Dokumentation

Inhaltsverzeichnis

Allgemeines

Dieses Pong-Spiel entstand im Zeitraum vom 05.11. bis 11.11.2012 im Rahmen des Conding Contest (http://www.coding-contest.de/).

Es ist veröffentlicht unter https://github.com/falkmueller/leemiapong
und läuft unter der Apache Lizense V2.

Das Spiel kann direkt gespielt werden auf http://leemia.de/pong/

Ausgabe

Umzusetzen war ein Server für ein Pong-Spiel, welcher gegebene API-Funktionen unterstützt oder eine KI, welche selbstständig entscheidet, wie es das Paddel bewegt, um zu gewinnen.

Die Anforderungen waren dabei wie folgt definiert:

- GET /game/:key/status <- Liefert den kompletten Status des Spielfeldes aus, also Ballposition, Ballbewegung und Spielstand.
- GET /game/:key/config <- Grundkonfiguration des Spiels (Größe des Spielfelds, des Balls, der Paddel).
- POST /game/:key/start <- Liefert *ok* zurück, wenn das Spiel gestartet wurde. Dies wird automatisch passieren, sobald sich zwei Spieler angemeldet haben.
- PUT /game/:key/player/:playername/ <- Meldet einen Spieler am Spiel an. Gibt eine geheime Zeichenkette zurück, die bei jeder Bewegung des Paddels von den Clients mitzuliefern ist.
- POST /game/:key/player/:playername/:secret/up/ <- Bewegt das eigene Paddel eine Einheit nach oben, das geht nur 10x pro Sekunde.
- POST /game/:key/player/:playername/:secret/down/ <- Bewegt das eigene Paddel eine Einheit nach unten, das geht nur 10x pro Sekunde.

Umgesetzt wurde ein Client, ein Server und eine KI.

1. Der Server

Der Server nimmt per REST-API Anfragen entgegen, wertet diese aus und gibt ggf. JSON-Objekte an den Client zurück.

1.1. API-Funktionen

/game/:key/status

Liefert ein JSON-Objekt, welches alle für den Client relevanten Informationen beinhaltet (u.a. Paddel-Positionen, Ballposition, Ballrichtung, Namen der Spieler, Spielstand, …).

/game/:key/config

Liefert ein JSON-Objekt, welchee die Spiel-Konfiguration enthält.
(u.a. Feldhöhe und -breite, Paddel-Größe, Ballradius, …).

/game/:key/start

Über den Aufruf wird das Spiel gestartet, bzw. nach Beendigung des Spieles neu gestartet. Der Aufruf liefert „OK“ zurück.

/game/:key/player/:playername/

Meldet einen Spieler am Spiel an. Gibt im Erfolgsfall einen Secret Key zurück, welchen der Spieler benötigt, um sein Paddel zu steuern

/game/:key/player/:playername/:secret/up/

Über diesen Befehl kann der Spieler sein Paddel um eine Einheit nach oben bewegen. Als Antwort kann dabei ein Fehler geliefert werden, wenn die Angaben nicht zum Spiel passen (z.B. falscher Secret Key) oder der Spieler versucht mehr als 10 mal pro Sekunde die Paddel-Position zu verändern.

/game/:key/player/:playername/:secret/down/

Über diesen Befehl kann der Spieler sein Paddel um eine Einheit nach unten bewegen. Als Antwort kann dabei ein Fehler geliefert werden, wenn die Angaben nicht zum Spiel passen (z.B. falscher Secret Key) oder der Spieler versucht mehr als 10 mal pro Sekunde die Paddel-Position zu verändern.

1.2. Umsetzung

Um das Spiel möglichst unabhängig zu halten und eine Installation zu vermeiden, habe ich mich gegen die Verwendung einer Datenbank entschieden. Zu Beginn wurde auf dem Server für jedes laufende Spiel eine Text-Datei angelegt, welche den serialisierten Spielstatus beinhaltet. Dies erwies sich jedoch im späteren Verlauf als unvorteilhaft, weshalb ich dazu überging, dass jedes Spiel eine eigene Session bekommt und sich somit 2 Spieler eine gemeinsame Session teilen.

