l

Inhaltsverzeichnis

**Inhalt**

[**Zielbestimmung**](#_gjdgxs) **2**

[**Produkteinsatz**](#_30j0zll) **3**

[**Produktumgebung**](#_1fob9te) **3**

[Software](#_3znysh7) 3

[Hardware](#_2et92p0) 3

[Windows](#_l8e6cd38t6mm) 3

[Mac](#_4mqipdkn42k3) 3

[Linux](#_6rljvr9wczyk) 3

[**Produktfunktionen**](#_6wdrc9f48rfh) **3**

[Fake Login](#_1d5digsf8x8f) 3

[Spiel](#_jculpkilz68v) 3

[Session Join](#_4pfe55dhdvmr) 4

[**Produktdaten**](#_slgqlvq2p6f2) **4**

[**Benutzeroberfläche**](#_rvjee78ib0xu) **4**

[**Globale Testszenarien und Testfälle**](#_qhm3aecvv92i) **5**

[**Entwicklungsumgebung**](#_ys9a29la18vl) **6**

[**PHP Sprach Features**](#_opb9m6i090gt) **6**

[Variablen](#_ejcdrc6ocez8) 6

[Strings](#_lstiskp8mdnp) 6

[Arrays](#_gvj10o1zob2b) 7

[Überladen](#_qvx6xuu0iqc9) 7

[Überschreiben](#_4xp9gq70vefb) 7

[Interfaces](#_1xc61h7yi9jh) 8

# Zielbestimmung

Das Projekt «Schiffe versenken» ist die Verwirklichung eines Spieles, welches über einen Browser verfügbar ist. Dabei soll es möglichst einfach sein, schnell ein Spiel zu starten und dieses dann direkt zu spielen.

Bei einem Spiel treten zwei Spieler gegeneinander an. Dabei muss man zuerst auf einem definierten Spielfeld mit 16 x 16 Feldern verschiedene, aneinanderhängende Blöcke mit einer variablen Länge zwischen 1 und 5 Blöcken verteilen. Natürlich haben beide Spieler dieselben Blöcke zum verteilen. Anschliessend bekommt man das Feld des Gegners dargestellt. Jedoch werden hier die Schiffe des Gegners nicht angezeigt. Abwechselnd kann man die verschiedenen Felder «aufdecken». Falls sich auf dem aufgedeckten Feld ein Schiff des Gegners befinden sollte, wird dies entsprechend markiert. Andernfalls war das Feld ein «Treffer ins Leere» und wird ebenfalls markiert.

Das Spiel ist dann gewonnen, wenn man alle Schiffe des Gegners aufgedeckt hat. Der Highscore wird an der verbrauchten Zeit gemessen.

Damit man mit seinem Kollegen spielen kann, wird eine URL dargestellt, welche man seinem Kollegen schicken kann, dass man mit Ihm zusammenspielen kann.

Optional soll ein Login nicht nur zur Dekoration sein. Die Highscores sollten dann über den mit dem Login verknüpften Account abgespeichert werden.

Optional könnte eine Globale Highscore Liste erstellt werden

# Produkteinsatz

Das Projekt dient der reinen Unterhaltung und soll keinen weiteren Mehrzweck haben. Zielgruppe sind 12- bis etwa 50-jährige Menschen. Diese sind im Umgang mit Browsern geübt und kommen damit einfach zurecht.

# Produktumgebung

## Software

Das Programm wird mit PHP auf Windows entwickelt. Auf die Verwendung eines Docker-Containers wird aus praktischen Gründen verzichtet, da das Wissen dazu nicht in der Gruppe vorhanden ist. Getestet wird das Spiel auf Windows und ggf. Linux. Jedoch sollte die Kompatibilität auf allen Betriebssystemen, welche von Google Chrome unterstützt werden, gegeben sein.

