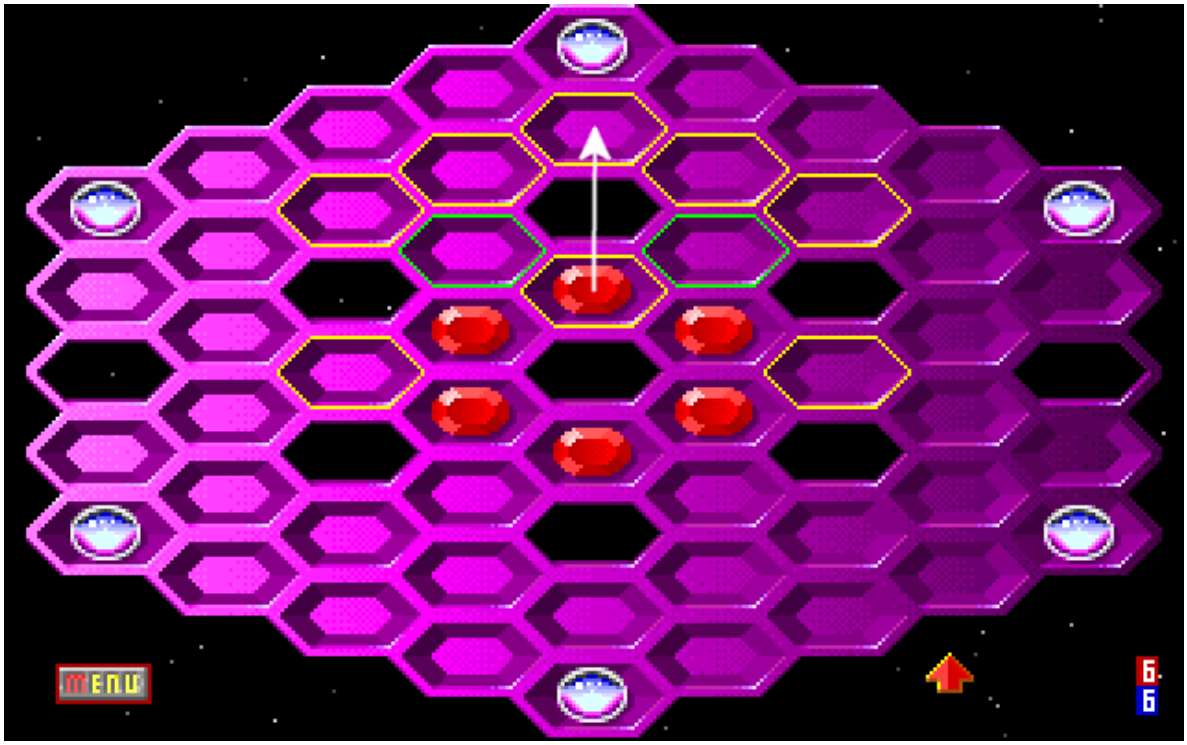


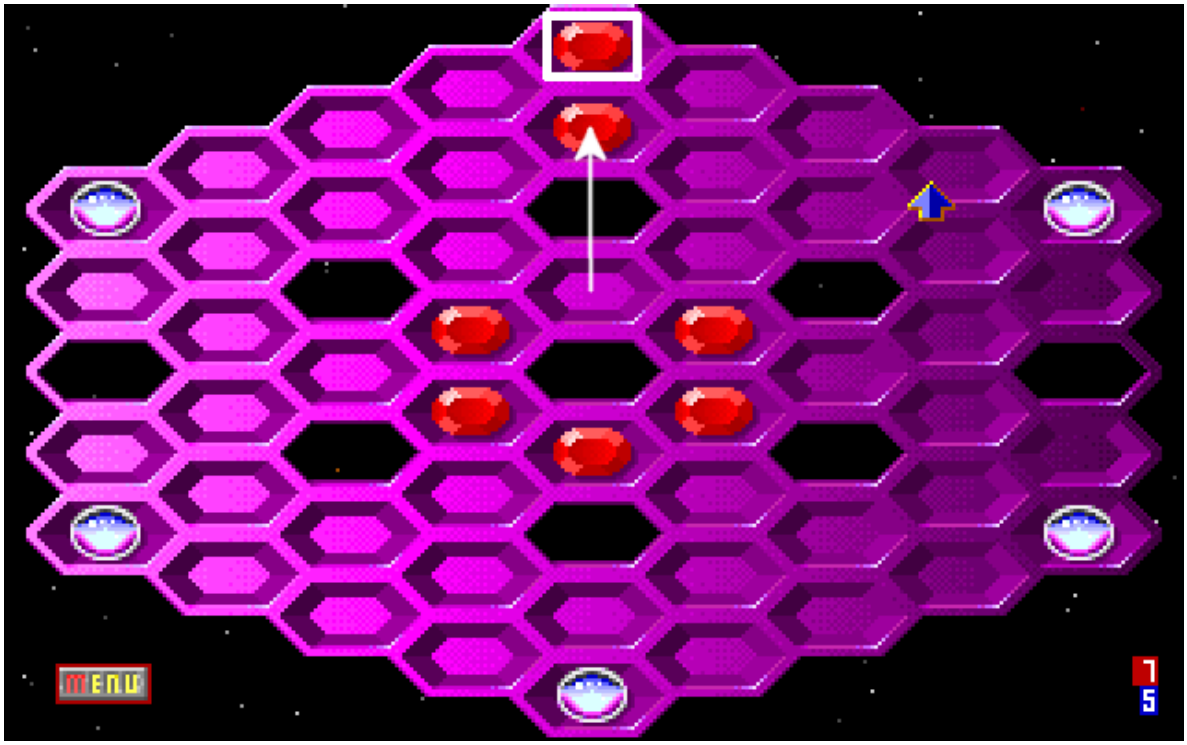
Spielsteine erobern

Wird ein Spielstein bewegt (ein oder zwei Felder weit) und es liegen gegnerische Spielsteine an den direkt benachbarten Kacheln an, so werden die gegnerischen Spielsteine zu eigenen Spielsteinen.