Da kein Cron-Job oder ähnliches dafür sorgt, dass in regelmäßigen Zeitintervallen (z.B. 0,1 Sekunden) die Routine läuft, welche die neue Ballposition bestimmt, den Ball bei Bandenkollision seine horizontale Richtung wechseln lässt, bei Paddel-Kollision die horizontale Richtung wechselt und bei horizontalen Verlassen des Spielfelder den Gegner einen Punkt gibt, muss dies beim regelmäßigen abruft des Status vom Client erledigt werden.

Dabei hatte des lesen und schreiben von Text-Dateien als Status-Speicher auf dem Server zu Konflikten geführt. Bei der Verwendung einer gemeinsamen Session für je ein Spiel ist es so, dass die Session gesperrt ist, wenn ein Spieler sie öffnet. Als gemeinsame Session Id wird dabei der Spiel-Key verwendet. Die Serveranfrage des anderen Spielers muss warten, bis sein Gegner die Session wieder freigegeben hat. Dies ist der Fall, wenn der Spieler die Serverantwort erhält oder wenn auf dem Server via „`session_write_close();`“ die Session geschlossen wird. Sobald die Session geschlossen ist, kann die Wartende Server-Anfrage sie öffnen. Dadurch, dass immer nur die anfrage eines Spielers am gemeinsamen Spiel bearbeitet wird und der andere Spieler warten muss, werden Kollisionen umgangen.

1.3. der Quellcode

Bis auf die index-Datei ist der Dateiname immer auch der Name der Klasse, welche sie beinhaltet.

Server/Index.php

> Nimmt die Anfrage entgegen. Beim Aufruf von Beispielsweise „/game/:key/status“ wird dies umgeleitet auf „server/index.php?q=:key/status“. Anschließend wird eine Response-Object erzeugt und zum Schluss dessen send-Funktion aufgerufen.

server/library/response.php

> Sie enthält die Response-Klasse. Diese erkennt Anhang der URL die gewünschte API-Funktion, führt diese aus und gibt deren Antwort zurück.

Server/library/configObj.php

> Diese Klasse beinhaltet die Konfigurationsdaten des Spiels.

Server/library/gameObj.php

> Ein Objekt dieser Klasse wird in die Spiel-Session gelegt. Es beinhaltet immer die aktuellen Spiel-Daten (Ballposition, etc.). Im Konstruktor werden die Standardwerte gesetzt und eine Get-Funktion gibt eine Kopie des Objektes zurück, jedoch ohne die Secret.Keys der Spieler, damit der Rückgabewert der Get-Funktion direkt an den Client gesendet werden kann.

Server/library/statusEnum.php

> Enthält die Aufzählung aller möglichen Spielstati (login, ready, started, finished)

Server/library/sessionManager.php

> Der Sessionmanager öffnet die gemeinsame Spielsession und liest das GameObj und das ConfigObj aus der Session, oder erzeugt neue, wenn keine in der Session vorhanden sind.

Server/library/player.php

> Beinhaltet die API-Funktionen zum Login und zur Paddel-Bewegung. Da die Funktionen kaum Verweise untereinander beinhalten, sind die Funktionen dieser Klasse statisch, damit sie ohne Erzeugung eines Objektes der Klasse verwendet werden können.

