Höhere Technische Bundeslehranstalt Wien 3, Rennweg

IT & Mechatronik

HTL Rennweg: Rennweg 89b

A-1030 Wien, Tel +43 1 24215-10, Fax DW 18

Diplomarbeit

ISOLA

ausgeführt an der  
Höheren Abteilung für  
Informationstechnologie/Ausbildungsschwerpunk  
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2018/2019

durch  
Johanna Kučera

unter der Anleitung von

Franz Stimpfl

Wien, September 2019

Anmerkung: Gender-Erklärung

Um eine bessere Lesbarkeit zu gewährleisten, werden personenbezogene Bezeichnungen nicht in männlicher und weiblicher Form angeführt. Alle personenbezogenen Bezeichnungen in männlicher Form beziehen sich gleichermaßen auf Frauen und Männer.

Kurzfassung

Isola ist ein rundenbasiertes Brettspiel, bei dem das Ziel der Spieler darin besteht, alle Gegner einzusperren und dadurch bewegungsunfähig zu machen. Dabei besteht jeder Spielzug aus dem Bewegen der eigenen Spielfigur auf ein angrenzendes Feld und dem anschließenden Entfernen eines beliebigen, nicht von einer Spielfigur besetzen Feldes.

Dieses ursprünglich für zwei Spieler entworfene Spiel wurde in diesem Projekt als Online-Multiplayer-Spiel umgesetzt und um einen Vier-Spieler-Modus erweitert. Zusätzlich dazu wurde ein einfach gehaltener Computergegner programmiert, der von den Spielern nach belieben als Gegner ausgewählt werden kann.

Abstract

Isola is a round-based board game where you try to trap your opponents on a single field of the board to render them unable to move. For each player their round consists of moving their piece on a neighbouring field and removing an unoccupied field from anywhere on the board.

This game, originally intended to be played by two players only, was implemented as an online-multiplayer game and extended to support a four-player-mode. Apart from that, players can choose to play against simple computer opponents.

Ehrenwörtliche Erklärung

Ich versichere,

* dass ich meinen Anteil an dieser Diplomarbeit selbstständig verfasst habe,
* dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe
* und mich auch sonst keiner unerlaubten Hilfe bzw. Hilfsmittel bedient habe.

Wien, am

Präambel

Die Inhalte dieser Diplomarbeit entsprechen den Qualitätsnormen für „Ingenieurprojekte“ gemäß § 29 der Verordnung des Bundesministers für Unterricht und kulturelle Angelegenheiten über die Reife- und Diplomprüfung in den berufsbildenden höheren Schulen, BGBl. Nr. 847/1992, in der Fassung der Verordnungen BGBl. Nr. 269/1993, Nr. 467/1996 und BGBl. II Nr. 123/97.

Liste der betreuenden Lehrer

DI, Franz Stimpfl – Hauptbetreuer

Inhaltsverzeichnis

[Anmerkung: Gender-Erklärung ii](#_Toc15569728)

[Kurzfassung iii](#_Toc15569729)

[Ehrenwörtliche Erklärung v](#_Toc15569730)

[Präambel vi](#_Toc15569731)

[1 Einleitung 1](#_Toc15569732)

[2 Ziele 2](#_Toc15569733)

[2.1 Hauptziele 2](#_Toc15569734)

[2.2 Optionale Ziele 3](#_Toc15569735)

[2.3 Nicht-Ziele 3](#_Toc15569736)

[3 Technologien 4](#_Toc15569737)

[3.1 Node.js 4](#_Toc15569738)

[3.2 socket.io 4](#_Toc15569739)

[3.3 Phaser 3 5](#_Toc15569740)

[4 Umsetzung 6](#_Toc15569741)

[4.1 Design 6](#_Toc15569742)

[4.1.1 Isometrische Darstellung 6](#_Toc15569743)

[4.1.2 Themen 7](#_Toc15569744)

[4.1.3 Spielfiguren 8](#_Toc15569745)

[4.1.4 Spiel & Website 8](#_Toc15569746)

[4.2 Website 10](#_Toc15569747)

[4.2.1 Allgemeines 10](#_Toc15569748)

[4.2.2 Tutorial 10](#_Toc15569749)

[4.3 Spiel 13](#_Toc15569750)

[4.3.1 Lobby 13](#_Toc15569751)

[4.3.2 Spielablauf 14](#_Toc15569752)

[4.3.3 Client-Server-Kommunikation 14](#_Toc15569753)

[4.3.4 Computergegner 15](#_Toc15569754)

[Literaturverzeichnis 17](#_Toc15569755)

[Abbildungsverzeichnis 17](#_Toc15569756)

[Stichwortverzeichnis 18](#_Toc15569757)

# Einleitung

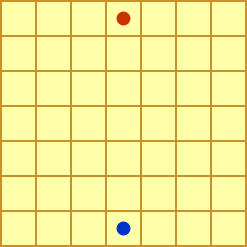
Isola ist ein Brettspiel für zwei Spieler mit relativ einfachen Regeln. Das Spielfeld besteht aus 7x7 Feldern, die Spielfiguren stehen sich gegenüber. Typisch für ein Brettspiel sind beide Spieler abwechselnd am Zug.

Abbildung ‎1.1: Spielfeld vor Beginn des Spiels

Jeder Zug besteht aus zwei Aktionen: Dem Bewegen der Spielfigur auf ein angrenzendes Feld (waagrecht, senkrecht oder diagonal) und dem Entfernen von einem beliebigen Feld auf dem gesamten Spielbrett. Die einzige Bedingung für die gewählten Felder ist, dass keine Spielfigur auf ihnen stehen darf. Kann ein Spieler seine Figur nicht mehr bewegen, hat er verloren. [[1]](#l_1)

Isola wird gerne verwendet, um zu lernen oder zu üben, wie man Computergegner programmiert. Das liegt daran, dass es sowohl wenige, einfache Regeln als auch ein konsistentes Ergebnis hat. Im Gegenteil zu anderen einfachen Spielen wie TicTacToe gibt es nämlich am Ende des Spiels immer einen Gewinner und einen Verlierer. Ein Gleichstand kann nicht auftreten.

