

Modul Ük-335 Mobile-Applikation realisieren

Autor: Matvej Levantsou

Berufsfachschule: CsBe

Experte: Manuel Sollberger



21. Mai 2023

# Projektinformationen

|  |  |
| --- | --- |
| **Abteilung** | Informatik (Applikation) |
| **Status** | In Arbeit |
| **Ausgabedatum** | 02.06.2023 |
| **Dokument** | Projektdokumentation |
| **Projektvorehen** | HERMES 5.1 |
| **Projektname** | Modul ÜK-335 LB1 |
| **Autor** | Matvej Levantsou |
| **Auftraggeber** | Manuel Sollberger |

# Versionierung

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Name** | **Beschreibung** |
| 0.1 | 22.05.2023 | Matvej Levantsou | Dokument erstellen und vorbereiten. Arbeiten an der Initialisierungsphase. |
| 0.2 | 23.05.2023 | Matvej Levantsou | Arbeiten an der Konzeptphase. |
| 0.5 | 24.05.2023 | Matvej Levantsou | Abschluss der Konzeptphase. |
| 0.6 | 25.06.2023 | Matvej Levantsou |  |
| 0.7 | 26.06.2023 | Matvej Levantsou |  |

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **Abkürzung** | **Bedeutung** |
| Bibliothek | Eine Sammlung von Routinen oder Objekten, die in der Programmentwicklung verwendet werden. |
| Cooldown | Abkühlen |
| Clicker-Spiel / Clicker | Ein Spiel, dessen Hauptzweck es ist, auf den Bildschirm zu tippen, um ein Ergebnis zu erhalten. |
| Cross-Platform/-ing | Dies ist eine Anwendung, die auf verschiedenen Geräten funktionieren kann. |
| DMG | Damage = Schaden |
| DPS | (Damage pro second) Schaden pro Sekunde |
| Framework | Wie eine Bibliothek, bietet aber viel mehr Funktionalität. |
| GB | Gigabyte |
| GUI | (Graphical User Interface) Das System der Mittel für die Interaktion des Benutzers mit dem Programm. |
| Level | Die Stufe |
| MB | Megabyte |
| RAM | Random Access Memory |
| Tap Slayer | Dies ist der Name des Spiels / der Anwendung. |
| Touch-Screen | Touch-Steuerung |
| UML | Einheitliche Modellierungssprache. |

# Inhaltsverzeichnis

[1 Projektinformationen 1](#_Toc136258139)

[2 Versionierung 1](#_Toc136258140)

[3 Abkürzungsverzeichnis 1](#_Toc136258141)

[4 Inhaltsverzeichnis 2](#_Toc136258142)

[5 Abbildungsverzeichnis 2](#_Toc136258143)

[6 Tabellenverzeichnis 2](#_Toc136258144)

[7 Quellenverzeichnis 2](#_Toc136258145)

[8 Arbeitsjournal 3](#_Toc136258146)

[9 Initialisierung 5](#_Toc136258147)

[9.1 Projektidee 5](#_Toc136258148)

[9.2 Vollständige Beschreibung der Idee 5](#_Toc136258149)

[9.2.1 Clicker 5](#_Toc136258150)

[9.2.2 Stufen und Monster 5](#_Toc136258151)

[9.2.3 Upgrades und Economy 6](#_Toc136258152)

[9.2.4 Fähigkeiten 6](#_Toc136258153)

[9.2.5 Partnern 6](#_Toc136258154)

[9.3 Projektziele 7](#_Toc136258155)

[9.4 Projektmethode 7](#_Toc136258156)

[9.5 Ist/Soll Situation 7](#_Toc136258157)

[9.5.1 Ist Situation 7](#_Toc136258158)

[9.5.2 Soll Situation 7](#_Toc136258159)

[9.6 Anforderungsanalyse 7](#_Toc136258160)

[9.7 Organisation der Arbeitsergebnisse 8](#_Toc136258161)

[9.7.1 Versionierung 8](#_Toc136258162)

[9.7.2 Der Zugriff 8](#_Toc136258163)

[9.7.3 Tägliche Sicherung 8](#_Toc136258164)

[9.7.4 Die Wiederstellung 8](#_Toc136258165)

[10 Konzept 10](#_Toc136258166)

[10.1 Die Deployment-Ziele 10](#_Toc136258167)

[10.2 Storyboard 11](#_Toc136258168)

[10.2.1 Hauptmenu 11](#_Toc136258169)

[10.2.2 Hero Fenster 12](#_Toc136258170)

[10.2.3 Characters Fenster 13](#_Toc136258171)

[10.2.4 Icon’s 14](#_Toc136258172)

[10.3 Codingstyling 15](#_Toc136258173)

[10.4 Testkonzept 17](#_Toc136258174)

[10.4.1 Testumgebungen und Testmitteln 17](#_Toc136258175)

[10.4.2 Testmethoden und was getestet wird 17](#_Toc136258176)

[10.5 Testfälle 18](#_Toc136258177)

[10.5.1 Echtes Ergebnis 20](#_Toc136258178)

[10.6 Systemdesign 20](#_Toc136258179)

[10.6.1 Dateistruktur 20](#_Toc136258180)

[10.6.2 Klassendiagramm (UML) 21](#_Toc136258181)

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Quellenverzeichnis

* Hermes Vorlage (Schweizerische Eidgenossenschaft) [22.05.2023] - <https://www.hermes.admin.ch/de/projektmanagement/verstehen/ubersicht-hermes/methodenubersicht.html>
* Coding Conventions -IPA [25.05.2023] –
* Hauptfigur-Asset [25.05.2023] - <https://assetstore.unity.com/packages/2d/characters/martial-hero-170422>
* Hintergrund-Assets [25.05.2023] - <https://assetstore.unity.com/packages/2d/environments/nature-pixel-art-base-assets-free-151370>
* Monster-Assets [27.05.2023] - <https://assetstore.unity.com/packages/2d/characters/monsters-creatures-fantasy-167949> , <https://assetstore.unity.com/packages/2d/characters/hero-nad-opponents-animation-140776#content>

