|  |
| --- |
|  |
| Viergewinnt |
|  |

|  |
| --- |
|  |

Andreas Nunberger

Ausbildung zum Fachinformatiker Anwendungsentwicklung

bei der Data Net Solutions GmbH

Mario Zeindlmeier

Ausbildung zum Fachinformatiker Anwendungsentwicklung

bei der Flughafen München GmbH

Fertigstellung am 10.02.2017

Inhaltsverzeichnis

[1. Projektauftrag 1](#_Toc474772654)

[1.1 Projektauftrag 1](#_Toc474772655)

[1.2 Projektumfeld 1](#_Toc474772656)

[1.4 Projektabgrenzung 1](#_Toc474772657)

[2. Projektplanung 1](#_Toc474772658)

[2.1 Ist-Zustand 1](#_Toc474772659)

[2.2 Soll-Konzept 1](#_Toc474772660)

[2.3 Vorgehensmodell 1](#_Toc474772661)

[2.4 Zeitplanung 1](#_Toc474772662)

[3. Projektdurchführung 1](#_Toc474772663)

[3.1 Implementierung 1](#_Toc474772664)

[3.1.1 Layout und Design 1](#_Toc474772665)

[3.1.2 Singleplayer 1](#_Toc474772666)

[3.1.3 Multiplayer (Hotseat) 1](#_Toc474772667)

[3.1.4 Online Multiplayer 1](#_Toc474772668)

[3.1.5 User Verwaltung 1](#_Toc474772669)

[3.1.6 Multiplayer Statistiken 1](#_Toc474772670)

[3.2 Qualitätssicherung 1](#_Toc474772671)

[4. Projektabschluss 1](#_Toc474772672)

[4.1 Soll-Ist-Vergleich 1](#_Toc474772673)

[4.2 Fazit 1](#_Toc474772674)

[4.3 Ausblick 1](#_Toc474772675)

[5. Anhang 1](#_Toc474772676)

[5.1 Quellen 1](#_Toc474772677)

[5.2 Abbildungsübersicht 1](#_Toc474772678)

[5.3 Zugangsdaten 1](#_Toc474772679)

1. Projektauftrag

**1.1 Projektauftrag**

Das Projekt Viergewinnt wird im Rahmen des Unterrichtsfaches Anwendungsentwicklung und Programmierung im Bereich Android / Java durchgeführt. Ausgeführt wird das Projekt beim Auftraggeber der Staatl. Berufsschule Freising im Unterrichtsraum 1A1 der Klasse IT12B. Es soll eine Spiel-App für das Betriebssystem „Android“ erstellt werden. Des Weiteren sind eine akribische Dokumentation der Arbeitsschritte und eine Präsentation für die Mitschüler Teil des Projekts. Das Projekt wird im Zeitraum von 05.12.2016 bis 14.02.2017 realisiert.

**1.2 Projektumfeld**

Flughafen München GmbH

Der 1992 am neuen Standort im Erdinger Moos eröffnete Flughafen hat sich binnen weniger Jahre zu einem der führenden europäischen Luftverkehrsdrehscheiben entwickelt. Im Frühjahr 2015 ist der Flughafen München zum ersten 5-Sterne-Flughafen Europas gekürt worden. Gemeinsam mit den 12 Beteiligungsunternehmen wird der zweitgrößte Airport in Deutschland betrieben. Ende 2015 wurde einen Beschäftigungsstand von 8.016 Mitarbeiter erreicht. Durch Kompetenz und Engagement der Mitarbeiter wurde ein Passagierrekord von 41,0 Millionen Fluggästen und einem Konzernumsatz von 1,25 Milliarden Euro erreicht.

Data Net Solutions GmbH

Die Data Net Solutions GmbH ist ein renommierter Softwarehersteller und Dienstleister im Gesundheitswesen in St. Wolfgang in Oberbayern. Unsere Produktlinie AMONDIS umfasst eine breite Palette an etablierten Lösungen für Logistik und Administration im Gesundheitswesen. Unsere Softwareprodukte stehen für Effizienz, Anwenderfreundlichkeit und vielseitige Einsetzbarkeit. Bundesweit vertrauen uns mehr als 500 Kliniken und Einrichtungen aller Größenordnungen bis zur Universitätsklinik und privaten Klinikkonzernen. Beratung und Umsetzung gehen bei uns Hand in Hand. Eine Aufgabe ist für uns erst dann gelöst, wenn unser Kunde sein Ziel erreicht hat.

**1.3 Projektziel**

Das Ziel des Projekts ist es, eine funktionstüchtige Android App mit einem Singleplayer, einem Multiplayer auf einem Gerät und einem Multiplayer über mehrere Geräte zu realisieren. Dazu sollen Statistiken im Online Multiplayer erfasst werden, ebenso müssen sich alle User Registrieren können, und nach einem erfolgreichen Login an Spielen teilnehmen können.

**1.4 Projektabgrenzung**

Der Projektstart Termin beläuft sich auf den 05.12.2016.

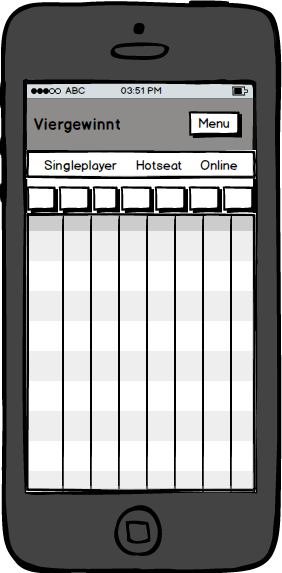
Das Projekt wurde auf Basis eines Agilen Projektmanagements aufgebaut. Bis zum 15.12.2016 wurden alle Sprits der Inkremente definiert und nur Bearbeitung den Projektmitgliedern zugewiesen. Bis Ende des Jahres 2016 soll in der Android App ein vollfunktionstüchtiger Multiplayer auf einem Gerät realisiert werden. Im zweiten Sprint wird ein Singleplayer gegen einen Computer Gegner bis einschließlich 15.01.2017 implementiert. Am abschließenden Schritt wird der Online Multiplayer bis 01.02.2017 eingebaut. Anschließend wurde die fertige App einem ausführlichen Test unterzogen. Eine Woche vor der endgültigen Produktivmachung werden die rückgemeldeten Fehler aus dem Test vom Projektteam ausgebaut. Der Projektabgabe Termin ist der 14.02.2017. Durch die sehr knappen Zeitrahmen kann im Rahmen des Projekts kein anschauliches App Layout erstellt werden. Dem Auftraggeber wurden die Funktionsbausteine höher Priorisiert.

2. Projektplanung

**2.1 Ist-Zustand**

Viergewinnt ist ein klassisches Tischspiel welches zu zweit gespielt werden kann. Ein Spielen ist in der klassischen Variante nur zu zweit und nur am gleichen Ort möglich. Das Spiel kann somit in seiner ursprünglichen Version nur im Multiplayer gespielt werden. In einem Klassenzimmer der Staatlichen Berufsschule Freising, befinden sich mehrere Android Smartphones des Typs Motorola G2. Auf keinem dieser Handys befindet sich eine App, welche das Spielen des Klassikers ermöglicht.

**2.2 Soll-Konzept**

Es soll im Zuge des Projekts allen Schülern möglich sein, auf ihrem Android Smartphone das Spiel Viergewinnt spielen zu können. Links auf dem Bild, ist das Soll-Konzept anhand eines ersten Mockups ersichtlich. Die Anforderungen wurden im Laufe des Projekts angepasst.

Im Rahmen des Projekts Viergewinnt wurde ein vollumfassendes Lastenheft erstellt. Aus dem Lasterhaft geht folgendes hervor:

1. Ein Singleplayer gegen eine künstliche Intelligenz muss in der Android App realisiert werden
2. Eine Registrierung über eine MySQL Datenbank muss möglich sein
3. Das Spielen gegen einen Freund auf demselben Gerät muss möglich sein

Abb. 1 Mockup Viergewinnt

1. Der Spielablauf muss sich an den Regeln des bekannten Viergewinnt orientieren
2. Es muss möglich sein, über das Internet auf mehreren Geräten ein Online Match zu bestreiten
3. Es muss möglich sein, seine Statistiken der Online Spiele zu verfolgen
4. Das Benutzer Passwort muss verschlüsselt in der MySQL Datenbank gespeichert werden
5. Dem User muss ermöglicht werden, sich aus der App auszuloggen.

**2.3 Vorgehensmodell**

Jedem Projektmitglied werden nach den individuellen Qualifikationen Arbeitspakete zugeteilt. Diese Programmierschritte sollen in dem zuvor definierten Zeitrahmen und innerhalb des geschätzten Stunden Budgets erfolgen. Die Durchführung dieser Programmierschritte erfolgt mithilfe eines inkrementellen Modells. Die Schritte der einzelnen Inkremente wurden wie folgt definiert:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1. **Inkrement** | 1. **Inkrement** | 1. **Inkrement** |
| Spielmodus | Lokaler Multiplayer | Singleplayer gegen eine künstliche Intelligenz | Online Multiplayer |
| Layout | Layout Grundlage | Layout Formatierung anhand des Mockups | Keine Anpassungen in diesem Inkrement |
| Registrierung/  Login | User Login über MySQL Datenbank | Registrierung in der App möglich | Passwort verschlüsselt in der Datenbank speichern |
| Datenbank Model | User Tabelle | Keine Anpassungen in diesem Inkrement | Multiplayer Tabelle |
| Fehlerbehebung | Keine Anpassungen in diesem Inkrement | Fehler die im Test des 1. Inkrements aufgetreten sind | Fehler die im Test des 2. Inkrements sowie im abschließenden Test aufgetreten sind |
| Dokumentation | Arbeitsschritte Dokumentieren | Arbeitsschritte Dokumentieren | Arbeitsschritte Dokumentieren |

**2.4 Zeitplanung**

| Arbeitsschritt | Dauer in Stunden |
| --- | --- |
| Installation und Konfiguration der benötigten Software | 1 Stunden |
| Planung / Erstellung eines Konzepts | 3 Stunden |
| Grundlegende Funktion der App implementieren | 11 Stunden |
| Server Bereitstellung und Installation | 4 Stunden |
| Multiplayer auf einem Gerät realisieren (Hotseat) | 5 Stunden |
| Multiplayer über mehrere Geräte – Android / Java | 11 Stunden |
| Multiplayer über mehrere Geräte – PHP | 4 Stunden |
| Singleplayer gegen einen Computer Gegner | 12 Stunden |
| Tests und Qualitätssicherung | 4 Stunden |
| Rückgemeldete Fehler ausbauen | 5 Stunden |
| Projektdokumentation | 10 Stunden |
| Gesamt Dauer aller Arbeitsschritte | 1. **Stunden** |

1. Projektdurchführung
   1. **Implementierung**

**3.1.1 Layout und Design**

Das Layout und das Desing der Nutzernavigation sind nur im groben implementiert. Viele Funktionen wurden aufgrund einer starken Zeitlichen Beschränkung nicht umgesetzt. Die Priorisierung des Auftraggebers lag bei der Layoutgestalltung nicht besonders hoch, weshalb nur ein schlichtes, aber in der Nutzerführung sehr leicht verständliches Grundgerüst gebaut wurde. Das Layout Konzept und die geplante Menüführung ist im Mockup (Abb.1) gut ersichtlich.

**3.1.2 Singleplayer**

In dieser Spielvariante kann gegen einen computeranimierten Spieler (auch genannt BOT) gespielt werden. Hierbei versucht der Spieler durch intelligente Spielzüge den BOT zu überlisten. Der Computerspieler wird mit mehreren Methoden und Abfragen versuchen den Spieler an einem möglichen Sieg zu hindern. Hierbei soll eine Reihe von vier gleichen Steinen des Gegners unterbunden werden. Wird keine Möglichkeit zum Gewinnen des Spieles gefunden, so wird der Spielzug des BOTs zufällig ermittelt.

**3.1.3 Multiplayer (Hotseat)**

Der Mutliplayer auf einem Handy wurde als erstes realisiert. Beide andern Spielmöglichkeiten basieren auf den Funktionen des Hotseat. Für den Aufbau des Spieles wurden im die Klassen "GameBoard" und "MainActivity" angelegt. Die erstgenannte Klasse ist für die nötigen Prüfungen im Spiel gedacht. Hier wird überprüft, ob ein Spieler gewonnen hat. Sollt dies der Fall sein werden die entscheidenden Steine mit einem eigenen Status markiert. In der "MainActivity" wird die Steuerung in der App übernommen. Es kann zwischen den einzelnen Spielmöglichkeiten gewechselt werden. Der Aufbau des Spielfeldes findet in dieser Klasse statt. Auch alle Änderungen andere Spieloberflächen werden umgesetzt. Sollte ein Spieler gewonnen haben, so werden alle Steine zurückgesetzt.