Dank des einfachen Spielprinzips ist es auch einfach, das Spiel zu erweitern. So können mit einem größeren Spielfeld und zwei weiteren Spielfiguren vier Spieler gemeinsam spielen. Falls einem das Spiel zu lange dauert, kann man einführen, dass auch die Felder, von denen ein Spieler weggeht, entfernt werden. Mit zusätzlichen Spielelementen könnten noch mehr Variationen hinzugefügt werden.

Sowohl die einfachen Regeln als auch die Erweiterungsmöglichkeiten führten dazu, dass Isola die Grundlage für dieses Projekt bildet.

# Ziele

Um bei einem Projekt schon zu Beginn zu wissen, was zu erreichen ist, werden schon am Anfang die Ziele definiert. Diese sind immer so formuliert, dass sie den gewünschten Endzustand beschreiben. Zusätzlich sollte das SMART-Schema eingehalten werden. Das heißt, dass jedes Ziel spezifisch, messbar, attraktiv, realistisch und terminisiert ist. [[2]](#l_2)

## Hauptziele

Die Hauptziele sind die wichtigsten Ziele eines Projektes. Sie beschreiben das Kernprojekt, das, was für die vollständige Funktionalität vorhanden sein muss. Sollten sie nicht alle erfüllt sein, ist das Projekt gescheitert.

* Ein Isola-Spiel im isometrischen Design ist Anfang September fertig erstellt und spielbar.
  + Es gibt einen Spielmodus für zwei und für vier Spieler.
  + Das Spiel ist ein Online-Multiplayer, d.h. es kann von mehreren Endgeräten aus gemeinsam gespielt werden.
* Das Spiel ist Anfang September in eine Webseite eingebunden, die die Spielregeln erklärt.
* Ein einfacher Computergegner ist Anfang September als Gegenspieler verfügbar.
* Es ist Anfang September möglich, einen Raum für ein Spiel zu erstellen.
  + Für die Anzahl der teilnehmenden Spieler kann zwei oder vier Spieler ausgewählt werden.
  + Die Anzahl an computergesteuerten Spielern kann ausgewählt werden.
  + Es gibt eine Lobby, in der die verfügbaren Räume zu sehen sind und man einen Raum aussuchen kann, in dem man mitspielen will.
* Das Diplomarbeitsbuch ist Anfang September fertig geschrieben.
* Eigene Grafiken für das Spiel sind Anfang September erstellt.

## Optionale Ziele

Die optionalen Ziele sind Ziele, die man gerne erfüllt hätte, die aber nicht über Gelingen und Scheitern des Projekts entscheiden. Sie beschreiben zusätzliche Funktionen, die das Kernprojekt erweitern und interessanter machen. Da diese Ziele nicht erfüllt werden müssen, können sie auch ohne fixe Termine definiert werden.

* Das Spiel kann auch als Local Co-Op gespielt werden.
* Bei verschiedenen Aktionen werden Animationen abgespielt.
* Das Aussehen der eigenen Spielfigur kann selbst ausgewählt werden.
* Es ist möglich, bei laufenden Spielen zuzuschauen.
* Es gibt einen Startschirm mit Ladebalken.
* Es gibt verschieden starke Computergegner (z.B. mit Minimax-Algorithmus)
* Es gibt Hintergrundmusik und Sounds bei Aktionen.
  + Die Lautstärke der Hintergrundmusik und der Sounds kann unabhängig voneinander verstellt und auch auf null gesetzt werden.

## Nicht-Ziele

Nicht-Ziele sind eine gute Möglichkeit, um ein Projekt einzugrenzen. Sie sollen verhindern, dass das Projekt in ungewünschte Richtungen abschweift. Für ein Projekt, bei dem nur ein Haus gebaut werden soll, kann hier zum Beispiel stehen, dass ein Garten angelegt wird. Trotz der Bezeichnung Nicht-Ziele werden auch diese Ziele positiv formuliert.

* Eine Chat-Funktion ist im Spiel verfügbar.
* Einstellungen und Statistiken werden über Browser-Sessions hinweg gespeichert.
* Es ist für den Benutzer möglich, eigene Grafiken/Sounds hochzuladen und im Spiel zu verwenden.
* Es gibt ein interaktives Tutorial.
* Das Spiel kann auch am Handy problemlos gespielt werden.

# Technologien

Um ein Projekt umzusetzen, werden verschiedene Technologien benötigt. Dieses Projekt basiert auf der Markup-Sprache HTML (Hyper Text Markup Language), die zum Erstellen der Strukturen einer Website verwendet wird, sowie der Formatierungssprache CSS (Cascading Style Sheets), in welcher das Aussehen der Website definiert wird. Zusätzlich dazu wird die Programmiersprache JavaScript verwendet, um alle Funktionalitäten einzubauen.

Obwohl man dieses Projekt mit genau diesen drei Technologien hätte umsetzen können, wurde dies nicht getan. Stattdessen wurden verschiedene JavaScript- Frameworks verwendet. Frameworks sind Sammlungen von grundlegenden Funktionen, die man weiterverwenden kann. Die wichtigsten werden nachfolgend ausführlich erklärt.

## Node.js

Node.js ermöglicht es, JavaScript serverseitig laufen zu lassen und so einen Software-Server zu erstellen. Im Gegensatz zu üblichen Modellen, die immer auf Antworten warten, dafür aber mehrere Anfragen gleichzeitig bearbeiten können, kann Node.js nur eine Aufgabe nach der anderen ausführen.

Allerdings arbeitet Node.js auch asynchron, das heißt, es wartet nicht auf Antwort vom Client. Synchron würde heißen, eine Anfrage zu erhalten, diese zu bearbeiten, die Antwort zu schicken und danach nichts anderes zu tun, als auf die nächste Antwort oder Anfrage vom Client zu warten. Um mit diesem System mehrere Clients gleichzeitig zu verwalten, müssen Anfragen parallel bearbeitet werden können. Node.js reagiert aber nur auf erhaltene Anfragen, der letzte Teil des synchronen Ablaufs fällt weg. Statt zu warten, wird die nächste Anfrage, eventuell von einem anderen Client, bearbeitet. [[3]](#l_3)

Node.js bietet damit einen JavaScript-Server, der als Grundlage für die auf socket.io basierende Client-Server-Kommunikation dient.