# Arbeitsjournal

Tabelle 1 Arbeitsjournal 22.05.2023

|  |  |  |
| --- | --- | --- |
| Tätigkeiten | Zeitaufwand (Std.) | |
| **SOLL** | **IST** |
| Einführung Projekt | 3 | 3 |
| Unity installieren und konfigurieren | 1 | 1 |
| Zeitplan erstellen | 2 | 2 |
| Dokumentation vorbereiten | 0.5 | 0.2 |
| Projektidee beschreiben | 0.5 | 0.2 |
| Arbeitsjournal erstellen | 0.2 | 0.2 |
| Projektanforderungen beschreiben | 0.3 | 0.3 |
| Projektziele festlegen | 0.2 | 0.2 |
| Führung des Arbeitsjournals | 0.3 | 0.3 |
| Organisation der Arbeitsergebnisse | 0 | 0.6 |
| Datensicherung | 0.2 | 0.2 |
| **Gesamte Tagesleistung** | 8.2 | 8 .2 |
| Probleme |  | |
| Obwohl ich mit Unity vertraut war, sind zusätzliche Kenntnisse erforderlich, um das Projekt zu realisieren. |  | |
| Hilfestellung |  | |
| Keine |  | |
| Reflexion |  | |
| Heute habe ich mich mit den Zielen des Projekts vertraut gemacht und sie analysiert, um einen Zeitplan zu erstellen. Die Vorbereitung des Dokuments und die Beschreibung der Idee dauerten weniger Zeit als geplant, daher konnte ich die Organisation der Arbeitsgebnisse noch umsetzen. | | |
| Nächste Schritte |  | |
| In die Phase der Konzepterstellung einsteigen. |  | |

Tabelle 2 Arbeitsjournal 23.05.2023

|  |  |  |
| --- | --- | --- |
| Tätigkeiten | Zeitaufwand (Std.) | |
| **SOLL** | **IST** |
| Ist / Soll Analyse | 0.3 | 0.3 |
| Die Deployment-Ziele definieren | 1 | 1 |
| Storyboard erstellen | 2 | 2 |
| Storyboard beschreiben | 0.5 | 0.5 |
| Führung des Arbeitsjournals | 0.3 | 0.3 |
| Organisation der Arbeitsergebnisse | 1 | 0 |
| Datensicherung | 0.2 | 0.2 |
| **Gesamte Tagesleistung** | 5.3 | 4.3 |
| Probleme |  | |
| Keine |  | |
| Hilfestellung |  | |
| Keine |  | |
| Reflexion |  | |
| Die Arbeit an der Konzeptphase wurde erfolgreich begonnen. Dank der effektiven Arbeit des letzten Tages ist es gelungen, den geplanten Arbeitsaufwand schneller als geplant zu erledigen. | | |
| Nächste Schritte |  | |
| Die Konzeptphase beenden und mit der Arbeit an der Implementierungsphase beginnen. |  | |

Tabelle 3 Arbeitsjournal 25.05.2023

|  |  |  |
| --- | --- | --- |
| Tätigkeiten | Zeitaufwand (Std.) | |
| **SOLL** | **IST** |
| Codingstyle beschreiben | 1 | 1 |
| Testkonzept erstellen | 2 | 1.5 |
| Testfälle erstellen | 2 | 2 |
| Systemdesign erstellen | 2 | 2 |
| Projekt erstellen und konfigurieren | 0.5 | 0.5 |
| Ressourcen suchen | 1 | 1 |
| Führung des Arbeitsjournals | 0.3 | 0.3 |
| Datensicherung | 0.2 | 0.2 |
| **Gesamte Tagesleistung** | 9 | 8.5 |
| Probleme |  | |
| Keine |  | |
| Hilfestellung |  | |
| Keine |  | |
| Reflexion |  | |
| Für heute wurde viel Arbeit geleistet. Die Phase des Konzepts ist abgeschlossen, wird aber wahrscheinlich später Nacharbeiten erfordern. | | |
| Nächste Schritte |  | |
| Arbeiten an der Umsetzung des Projekts |  | |

Tabelle 4 Arbeitsjournal 26.05.2023

|  |  |  |
| --- | --- | --- |
| Tätigkeiten | Zeitaufwand (Std.) | |
| **SOLL** | **IST** |
| Umsetzung des Projekts | 4 | 4 |
| Führung des Arbeitsjournals | 0.3 | 0.3 |
| Datensicherung | 0.2 | 0.2 |
| **Gesamte Tagesleistung** | 4.5 | 4.5 |
| Probleme |  | |
| Keine |  | |
| Hilfestellung |  | |
| Keine |  | |
| Reflexion |  | |
| Die Arbeit an der Umsetzung des Projekts wurde begonnen. Das Projekt wurde erstellt und konfiguriert, um mit mobilen Geräten auf dem Android-System zu arbeiten. Die GUI der Anwendung und die Animation beim Anklicken wurden implementiert. | | |
| Nächste Schritte |  | |
| Umsetzung der Angriffsfunktionen und des Auftretens von Monstern. |  | |

# Initialisierung

## Projektidee

Das Spiel wird als **Tap Slayer** bezeichnet, da der Spieler verschiedene Kreaturen zerstören muss, indem er auf sie tippt(tap). Wenn Benutzer auf den Bildschirm tippen, erhalten die Monster etwas Schaden und sterben nach dem Verlust aller Gesundheit, indem sie dem Spieler eine Währung geben, um den Charakter zu verbessern. Das Spiel ist in Stufen unterteilt, in denen jeweils 10 Monster enthalten sind und das letzte Monster der Stufe stärker ist als alle Vorgänger. Nachdem das letzte Monster in der Stufe getötet wurde, geht der Spieler mit stärkeren Monstern in die nächste Stufe. Für die erhaltene Währung kann der Spieler seinen Charakter auf verschiedene Arten verbessern (z. B. indem er seinen Schaden durch Tippen erhöht). Der Spieler hat auch die Möglichkeit, Partner zu kaufen, die Monstern passiv Schaden zufügen. Ausserdem wird dem Spieler mehrere Fähigkeiten zur Verfügung stehen, die ihn auf verschiedene Arten verstärken werden, aber sie werden für eine Weile aufgeladen. Der Spieler erhält 3 Fähigkeiten, die er verbessern kann. Die erste Fähigkeit verursacht viel Schaden (10xDMG). Die zweite erhöht den Schaden beim tippen (2xTip DMG). Der dritte erhöht den Schaden von Partnern(2xPartnern DMG). Auch um die Progression der Spieler zu speichern, wird der Fortschritt automatisch nach jedem getöteten Monster gespeichert. gespeichert. Abhängig von der Geschwindigkeit der Entwicklung des Spiels ist es möglich, verschiedene neue Features einzugeben. Das Spiel ist nur in vertikaler Position verfügbar, da es keinen Sinn macht, ein Clicker-Spiel horizontal zu machen.

