

Modul 150

Dokumentation zum Projekt Wallstreet-Clicker



6. Juni 2021

Wallstreet-clicker

Majd Amro, Luke Kronenberg, Richard Leist, Samuel Porchet

Inhalt

[Kurzfassung 2](#_Toc73565559)

[Ausgangssituation 2](#_Toc73565560)

[Umsetzung 2](#_Toc73565561)

[Ergebnis 2](#_Toc73565562)

[Projektidee 2](#_Toc73565563)

[Generelle Informationen 2](#_Toc73565564)

[Projektmanagement und Planung 2](#_Toc73565565)

[Projektmanagement-Methode 2](#_Toc73565566)

[Planung 2](#_Toc73565567)

[Wissensbeschaffung 2](#_Toc73565568)

[Konzept 3](#_Toc73565569)

[Systemgrenzen, Schnittstellen und der Bezug zur Aussenwelt 3](#_Toc73565570)

[Anforderungen & Testkonzept 4](#_Toc73565571)

[Arbeitsjournal 7](#_Toc73565572)

[Arbeitsjournal – Woche 1 7](#_Toc73565573)

[Arbeitsjournal – Woche 2 8](#_Toc73565574)

[Arbeitsjournal – Woche 3 9](#_Toc73565575)

[Arbeitsjournal – Woche 4 10](#_Toc73565576)

[Arbeitsjournal – Woche 5 11](#_Toc73565577)

[Arbeitsjournal – Woche 6 12](#_Toc73565578)

[Arbeitsjournal – Woche 7 13](#_Toc73565579)

[Reflexion 13](#_Toc73565580)

[Testverfahren und Testresultate 13](#_Toc73565581)

[Testverfahren 13](#_Toc73565582)

[Testresultate 14](#_Toc73565583)

# Kurzfassung

## Ausgangssituation

Für das Modul 150, betreut durch Adrian Imboden, wurde uns der Auftrag erteilt, wie in der IPA ein Projekt zu implementieren und dieses zu dokumentieren. Gegensätzlich zur IPA konnte dieses allerdings in Gruppen umgesetzt werden. Als Projekt entschieden wir uns für ein Hommage des Spieles Cookie-Clicker in einer abgewandelten Form.

## Umsetzung

Über sieben Wochen hinweg konnten wir unser geplantes Projekt nun umsetzen. Die genaue Entwicklung unseres Projektes kann man im Arbeitsjournal, sowie als auch im Rest der Dokumentation nachlesen. Die Umsetzung erfolgte unter einer klaren Aufgabenteilung, unter welcher jeder die Möglichkeit hatte, sein potential bestmöglich auszuschöpfen.

## Ergebnis

Als Ergebnis erhielten wir ein knapp, aber rechtzeitig fertig gewordenes Produkt inklusive zugehöriger Dokumentation, mit welchem wir das Modul zufrieden abschliessen können. Bei der Implementierung des Projektes haben wir es geschafft die Funktionalitäten alle so wie gewollt zu integrieren, wohingegen wir uns beim Design uns später auf einen simpleren Look im Pixelart-Style aufgrund von Zeitmangel zu einigen hatten.

# Projektidee

Zu Beginn des Projektes mussten wir uns für ein Projekt entscheiden, bei welchem es für alle interessant ist, an der Entwicklung mitzuwirken. Recht schnell einigten wir uns darauf ein Spiel zu entwerfen, welchen den zeitlosen Klassiker Cookie-Clicker mit dem neumodernen Hype um den Aktien- und Cryptomarkt vereint. Den Wallstreet-Clicker.

# Generelle Informationen

## Projektmanagement und Planung

### Projektmanagement-Methode

Um nicht nur blind drauflos zu programmieren und zu dokumentieren haben wir uns dafür entschieden als Projektmanagement-Methode das Scrum-Framework zu nutzen. Diese Entscheidung hatten wir durch zwei Kriterien einfach treffen können.

1. Scrum ist das Projektmanagement-Framework, zu welchem wir in der Schule am meisten gelernt haben und mit welchem wir somit am besten umgehen können.
2. Das Programm Notion, ein All-in-One-Produktivitätstool, hat eine integrierte Funktion, mit welchem wir gemeinsam per Cloud-Synchronisation Scunser Scrum-Board managen und anpassen können.