## socket.io

socket.io ist ein JavaScript-Framework, dass eine eventbasierte Echtzeit-Kommunikation zwischen Client und Server ermöglicht. Dazu werden sowohl am Client als auch am Server Events definiert, auf die jeweils mit einem bestimmten Verhalten reagiert werden soll. Zum besseren Verständnis hier ein Beispiel. Am Server gibt es den folgenden Code:

socket.on(“im\_fine“, function() {

socket.emit(“good\_to\_hear”);

});

Der Client wird hier als socket bezeichnet. Wenn von diesem ein Event mit dem Namen „im\_fine“ kommt, soll der Server reagieren und ein Event namens „good\_to\_hear“ zurückschicken.

Am Client ist der folgende Code zu finden, der das Event am Server auslöst:

socket.emit(“im\_fine“);

Auch hier kann mit der Syntax socket.on auf andere Events vom Server reagiert werden.

Eine zusätzliche Funktion von socket.io ist, dass Clients Räumen beitreten können. Vom Server aus kann man Events dadurch auf drei verschiedene Arten schicken: an alle verbundenen Clients, an alle Clients in einem Raum oder an einen bestimmten Client. [[4]](#l_4) Diese Funktionalität ist gerade für dieses Projekt eine große Hilfe, um mehrere Spiele gleichzeitig am Laufen haben zu können.

## Phaser 3

Phaser 3 ist ein Spiele-Framework, basierend auf JavaScript, dass WebGL (Web Graphics Language) und Canvas unterstützt. [[5]](#l_5)

Canvas ist eine der Möglichkeiten, im Browser Bilder darzustellen und interaktive Grafiken zu erzeugen. WebGL erweitert diese Funktionalitäten und bietet eine plattformunabhängige Basis. [[6]](#l_6) Daher verwendet Phaser 3 auch hauptsächlich WebGL, während Canvas, falls nicht ausdrücklich im Programm angegeben, nur als Ausweichmöglichkeit genommen wird, wenn WebGL nicht funktionieren sollte.

Phaser 3 ermöglicht einen einfachen Umgang mit Grafiken, verschiedenen Szenen und Interaktionen des Benutzers. Es bietet viele Funktionalitäten, die das Erstellen von Spielen maßgeblich erleichtern, und ist daher eine gute Wahl, um schnell und einfach Spiel zu erstellen.

# Umsetzung

## Design

### Isometrische Darstellung

Wenn man Brettspiele am Computer umsetzen will, hat man eine grundlegende Entscheidung zu treffen: stellt man das Brett von oben rein zweidimensional dar, oder nimmt man eine dreidimensionale Darstellung?

Die zweite Möglichkeit bietet wesentlich mehr Freiheit in der Gestaltung, und ist meistens auch interessanter anzuschauen. Daher wurde auch für dieses Projekt ein dreidimensionaler Ansatz in der Darstellung gewählt. Eine relativ einfache Methode, dies zu erreichen, ist über eine isometrische Darstellung.

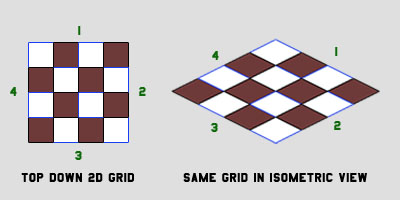
Isometrie wird sehr gerne für Computerspiele verwendet, da die Berechnung dieser Darstellung im Vergleich zu anderen dreidimensionalen Optionen relativ einfach ist. Um eine von oben betrachtete quadratische Ebene isometrisch darzustellen, muss sie erst um 45° um ihren Mittelpunkt rotiert werden, und dann um 30° nach hinten gekippt werden. Dadurch erhält man einen Rhombus, der halb so hoch wie breit ist – eines der Hauptmerkmale einer isometrischen Darstellung. [[7]](#l_7)

Abbildung ‎4.1: Umwandlung einer Ebene in isometrische Darstellung

### Themen

Um mehr Abwechslung zu bieten, wurden drei verschiedene Themen entworfen und eingearbeitet. Da die Spielfelder Blöcke sind, die auch entfernt werden können, wurde darauf Rücksicht genommen, dass diese Eigenschaft auch zum Thema passt.

#### Weltraum

Das erste Thema ist eine Weltraum-Szene. Ein Sternenhimmel bildet den Hintergrund, während die Spielfiguren auf Metallblöcken, die an ein Raumschiff erinnern sollen, stehen. Als Hintergrundfarbe für die Website und diverse Menüs wurde ein helles Grau gewählt, die Schrift ist in Schwarz gehalten und dadurch gut lesbar.

Abbildung ‎4.2: Weltraum-Thema

#### Wasser

Ein weiteres Thema stellt eine Wasserlandschaft dar. Auf einer Wasserfläche als Hintergrund schwimmen lauter Holzkisten, auf denen die Spieler stehen können. Die hellgelbe Hintergrundfarbe soll an einen Sandstrand erinnern, die in Schwarz gehaltene Schrift ermöglicht eine gute Lesbarkeit.

Abbildung ‎4.3: Wasser-Thema

#### Lava

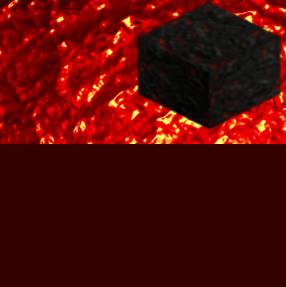
Das letzte Thema ist in einer Lavaumgebung angesiedelt. Auf einem feurig glühenden See aus Lava schwimmen dunkle, aus erstarrter Lava bestehende Blöcke. Ein dunkles Rot als Hintergrundfarbe ergänzt das feurige Setting. Um die Lesbarkeit der Schrift zu gewährleisten, ist diese weiß gefärbt.