Theoretisch ist es möglich, ein Fähigkeitsbaum und Ausrüstung für den Charakter zu implementieren. Das Ziel des Spiels ist es, die höchstmögliche Stufe zu erreichen.

## Vollständige Beschreibung der Idee

### Clicker

Das Hauptziel des Spielers ist es, auf den Bildschirm zu klicken/tippen. Durch Anklicken des Bildschirms fügt der Spieler Feinden Schaden zu, tötet sie dadurch und verdient Währung. Die Währung kann der Spieler für verschiedene Verbesserungen verwenden, da die Gegner mit jeder Stufe stärker und stärker werden. Es besteht auch die Möglichkeit, passiven Schaden zuzufügen, nachdem Partner verbessert wurden.

Formeln:

1. Start Tap DMG = 5
2. Start Partnern DMG pro Sekunde = 0

### Stufen und Monster

Das Spiel besteht aus Stufen (beginnend mit dem ersten), von denen jeder 10 Gegner hat. Jeder Gegner hat eine bestimmte Menge an Gesundheit. Wenn die Gesundheit auf Null gefallen ist, stirbt der Gegner und ein neuer Gegner erscheint. Dies wird wiederholt, bis der zehnte Gegner (der Boss) erscheint. Der Boss hat eine erhöhte Menge an Gesundheit. Nur Bosse haben ein grosses Modell (wie im Storyboard), die anderen Gegner haben ein Standardmodell. Der Boss gibt mehr Währung als andere Gegner, wenn er stirbt. Mit jeder Stufe werden die Gegner stärker, geben aber auch mehr Währung.

Formeln:

1. Start Komplexitätsmodifikator = 0%
2. Komplexitätsmodifikator pro eine Stufe = 10%
3. Startgesundheit der Gegner = 50
4. Gesundheitsmodifikator pro eine Stufe = 25%
5. Gesundheitsmodifikator des Boss = 10X
6. Währungsmodifikator des Boss = 10X
7. Währungsmodifikator pro eine Stufe = 5%
8. Gold für das Töten des Feindes = 10

### Upgrades und Economy

Um stärkere Gegner zu besiegen, muss sich der Spieler verbessern. Für Verbesserungen benötigt der Spieler die Währung, die er erhält, wenn er Gegner tötet. Der Spieler kann den Tapsschaden, den Fähigkeitsschaden und den Partnerschaden verbessern.

Formeln:

1. Preis für die Verbesserung des Tapschadens = 10
2. Preis für die Verbesserung der ersten Fähigkeit = 30
3. Preis für die Verbesserung der zweiten Fähigkeit = 50
4. Preis für die Verbesserung der dritten Fähigkeit = 70
5. Kaufpreis des ersten Partners = 20
6. Kaufpreis des zweiten Partners = 50
7. Kaufpreis des dritten Partners = 100
8. Kaufpreis des vierten Partners = 200
9. Kaufpreis des fünften Partners = 500
10. Preismodifikator für alle Verbesserungen= 20%
11. Schadensmodifikator pro Tap = +5 / und 10% mehr beim nächsten Mal
12. Modifikator der ersten Fähigkeit = +2.0X
13. Modifikator der zweiten Fähigkeit = +0.5X
14. Modifikator der dritten Fähigkeit = +0.5X
15. Schadensmodifikator des ersten Partners = +3
16. Schadensmodifikator des zweiten Partners = +5
17. Schadensmodifikator des dritten Partners = +10
18. Schadensmodifikator des vierten Partners = +20
19. Schadensmodifikator des fünften Partners = +35

### Fähigkeiten

Der Spieler kann verschiedene Fähigkeiten mit unterschiedlichen Effekten verwenden. Die erste Fähigkeit (Heavy Strike) verursacht 10X des Schadens des Spielers und hat eine Cooldown von 10 Sekunden. Die zweite Fähigkeit (Enhancement) erhöht den Schaden des Spielers um 20 Sekunden um 3X und hat eine Cooldown von 60 Sekunden. Die dritte Fähigkeit (War cry) erhöht den Schaden aller Partner um 20 Sekunden um 3X und hat eine Cooldown von 60 Sekunden.

### Partnern

Partner verursachen einmal pro Sekunde passiv Schaden. Dieser Schaden erhöht sich mit der Stufe der Partner und mit dem Einsatz von War cry.

## Projektziele

Das Ziel des Projekts ist, dass eine Multiplattform-Anwendung mit der Unity-Engine implementiert werden muss. Da unter den Bedingungen der Aufgabe keine spezifischen Anforderungen an das Thema und die Komplexität der Aufgabe erforderlich sind, wurde ein eigenes Thema ausgewählt. Unity ermöglicht zwar die Erstellung verschiedener Anwendungen, ist aber auf die Entwicklung von Videospielen spezialisiert. Deshalb wurde die Entscheidung getroffen, ein Videospiel zu erstellen.

Da ich mit der Unity-Engine wenig vertraut bin, wurde die Entscheidung getroffen, ein Clicker-Spiel zu entwickeln, da es eines der einfachsten Genres ist, aber dennoch faszinierend ist. Der Zweck solcher Spiele ist sehr einfach, der Spieler muss nur auf die Maus oder den Bildschirm des Telefons klicken und etwas Ergebnis erhalten.

Während der Arbeit am Projekt müssen die folgenden Ziele erreicht werden:

* Die Anwendung sollte auf verschiedenen Plattformen gleichermassen gut funktionieren.
* Das Genre für das Spiel ist ein Clicker-Spiel.
* Bugs und Fehler werden minimiert.
* Die Benutzeroberfläche ist genug benutzerfreundlich.
* Der Spieler hat die Möglichkeit, sich selbst zu verbessern.
* Die Entwicklung von Multiplattform-Anwendungen mit Unity zu lernen.

## Projektmethode

Ich verwende als Projektmethode HERMES 5.1.

HERMES ist die Projektmanagementmethode für Projekte im Bereich der Informatik, der Entwicklung von Dienstleistungen und Produkten sowie der Anpassung der Geschäftsorganisation. HERMES unterstützt die Steuerung, Führung und Ausführung von Projekten verschiedener Charakteristiken und Komplexität. HERMES hat eine klare, einfach verständliche Methodenstruktur, ist modular aufgebaut und erweiterbar. Nachfolgend sind die wesentlichen Methodenelemente und ihr Zusammenspiel beschrieben.