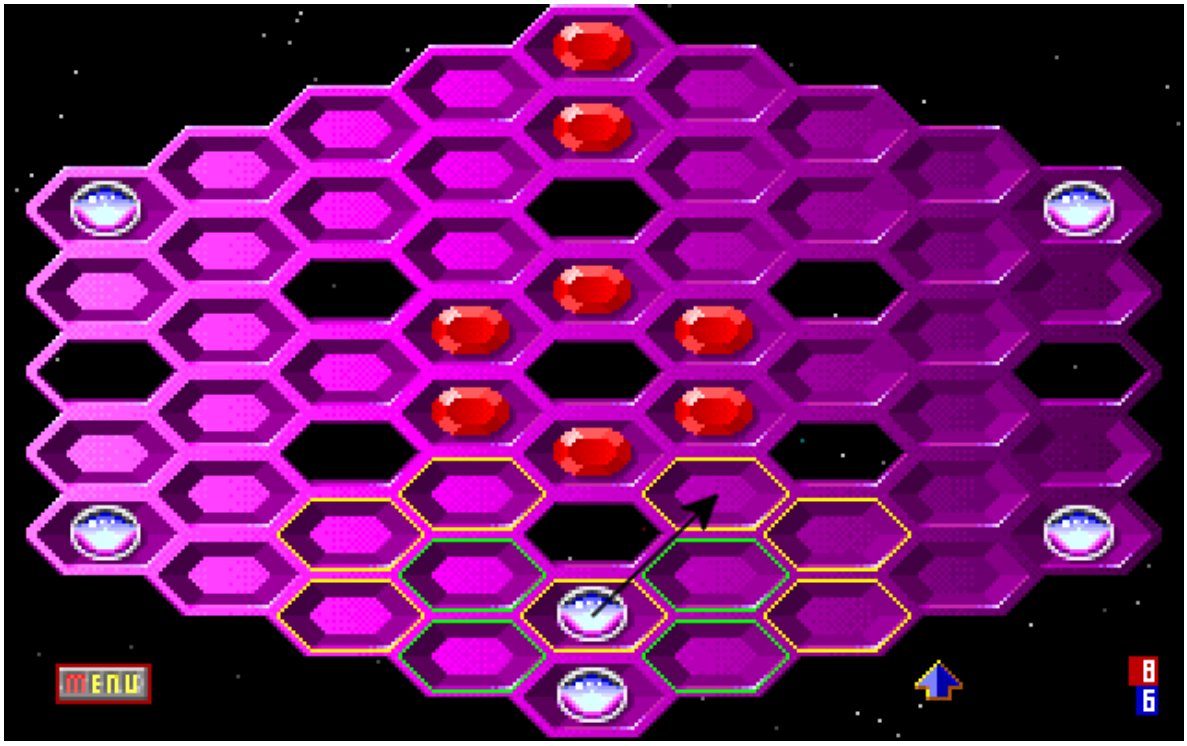
Beispielhafte Attacken

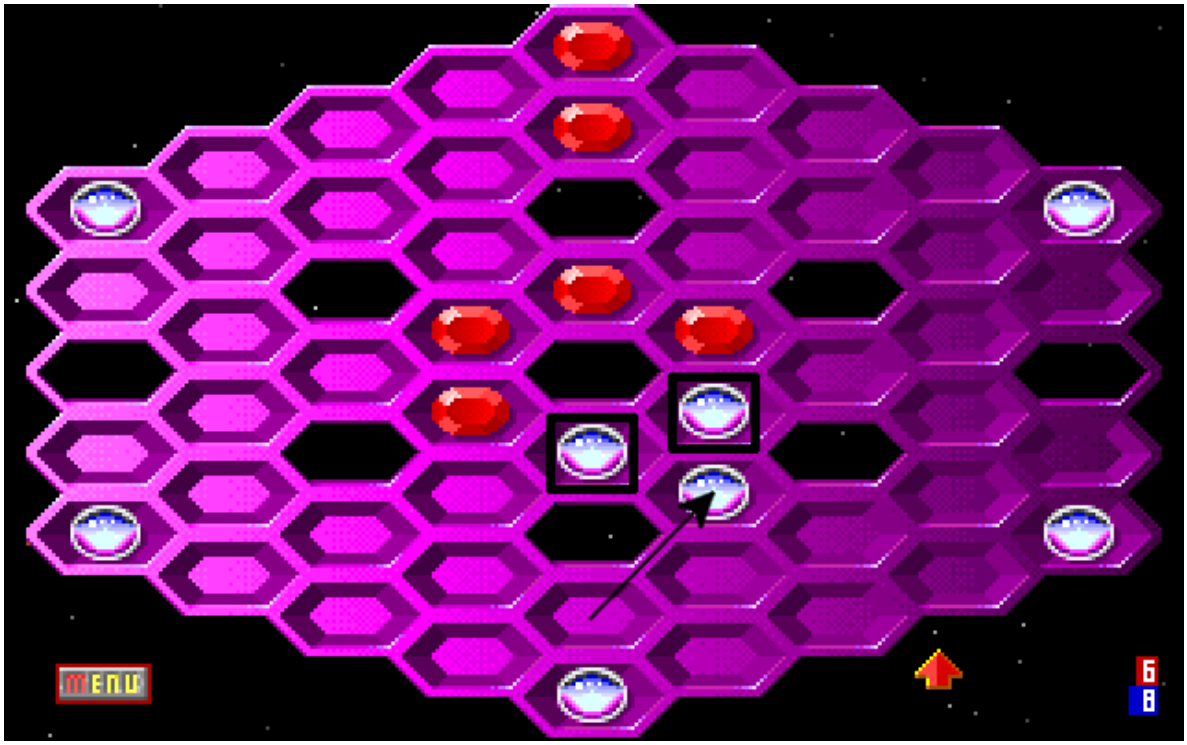
Attacke 1



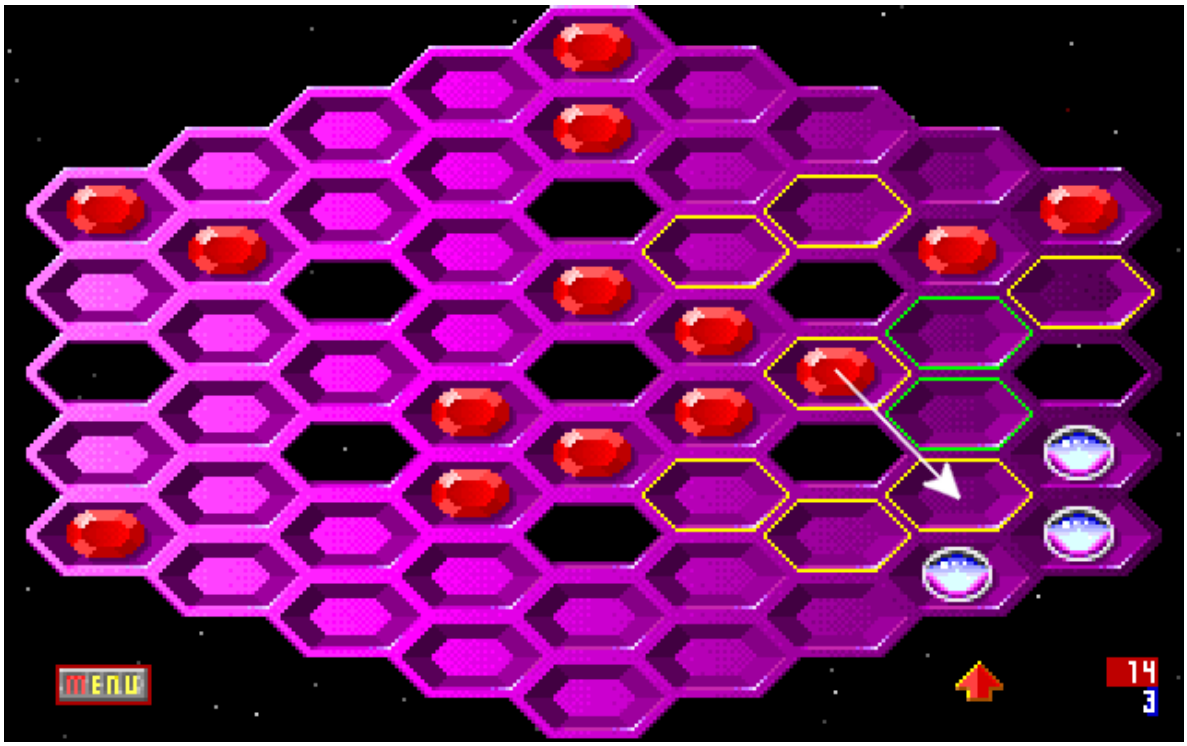


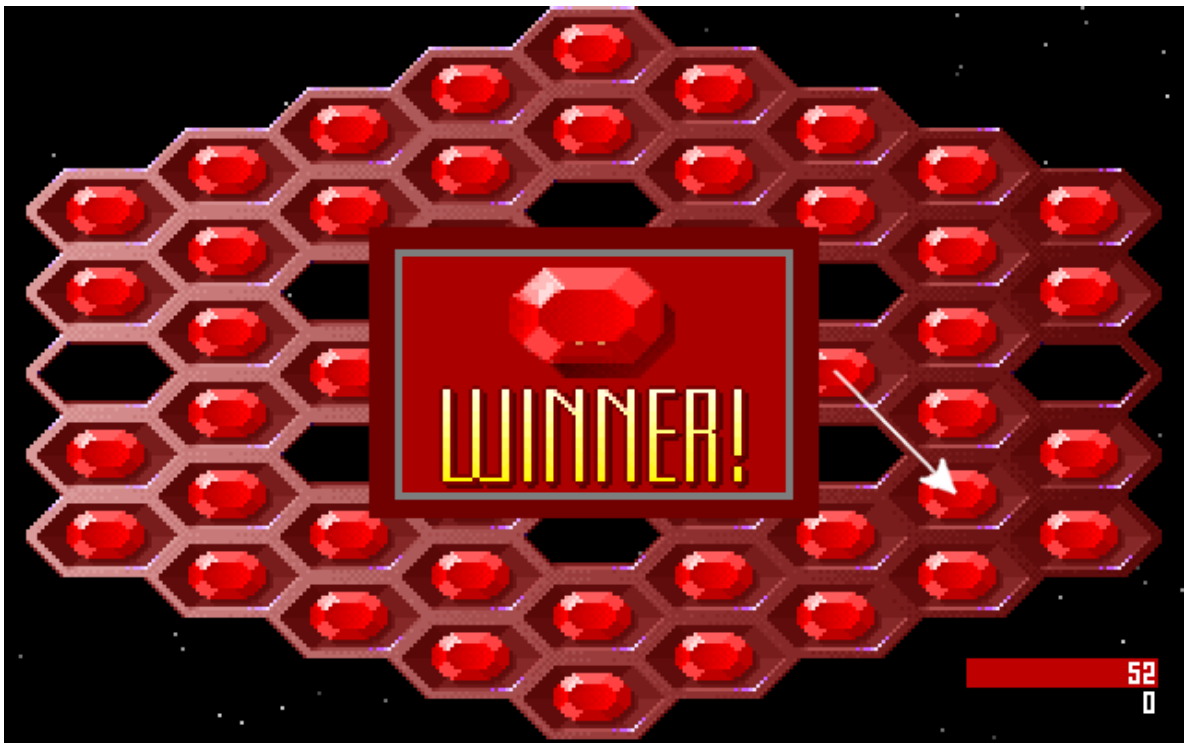
Attacke 2





Attacke 3 (mit Siegbedingung)





Siegbedingungen

1. Ein Spieler erobert alle Spielsteine des Gegenspielers, sodass nur noch Spielsteine einer Farbe auf dem Spielbrett vorhanden sind.
2. Es kann kein weiterer Zug ausgeführt werden, es gewinnt der Spieler mit den meisten Spielsteinen.

Spielverlauf

Es beginnt Spieler 1 (hier Spieler mit den roten Spielsteinen) und macht den ersten Zug.

Danach führt Spieler 2 seinen Spielzug aus.

Dieser Vorgang wird solange wiederholt, bis eine Siegbedingung eintritt.

Wichtig: Jeder Spieler darf nur einen Spielstein pro Spielzug bewegen.

Vorgaben zum Entwicklungsprozess

Die Implementierung muss alleine erfolgen.
Das Projekt muss in einem Git Repository auf gitlab.informatik.uni-ulm.de versioniert werden.
Dokumentieren Sie Ihren Quellcode.
Es wird eine Abnahme des Projekts erfolgen, Ihr Tutor / Ihre Tutorin wird Ihnen eventuell gezielt Fragen zu dem von Ihnen verfassten Quellcode stellen. Sie müssen in der Lage sein, diese Fragen zu beantworten.
Eine erfolgreiche Abnahme des Einzelprojekts ist Voraussetzung zur Teilnahme am zweiten Teil des Softwaregrundprojekts (Sopra).