Abbildung ‎4.4: Lava-Thema

### Spielfiguren

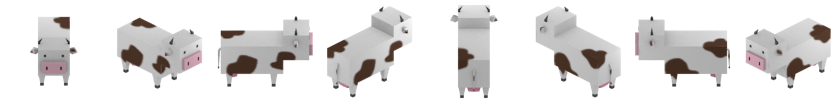
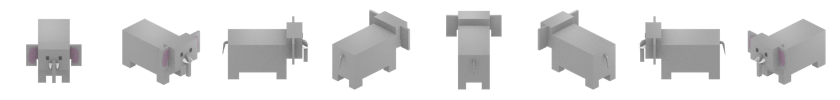
Um den Spielfiguren ein nettes Aussehen zu geben, wurden das Thema Tiere gewählt. Bei der Erstellung der Figuren wurde darauf geachtet, so wenige runde Elemente wie möglich zu haben, um die Tiere einfach zu halten und ein durchgängiges Design zu bilden. Die Spielfiguren wurden mit dem 3D-Programm Blender gemacht.

Abbildung ‎4.6: Spritesheet des Elefanten

Abbildung ‎4.5: Spritesheet der Kuh

Um die Tiere in jede Richtung schauen lassen zu können, wurden Spritesheets erstellt. Das sind Bilder, die aus mehreren gleich großen Einzelbildern bestehen. Auf diese Einzelbilder kann dann im Code direkt zugegriffen werden. Diese Vorgehensweise hat den Vorteil, dass man wesentlich weniger Bilder laden muss. Insgesamt wurden vier verschiedene Spielfiguren erstellt: Kuh, Elefant, Raupe und Schwein.

### Spiel & Website

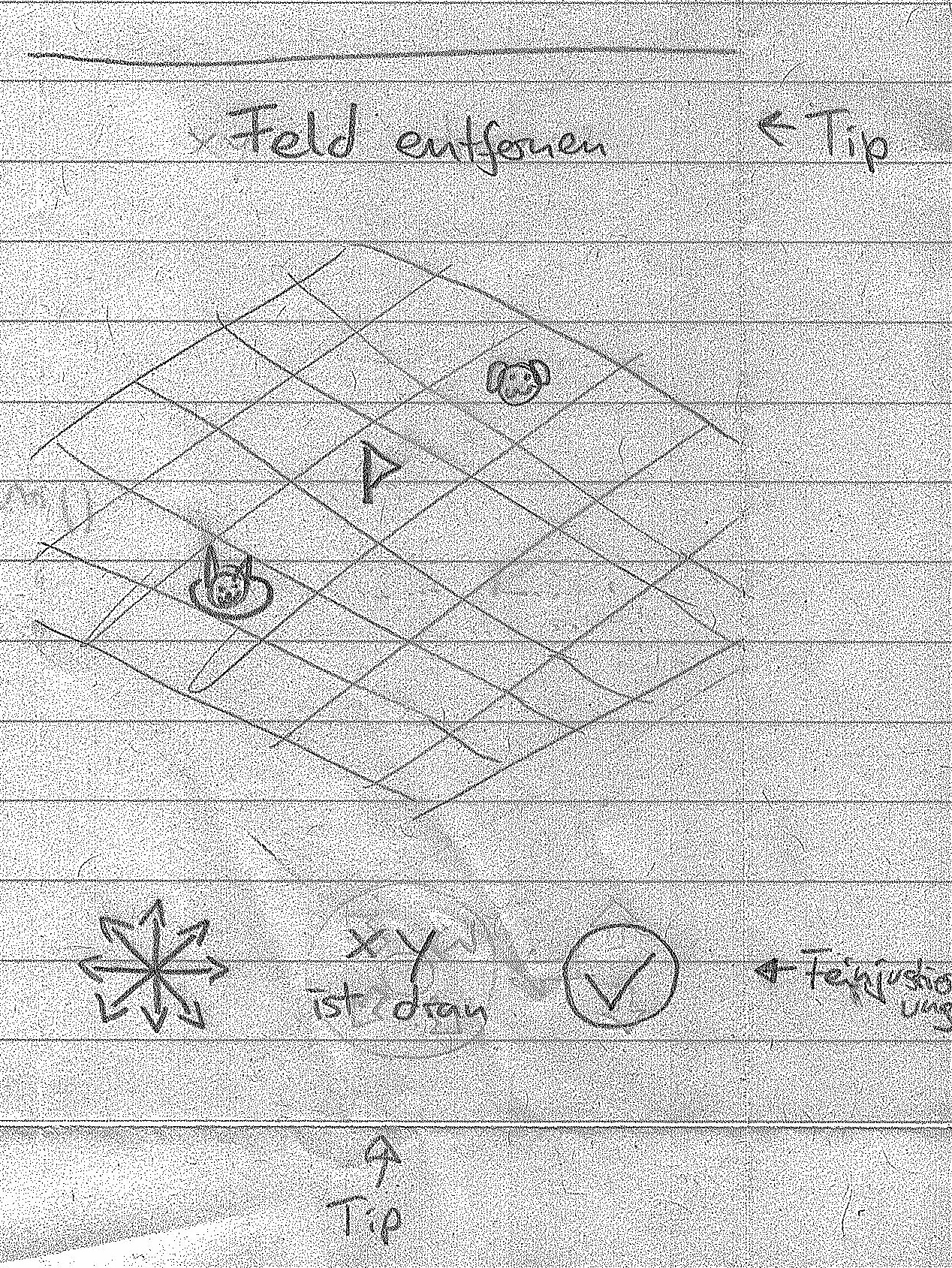
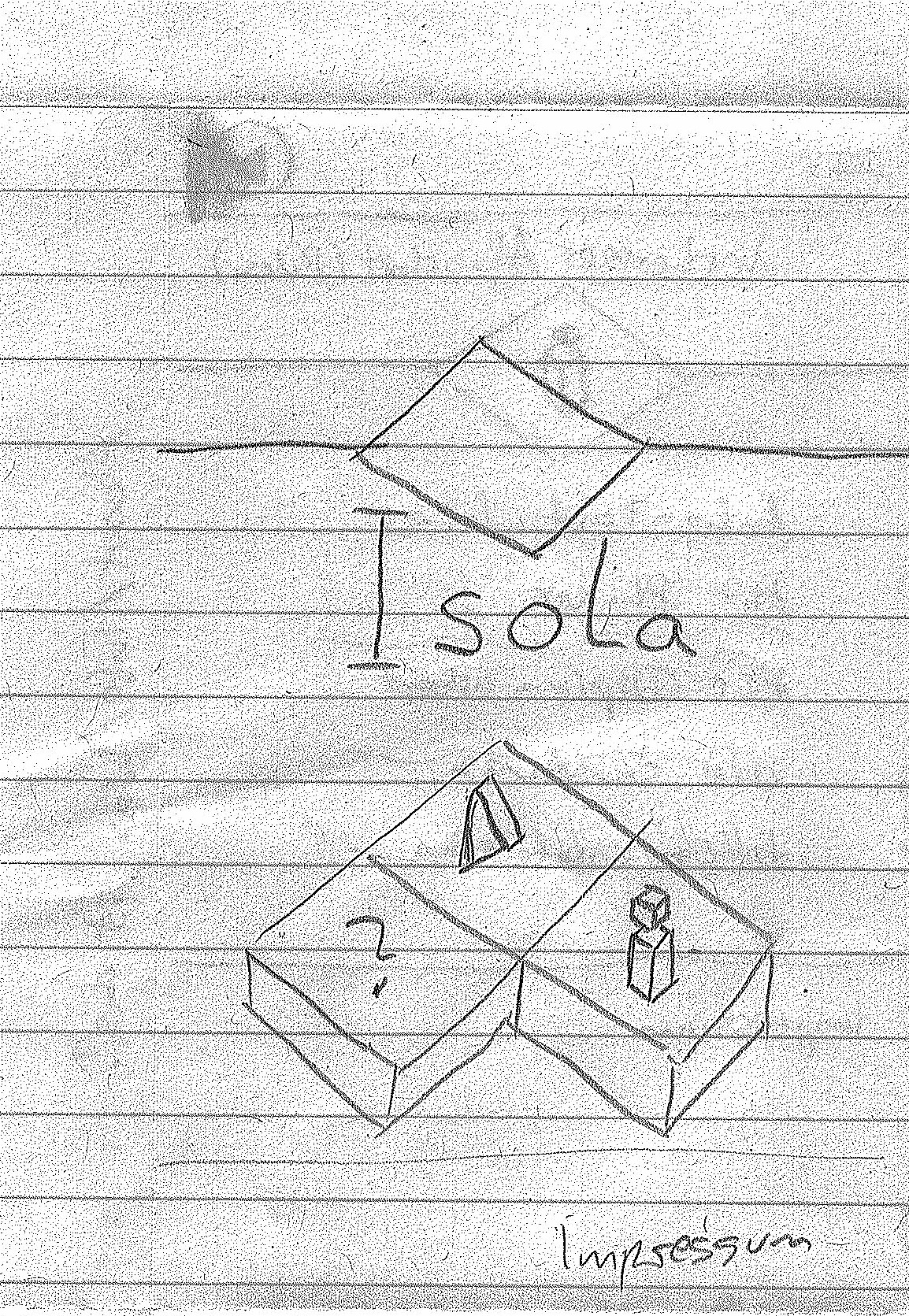
Wenn eine Website entworfen werden soll, ist immer die Frage, ob mobile Geräte unterstützt werden sollen oder nicht. Anfangs war die Idee, nach dem üblichen Mobile-First-Ansatz vorzugehen, bei dem Website und Spiel zuerst für mobile Geräte entworfen werden und dann für Laptops und PCs abgeändert werden, um auch dort ein gutes Bild zu geben. Doch nach den ersten Entwürfen war schnell klar, dass es sehr umständlich wird, das Spiel für mobile Geräte zu optimieren. Schon für den Zwei-Spieler-Modus, der ein relativ kleines Spielfeld hat, müssten zusätzliche Elemente zur Feinjustierung der Auswahl bereitgestellt werden. Als dann Recherchen über mögliche Methoden zur Kennzeichnung der Auswahl eine Veränderung der Felder, wenn die Maus über diesem ist, als schnelle und einfache Möglichkeit ergaben, wurde die Optimierung für mobile Geräte endgültig verworfen.

