

# Projektdokumentation Dame 2.0

---

Projekt A von Annika Bode (MatrNr.: 2116458) und Jan Ryklikas (MatrNr.: 2115506),  
WiSe 2016/2017, Tutoren: Herr Alpers und Professor Plaß

Dame 2.0 ist die Erweiterung des klassischen Dame Brettspiels. Hierbei spielen bis zu 6 Spieler auf einem wabenähnlichen Spielfeld gegeneinander. Wie auch beim Dame Brettspiel zieht der Spieler vorwärts, überspringt und schlägt dabei andere Spielsteine und erhält eine Dame, wenn er am anderen Spielfeldrand angekommen ist. Hinzu kommt eine Anzahl von Spielkarten, welche das Spiel noch spannender machen sollen. Sie enthalten Anweisungen, die sogenannten Mauern aufzubauen, zu verschieben oder abzureißen. Der Spieler, der als letztes noch Spielsteine auf dem Feld hat ist der Gewinner.

## Aufgabenbeschreibung

### Spielvorbereitung

Der User kann einen Account erstellen, womit er spielen kann. Dies bietet den Vorteil, dass er seine Spielstatistiken speichern und diese bei Bedarf auswerten kann. Die Erstellung eines Accounts ist aber nicht zwingend notwendig. Man kann auch als Gast das Spiel nutzen. Hierzu muss der User nur einen Usernamen wählen und kann losspielen. Nach dem Login als Gast oder als User kann ein Spielstein generiert und Mitspieler eingeladen werden. Der Spielstein wird dynamisch von PHP generiert und als Bilddatei gespeichert. Dieses kann später abgerufen werden, um es auf dem Spielfeld angezeigt zu bekommen.

### Spielfeld

Das Spielfeld besteht aus dem 6-eckigen Spielbrett, den Kartenstapeln und der Spielanweisung für den Spieler. Hier werden die unterschiedlichsten Informationen der einzelnen Komponenten zusammengetragen und ausgegeben. Das Spielbrett wird mit CSS generiert.

Die Spielfiguren der Spieler werden als Symbole mit der eigenen Farbe dargestellt, Mauern als braune Flächen und angeklickte Felder Orange, um dem Spieler ein Feedback über seine Eingabe zu geben.

## Kartenstapel

Der Kartenstapel setzt sich aus 50 Karten zusammen, die sich bei jeder neuen Partier neu zusammenwürfeln. Dazu gibt es eine festgelegte Formel. Die Karten werden in „Setzen“, „Verschieben“ und „Entfernen“ unterschieden. Zusammen mit einer Anzahl und der Endung „.png“ werden die Karten im Kartenstapel gespeichert.

```
//setzen: 7/25 des Stapels sind Setzen-Karten
$setzenKarten = array("EinsSetzen", "ZweiSetzen", "DreiSetzen", "VierSetzen");
//verschieben: 9/25 des Stapels sind Verschieben-Karten
$verschiebenKarten = array("EinsVerschieben", "ZweiVerschieben", "DreiVerschieben", "VierVerschieben");
//entfernen: 9/25 des Stapels sind Entfernen-Karten
$entfernenKarten = array("EinsEntfernen", "ZweiEntfernen", "DreiEntfernen", "VierEntfernen");
$anzahlSetzenKarten = 7/25*$this->limit;
$anzahlVerschiebenKarten = 9/25*$this->limit;
$anzahlEntfernenKarten = 9/25*$this->limit;
```

## Installationsanleitung

Um das Spiel zu installieren ist zunächst ein Apache2 Server mit installiertem PHP5.6 Modul erforderlich, sowie ein FTP Zugang zu selbigem.

In das Hauptverzeichnis des Servers werden nun die Spieldaten kopiert und anschließend wird mit dem Web Browser die Seite aufgerufen.

Für erweiterte Features wie Login und das Speichern von Nutzerdaten wird weiterhin ein MySQL Server benötigt, auf diesem muss eine Datenbank eingerichtet sein und die Zugangsdaten müssen in den Spieldateien hinterlegt werden. Die Einrichtung der Tabellen erfolgt automatisch.

Das Spiel benötigt keine Aktive Internetverbindung sofern der Server im Lokalen Netz oder auf der Maschine des Nutzers läuft.

Das Spiel kann mit allen gängigen Internet Browsern gespielt werden.

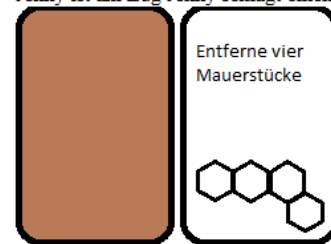
## Bedienungsanleitung

Nachdem ein Internet Browser geöffnet und eine Verbindung zum PHP-Server hergestellt wurde, gelangt der User auf die Index Seite. Dort hat er die Möglichkeit einen Account anzulegen, sich einzuloggen oder als Gast zu spielen. Nach erfolgreichem Anmelden mit einer der drei Möglichkeiten, gelangt der User in die Einstellungen. Hier kann er seinen Spielstein gestalten, indem er zwischen 3 verschiedenen Formen wählt und diesen dann mit einer individuellen Farbe einfärbt. Außerdem kann der User entscheiden, ob er mit anderen zufälligen Spielern eine Partie spielen oder ob er Freunde zu einer privaten Partei einladen möchte. Nun gelangt der Spieler auf das Spielfeld und kann seine Partie spielen. Das Spielfeld beinhaltet das Spielbrett auf dem Spielsteine und die Mauerstücke gesetzt werden können. Außerdem befinden sich hier auch der Kartenstapel und ein Button, der zu den Optionen führt. In den Optionen findet der User die Spielregeln.

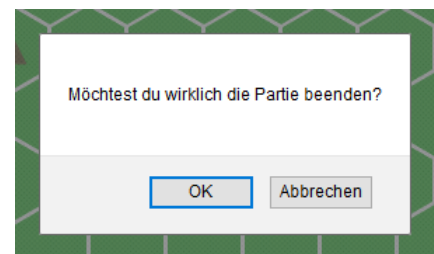
Der Spielaufbau beinhaltet 6 Spieler, die jeweils 6 Spielsteine an einer Kante des Spielfelds haben. Eine Spielrunde beginnt mit dem Ziehen einer Karte. Dann kann der User einen seiner Spielsteine setzen und ggf. einen gegnerischen Spielstein schlagen. Mit dem Betätigen des Buttons „Zug beenden“ wird das Ziehen des Spielsteins beendet. Der Statustext oberhalb des Kartenstapels soll dem User einen Überblick verschaffen, was als nächstes zu geschehen hat. Als nächstes wird die Anweisung auf der Karte ausgeführt und mit dem Button „Mauer platzieren“ übernommen. Der Zug des Spielers ist nun beendet und der nächste Spieler ist an der Reihe. Wenn ein Spielstein die gegenüberliegende Kannte des Spielfelds erreicht, wird dieser zu einer Dame umgewandelt. Die Dame darf in alle Richtungen ziehen und schlagen.

Sollte der User vorzeitig die Runde beenden wollen, kann er auf den Button „Aufgeben“ klicken. Es erscheint ein Popup mit der Frage, ob der User wirklich aufhören möchte. Bei Bestätigung wird die Partie beendet.

Jenny ist am Zug Jenny schlägt einen Gegner!



Mauer so platzieren?  
Zug beenden!  
Spiel aufgeben?  
reset



## Systemarchitektur

Unsere Systemarchitektur ist eine Client-Server Architektur.

Der Apache Server interpretiert die PHP Dateien und liefert dem Browser eine HTML Seite aus, die angezeigt wird. Der Nutzer interagiert nun mit dieser HTML Seite über Schaltflächen.

Um das ständige neu laden der Seite zu verhindern werden einige Eingaben mittels JavaScript an ein spezielles PHP Script auf dem Server geschickt und sofort ausgewertet, um dem Nutzer ein Feedback anzuzeigen z.B. Verfärbung von angeklickten Schaltflächen (AJAX).

Unsere Softwarearchitektur orientiert sich sehr grob an dem „Model-View-Controller“ Modell auch „MVC“ genannt, wobei in unserem Fall keine Templates als Views hinterlegt werden, sondern das HTML in den Models generiert wird.

Unsere Verzeichnisstruktur umfasst 4 Verzeichnisse:

- Bilder           enthält alle Bilddateien
- Klassen       enthält alle PHP Klassen
- Includes       enthält alle Dateien die irgendwo eingebunden werden (z.B. CSS, JS)
- Web           enthält alle PHP Dateien die direkt vom Browser aufgerufen werden

Die Controller, die im Endeffekt die Seite generieren, arbeiten ausschließlich mit den Methoden der Klassen.

Alle Daten der laufenden Spielsitzung, wie Spieler, Felder und Positionen, werden im Browser Cache gespeichert (PHP Session) und bei der Initialisierung des Controllers in den Speicher geladen.

Das JavaScript läuft Client-Seitig und ist die Schnittstelle zwischen User Interface und PHP. Alternativ könnte man auch ohne JavaScript mit Formularen arbeiten, das ist jedoch nicht mehr zeitgemäß und schwer zu bedienen.

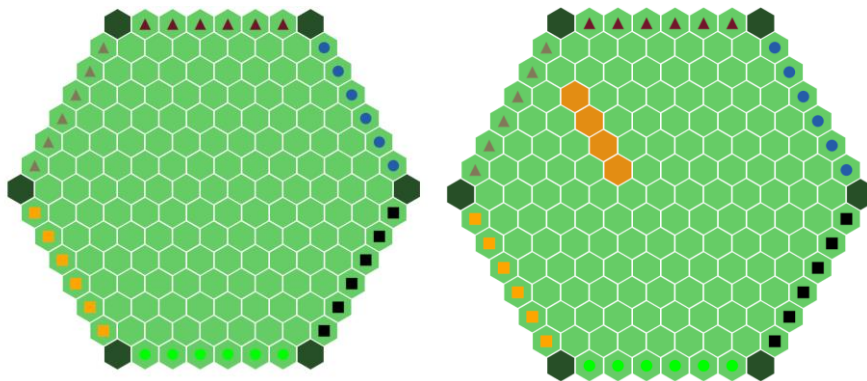
## Ein genauerer Blick auf das Spielfeld

Besonders sticht bei unserem Spiel das Spielfeld hervor, welches sich aus 169 Hexagonalen Flächen zusammensetzt.

Dieses Feld wird ausschließlich mithilfe von HTML und CSS generiert indem man zuerst ein rechteckiges <div> erzeugt und oben und unten zwei weitere <div>. Diese werden nun auf ihren unteren/oberen Rahmen reduziert, der bei einer <div>-Höhe von 0 spitz zuläuft (siehe Abb.), man kann nun diese 2 Ecken und das Rechteck in einen Container zusammenfassen und ein Hexagon entsteht.



Nun generieren wir ein ebenfalls Hexagon-förmiges Spielfeld aus diesen einzelnen Feldern.



Die einzelnen Felder können unterschiedliche Zustände annehmen, so sind z.B. die dunklen Flächen nicht klickbar und auf an den Kanten stehen die Spielfiguren.

Die Zustände der Felder werden aus einem Array von Feldobjekten im PHP ausgelesen und beim Generieren auf das Spielfeld übertragen.

## Projektauswertung

Das Projekt wurde im Sommer Semester 2016 angefangen unter der Leitung von Markus Alpers. Vorgabe war ein interaktives Spiel, welches ausschließlich mit HTML, PHP und ggf. MySQL realisiert werden sollte. CSS und andere Programmiersprachen durften nicht genutzt werden. Die Projektidee und die Strukturentwicklung nahmen einen großen Teil der Zeit ein, da immer wieder Korrekturen gemacht werden mussten. Im Wintersemester war die Planungsphase größtenteils beendet und die Programmierphase konnte endlich beginnen. Zuerst sollte der HTML-Teil programmiert werden und dann der PHP- und MySQL-Teil. Während der Leitung von Herrn Alpers gab es ein regelmäßiges Feedback zum aktuellen Stand. Nach der Übergabe der Verantwortung an Professor Pläß wurden die Vorgaben geändert und so konnten viele Unwegsamkeiten in der Programmierung gelöst werden. Mit CSS konnte eine bessere Darstellung des Spielfelds und deren Spielsteinen entwickelt werden und mit Javascript Funktionen erstellt werden, die ein neu laden der Seite nicht mehr so oft erforderlich machen. Die Gruppenarbeit konnte bis zur Ablösung durch Professor Pläß nicht stattfinden, sodass danach erst eine Gruppe gebildet werden konnte und die Arbeit fortgesetzt werden konnte.

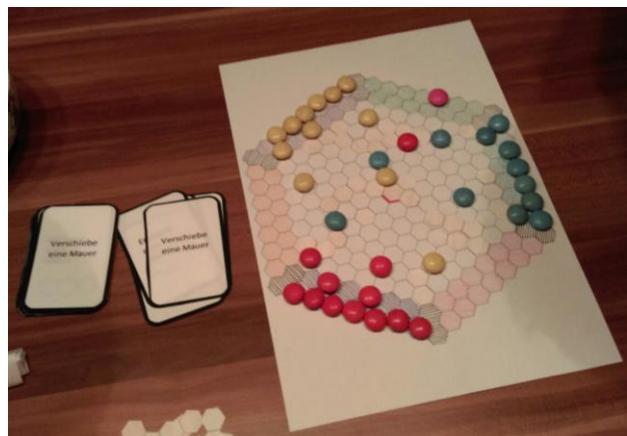
Das Arbeiten mit Github war leicht und unkompliziert. Sowohl die Gruppenarbeit als auch die Feedbacks wurden hierrüber koordiniert. Die von Github entwickelte Entwicklungsumgebung „Atom“ ist leicht zu bedienen und in Verbindung mit PHP sehr nützlich.

Die Ordnerstruktur hat sich im Laufe des Projekts geändert und auch die Schwerpunkte der einzelnen Dateien haben sich mehrfach geändert. Mittlerweile haben die Includes, Klassen, Webansichten und Bilder einen eigenen Ordner.

## Test

Um das Spiel zu testen, wurde es als Brettspiel aufgebaut und mit 3 Spielern ausprobiert. Die Ergebnisse aus diesem Test werden im Ordner „Balancing“ zusammengefasst. Der Test war sehr aufschlussreich und hat zur Verbesserung des Spielerlebnisses geführt:

- Das Spielfeld wurde um die äußeren Felder verkleinert.
- Die Mauerkarten wurden angepasst, um ein flüssigeres Spiel zu ermöglichen.
- Die Anzahl der Spielsteine wurde mehr als halbiert, um die Spielzeit der Partie zu verkürzen.



## Schwierigkeiten

- Durch die erst spät angefangene Gruppenarbeit war der Ideenaustausch eingeschränkt.
- Die Feedbacks und Aufgabenstellungen waren teilweise unverständlich und wenig hilfreich.
- In der Vorlesung gab es keine Einführung in PHP und HTML. Dies wäre wünschenswert gewesen.
- Die zuerst vorgegebene Systemstruktur war kompliziert und schlecht umsetzbar.

## Ein Blick in die Zukunft

- Erweiterte Spieloptionen
  - o Implementieren einer Hintergrundmusik
  - o User kann Spielfeldfarbe verändern
- Einfügen von NPCs, um auch als Singleplayer spielen zu können.
- Die Spielstatistik muss noch dargestellt werden.
- Die Umwandlung in Damen muss noch implementiert werden.

## Quellen

Zur besseren Darstellung des Spielfelds wurde das CSS-Hexagon-Prinzip genommen und für die eigene Anwendung abgeändert: (zu finden in der Datei /includes/style.css)

<https://jtauber.github.io/articles/css-hexagon.html>