## Hardware

Um das Spiel spielen zu können, muss mindestens Google Chrome auf dem Computer installiert sein. Eine Unterstützung von mobilen Geräten ist nicht vorgesehen. Die Voraussetzungen von Google Chrome sind auf dessen Produktseite zu finden:

<https://support.google.com/chrome/a/answer/7100626?hl=de>

#### Windows

Zur Verwendung von Chrome unter Windows benötigen Sie Folgendes:

* Windows 7, Windows 8, Windows 8.1, Windows 10 oder höher
* Einen SSE2-fähigen Intel Pentium 4-Prozessor oder höher

**Hinweis**: Für Server ist Windows Server 2008 R2 oder Windows Server 2012 erforderlich.

#### Mac

Zur Verwendung von Chrome auf einem Mac benötigen Sie Folgendes:

* OS X Yosemite 10.10 oder höher

#### Linux

Zur Verwendung von Chrome unter Linux benötigen Sie Folgendes:

* Ubuntu (64 Bit) 14.04 oder höher, Debian 8 oder höher, openSUSE 13.3 oder höher oder Fedora Linux 24 oder höher
* SSE2-fähigen Intel Pentium 4-Prozessor oder höher

# Produktfunktionen

## Fake Login

Nach dem Starten der Applikation (das Aufrufen der initialen URL) wird ein Login Screen präsentiert. Dabei wird jedoch lediglich der Name des Spielers festgelegt. Ein effektives Login mit Logindaten auf dem Server werden höchstens als Bonus implementiert.

## Spiel

Nachdem der Benutzer durch das Login gekommen ist, kann dieser einen automatisch erstellten Link an einen Freund schicken, mit welchem dieser dem Spiel als Gegner beitreten kann. Das restliche Spielprinzip ist unter “Zielbestimmung” bereits definiert.

## Session Join

Der Benutzer kann seinem Freund einen Link schicken, über welchen dieser der Spielsession des Benutzers beitreten kann.