Abbildung ‎4.7: Erster Entwurf des Spiels

Abbildung ‎4.8: Erster Entwurf der Startseite

Eines der Hauptelemente, das aus den Entwürfen geblieben ist, sind die Blöcke auf der Website sowie als Kennzeichnung für verschiedenste Menüs.

Abbildung ‎4.10: Kennzeichnung eines Menüs

Abbildung ‎4.9: Das Hauptelement der Website

Alle Menüs im Spiel sind mit einfachen und bekannten Icons ausgestattet, der Text wurde auf ein Minimum beschränkt. So gibt es ein Plus, um etwas neues hinzuzufügen, ein Symbol einer Person, um einen menschlichen Spieler anzuzeigen und ein Symbol eines Desktops, um auf einen Computergegner hinzuweisen.

## Website

### Allgemeines

Ein Spiel allein zu programmieren und online zu stellen ist zwar möglich, aber es ist schöner, wenn es in einer passenden Website eingebunden ist. Dort können dann auch zusätzliche Informationen zu finden sein, und es ist ein guter Platz, um ein Tutorial unterzubringen.

Die Website und auch das Spiel sind vollständig in Englisch gehalten, um einer breiteren Masse an Personen zugänglich zu sein.

### Tutorial

Um den Aufwand eines interaktiven Tutorials zu sparen und trotzdem dafür sorgen zu können, dass sich alle Spieler auskennen und zurechtfinden, wurde ein textbasiertes Tutorial mit Bildern, Animationen in die Website eingebaut. Um die Länge der Seite nicht zu extrem werden zu lassen, ist das Tutorial in mehrere Bereiche gegliedert, von denen immer nur einer angezeigt wird. Im Folgenden ist das übersetzte Tutorial ohne die Animationen zu lesen.

#### Generelles

In diesem rundenbasierten Spiel ist das Ziel, Ihre Gegner auf einem einzelnen Feld einzusperren und die letzte Person zu sein, die sich noch bewegen kann.

Um an einem Spiel teilzunehmen, müssen Sie einem Raum beitreten, der noch leere Plätze hat. Das Spiel startet, sobald genügend Spieler dem Raum beigetreten sind. Es besteht die Wahl zwischen Zwei-Spieler und Vier-Spieler-Räumen.