Server/library/game.php

> Beinhaltet die API-Funktionen der Abfrage des Spielstatus und der Spielstarts. Beim Abfragen des Status wird die Funktion „run“ ausgeführt, welche beispielsweise überprüft, ob die Wartezeit vor dem Spielbeginn vergangen ist und die Zähler der Paddel-Bewegungen jede Sekunde zurücksetzt. Die wichtigste Aufgabe der Funktion ist jedoch das Ausrechnen, wie viele Ball-Bewegungen anhand der Zeitdifferenz der Anfragen getätigt werden müssen und dies mit Aufruf der Funktion „Step“ in einer Schleife zu erledigen.
> Die Funktion „Step“ arbeitet jedes mal folgende Reihe ab:

1. Wenn Ball auf Spielfeld ganz rechts oder links:
   - Prüfe ob Paddel-Berührung
     - Wenn ja, dann ändere horizontale Ball-Richtung
     - Wenn nein, dann bekommt der Gegner einen Punkt und es wird überprüft, ob ein Spieler gewonnen hat.
2. Wenn Ball am oberen oder unteren Spielfeldrand ist, dann ändere die vertikale Ball-Richtung
3. Beschleunigt den Ball
4. Bewegt den Ball auf die nächste Position

2. Die KI

Dem Spieler soll es möglich gemacht werden, gegen eine künstliche Intelligenz zu Spielen. Anfänglich sollte die KI über einen Cron-Job oder eine regelmäßige Anfrage zum Handeln bewegt werden. Da dies aber nicht bei jedem System gewährleistet ist, habe ich mich entschlossen die KI nicht separat zu bauen, sondern an das Spiel zu koppeln. So wird, wenn bei einem Spieler der Status des Spieles abgefragt wird, die KI angestoßen, damit diese ihre Berechnungen anstellen und eventuelle Paddel-Bewegungen durchführen kann. So wird auch nicht die Serverlast dadurch erhöht, dass ein Cron-Job regelmäßig Anfragen schicken muss.

2.1. API-Funktionen

/game/:key/robot

Meldet am Spiel einen Robot, also die KI, an. An einem Spiel kann nur eine KI angemeldet werden, der Gegenspieler muss also ein Mensch sein, oder eine andere KI auf einem anderen Server.

2.3. der Quellcode

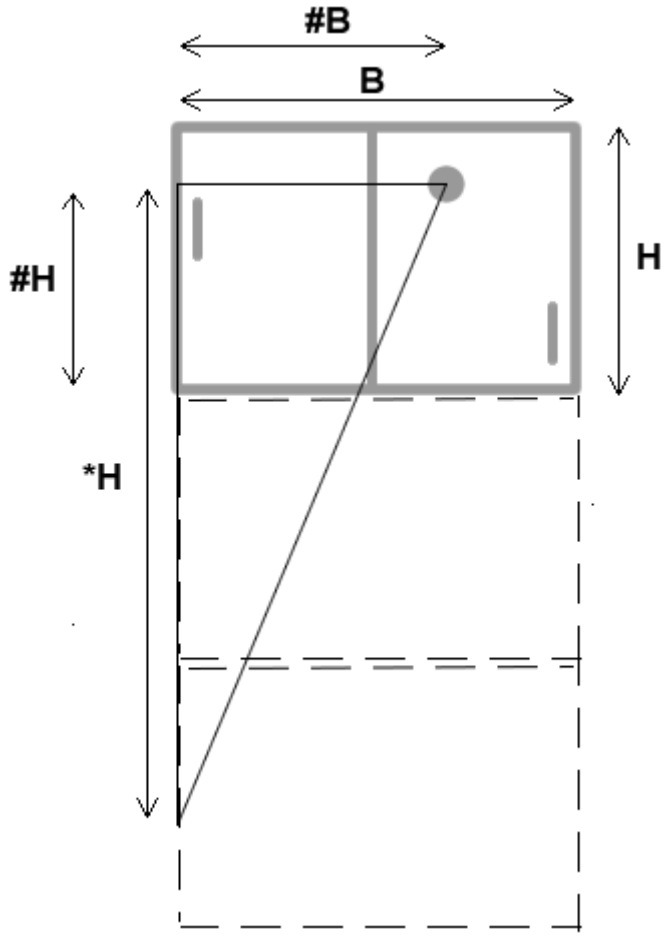
Server/library/robotObj.php

Ein Objekt dieser Klasse wird in die Session des Spiels abgelegt. Es beinhaltet unter Anderem den Secret-Key der KI, damit diese Ihr Paddel bewegen kann.