**3.1.4 Online Multiplayer**

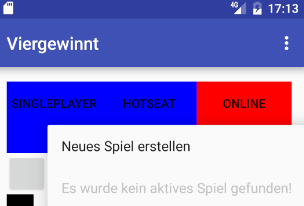
Abschließend wird der Online Multiplayer implementiert, um das Spielen über das Internet zu ermöglichen wurde eine Webspace Hoster gesucht, der sich im Kostenrahmen bewegt. Nach einer ausführlichen Recherche wurde ein guter und zugleich auch kostenloser Webspace Anbieter gefunden. Über den Anbieter können die benötigten PhP Skripte sowie eine MySQL Datenbank für den Betrieb des Multiplayers gehostet werden. Für die Plattform Android wurde im Programm Android Studio eine Klasse *GetUrlData* definiert, diese unterstützt die Klasse *PhpConnect* welche alle Funktionen beinhaltet, die für die erfolgreiche Abfrage der PhP Schnittstelle benötigt werden. Für den Multiplayer wurde in der Konzept Erstellung festgelegt, dass alle Datenbank anfragen über ein PhP Script laufen müssen. Die Hypertext Preprocessor Sprache beinhaltet einfache MySQL anfragen mit denen es möglich ist, Daten des aktiven Multiplayer Games in einer Datenbank zu hinterlegen. Auf dem MySQL Server werden die Benutzernamen, die gesetzten Seine und das Datum des Spieles hinterlegt. Alle Spiele, die den Status 1 haben, werden von der PhP Klasse als aktive gekennzeichnet und somit auch in der App als betretbares Spiel angezeigt (Abb.2). Sobald ein Spiel 2 Spieler beinhaltet wird der Status auf 0 gesetzt, dies bedeutet, dass das Spiel von anderen Android Geräten nicht mehr gefunden werden kann. Setzt ein Spieler einen Stein, wird dieser über die *PhpConnect* Klasse an den Server gesendet, dieser hinterlegt die Sein ID in der Datenbank, ein vordefiniertes Zeit interfall Prüft bei den Gegenspieler immer wieder über das PhP Script, ob bereits ein neuer Stein in der Datenbank vermerkt ist. Sobald dies der Fall ist, wird diese Seite im Game Board des Spielers angezeigt und ihm wird die Möglichkeit gegeben einen Stein zu setzen. Sollte der Gegner bereits vier Steine in einer Reihe haben, gibt es keine Möglichkeit mehr seine Spielfigur zu platzieren, daraus folgt eine Aufruf der des PhP Scripts, indem der Spieler als Lose Player gegenzeichnet wird. Wie im in den zwei Vorgehenden Spielmodi wird auch im Multiplayer bei beiden Spielern die Stein Reihe maskiert, mit der gewonnen wurde.

Abb.2 Keine aktiven Spiele

**3.1.5 User Verwaltung**

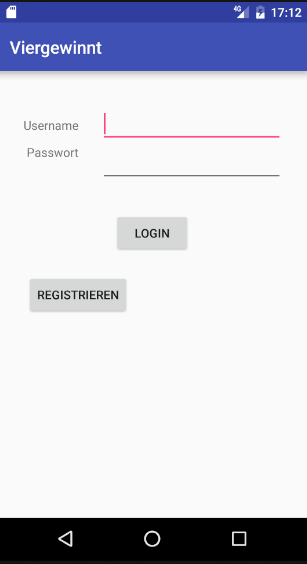
Die Userverwaltung wird online über ein PhP Script realisiert, wenn sich ein User in der App Registriert (Abb. 4), wird wie im Spielmodus Multiplayer vorgegangen, jedoch wird ein andres Script aufgerufen. An den Server werden Username und Passwort übergeben, dieser Legt einen Datensatz in der Datenbank an, wobei er das Passwort auf „sha256“ verschlüsselt. In Android wird bereits die Übereinstimmung der zwei zu eingebenden Passwörter geprüft. Sollte das Anlegen eines Datensatzes nicht möglichsein, existiert der Benutzername bereits, da dieser als Primary Key definiert wurde und somit einzigartig sein muss. Nach Android wird vom PhP Script eine „false“ zurückgegeben, sollte dies der Fall sein, wird der User darüber informiert, dass er einen anderen Usernamen verwenden sollte. Beim Login (Abb. 3) eines Benutzers wird das Passwort und der Benutzername an den Server gesendet, wenn der Rückgabe wert „true“ beinhaltet, wird der Benutzer auf den Screen des Hotseat Spieles weitergeleitet.

Abb.3 Login Screen

**3.1.6 Multiplayer Statistiken**

In der App werden nur Statistiken, des Online Multiplayers erfasst. Da alle Spiele auf in einer MySQL Datenbank archiviert werden, kann über diese auf diese Tabelle ein einfacher Select ausgeführt werden können. Es werden jeweils die Verlorenen und Gewonnenen Spiele auf addiert. In der Statistik gibt es zusätzlich zu den gewonnenen und verlorenen Spielen auch einen Punkt, welcher die Anzahl aller gespielten Spiele anzeigt. Dieser Wert wird in der App direkt berechnet, es werden dazu die Verlorenen und gewonnenen Spielaktivitäten addiert.

* 1. **Qualitätssicherung**

Um einen hohen Qualitätsanspruch gerecht zu werden, wurde während der Projekt Phase, insbesondere im Zuge der Realisierung, auf die Arbeitsweise geachtet. Im Projekt wurden folgende Qualität Merkmale beachtet:

* Terminvorgaben zu den zuvor definierten Sprints einhalten
* Programm Konventionen der Sprache Java einhalten, um eine reibungslose Zusammenarbeit im Projektteam ermöglichen zu können.
* Einsatz des Versionierungstools GitHub
* Jeden Implementierung Schritt ausführlich testen und Fehler direkt in das nächste Inkrement aufnehmen
* Code-Review

1. Projektabschluss
   1. **Soll-Ist-Vergleich**

Um die benötigte Zeit mit der Ursprüngliche Zeitplanung zu vergleichen, wurde mit Hilfe der untenstehenden Tabelle eine Übersicht erstellt.

| Arbeitsschritt | Zeitplanung | Benötigte Zeit |
| --- | --- | --- |
| Installation und Konfiguration der benötigten Software | 1 Stunde | 1 Stunde |
| Planung / Erstellung eines Konzepts | 3 Stunden | 2 Stunden |
| Grundlegende Funktion der App implementieren | 11 Stunden | 15 Stunden |
| Server Bereitstellung und Installation | 4 Stunden | 1 Stunden |
| Multiplayer auf einem Gerät realisieren (Hotseat) | 5 Stunden | 5 Stunden |
| Multiplayer über mehrere Geräte – Android / Java | 11 Stunden | 14 Stunden |
| Multiplayer über mehrere Geräte – PHP | 4 Stunden | 4 Stunden |
| Singleplayer gegen einen Computer Gegner | 12 Stunden | 13 Stunden |
| Tests und Qualitätssicherung | 4 Stunden | 4 Stunden |
| Rückgemeldete Fehler ausbauen | 5 Stunden | 3 Stunden |
| Projektdokumentation | 10 Stunden | 10 Stunden |
| Gesamt Dauer aller Arbeitsschritte | 1. **Stunden** | **72 Stunden** |

In der Ursprünglichen Zeitplanung konnten nicht alle Arbeitsschritte fristgerecht umgesetzt werden. Alle rot maskierten Felder haben einen erhöhten Zeitaufwand mit sich gezogen. Bei den grün maskierten Arbeitsschritten konnte eine Zeitersparnis verbucht werden. Weshalb ein gesamt Arbeitsaufwand von 72 Stunden entstanden ist. Die vom Projekt abgedeckte Arbeitszeit wurde somit um 2 Stunden überschritten.

* 1. **Fazit**

Alle Anforderungen, die der Auftraggeber im Anforderungskatalog definiert hat, wurden mit voller Funktionstüchtigkeit realisiert. Dem Layout wurde ein niedriger stellen Wert zugewiesen und wurde somit als alternativer Punkt betrachtet. Durch einen straffen Zeitrahmen von 70 Stunden, konnte jedoch das Layout nicht innerhalb der definierten Vorgaben umgesetzt werden. Der Online Multiplayer konnte nicht in der geschätzten Zeit realisiert werden. Es gab einige Probleme mit der Server Verbindung, aus der Android App wurde das PhP Script lange Zeit nicht richtig aufgerufen. Aufgrund einiger Probleme im Android Quellcode wurden unter diesem Punkt die Zeitlichen Vorgaben um in etwa 3 Stunden überschritten. Bei den Grundlegenden App Funktionen wurden im dritten Inkrement einige Fehler gefunden, welche einen erhöhten bearbeitungsaufwand nach sich zogen. Viel Zeit konnte bei der Server Bereitstellung gespart werden, da ein guter und leicht skalierbarer Anbieter gefunden wurde. Die Allgemeine Fehlerbeseitigung ging leicht von der Hand, wenige Fehler wurden im abschließenden Test rückgemeldet. Viele zusätzliche Funktionen wurden eingebaut, um sprechende Fehlermeldungen und somit eine höhere Akzeptanz unter den Usern zu erreichen. Durch die einfache Navigation ist es leicht für jeden neu Einsteiger, die Funktionen der App auf einen Blick zu erkennen. Als abschließendes Fazit lässt sich festhalten, dass der Großteil der Realisierung weitestgehend reibungslos abgelaufen ist. Seitens des Projekt Teams wurde viel Wert auf eine fehlerfreie Implementierung aller Elemente gelegt.

* 1. **Ausblick**

Die Android Applikation des Spieles Viergewinnt ist flexible Programmiert und kann besonders durch eine ausführliche Code Dokumentation jederzeit um zahlreiche Features erweitert werden. Das Grundgerüst des Quellcodes muss bei einer Erweiterung nicht verändert werden. Die App bietet das Potential um Folgeprojekte schnell und flexibel durchführen zu können. Daraus resultierende Folgeprojekte sind:

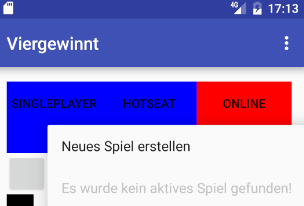
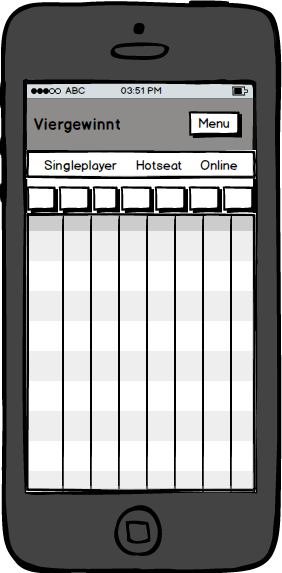
* Layout Anpassungen: Durch den Auftraggeber wurde die Funktionalität höher priorisiert, weshalb das Layout während des Realisierungsprojektes nicht unter eine große Beachtung viel. In einem Zukunftsprojekt kann ein Navigationskonzept, sowie eine anschauliche Menü Führung konzeptioniert werden. Am Grundlegenden Spiel Layout ist anhand der Image Views die Implementierung von Grafiken leicht realisierbar.
* Verschieden KI-Schwierigkeitsstufen: Im aktuellen Release wurde im Singleplayer nur eine Schwierigkeitsstufe realisiert. Diese Stufe lässt sich als eine mittlere Schwierigkeit eingruppieren.
* Verschlüsselte Übertragung: Aktuell werden die Benutzerdaten ohne Verschlüsslung an den Server übertragen, dort wird im PhP Script eine hash Verschlüsselung des Passwortes durchgeführt. Um einen vollkommenen Schutz der Benutzerdaten zu ermöglichen, sollten die privaten Daten von der Android App direkt schlüsselt werden.

1. Anhang
   1. **Quellen**

<http://stackoverflow.com/questions/33229869/get-json-data-from-url-using-android>

* 1. **Abbildungsübersicht**

|  |  |
| --- | --- |
| Abb. 1 Mockup Viergewinnt | Abb.2 Keine aktiven Spiele |



* Abb.2 Keine aktiven Spiele
  1. **Zugangsdaten**
* MySQL-Server:
  + Link: <http://phpmyadmin.wirgewinnt.square7.ch/>
  + Benutzername: wirgewinnt
  + Passwort: ABCD
* FTP-Webserver:
  + Host: wirgewinnt.square7.ch:21
  + Benutzername: wirgewinnt
  + Passwort: asdfg-01
* GitHub:
  + Link: https://github.com/gamebreak3r/wirgewinnt