### Planung

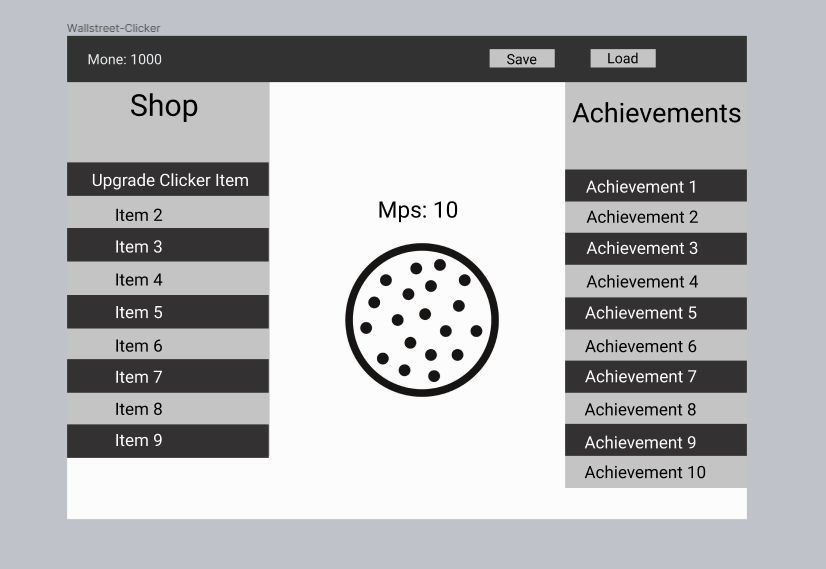
Um in kurzer Zeit eine bestmögliche Qualität zu erreichen, haben wir uns dafür entschieden, neben einem Excel-Zeitplan eine klare Aufgabenteilung zu verfolgen. So legten Luke und Samuel den Fokus komplett auf das Programmieren, wohingegen Majd und Richard sich beinahe ausschliesslich um das Dokumentieren kümmerten. Mit dieser Aufteilung konnte jeder perfekt seine Stärken ins Team einbringen und zu einem bestmöglichen Gesamtresultat beitragen.

## Wissensbeschaffung

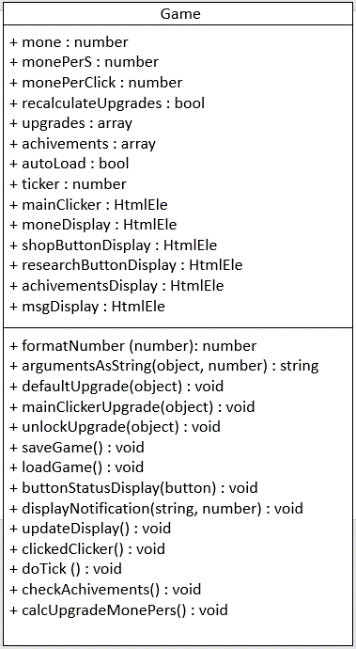
Die Wissensbeschaffung fungierte vor allem über zwei Kanäle: Durch bisherigen Unterricht oder sonstige Wissensbeschaffung und die grosse Macht von Google. Ersteres ist schwer nachträglich zu dokumentieren, da bereits erlangtes Wissen zum allgemeinen Repetour von uns gehört und somit nicht mehr nachgesucht muss und dokumentiert werden kann.

## Konzept

Damit jeder der Gruppenmitglieder eine ähnliche Vorstellung zu dem Projekt hat, haben wir ein Lo-Fi Mockup erstellt, welches den ersten Prototypen unseres Projektes darstellt:



Zusätzlich dazu haben wir eine UML-Tabelle erstellt:



## Systemgrenzen, Schnittstellen und der Bezug zur Aussenwelt

* Frameworks: Keine
* Programmiersprachen: "HTML", "SCSS", JS
* Technologien: Visual Studio Code, GitHub Desktop auf Ehre