server/library/robot.php

Bei der Erzeugung wird die KI am Spiel angemeldet und in der Session das robotObj abgelegt, falls die KI noch nicht am Spiel teilnimmt.
Die Status-Anfrage des anderen Spielers sorgt dafür, dass die Klasse robot aufgerufen wird und die Funktion „run“ ausgeführt wird. Diese berechnet ihre neue Paddel-Position und ruft die „up“ und „down“ Funktionen auf um die Paddel-Position des KI-Paddels zu verändern.

2.4. Berechnungen



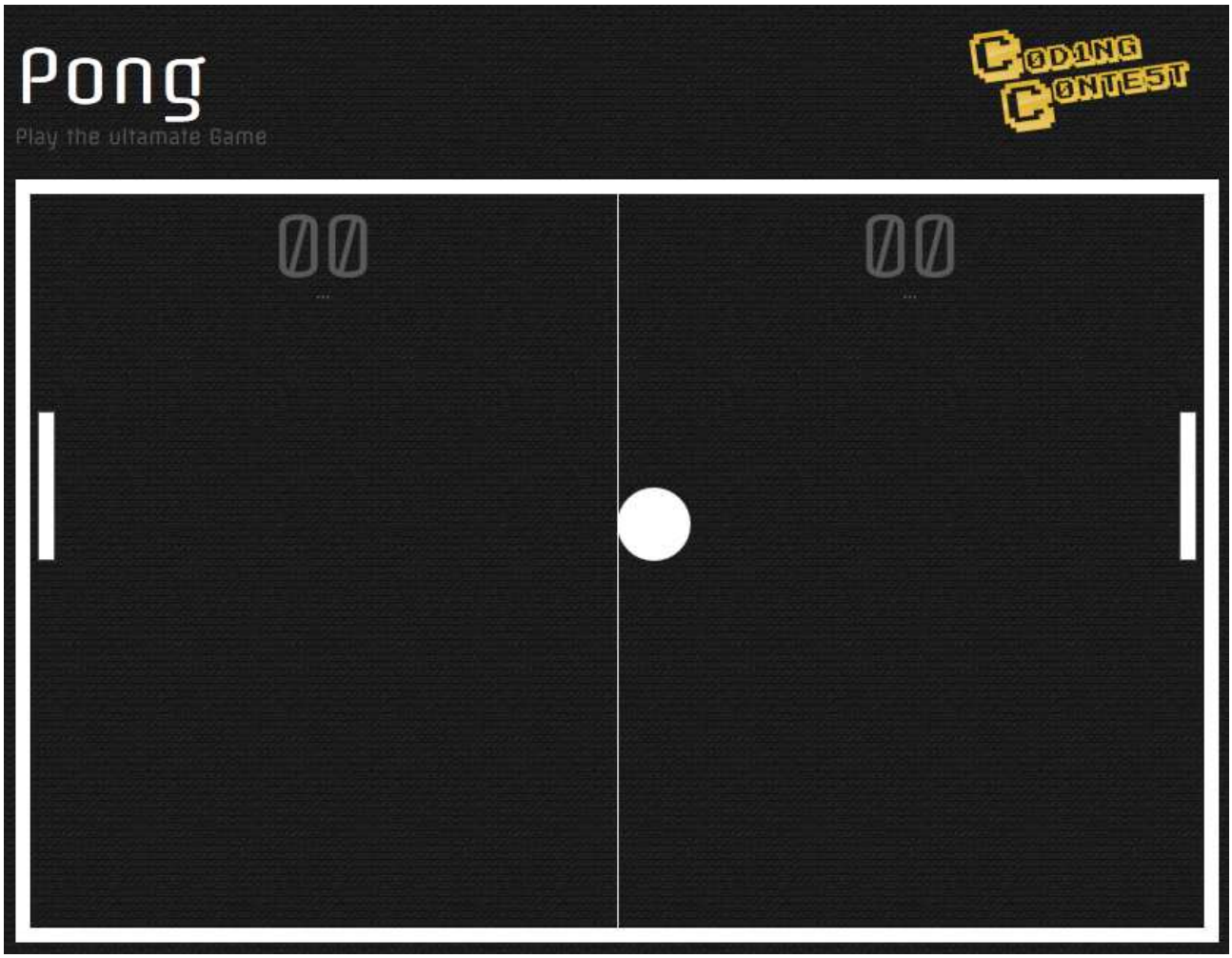
Die Berechnung der KI ist im Wesentlichen der mathematische Strahlensatz. Anhand der X und Y Werte von BallDelta (Ball-Verschiebungs-Vektor) und der Ballposition auf dem Spielfeld wird berechnet, an welcher Position das Paddel positioniert werden muss, damit es den Ball trifft. Zu beachten ist dabei, dass, je nach dem, wie oft der Ball an den Banden reflektiert wird, die berechnete Position auch gespiegelt sein kann (berechnete Paddelhöhe oder Spielfeldhöhe minus berechnete Paddelhöhe).