# Produktdaten

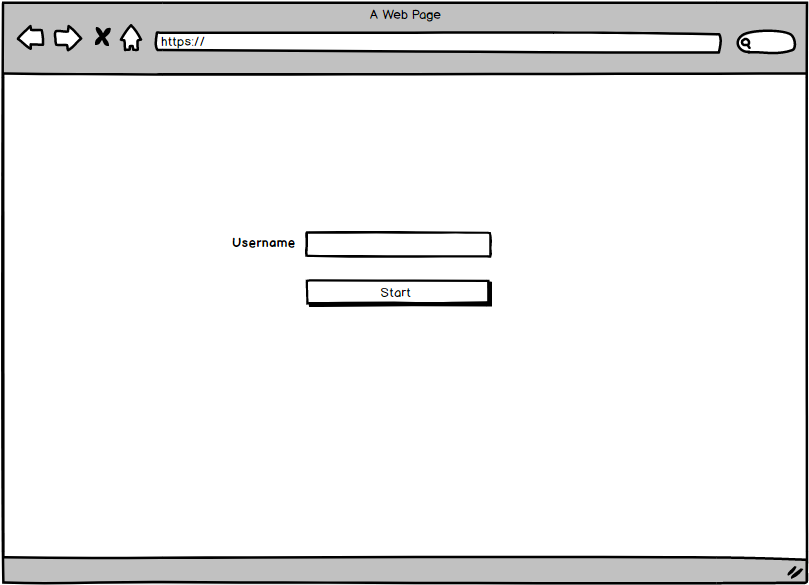
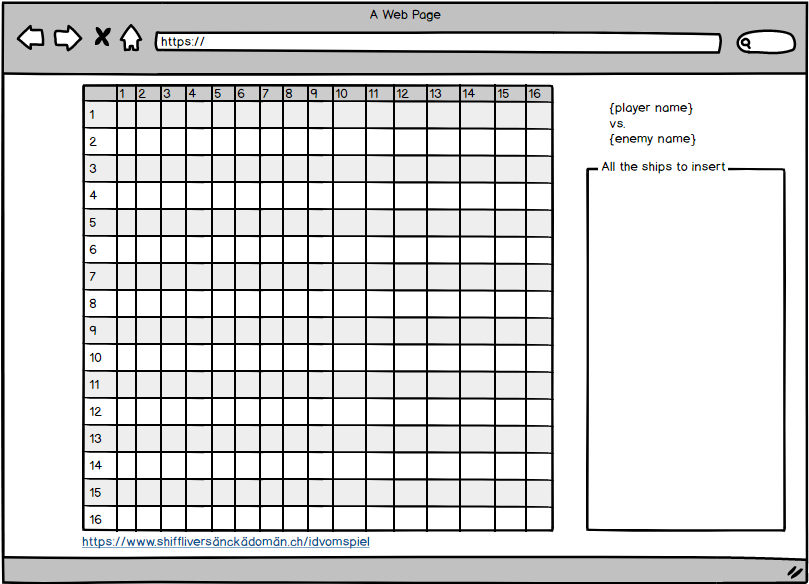
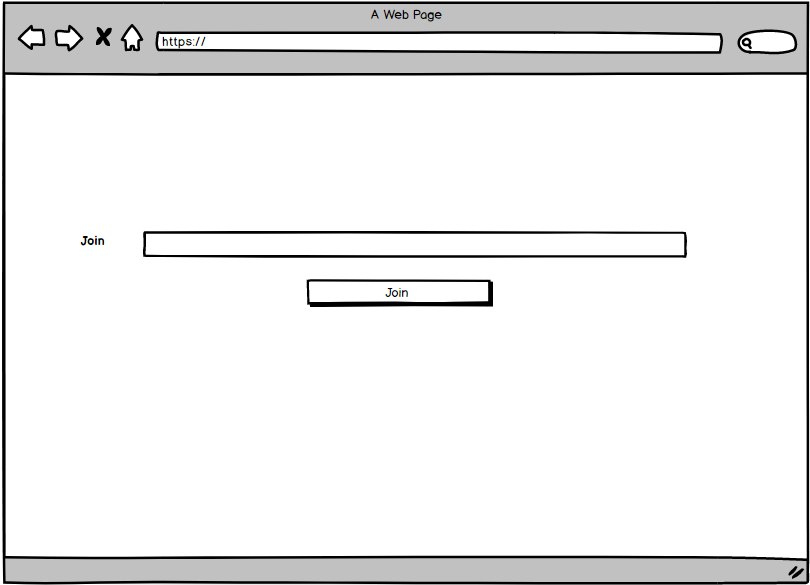
Das Produkt erhebt in seiner grundsätzlichen Form keine Daten. Es wird lediglich der Benutzername für die Session gespeichert. Nach Beendigung des Spiels wird die Session (und der Benutzername) gelöscht.

Falls der optionale Highscore noch implementiert wird, wird für diesen der Benutzername sowie der Punktestand persistent gespeichert.

Falls das Login umgesetzt wird, wird hierfür ein Benutzername sowie ein Passwort gespeichert. Dieser Benutzername wird dann ggf. mit einem Highscore (oder allgemein mit einem Score) verbunden sein. Eventuell werden noch Metadaten wie der letzte Loginzeitpunkt oder Erzieleungsdatum des Highscores gespeichert

# Benutzeroberfläche

Die Benutzeroberfläche der Applikation besteht aus drei Fenstern. Das erste Fenster ist ein aus einem Benutzernamen Feld und einem Start Button bestehendes Login Fenster. Im Spiel Fenster findet man auf der linken Seite ein Raster mit 16 x 16 Feldern, in welches man seine Schiffe platzieren muss. Am rechten Rand findet man seinen Benutzernamen sowie den seines Gegners und alle eigenen Schiffe, welche noch platziert werden müssen. Unterhalb des Spielfeldes befindet sich ein Link, welcher an Freunde verschickt werden kann. Mit diesem ist es möglich, dem Spiel des Freundes als Gegner beizutreten. Alternativ kann man diesen Link auch im letzten Fenster einfügen und auf den sich unter dem Eingabefeld befindenden Join Button klicken.