# Anforderungen & Testkonzept

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-01** | **Mone durch aktiven Spieleingriff** | | | |
| Als Spieler will ich, dass man durch das Klicken auf den Button Mone (Spielwährung) bekommt, um sich weitere Upgrades im Shop kaufen zu können. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Bei Klick auf Button bekommt Spieler 1 (ohne Upgrades) Mone. * Die Anzeige des Monestandes wird aktualisiert. | | | | |
| **Priorität:** | | **1** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-02** | **Klickmultiplikator Upgrades (Leitfrage 1, Teil 1)** | | | |
| Als Spieler will ich, dass es im Shop Upgrades gibt, welche für Mone gekauft werden können, die mein Mone/Klick erhöhen, um für einen Klick besser entlohnt zu werden. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Mindestens ein Upgrade existiert, welches den Mone/Klick Wert verbessert. * Das Upgrade kann nur bei genügend Mone gekauft werden. * Das Upgrade kann unendlich oft gekauft werden, wird aber stetig zu einem gewissen Multiplikator verteuert. * Das Upgrade zeigt den Multiplikator im Shop an. * Das Item funktioniert wie erwartet und die Anzeige des Monestandes wird aktualisiert. | | | | |
| **Priorität:** | | **1** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-03** | **Items für passives Mone (Leitfrage 1, Teil 2)** | | | |
| Als Spieler will ich, dass ich Items kaufen kann, welche mir passives Mone geben, um auch ohne aktives Eingreifen ins Spielgeschehen mehr Mone zu verdienen, um mir meinen Spielalltag zu erleichtern. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Mindestens ein Item existiert, welches passives Mone generiert. * Das Item kann nur bei genügend Mone gekauft werden. * Das Item kann unendlich oft gekauft werden, wird aber stetig zu einem gewissen Multiplikator verteuert. * Die Statistiken des Items werden im Shop angezeigt. * Das Item funktioniert bei geöffneter Website wie erwartet. | | | | |
| **Priorität:** | | **1** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-04** | **Achievements (Leitfrage 4)** | | | |
| Als Spielmeister will ich, dass Achievements existieren, damit der Spieler Interesse daran verspürt stetig weiterzuspielen und neue Meilensteine zu erreichen. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Mindestens zehn Achievements existieren. * Dem User wird das Erreichen eines Achievements mitgeteilt. * Erreichte Achievements werden mit grüner Farbe hinterlegt. * Die Achievements sind abwechslungsreich. (Mindestens 3 verschiedene Arten) * Jedes Achievement wurde auf Funktionalität getestet und funktioniert. | | | | |
| **Priorität:** | | **2** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-05** | **Speicherung des Spielstandes (Leitfrage 3)** | | | |
| Als Spieler will ich, dass meinen Spielstand jederzeit speichern und neu laden kann, damit ich zu einem beliebig früheren Spielstand zurückkehren kann oder einen neuen beginnen kann. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Bei einem Klick auf den Save-Button wird der Spielstand als Json im Localstorage gespeichert. * Der User besitzt einen Speicherslot, welcher bei einer zweiten Speicherung überschrieben wird. * Bei einem Klick auf den Load-Button wird der Spielstand zum letzten Speicherstand geladen. * Beim Neuladen der Seite startet der Spieler wieder von neu, kann aber, wenn er möchte, seinen gespeicherten Spielstand laden. | | | | |
| **Priorität:** | | **1** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-06** | **Mathematisches Konzept (Leitfrage 2)** | | | |
| Als Spielmeister will ich, dass die Items im Shop ungefähr dem Spielfortschritt entsprechen und nicht zu stark oder zu schwach sind, damit der Spieler sein Interesse behält. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Die Startpreise der Items sind präzise definiert. * Die Folge-Preise ergeben sich aus einer mathematischen Wachstumsformel. * Jedes Item hat seine eigenen Preise und oder Wachstumsformel. | | | | |
| **Priorität:** | | **1** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-07** | **Gute Spieldarstellung** | | | |
| Als Spieler will ich, dass die Website schön dargestellt ist, damit ich mich einfach zurechtfinde. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Die Website wird mit der Schwarz-Weiss Methode implementiert (Zuerst Schwarz-Weiss, dann Blur-Test, dann färben) * Das fertige Design analysieren und auf Gestaltungsgesetze überprüfen. * Der Text steht in gutem Kontrast zur Hintergrundfarbe * Das Design wird von mehreren subjektiven Meinungen beurteilt. | | | | |
| **Priorität:** | | **3** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-08** | **Übersichtlicher Shop (Mehr oder weniger Leitfrage 1)** | | | |
| Als Spieler will ich, dass der Item Shop übersichtlich ist, damit ich mich auch beim Kauf von Items gut zurechtfinden kann. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Der Shop besitzt eine Sortierung. (Zuerst das Upgrade des Clickers, danach die Items, sortiert nach Preis) * Es werden Informationen zum Item ausgegeben. (Name, Anzahl, Mone/Sec, Kosten) | | | | |
| **Priorität:** | | **2** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-09** | **Erforschen von Items** | | | |
| Als Spielmeister will ich, dass ein Spieler manche Items zuerst erforschen muss, bevor er diese kaufen kann, damit das Spiel vielfältiger wirkt und mehr Abwechslung bietet. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Um manche Items kaufen zu können, müssen diese zuerst freigeschaltet / freigekauft) werden * Das Erforschen von Items kann in der upgrades.js Datei definiert werden. * Das Erforschen funktioniert erst, wenn der Spieler genügend Mone hat. * Das Erforschen eines einzelnen Items funktioniert nur einmal, danach: * Tritt das erforschte Item im Shop auf und kann gekauft werden | | | | |
| **Priorität:** | | **3** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-10** | **Auto-Load** | | | |
| Als Spieler will ich, dass ich auswählen kann, dass beim Starten der Website automatisch mein Spielstand geladen wird, damit ich diesen nicht manuell laden muss. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Es besteht ein On/Off Toggle, welcher diese Möglichkeit zur Verfügung stellt * Dieser Funktioniert ordnungsgemäss | | | | |
| **Priorität:** | | **4** | **Status (fertig)** | **Ja** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US-11** | **Sicherheit** | | | |
| Als Spielmeister will ich, dass es niemandem möglich ist, sein Mone und damit seinen Score, zu manipulieren, damit das Spiel normal gespielt werden muss. | | | | |
| Abnahmekriterien / erfüllte Testfälle:   * Nicht perfekt zu testen, da potenziell immer wieder neue "Glitches" gefunden werden könnten. * Testing durch Blackbox: Schauen ob es niemand hinbekommt. | | | | |
| **Priorität:** | | **4** | **Status (fertig)** | **Nein** |