Durch die Berechnung kann die KI schon wenn der Spieler den Ball zurückwirft zielstrebig auf die Position gehen, an der das Paddel den Ball treffen wird.

Da jedoch der Ball bei diesem von mir umgesetzten Server nicht immer genau an den vertikalen Koordinaten „0“ und „Spielfeldhöhe“ reflektiert wird, sondern es durch die Größe des Richtungsvektors und durch die schrittweise Verschiebung des Balles um diesen, auch dazu kommen kann, dass der Ball ein paar Pixel später reflektiert wird, kann die KI das Paddel nicht genau positionieren und muss bei jeder Reflexion nachjustieren.

Dadurch kann sie manchmal das Paddel nicht mehr rechtzeitig auf die genaue Position bringen und der Gegner bekommt einen Punkt. Dieses Problem habe ich jedoch beabsichtigt nicht weiter betrachtet, da das Spiel für den Menschen sonst langweilig wäre, wenn er gegen die KI nie punkten würde.

3. Der Client



Um den Server zu testen und etwas anschauliches zu schaffen habe ich einen Client für den Server entwickelt. Die API wird immer per „/game“ angesprochen. Ohne dies wird nicht die API angesprochen und der Server sendet den Spiel-Client an den Browser.

3.1. Funktionen und Besonderheiten

AJAX Crawled URLs

Zu Beginn bekommt der Nutzer nur eine Seite geliefert, welche einen Header und einen Footer beinhaltet, aber nicht den wesentlichen Hauptinhalt. Dieser wird per Ajax nachgeladen. Links auf der Seite verweisen lediglich auf Anker (Zum Beispiel „#info“). Beim Klick auf den Link, wird der Anker-Tag in der U>RL gesetzt und per Ajax der gewünschte Inhalt geladen. Dies hat den Vorteil, das nur das geladen wird, was sich an den Seiten auch unterscheidet. Durch den Anker-Tag kann nun der Nutzer auch die Seite in seine Favoriten im Browser eintragen und bekommt beim Aufruf den zum Anker passenden Inhalt geladen. Des Weiteren verrät ein Meta-Tag auf der Seite Suchmaschinen wie Google, dass es sich um eine mit Ajax nachladende Seite handelt. Da die Crawler der Suchmaschinen kein Javascript interpretieren können, werden die Hauptinhalte nicht in einer Suchmaschine erfasst. Durch den Meta-Tag verfolgt die Suchmaschine nun aber nicht den Link „#info“, sondern „?_escaped_fragment_=info“. Der Server gibt nun anhand des GET-Parameters gleich die komplett gerenderte Seite zurück und der Crawler kann den Inhalt in die Suchmaschine aufnehmen.