## Ist/Soll Situation

### Ist Situation

Eine Multiplattform-Anwendung muss mit Unity realisiert werden. Um das zu tun, muss Unity installiert werden, einige Guides gelesen werden und das Projekt mithilfe von Ressourcen aus dem Unity Asset Store implementiert wird.

### Soll Situation

Alle Projektziele wurden erfolgreich erreicht und die Anforderungen wurden erfüllt.

## Anforderungsanalyse

Die Anforderungen an die Applikation sind folgende:

* Die Anwendung sollte auf verschiedenen Plattformen gleichermassen gut funktionieren (Cross-Platforming)
* Alle Ressourcen müssen aus dem Unity asset Store stammen oder selbst erstellt werden.
* Die Unity-Engine muss verwendet werden, um die Anwendung zu entwickeln.
* Die Anwendung muss mit verschiedenen Tests getestet werden.
* Die GUI sollte benutzerfreundlich und ergonomisch sein.
* Alle Codestyle Richtlinien werden eingehalten.
* Der Standard EN-ISO 9241-110 wurde vollständig eingehalten.
* Das Design der grafischen Benutzeroberfläche muss responsable sein.
* Safe Area verhindert nicht die Verwendung der App.
* Die Daten werden vollständig und korrekt persistiert.

## Organisation der Arbeitsergebnisse

### Versionierung

Git Hub wurde als Versionsverwaltungslösung ausgewählt, weil es eine einfache und zuverlässige Methode zum Speichern und Wiederherstellen von Daten ist.

A picture containing text, screenshot, line, font

Description automatically generated

Mit der **<>**-Taste kann eine beliebige Version des Projekts abgerufen werden.

### Der Zugriff

Der Zugriff auf die Daten ist von jedem autorisierten Gerät aus möglich, sofern das Internet verfügbar ist. Das ist dank Git Hub möglich.

Link auf das Repository - <https://github.com/mat414748/MobileProject>

### Tägliche Sicherung

Jeder Commit behält die aktuelle Version des Projekts und der Dokumentation bei.

A picture containing text, line, font, screenshot

Description automatically generated

### Die Wiederstellung

Im Falle eines Datenverlustes können diese einfach wiederherstellen werden, indem eine Kopie aus dem Repository herunterladen wird.

A screenshot of a computer

Description automatically generated with medium confidence

# Konzept

## Die Deployment-Ziele

Im Zusammenhang mit dem Projektthema (Entwicklung von Cross-Platform-Anwendungen) muss die Anwendung auf verschiedenen Geräten und Systemen verfügbar sein.

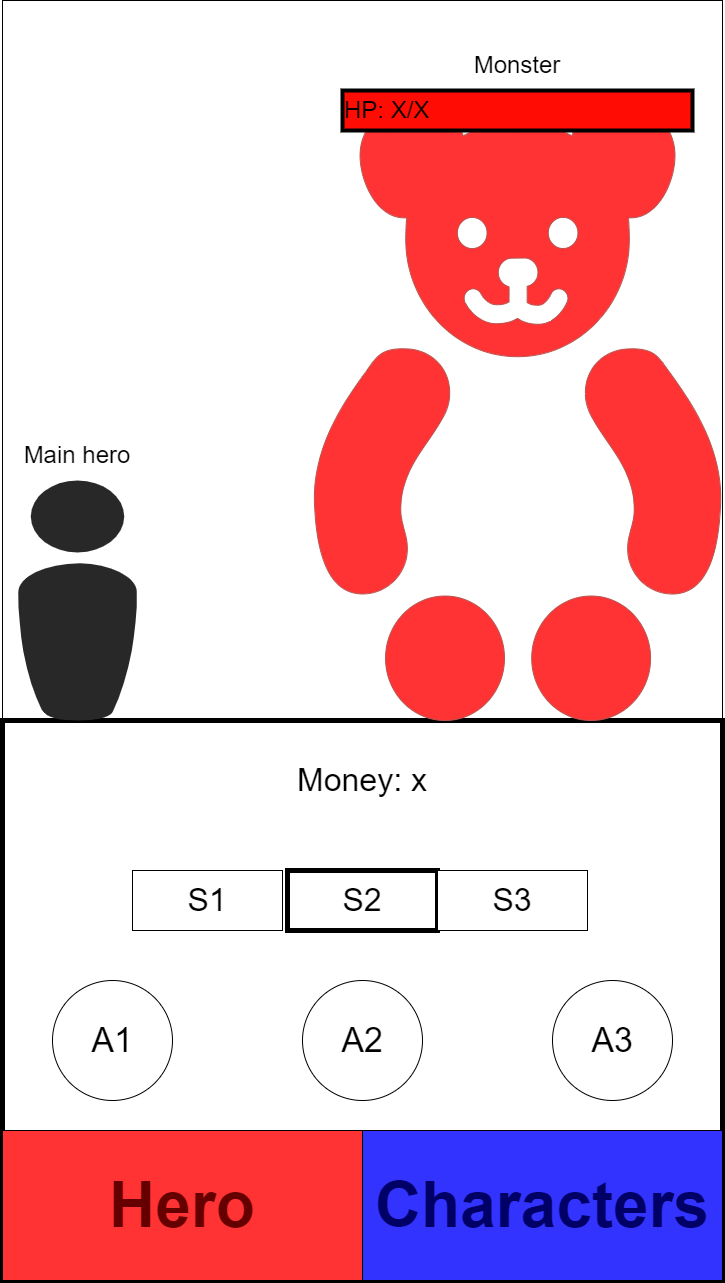
Geräte, die die Anwendung genau unterstützen sollten, sind Telefone, Laptops und Computer. Theoretisch kann die App auf fast jedem Gerät funktionieren, das Unity unterstützt, aber das ist nicht notwendig. Die Mindestanforderungen an das Gerät sind nicht hoch: 1-2 GB RAM, etwa 100 MB Speicher und Touch-Screen (nur für Telefone).

Systemen, die die Anwendung genau unterstützen sollten, sind Windows und Android. Optional kann die App auf IOS und Linux Systemen funktionieren. Die Android-Version sollte mindestens 8.0.0 und Windows-Version muss mindestens 7 sein.