# Globale Testszenarien und Testfälle

Bei unserem Projekt gibt es mehrere Testszenarien. Folgende Test müssen erfüllt sein:

|  |  |
| --- | --- |
| Testfall 1: Login | |
| Ziel | Der Benutzer soll sich mit einem beliebigen Benutzernamen anmelden können. |
| Akteur | Neuer Benutzer |
| Vorbedingungen | Der Server wurde gestartet.  Der Websocket wurde auf dem Server gestartet. |
| Ablauf | * Der User öffnet die Website im Browser * Der User verwendet einen beliebigen Benutzernamen für das Login * Der User klickt auf Start |
| Nachbedingungen | Die Hauptseite der Applikation mit dem Spielfeld und dem gewählten Benutzernamen erscheint auf dem Bildschirm. |
| Sonderfall |  |

|  |  |
| --- | --- |
| Testfall 2: Beitritts Link verschicken | |
| Ziel | Der Benutzer soll einen Link zum Beitritt zu seinem Spiel versenden können. |
| Akteur | Eingeloggter Benutzer |
| Vorbedingungen | Der Server wurde gestartet.  Der Websocket wurde auf dem Server gestartet.  Der Benutzer ist eingeloggt.  Ein Spiel wurde gestartet. |
| Ablauf | * Der User kopiert den Link unterhalb des Spielfeldes * Der User sendet den Link an einen Freund |
| Nachbedingungen | Der Freund kann per Klick auf den Link dem Spiel beitreten. |
| Sonderfall |  |

|  |  |
| --- | --- |
| Testfall 3: Per Link beitreten | |
| Ziel | Der Benutzer soll per Link einem Spiel beitreten können. |
| Akteur | Neuer Benutzer |
| Vorbedingungen | Der Server wurde gestartet.  Der Websocket wurde auf dem Server gestartet.  Mit einem Anderen Browser wurde ein Spiel geöffnet. |
| Ablauf | * Der User klickt auf den Spiel Link |
| Nachbedingungen | Die Hauptseite der Applikation mit dem Spielfeld und dem eigenen sowie dem Benutzernamen des Gegners erscheint. |
| Sonderfall |  |

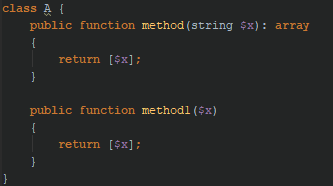
|  |  |
| --- | --- |
| Testfall 4: Spielen | |
| Ziel | Der Benutzer soll das Spiel spielen können |
| Akteur | Eingeloggter Benutzer |
| Vorbedingungen | Der Server wurde gestartet.  Der Websocket wurde auf dem Server gestartet.  Der Benutzer ist eingeloggt.  Ein Spiel wurde gestartet.  Ein anderer Spieler ist beigetreten. |
| Ablauf | * Beide User verteilen Ihre Schiffe auf dem Spielfeld * Die Benutzer spielen zusammen das Spiel |
| Nachbedingungen | Das Spiel soll funktionieren. Somit soll am Schluss ein Sieger hervorgehen. |
| Sonderfall |  |

# Entwicklungsumgebung

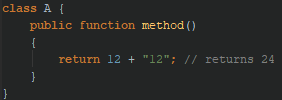
Die Applikation wird auf Windows PCs mit XAMPP entwickelt, jedoch auf einem Linux Server installiert. Da PHP nicht Abhängig vom Betriebssystem ist, macht dies keinen Unterschied.