Animation und Sound

Damit die Seite optisch und akustisch ansprechend ist, wird der Inhalt beim Klick auf einen Link nicht einfach durch den neuen Inhalt ersetzt, sondern der alte Seiteninhalt wird nach oben herausgefahren und der neue herein. Dies geschieht mittels JQuery Animation. Zudem wird in HTML5 fähigen Browsern ein Klang zum Effekt angespielt. Auch während des Spiels ertönt ein Gong, wenn ein Spieler einen Punkt macht. Etwaige Meldungen und Fehlermeldungen werden mittels JQuery BlockUI eingeblendet, statt eines gewöhnlichen alerts.

CSS3

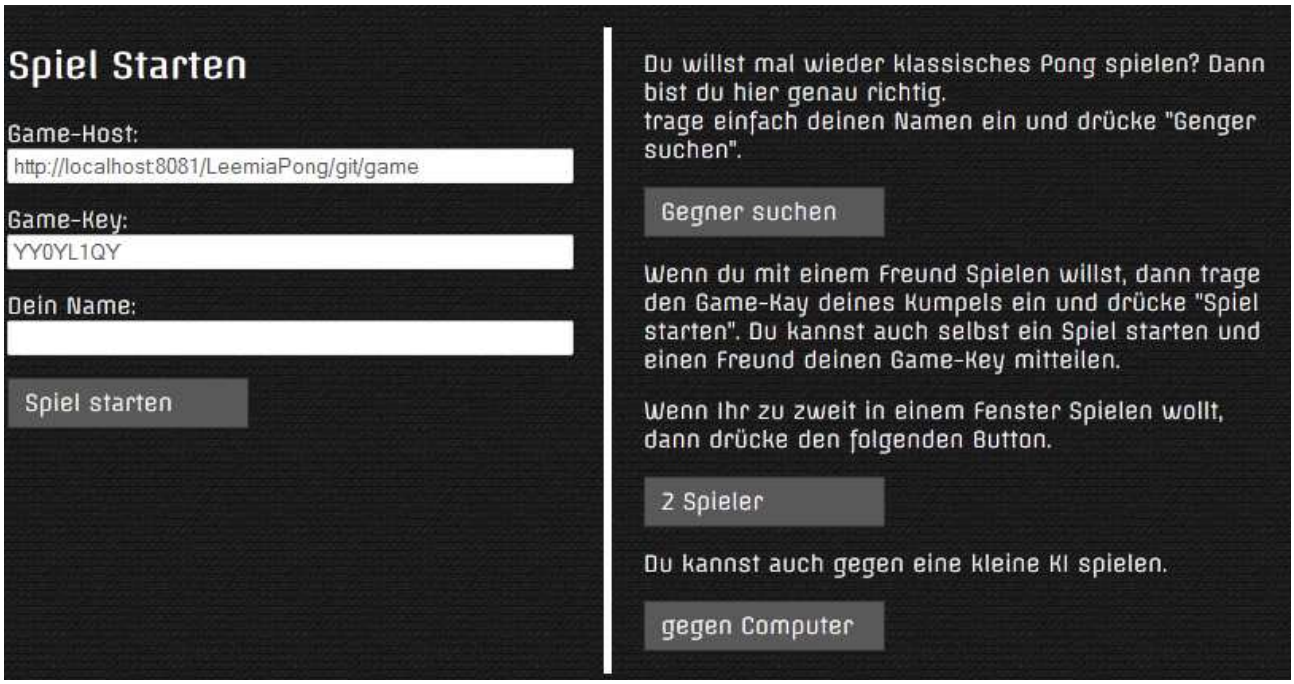
Das Spiel ist für CSS3 fähige Browser optimiert. Zum Einen wird eine Google-Font eingebunden. Zum Anderen ist beispielsweise der Spielball bei älteren Browsern eventuell viereckig.