## Storyboard

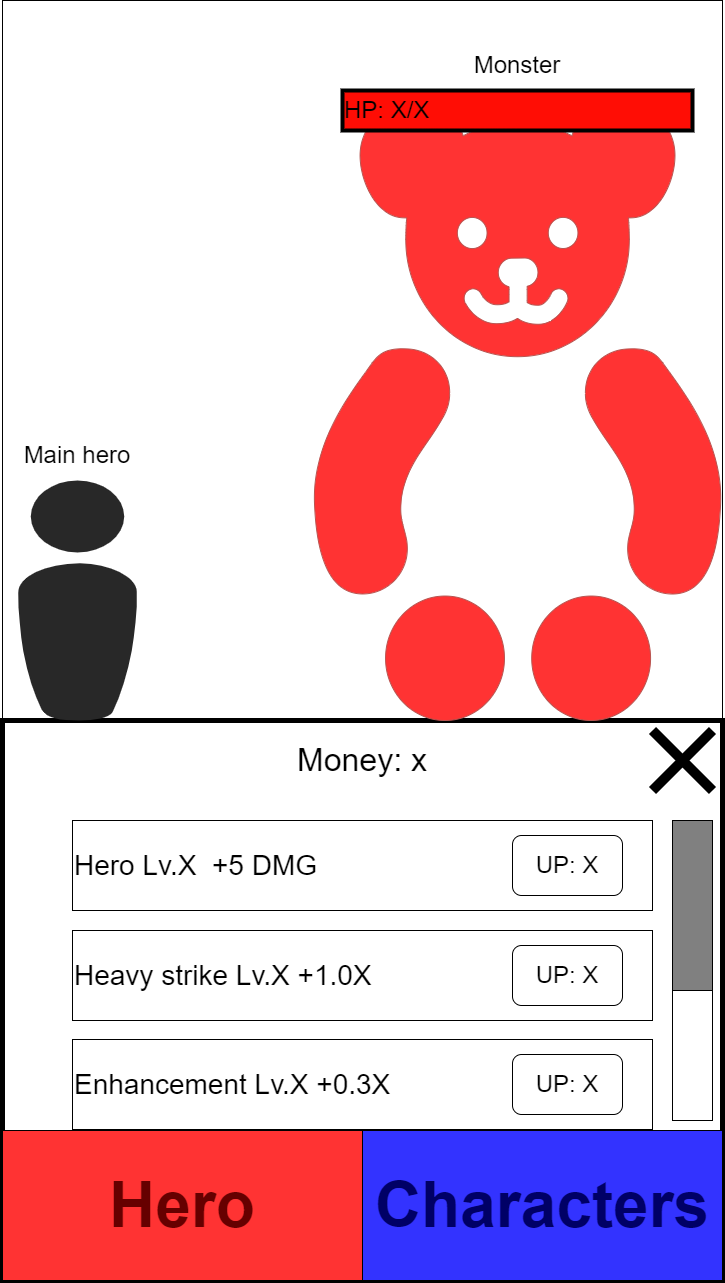
### Hauptmenu

1. Monster Name – Der Name des Gegners und sein Modell.
2. HP: X/X – Ein Streifen, der die aktuelle Gesundheit des Monsters anzeigt.
3. Main Hero –Das Modell der Hauptfigur.
4. Monster X/10 – Die Anzahl der Monster, die Spieler gewinnen müssen, um zur nächsten Stufe zu gelangen
5. S1, S2, S3 – Stufen (Z.B. Die zweite Stufe ist jetzt aktiv.)
6. Money: X – Die Menge an Währung, die dem Benutzer derzeit zur Verfügung steht.
7. A1, A2, A3 – Icon’s für die Verwendung von Fähigkeiten.
8. Hero – Wenn ein Spieler darauf klickt, wird das Menü aufgerufen, um den Hauptfigur und seine Fähigkeiten zu verbessern.
9. Characters – Wenn ein Spieler darauf klickt, wird das Menü aufgerufen, um den Partnern zu verbessern.