# PHP Sprach Features

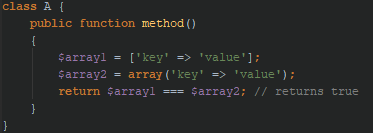
## Variablen

Variablen können, müssen aber nicht explizit typisiert werden.  
In diesem beispiel retournieren beide Methoden einen Array mit dem String $x. Wenn die erste methode “method” mit einem Array (oder Object, welches kein String ist) aufgerufen wird, gibt es einen “Fatal Error”. Ebenfalls würde es diesen Fehler geben, falls die Methode einen Wert, welcher nicht dem Datentyp “array” entspricht, zurückgibt.  
Bei der zweiten Methode “method1” wird jeder Datentyp akzeptiert und der Rückgabewert ist ebenfalls nicht definiert.  


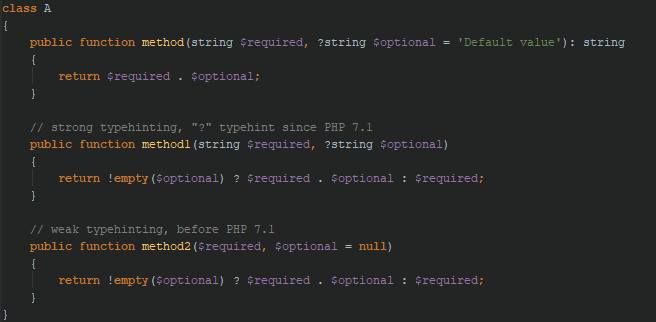
## Strings

Numerische Strings können wie Integer in Kalkulationen verwendet werden.  
Bei dem Beispiel wird von PHP automatisch der Wert ausgerechnet. Es ist kein Typencasting nötig.  


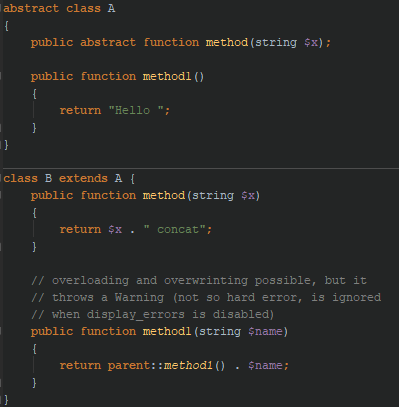
## Arrays

Arrays werden mittels “[]” oder “array()” initialisiert. Beide Arrays sind vom Datentyp (logischerweise) und vom Aufbau her identisch.  