Abbildung ‎4.12: Ein Vier-Spieler-Raum mit zwei freien Plätzen und Computergegnern an zweiter und vierter Stelle

Abbildung ‎4.11: Ein voller Zwei-Spieler-Raum

Um die verschiedenen Spieler unterscheiden zu können, sind die Figuren mit Farben markiert. Der Spieler, der gerade an der Reihe ist, hat eine stärkere Farbe als die anderen.

Abbildung ‎4.14: Dieser Spieler war schon an der Reihe

Abbildung ‎4.13: Dieser Spieler ist an der Reihe

Falls ein Spieler die Verbindung verliert, während er in einem Raum ist, wird er aus diesem Raum entfernt. Falls das Spiel in diesem Raum schon begonnen hat, wird es abgebrochen, da auf einmal ein Spieler fehlt.

#### Bewegung

Wenn Sie an der Reihe sind, müssen Sie sich zuerst bewegen. Sie können auf jedes leere benachbarte Feld gehen, sowohl gerade als auch diagonal. Die grüne Markierung zeigt, ob ein Feld gültig ist oder nicht.

(Hier ist eine Animation, in der die grüne Markierung gezeigt wird)

Nachdem Sie Ihren Charakter bewegt haben, müssen Sie ein Feld entfernen. Das kann jedes beliebige leere Feld auf dem gesamten Spielbrett sein. Die blaue Markierung zeigt, ob ein Feld gültig ist oder nicht.

(Hier ist eine Animation, in der die blaue Markierung gezeigt wird)

Sobald Sie ein Feld entfernt haben, ist Ihr Zug vorbei und der nächste Spieler ist an der Reihe.

#### Gewinnbedingungen

Um zu gewinnen, müssen Sie die letzte Person sein, die sich noch bewegen kann. Falls die letzten beiden Spieler auf drei verbundenen Feldern eingesperrt sind, gewinnt der Spieler, der das leere Feld entfernt, da er der letzte war, der seine Runde beenden konnte.

(Hier ist eine Animation, die die oben beschriebene Situation zeigt)

#### Räume

Es können immer nur 12 Räume gleichzeitig existieren. Daher wird ein Raum, in dem fünf Minuten lang keine Aktion passiert, entfernt, um Platz für aktivere Räume zu schaffen. Aktionen, die diese Zeit zurücksetzen, sind:

* Das Beitreten eines neuen Spielers in einen Raum,
* das Bewegen einer Spielfigur
* und das Entfernen eines Feldes.

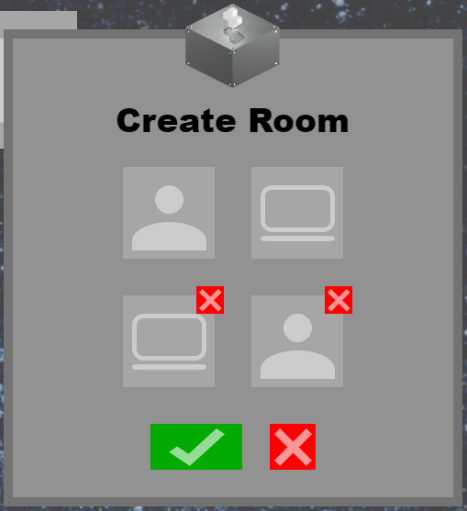
Wenn Sie den Plus-Knopf drücken, können Sie einen neuen Raum erstellen. Sie können dabei die folgenden Einstellungen tätigen: Gesamtanzahl der Spieler (2 oder 4), Anzahl der Computergegner und die Startpositionen der Computergegner.

Abbildung ‎4.17: Ein Vier-Spieler-Raum mit dem 2. & 3. Spieler als Computergegner

Abbildung ‎4.16: Ein Zwei-Spieler-Raum mit nur menschlichen Spielern

Abbildung ‎4.15: Dieser Knopf erstellt einen neuen Raum

Nachdem Sie die Einstellungen festgelegt haben, können Sie das Hakerl anklicken, um den Raum zu erstellen. Falls Sie in die Lobby zurückkehren wollen, ohne den Raum zu erstellen, drücken Sie einfach den x-Knopf.

Bitte verstehen Sie, dass aufgrund der Raumbegrenzung Ihr Raum nicht erstellt werden könnte. Das kann geschehen, wenn mehr Personen versuchen, einen Raum zu erstellen, als die maximale Anzahl erlaubt. In diesem Fall werden nur die Räume erstellt, deren Erstellungsanfragen am schnellsten beim Server ankamen.

## Spiel

### Lobby

#### Allgemeines

Das erste, was nach Starten des Spiels zu sehen ist, ist die sogenannte Lobby. Diese dient dazu, den Spielern einen Überblick über bereits vorhandene Räume zu geben. Dadurch können sie sich entscheiden, ob und welchem Raum sie beitreten wollen: Einen Zwei-Spieler-Raum, in dem jemand auf einen Herausforderer wartet, oder lieber einen Vier-Spieler-Raum mit zwei Computergegnern.

Falls alle Optionen nicht den Wünschen der Spieler entsprechen oder keine freien Plätze mehr vorhanden sind, kann auch ein eigener Raum erstellt werden, bei dem die gewünschten Einstellungen vorgenommen werden.

#### Raumbegrenzung

Um den Server nicht durch die Erstellung von zu vielen Räumen zu belasten, wurde eine Maximalanzahl von Räumen, die gleichzeitig existieren können, eingebaut.

Um Problemen wie einer Beanspruchung aller Räume von einer Person etwas vorzubeugen, gibt es zusätzlich ein Zeitlimit: Räume, in denen fünf Minuten lang nichts neues geschieht, werden gelöscht.

Dazu wird für jeden Raum der Zeitpunkt der letzten Aktion gespeichert. Jede Minute überprüft der Server, ob ein Raum das Zeitlimit überschritten hat und wenn ja, zerstört ihn. Dadurch kann die tatsächliche Zeit zwischen 5 und 6 Minuten schwanken.

#### Funktionen

Das Betreten eines Raumes und die damit einhergehende Teilnahmen an dem Spiel in diesem Raum ist äußerst einfach. Mit einem Klick auf das Symbol des Raumes kann man sich in diesen begeben. Sollte man den letzten freien Platz füllen, beginnt das Spiel sofort. Ansonsten kann man den Raum auch wieder verlassen, um einem anderen Spiel beizutreten.

Um einen neuen Raum zu erstellen, muss man auf das Plus-Symbol klicken. Danach erscheint ein Menü, in dem man auswählen kann, ob zwei oder viel Spieler an dem Spiel teilnehmen und wie viele der Gegner vom Computer gesteuert sein sollen. Außerdem kann bereits beim Erstellen des Raumes festgelegt werden, an welcher Position die Computergegner stehen sollen. Um unnötiges Rechnen auf der Serverseite zu verhindern, kann kein Raum nur mit Computergegnern gefüllt werden, es muss mindestens ein menschlicher Spieler teilnehmen.

Aus technischer Sicht ist jeder Raum eine Sammlung von Eigenschaften und Funktionen, um den Raum richtig anzeigen zu können. Eine der wichtigsten Eigenschaften ist hier der Name des Raumes, welcher unbedingt eindeutig sein muss. Dieser wird nämlich dazu verwendet, um die Spieler einem Raum zuzuordnen. Sollte dieser also nicht eindeutig sein, bekommen die Spieler von zwei Räumen nur die Informationen für einen, was dazu führt, dass die Hälfte der Spieler nicht spielen, sondern nur zusehen kann.

### Spielablauf

Sobald das Spiel begonnen hat, sind alle Teilnehmer der Reihe nach am Zug. Der aktuelle Spieler ist dabei mit einer stärkeren Farbe gekennzeichnet. Ein Zug besteht aus zwei Teilen. Zuerst muss man die Spielfigur auf ein leeres, angrenzendes Feld bewegen. Das kann waagrecht, senkrecht und diagonal sein. Anschließend muss ein beliebiges, nicht besetztes Feld entfernt werden. Zur Auswahl stehen in dieser Phase alle leeren Felder auf dem gesamten Spielfeld. Das Entfernen eines Feldes bedeutet den Abschluss des eigenen Zuges und der nächste Spieler ist an der Reihe.

Sollte ein Spieler den eigenen Zug nicht mehr beginnen können, ist er ausgeschieden. Dieser Fall tritt ein, wenn es keine leeren, angrenzenden Felder mehr gibt. Sobald nur noch ein Spieler sich bewegen kann, oder wenn alle Spieler außer einem ausgeschieden sind, hat dieser letzte Spieler gewonnen.

### Client-Server-Kommunikation

Um ein Online-Multiplayer-Spiel zu ermöglichen, müssen die Änderungen, die auf einem der Clients gemacht werden, zu allen anderen betroffenen Clients gelangen. Dies geschieht über einen zentralen Server.

#### Aufgaben des Servers

Die Hauptaufgabe des Servers ist es, immer einen aktuellen Stand zu den laufenden Spielen zu haben. Er muss wissen, welche Clients in welchem Spiel sind, wie das Spielfeld gerade aussieht und wer gerade an der Reihe ist. Des Weiteren ist er für das Aufräumen zuständig. Das heißt, er sorgt dafür, dass Räume, die bereits beendet sind oder die aus anderen Gründen geschlossen werden, richtig gelöscht und alle Clients, die sich in diesem Raum befanden, richtig aus dem Raum entfernt werden. Zusätzlich laufen auch die Computergegner auf dem Server.

Wenn der Server gerade nichts zu tun hat, wartet er auf ein Event von einem Client. Je nach dem, welches Event ankommt, führt er unterschiedliche Operationen aus und schickt ein passendes Event an alle beteiligten Clients zurück.

#### Aufgaben des Clients

Der Client ist hauptsächlich für das korrekte Anzeigen des Spielzustandes und das Weitergeben von Benutzerinteraktionen verantwortlich. Er nimmt alle Informationen, die er vom Server bekommt, zeigt sie an und wartet dann auf neue Informationen. Diese können entweder wieder vom Server kommen oder eine Interaktion des Spielers sein. Nach jeder Änderung im Zustand des Spielfeldes informiert der Client den Server, so dass der aktuelle Stand zentral gespeichert ist.

#### Beispiel

Client A und Client B spielen gegeneinander. Das Spiel hat gerade erst angefangen, Client A ist an der Reihe. Der Client stellt das noch vollständige Spielfeld dar. Nach einer Weile passiert etwas, das für den Server relevant ist: ein Klick vom Benutzer auf eines der Felder. Nachdem Client A festgestellt hat, dass das Feld einem gültigen Zug entspricht, schickt er an den Server: „Ich will ziehen“, und sagt ihm die gewünschte Position. Zur Sicherheit überprüft auch der Server, ob die Position gültig ist. Danach trägt er die Änderung des Spielstandes bei sich ein und schickt an beide Clients: „Jemand ist gezogen“. Zusätzlich dazu gibt er das neue Aussehen des Spielfeldes mit, sowie die Information, dass Client A an der Reihe ist.

Client A und B stellen das erhaltene Spielfeld dar, und weil Client A sieht, dass soeben ein Zug gemacht wurde und er an der Reihe ist, erlaubt er die Auswahl eines zu löschenden Feldes vom Benutzer. Sobald dieser eine gültige Wahl getroffen hat, schickt er an den Server: „Ich will ein Feld löschen“, gemeinsam mit der Position des Feldes. Nach nochmaliger Überprüfung wird das Feld in den Spielinformationen gelöscht. Bevor jetzt ein Event zurückgeschickt wird, muss der Server überprüfen, ob nicht vielleicht schon jemand gewonnen hat. Wäre das der Fall, würde er ein passendes Event ausschicken. Da das Spiel aber gerade erst begonnen hat, ist das nicht der Fall und er schickt an beide Clients: „Ein Feld wurde gelöscht“. Wie vorhin gibt es die zusätzliche Information über das neue Aussehen des Spielfeldes und dass Client B an der Reihe ist.

Wieder stellen beide Clients das neue Spielfeld dar. Client A sieht, dass er nicht an der Reihe ist, und wartet daher auf ein neues Event vom Server. Andererseits merkt Client B, dass er an der Reihe ist. Da das letzte erhaltene Event die Bestätigung einer Löschung war, muss er seinem Benutzer eine Bewegung ermöglichen. Sobald dieser Input vom Benutzer erfolgt ist, fängt das ganze wieder vorne an.

### Computergegner

Damit das Spiel auch gespielt werden kann, wenn außer einem niemand spielt, wurden Computergegner eingebaut. Damit der Server jedoch nicht unnötig am Rechnen ist, ist es nicht möglich, einen Raum mit nur computergesteuerten Gegnern zu erstellen.

Die Computergegner funktionieren so, dass der Server jedes Mal, nach dem ein Feld entfernt wurde, nachschaut, ob der nächste Spieler ein Mensch – also ein Client – ist. Ist das nicht der Fall, startet er sofort nach dem Senden der nötigen Events eine Funktion, die berechnet, wohin sich der computergesteuerte Gegner bewegen soll. Nachdem die Clients über den Zug des Computergegners informiert wurden, wird eine weitere Funktion aufgerufen, die feststellt, welches Feld entfernt wird. Danach kommt wieder die Überprüfung des nächsten Spielers.

Für die erste Version des Spiels wurde ein relativ einfach gehaltener Computergegner entwickelt. Daher ist er auch nicht allzu schwer zu besiegen. Um festzustellen, wohin er gehen soll, schaut sich der Computer alle Felder an, die rund um ihn sind und merkt sich alle, die die größte Anzahl an umgebenden Feldern haben. Aus diesen wählt er dann zufällig eines aus. Das Feld, welches entfernt werden soll, wird sehr ähnlich ausgewählt. Der Computer schaut, welcher der Spieler noch die meisten Felder rund um sich hat. Falls es mehrere Spieler mit gleich vielen Feldern gibt, wird zufällig einer ausgewählt. Danach tut der Computer so, als ob er an der Stelle des gewählten Spielers steht und gerade gehen will. Das Feld, das diesmal ausgewählt wird, wird dann gelöscht. Er versucht also, von dem Spieler mit dem größten Vorteil das beste Feld wegzunehmen.

Bessere Computergegner würden komplexere Algorithmen verwenden, um mögliche Züge und deren Folgen zu analysieren. Von dem günstigsten Ende wird dann der erste Zug ausgeführt. Dabei können nach einiger Zeit alle Enden, die vom jetzigen Stand möglich sind, herausgefunden werden. Je größer und vollständiger das Spielfeld ist, desto länger dauert dieser Prozess allerdings. Daher wird bei solchen Methoden normalerweise vorgegeben, wie viele Züge der Computer vorausberechnen soll oder wie lange er maximal brauchen darf. Je größer diese einschränkende Zahl ist, desto besser und schwieriger zu besiegen ist der Computer.

Literaturverzeichnis

[1] <https://www.hackerearth.com/problem/multiplayer/isola/> 30.07.2019 14:30

[2] <https://www.projektmanagementhandbuch.de/handbuch/projektinitiierung/zieldefinition/> 30.07.2019 15:00

[3] [https://whatis.techtarget.com/definition/Nodejs 01.08.2019 7:00](https://whatis.techtarget.com/definition/Nodejs%2001.08.2019%207:00)

[4] [https://github.com/socketio/socket.io 01.08.2019](https://github.com/socketio/socket.io%2001.08.2019) 7:30

[5] <https://github.com/photonstorm/phaser> 01.08.2019 8:30

[6] <https://whatis.techtarget.com/definition/WebGL> 01.08.2019 8:30

[7] <https://gamedevelopment.tutsplus.com/tutorials/creating-isometric-worlds-primer-for-game-developers-updated--cms-28392> 01.08.2019 9:30

Abbildungsverzeichnis

[Abbildung ‎1.1: Spielfeld vor Beginn des Spiels 1](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566137)

[Abbildung ‎4.1: Umwandlung einer Ebene in isometrische Darstellung 6](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566138)

[Abbildung ‎4.2: Weltraum-Thema 7](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566139)

[Abbildung ‎4.3: Wasser-Thema 7](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566140)

[Abbildung ‎4.4: Lava-Thema 7](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566141)

[Abbildung ‎4.7: Erster Entwurf des Spiels 8](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566142)

[Abbildung ‎4.8: Erster Entwurf der Startseite 9](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566143)

[Abbildung ‎4.12: Ein Vier-Spieler-Raum mit zwei freien Plätzen und Computergegnern an zweiter und vierter Stelle 10](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566144)

[Abbildung ‎4.11: Ein voller Zwei-Spieler-Raum 10](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566145)

[Abbildung ‎4.14: Dieser Spieler war schon an der Reihe 11](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566146)

[Abbildung ‎4.13: Dieser Spieler ist an der Reihe 11](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566147)

[Abbildung ‎4.17: Ein Vier-Spieler-Raum mit dem 2. & 3. Spieler als Computergegner 12](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566148)

[Abbildung ‎4.16: Ein Zwei-Spieler-Raum mit nur menschlichen Spielern 12](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566149)

[Abbildung ‎4.15: Dieser Knopf erstellt einen neuen Raum 12](file:///C:\Users\Johanna\Desktop\Schule\DA\Iso\Dokumente\isola_diplomarbeitsbuch.docx#_Toc15566150)

Stichwortverzeichnis

Canvas 5

Client 4, 5, 14, 15

Computergegner iii, 1, 2, 3, 9, 12, 13, 14, 15, 16, 17

CSS 4

Frameworks 4

HTML 4

Isola iii, iv, 1, 2

isometrische Darstellung 6, 17

JavaScript 4, 5

Mobile-First 8

Node.js 4

Phaser 3 5

Raum 2, 5, 10, 11, 12, 13, 14, 15, 17

Server 4, 5, 13, 14, 15

socket.io 4, 5, 17

Spiel iii, 1, 2, 3, 5, 8, 9, 10, 11, 13, 14, 15

Spielfiguren 1, 7, 8

Thema 7, 8, 17

Tutorial 3, 10

WebGL 5, 17

Website 4, 7, 8, 9, 10

Ziele 2, 3