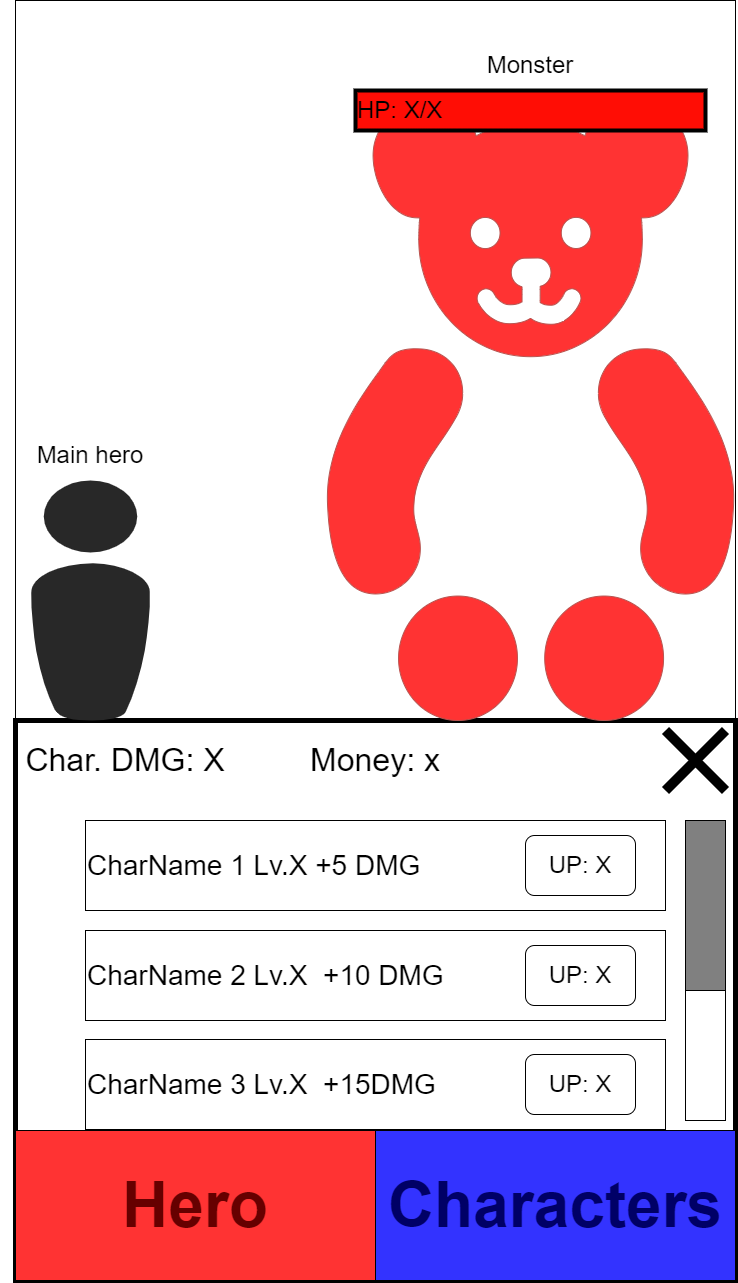
### Hero Fenster

1. Tap DMG: X – Schaden beim Tippen.
2. Hero, Heavy Strike, Enhancement, War Cry – Dies ist der Name der Fähigkeiten, die der Spieler nutzen und verbessern kann.
3. LV. X – Aktuelle Fähigkeiten- und Hauptfigur-Level.
4. +5 DMG, +1.0X – Zeigt an, wie stark die Fähigkeit oder der Hauptfigur wird, wenn er um Stufe 1 verbessert wird.
5. UP: X – Zeigt an, wie viel Währung zur Verbesserung benötigt wird
6. **X** - Schliesst das Fenster



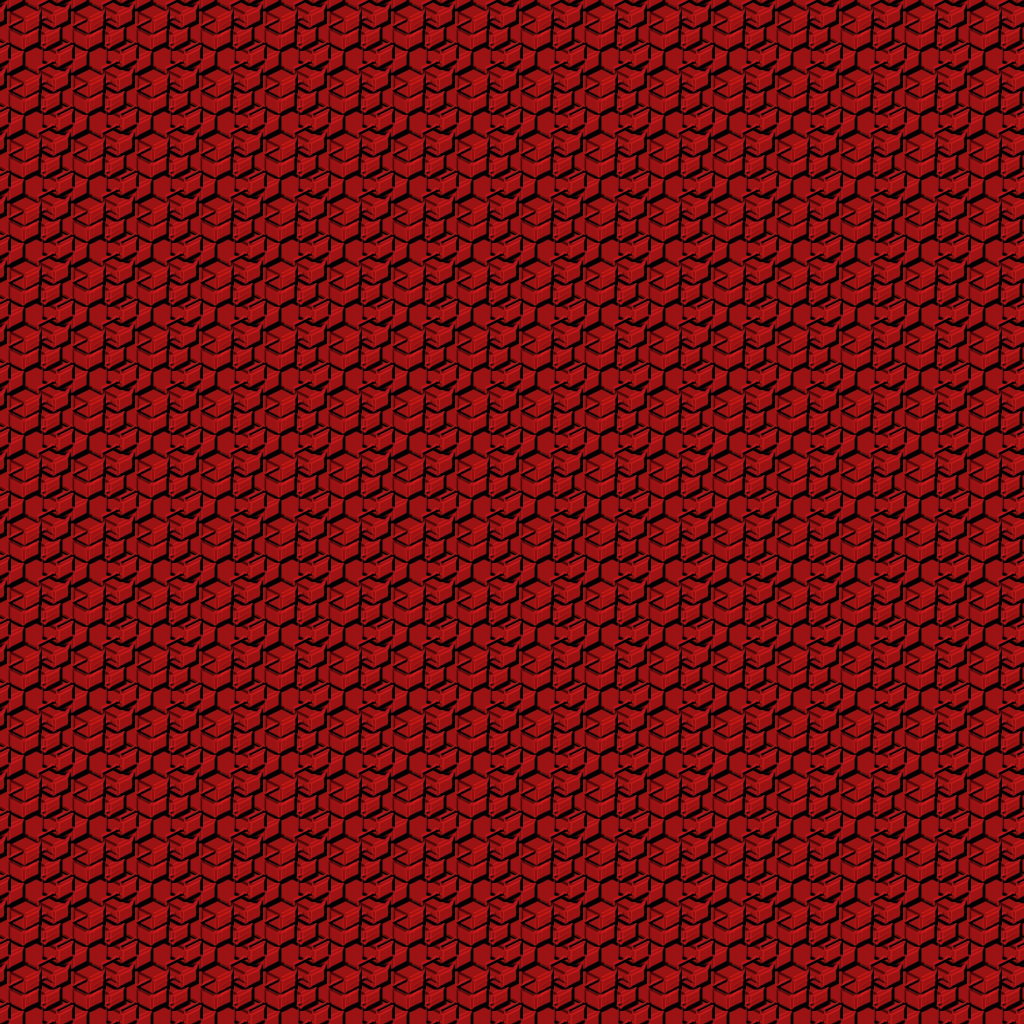
### Characters Fenster

1. Char. DMG – Der Schaden, den Partner in einer Sekunde verursachen.
2. CharName 1,2,3 – Namen der Partner

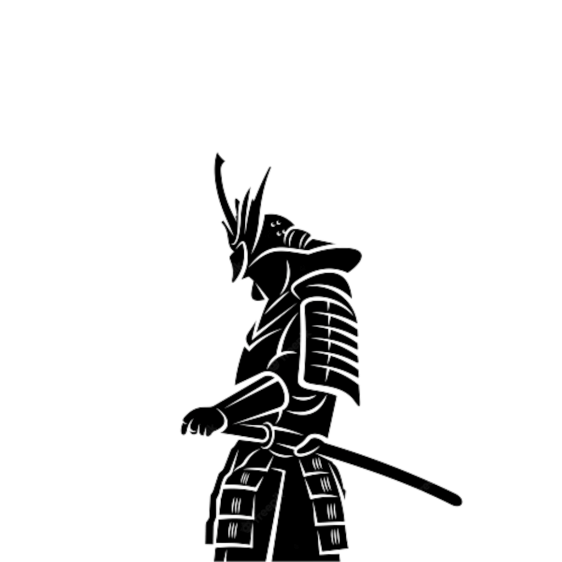


### Icon’s

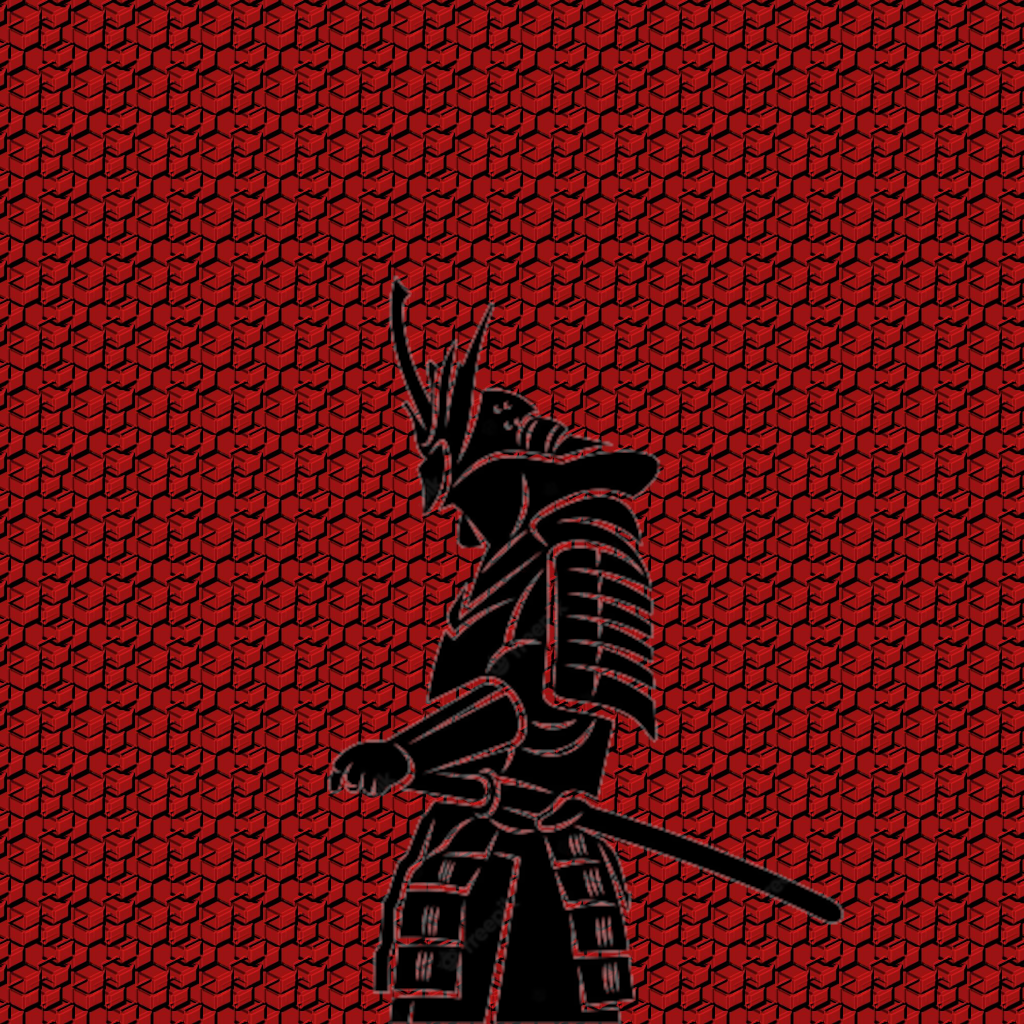
Background:



Font:



Vollbild:



## Codingstyling

Für korrekten und lesbaren Code wurden Regeln erstellt, die den Schreibstil des Codes regeln. Folgende Regeln werden beim Schreiben von Code beachten:

1. Sonderzeichen sowie Umlaute (äöü) und Zahlen dürfen in Bezeichnungen nicht verwendet werden
2. Kommentare nur in englischer Sprache
3. Klassen/Interfaces:
   1. Sind verständlich gewählt und entsprechen ihrer Bedeutung
   2. Sind PascalCase (auch Upper-CamelCase bezeichnet) bsp. MyClass
   3. Werden Interfaces verwendet, so werdenKlassenals MyClassImpl bezeichnet
4. Properties:
   1. Sind verständlich gewählt und entsprechen ihrer Bedeutung
   2. Sind PascalCase (auch Upper-CamelCase bezeichnet) bsp. MyProperty
5. Methoden/Funktionen:
   1. Sind verständlich gewählt und entsprechen ihrer Bedeutung
   2. Sind CamelCase (auch Lower-CamelCase bezeichnet).
   3. Beginnen immer mit einem Verb
   4. Erfüllen genau eine Aufgabe(Single Responsibility)und habenohne Begründungnie mehr als 20 Zeilen Code
   5. Methoden werden so erstellt, dass Sie mit einem Unit-Test getestet werden können
6. Variablen:
   1. Sind verständlich gewählt und entsprechen ihrer Bedeutung
   2. Variablen mit einem Zeichen sind nur Zählvariablen vorbehalten bsp. i = 3
   3. Keine Nummerierungen (numerisch und alphanumerisch wie v1 oder vEins)
   4. Sind CamelCase (auch Lower-CamelCase bezeichnet) bsp. myVar.
   5. Konstanten sind UPPERCASE. Besteht die Konstante aus mehreren Begriffen, so sind die Begriffe mit einem Underline „\_“ zu trennen
7. Kommentare:
   1. Klassen/Interfaces:
      1. Jede Klasse/Interface beinhaltet einen Klassenkommentar mit dem Zweck der Klasse und dem Namen des Autors
   2. Methoden:
      1. Jede öffentliche Methode, mit Ausnahme von Getter-und Setter-Methoden werden kommentiert
      2. Werden Klassen abgeleitet oder implementieren Interfaces, so reicht es, wenn die Methoden der abgeleiteten Klasse oder implementierten Interfaces kommentiert sind
      3. Der Zweck der Methode, sowie Parameterund Rückgabe sind kommentiert
   3. Variablen:
      1. Konstanten und Enums sind kommentiert und beschreiben ihren Zweck
   4. Allgemeine Code Kommentare:
      1. Inline Kommentare sind nur im Ausnahmefall zu verwenden(Bsp. um zu beschreiben, warum etwas so gemacht wurde)
8. Verwendung von Entwurfsmustern/Prinzipien:
   1. DRY ist zwingend einzuhalten (Dont Repeat Yourself, keine Redundanzen)
   2. KISS ist zwingend einzuhalten (Keep it small and simple, nur nötiger Code, kein Code auf Vorrat)
   3. Werden Entwurfsmuster umgesetzt, so muss man sich an die Bezeichnungen der jeweiligen Muster halten

## Testkonzept

### Testumgebungen und Testmitteln

Das folgende System wurde zum Testen der Anwendung verwendet:

1. Computer:
   1. Betriebssysteme – Windows 11 22H2
   2. CPU – i7-1165G7
   3. RAM – 16GB
   4. Grafikkarte – NVIDIA GeForce MX330
   5. SSD – 512GB
2. Telefon:
   1. Betriebssysteme – Android 8.1.0
   2. Modell – Samsung Galaxy J7

Die Anwendung benötigt keinen Internetzugang, um ordnungsgemäss zu funktionieren. Alle Tests werden auf realen und nicht virtuellen Maschinen durchgeführt.

Für Unit-Tests wird die xunit-Bibliothek verwendet und für alle anderen Tests wird das Unity Test Framework verwendet.

### Testmethoden und was getestet wird

Die folgenden Arten von Tests werden verwendet, um das Programm vollständig zu testen:

1. Unit Testen. Wird von den Entwicklern selbst durchgeführt, um eine Funktion oder Methode zu testen.
2. Black Box (Usability) Testen. Tests, die ausschliesslich auf der externen Schnittstelle des zu testenden Systems basieren. Für solche Tests ist es am besten, Personen zu verwenden, die mit der App nicht vertraut sind.
3. Funktionale Testen. Es wird überprüft, ob die Funktionalität der Anwendung korrekt funktioniert.

Die folgenden Arten von Tests werden nicht angewendet, verwendet, um das Programm zu testen:

1. Integration Testen. Da die Anwendung ein einheitliches Ganzes (Build) ist, ist ein solcher Test einfach nicht möglich, da es keinen zweiten Programmteil gibt.
2. System Testen. Die Anwendung ist nicht sehr anspruchsvoll für die Ressourcen des Systems und funktioniert auf allen Systemen, die für die Mindestanforderungen geeignet sind.

## Testfälle

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-001 | Starten der Anwendung |
| **Beschreibung** | Die Anwendung wird korrekt und ohne Probleme gestartet. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer 2. Ausführen der Anwendung auf einem Telefon | |
| **Erwartetes Ergebnis** | Die Anwendung wird ohne Probleme auf dem Computer und Telefon gestartet und ausgeführt. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-002 | Tippen auf den Bildschirm |
| **Beschreibung** | Das Tippen auf den Bildschirm funktioniert und führt zu einem Ergebnis. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer 2. Klicken Sie mit der linken Maustaste auf den Bildschirm. 3. Ausführen der Anwendung auf einem Telefon 4. Tippen Sie auf den Bildschirm. | |
| **Erwartetes Ergebnis** | Wenn ein Spieler auf den Bildschirm tippen, werden die Monster beschädigt. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-003 | Stufen wechseln |
| **Beschreibung** | Nach der Zerstörung von 10 Monstern ist die nächste Stufe | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Die Stufen ändern sich korrekt. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-004 | Fähigkeiten einsetzen |
| **Beschreibung** | Der Benutzer kann alle Fähigkeiten verwenden. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Alle Fähigkeiten können verwendet werden und ihre Wirkung ist korrekt | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-005 | Hauptfigur und Fähigkeiten verbessern |
| **Beschreibung** | Der Benutzer kann alle Fähigkeiten und Tippen-DMG verbessern. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Die Verbesserungen des Hauptfigur und der Fähigkeiten wirken sich wirklich auf ihre Wirksamkeit aus. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-006 | Partnern verbessern |
| **Beschreibung** | Der Benutzer kann alle Partnern verbessern. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Partnerverbesserungen erhöhen ihre DPS. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-007 | DMG-Ziffern sind korrekt |
| **Beschreibung** | Hauptfigur- und Partner-DMG werden korrekt gezeigt. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Der gesamte Schaden wird gezählt und korrekt angezeigt | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-008 | Währung |
| **Beschreibung** | Der Spieler kann Geld verdienen und ausgeben. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Der Spieler erhält eine Währung für das Töten von Monstern und kann sie ausgeben. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-009 | Komplexität |
| **Beschreibung** | Die Komplexität des Spiels entspricht dem Konzept und ist nicht übermässig. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Die Komplexität erhöht sich allmählich, wird aber nicht überflüssig. | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-010 | EN-ISO 9241-110 |
| **Beschreibung** | Der Standard EN-ISO 9241-110 wurde vollständig eingehalten. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Das Programm entspricht dem Standard und ist angenehm zu bedienen | |

|  |  |  |
| --- | --- | --- |
| **ID / Bezeichnung** | T-011 | Safe area |
| **Beschreibung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testvoraussetzung** | Der Computer und das Telefon entsprechen den Mindestanforderungen. | |
| **Testschritte** | 1. Ausführen der Anwendung auf einem Computer | |
| **Erwartetes Ergebnis** | Die Verwendung des Programms ist mit Safe Area in keiner Weise schwierig. | |

### Echtes Ergebnis

|  |  |
| --- | --- |
| Test № | Ergebnis |
| 001 | Wie erwartet |
| 002 | Wie erwartet |
| 003 | Wie erwartet |
| 004 | Wie erwartet |
| 005 | Wie erwartet |
| 006 | Wie erwartet |
| 007 | Wie erwartet |
| 008 | Wie erwartet |
| 009 | Wie erwartet |
| 010 | Wie erwartet |
| 011 | Wie erwartet |

## Systemdesign

### Dateistruktur

* Assets - Der Hauptordner enthält alle notwendigen Komponenten für die Entwicklung.
  + Adaptive Performance - Automatisch erstellte Datei.
  + Editor - Automatisch erstellte Datei.
  + Plugins - Automatisch erstellte Datei.
  + Icons – Speichert Icon’s (Drei weil Android) für die Anwendung.
  + Prefabs - Speichert alle Objektvorlagen.
  + Scenes – Speichert alle Scenes (Aber eigentlich wird nur eine Szene gebraucht)
  + Scripts - Speichert alle Skripts
    - UI - Speichert Skripts, um Probleme mit Safe Area zu vermeiden
  + TextMesh Pro - Wird für die Arbeit mit Textfeldern benötigt.
  + External assets - Ordner zum Speichern der heruntergeladenen Assets.
    - Martial Hero - Asset und Animationen des Protagonisten
    - Nature\_pixel\_art\_assets – Assets für die Hintergrund.
    - Monster Pack 1 – Monster-Assets-Set 1
    - Monster Pack 2 – Monster-Assets-Set 2

### Klassendiagramm (UML)