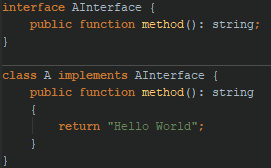
## Überladen

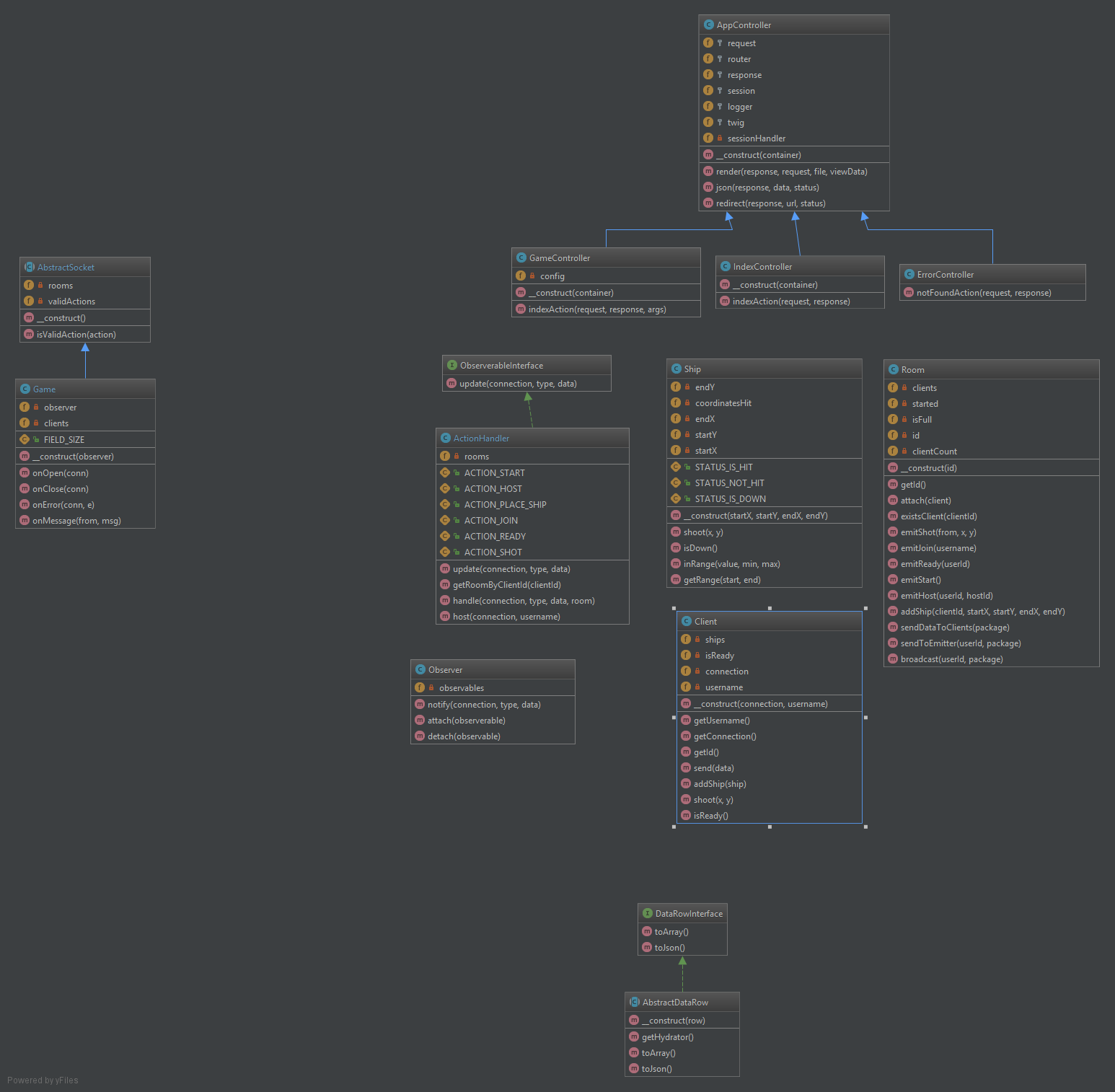
Methoden werden für gewöhnlich nicht überladen, sondern erhalten optionale Parameter. Dabei ist es möglich, bereits bei der Auflistung der Parameter einen Standardwert zu übergeben (siehe “method”). Wenn man einen Parameter vollkommen optional machen möchte, kann man diesen per default auf NULL setzen oder mittels des “?” Typehints optional machen. Wenn kein Standardwert definiert wird, entspricht dieser (falls er beim Methodenaufruf nicht definiert wird) dem Wert NULL. 

## Überschreiben

Das überschreiben in PHP funktioniert genau gleich wie in Java. Man kann eine Methode der Superklasse in der Kindklasse implementieren. Die Methode in der Kindklasse wird dann ausgeführt. Ebenfalls sind Abstrakte Methoden möglich, welche in der Klasse selbst implementiert werden müssen. Will man dies nicht, muss man die Klasse Abstrakt definieren, was bedeutet, dass diese Klasse nicht instanziierbar ist.   
Der Aufruf der Superklasse erfolgt nicht wie in Java mit super, sondern mit parent::methodName().  


## Interfaces

Wie in Java gibt es in PHP ebenfalls die Möglichkeit, Interfaces zu definieren und zu implementieren.  


****