# Arbeitsjournal

## Arbeitsjournal – Woche 1

Gruppe: Wallstreet-Clicker  
Tag: 20.04.2021

**Tätigkeiten:**

* Idee / Konzept entworfen
* Leitfragen formuliert
* Erste Entscheidungen zu Design gemacht
* GIT-Repo erstellt
* Arbeitsjournal Woche 1

**Ablauf:**

Zunächst haben wir uns überlegt was für ein Projekt wir umsetzen wollen. Diese Entscheidung war recht schnell getroffen, da wir alle das Prinzip von Idle-Games à la Cookie Clicker cool finden und es interessant finden, selbst mal ein Projekt dieser Art zu entwickeln. Daraufhin mussten wir selbstständig vier Leitfragen überlegen und diese zur Korrektur abgeben. Als diese nach kleineren Änderungen genehmigt wurden, haben wir uns erste Gedanken zum Design gemacht und das Git-Repo erstellt.

**Reflexion:**

Bisher lief soweit alles gut und wir sind allesamt recht motiviert für das Projekt. Wir stehen allerdings am Anfang und können daher noch nicht so viel dazu sagen, wie das Projekt verlaufen wird. Allerdings ist uns aufgefallen, dass wir unsicher bezüglich CI/CD sind und diese Lücken nächstes Mal klären müssen.

**Nächste Schritte:**

CI/CD aufsetzen, weiter am Code programmieren, Design als Mockup festigen.

## Arbeitsjournal – Woche 2

Gruppe: Wallstreet-Clicker  
Tag: 27.04.2021

Tätigkeiten

* CI/CD aufgesetzt
* Clicker-Funktionalität implementiert
* Clicker-Anzeige implementiert
* Animation für Clicker erstellt
* Anzeige des aktuellen Punktestandes implementiert
* Anzeige von Items / Upgrades erstellt
* Test-Item für passives Mone/Second erstellt
* Mockup entworfen
* Worddokument für Dokumentation erstellt
* Grundformatierung vorgenommen
* Erste Ideen zu Items gemacht
* Scrum-Board aufgesetzt
* Arbeitsjournal Woche 2

Ablauf:

Heute haben wir angefangen zu programmieren und während dem auch an der Doku weitergearbeitet. Zu grossen Teilen haben Majd und Richard an der Dokumentation bzw. an den organisatorischen Aufgaben beteiligt. Während Luke und Samuel programmiert haben. Teilweise haben wir auch die Strategie des Pair-Programmings genutzt, um sauberen Code zu schreiben. Am Ende des Vormittages haben wir unsere Erkenntnisse des Tages ausgetauscht und diese im Arbeitsjournal dokumentiert.