Long Polling

Da nicht jeder Server neuere Technologien wie Websockets unterstützt, werden zum Beispiel Verbindungen mit dem Server offengehalten, bis dieser eine Antwort sendet.

3.2 Das Spiel im Browser

3.2.1. Startseite



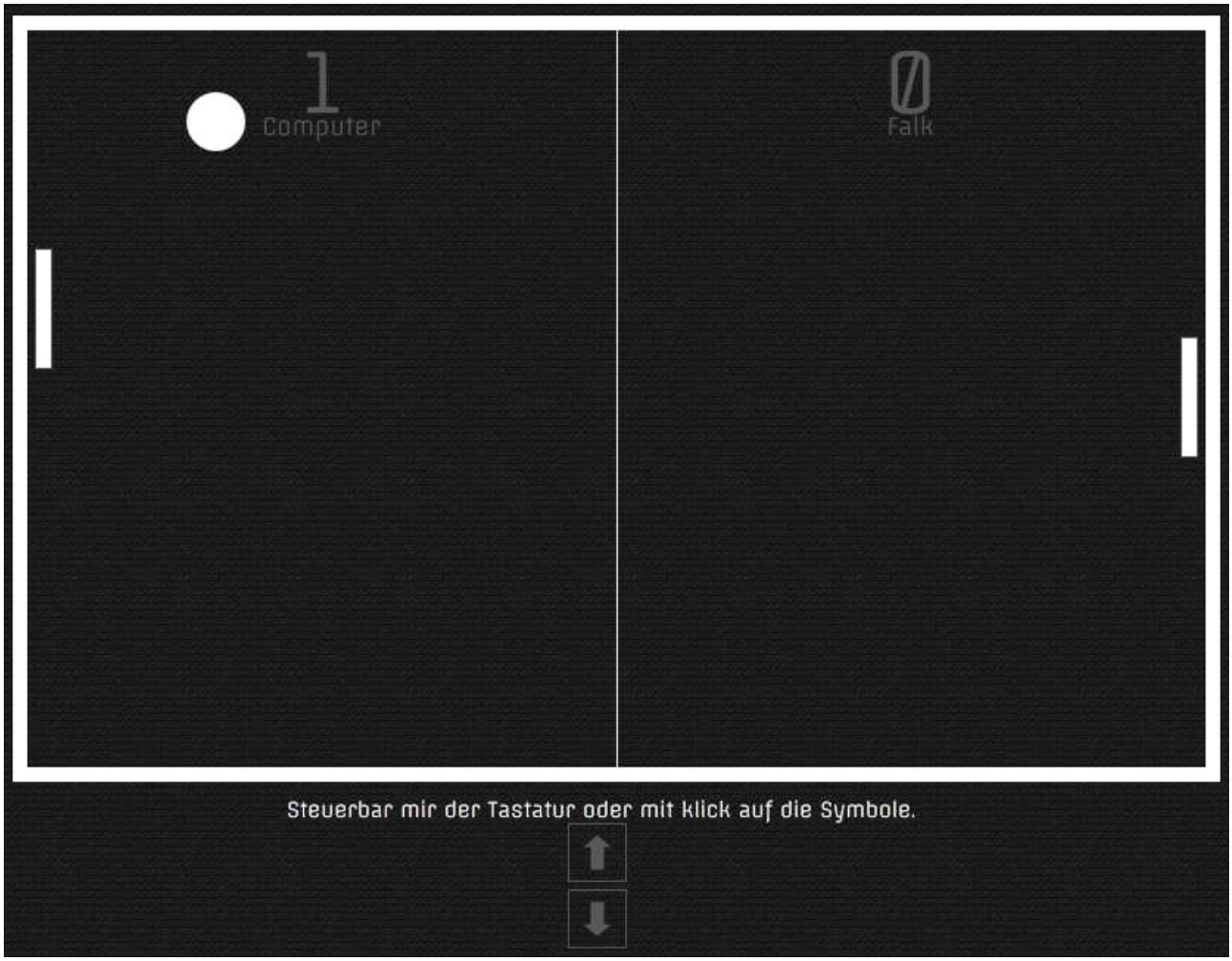
Zu Beginn muss man den Game-Host angeben, welcher für das Spiel verwendet werden soll. Hier steht standardmäßig die URL des Servers, zu dem mitgelieferten Server.

In das Feld „Dein Name“ muss der Spieler seinen Namen eingeben.

Der Weitere Verlauf hängt nun davon ab, wie bzw. gegen wen man spielen möchte:

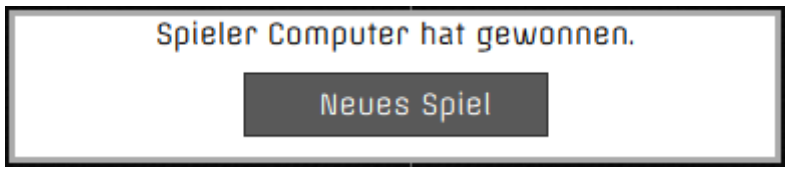
- **Spiel gegen die KI:**
  Man muss nur den Button „gegen Computer“ drücken und das Spiel beginnt.
- **Spiel gegen einen unbekannten Spieler:**
  Zunächst drückt man den Button „Gegner suchen“. Nun wird das Feld „Game-Key“ ausgeblendet, da der Webservice, welcher den Gegner sucht, den gemeinsamen Key mit dem Gegner zurückgibt. Per Long Polling wird nun so lange eine Verbindung zum Server offengehalten, bis ein Gegner gefunden ist.
- **Spiel gegen einen Bekannten**:
  Wenn man zu einen Spiel eingeladen wurde, gibt man einfach den Host und den Game-Key an, welchen man mitgeteilt bekommen hat und klickt auf „Spiel starten“ oder man teilt einem Freund seinen Host und Game-Key mit. Nachdem man auf „Spiel starten“ geklickt hat, wird man zum Spielfeld geleitet. Das Spiel beginnt jedoch erst, wenn der Gegner sich angemeldet hat.
- **Zusammen an einem PC spielen:**
  Klickt man den Button „2 Spieler“, wird ein weiteres Eingabefeld „Name Spieler Zwei“ eingeblendet. In dieses muss der zweite Spieler seinen Namen eintragen. Nach Klick auf „Spiel starten“ wird man zum Spielfeld geleitet. Der Linke Spieler kann die Symbole links unter dem Spielfeld zum steuern seines Paddels verwenden oder die Tasten „W“ und „S“ auf der Tastatur. Der rechte Spieler kann die Symbole rechts unter dem Spielfeld verwenden oder die Auf- und Ab-Pfeil-Tasten der Tastatur.

3.2.2. Das Spielfeld



Das Spiel kann jeweils über die Icons oder die Tastatur gesteuert werden. Zu beachten ist dabei, dass die Tastenbefehle nur erkannt werden, wenn sich der Mauszeiger innerhalb des Browsers befindet, bzw. der Browser im Vordergrund ist.

Sobald ein Spieler 10 Punkte hat, ist das Spiel beendet.



Es erscheint eine Nachricht und das Spiel kann über den Button neu gestartet werden.

Bitte beachten Sie, dass Sie nicht die Seite aktualisieren oder per „F5“ neu laden. Dabei geht im Hintergrund Ihr geheime Schlüssel verloren. Sie können anschließend nur noch zuschauen, aber nicht mehr Ihr Paddel steuern. In diesem Falle gehen Sie zurück auf die Startseite und starten ein neues Spiel.