Reflexion:

Auch heute war ein produktiver und effizienter Tag, allerdings ist uns beim Programmieren aufgefallen, dass wir die Items / Upgrades noch sauberer implementieren sollten. Zusätzlich dazu haben wir beim Mockup und beim Planen bzw. Organisieren ein wenig zu viel Zeit verschwendet. Auch ausgelöst dadurch, dass die Gruppe nicht vollständig anwesend war und dadurch teilweise von zu Hause aus mitgearbeitet werden musste.

Nächste Schritte:

* Alle Upgrades in einem File
* Weiter programmieren
* Erste Dokumentationsinhalte erarbeiten

## Arbeitsjournal – Woche 3

Gruppe: Wallstreet-Clicker  
Tag: 04.05.2021

**Tätigkeiten:**

* Alle Upgrades im upgrades.js File
* MpS wird nun auf Website angezeigt
* Funktion von passivem MpS komplett umschrieben
* Multiplikator für den aktiven Clicker als Item verfügbar
* FormatNumber hinzugefügt
* ButtonStatusDisplay Funktion hinzugefügt
* Styling leicht geändert
* Bugfixes
* Mockup überarbeitet
* Konzept der Item-Liste vollständig entworfen
* Einleitung und andere kleine Kapitel in Dokumentation geschrieben
* Arbeitsjournal Woche 3

**Ablauf:**

Heute konnten wir wieder sehr produktiv arbeiten. Bis zum jetzigen Stand läuft es rund und effizient. Wir hoffen, dass es auch dabeibleiben wird. Zusätzlich dazu haben wir heute die Liste für die Items fertiggestellt und haben damit einen guten Überblick dazu, welche Items am Ende im Spiel kaufbar sein werden. Am Ende haben wir wie jede Woche das Arbeitsjournal geschrieben.

**Reflexion:**

Wir sind auf Kurs und arbeiten mit guten Methoden und Techniken (Scrum, Pair-Programming etc.) wir denken, dass das Projekt zum aktuellen Stand sehr gut rauskommen kann und wird.

Nächste Schritte

* Weiter programmieren
* User-Stories definieren

## Arbeitsjournal – Woche 4

Gruppe: Wallstreet-Clicker  
Tag: 11.05.2021

**Tätigkeiten:**

* Brainstorming
* Erforschungssystem hinzugefügt
* Erforschungssystem mit Buttons bestückt
* Funktion zum Sperren eines Upgrades hinzugefügt
* Cost-Increase per buy hinzugefügt
* Style komplett überarbeitet
* Bugfixes & kleine Änderungen
* User-Stories für Dokumentation
* Testkonzept
* Arbeitsjournal Woche 4

**Ablauf:**

Zunächst haben wir uns heute zusammengetan und uns überlegt, wie wir unser bisheriges Spiel noch spannender gestallten könnten. Nach einer guten Einheit des Brainstormings haben wir uns dazu entschieden, das Erforschungssystem zu implementieren und zusätzlich das Design nochmal zu überarbeiten. Um in der Dokumentation ein vernünftiges Testkonzept zu haben, haben wir uns dafür entschieden, dass wir User-Stories aufstellen und die dabei zusätzlich aufgestellten Abnahmekriterien gleichzeitig dafür nutzen, um zu entscheiden, ob ein Testfall erfolgreich war oder nicht.

**Reflexion:**

Wir haben auch heute wieder produktiv gearbeitet. Wir sind auf Kurs und sind mit unseren Arbeitsmethoden sehr zufrieden. Auch wenn es bisher immer wieder vorkam, dass jemand von uns gefehlt hat, so hat diese Person immer zeitnah oder sonst kurzzeitig versetzt mitgearbeitet. Dies ist generell wichtig zu erwähnen. Das Arbeitsjournal bezieht sich immer auf die ganze Woche und nicht nur explizit auf den einen Arbeitstag, welchen wir im Unterricht hatten.

**Nächste Schritte:**

* Save Funktionalität implementieren
* User-Stories und Testkonzept fertigstellen

## Arbeitsjournal – Woche 5

Gruppe: Wallstreet-Clicker  
Tag: 18.05.2021

**Tätigkeiten:**

* Save / Load Game Funktionalität
* Nummer Formatierung verbessert
* User-Stories fertig gestellt
* Arbeitsjournal Woche 5

**Ablauf:**

Auch heute haben wir uns in unsere zwei Untergruppen eingeteilt und weiterprogrammiert, respektive an der Dokumentation gearbeitet. Beim Implementieren der Save & Load Funktionalität gab es einiges an Problemen, welche wir zunächst klären mussten und was viel Zeit gefressen hatte. Beim Dokumentieren funktionierte alles soweit ohne Probleme, aber sehr langsam.

**Reflexion:**

Das Programmierteam hatte heute sehr mit der Load / Save Funktionalität zu kämpfen, währenddessen beim Programmierer Team nicht so viel Motivation vorhanden war. Wir haben heute eher langsam und nicht so gut wie in den letzten Wochen gearbeitet und sollten dies definitiv wieder verbessern, vor allem da wir so kurz vorm Ende stehen.

**Nächste Schritte:**

* Achievements hinzufügen
* Safe & Load komplett ausbauen
* Testing

## Arbeitsjournal – Woche 6

Gruppe: Wallstreet-Clicker  
Tag: 25.05.2021

**Tätigkeiten:**

* Achievements Funktion
* Safe & Load verbessert
* Testing angefangen
* Formatierung Dokumentation verbessert

**Ablauf:**

Heute haben wir die Achievements Funktion komplett erschaffen und somit den Meilenstein erreicht, dass wir mehr oder weniger fertig sind. Dann haben wir die offizielle Phase des Testings eingeleitet (auch schon vorher haben wir immer mal wieder ein wenig getestet).

**Reflexion:**

Heute waren wir wieder produktiver als letztes Mal, allerdings sind wir noch nicht fertig mit dem Testing und müssen noch mehr Inhalte implementieren.

**Nächste Schritte:**

* Testing beenden
* Inhalte vollständig
* Dokumentation vollständig

## Arbeitsjournal – Woche 7

Gruppe: Wallstreet-Clicker  
Tag: 01.06.2021

Geplante Tätigkeiten:

* Noch in 7. Woche drin, schreiben AJ am Ende.

Ablauf

Reflexion

Nächste Schritte

# Reflexion

Das Projekt war für uns eine gute Vorbereitung für die auf uns zukommende IPA. Dazu müssen wir allerdings zugeben, dass wir das Konzept der Gruppenarbeit ein wenig ausgenutzt haben. Die Dokumentationsstarken Personen nur an der Dokumentation zu lassen und die im Coding starken Personen die komplette Verantwortung über den Code zu geben, war zwar eine sehr raffinierte Idee unsererseits, hat aber natürlich dafür gesorgt, dass nur Stärken weiter ausgebaut wurden und Schwächen nicht verbessert, werden konnten.

(Noch nicht komplett fertig logischerweise)

# Testverfahren und Testresultate

## Testverfahren

Aufgrund von mangelnder Zeit haben wir uns für ein einfaches Testverfahren entschieden: Whitebox-, sowie Blackbox-Testing. Dabei probiert der Programmierer selbst Fehler in seinem Code, welcher ihm logischerweise bekannt sein müsste, zu provozieren und zu finden (Whitebox) oder einer Person, welche den Code nicht kennt, wird die Aufgabe gegeben möglichst viele Fehler zu provozieren. (Blackbox)

## Testresultate

Für jede User-Story haben wir, teils auch bereits während der Entwicklung, getestet. Anbei eine Übersicht mit allen Testresultaten:

Resultate sind noch nicht komplett fertig, wird aber so aussehen:

Userstory X

|  |  |  |  |
| --- | --- | --- | --- |
| **Projektversion** | **Tester** | **Testart** | **Ergebnis** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |