Ein Bild, das Text, Logo enthält.

Automatisch generierte Beschreibung

**Autoren:** Sandro Gerber

**Dozent:** Christian Klauenbösch

**Veröffentlichung:** 20.06.2024

Eternal Depths

Mobile Web RPG Game

Inhaltsverzeichnis

[1 Projektidee und Geschäftsregeln 3](#_Toc169815567)

[1.1 Spielregeln 3](#_Toc169815568)

[1.2 Abgrenzung 3](#_Toc169815569)

[2 Technologie Stack 4](#_Toc169815570)

[3 Anforderungen 5](#_Toc169815571)

[3.1 Funktionale Anforderungen 5](#_Toc169815572)

[3.2 Nicht-funktionale Anforderungen 10](#_Toc169815573)

[4 Struktur und Planung 11](#_Toc169815574)

[4.1 Wireframe 11](#_Toc169815575)

[4.2 Spielzyklus 13](#_Toc169815576)

[4.3 Modularisierung 14](#_Toc169815577)

[4.3.1 Atomic Design 14](#_Toc169815578)

[4.4 Route Guard 15](#_Toc169815579)

[4.5 Backend Konfiguration 15](#_Toc169815580)

[5 Abschlussbericht 16](#_Toc169815581)

[5.1 Erfüllte Anforderungen 16](#_Toc169815582)

[5.2 Zukunftsblick 16](#_Toc169815583)

[5.3 Reflexion 17](#_Toc169815584)

[6 Weitere Dokumentations Resourcen 18](#_Toc169815585)

# Projektidee und Geschäftsregeln

Das Projekt umfasst die Entwicklung eines web-basierten, mobilen RPG-Spiels, das rundenbasierte Kämpfe in einem zufällig generierten Dungeon-Setting bietet. Spieler beginnen auf Level 1 des Dungeons und kämpfen gegen Gegner, wobei sie nach jedem Kampf zwischen HP-Regeneration oder einem Gegenstand wählen können. Spieler können bis zu drei spezielle Attacken ausrüsten, müssen aber bei der Erlernung neuer Attacken ggf. eine bestehende ersetzen. Dungeons und Gegner sind zufällig generiert, und die Gegner-Stats skalieren mit dem Fortschritt des Spielers. Ein Leaderboard zeigt die Spieler, die am weitesten gekommen sind.

## Spielregeln

* Rundenbasierte Kämpfe.
* Zufällig generierte Gegner.
* Auswahl zwischen HP-Regeneration oder Gegenstand nach jedem Kampf.
* Maximal drei spezielle Attacken pro Spieler.
* Level-Up ermöglicht Attributserhöhung oder Erlernen neuer Attacken.
* Gegner-Stats skalieren mit Spielerfortschritt.
* Alle 5 Begegnungen erscheint ein stärkerer Gegner.
* Dungeons können nur nach einem Kampf verlassen werden.
* Ausrüstung ohne Inventar; direkter Ersatz bei Neufindung.
* Charakterattribute
  + Leben: Maximal Anzahl Lebenspunkte
  + Angriffswert: Schadenspunkte bei Angriff
  + Schnelligkeit: Bestimmung des ersten Angriffs im Kampf
  + Präzision: Möglichkeit für kritischer Treffer
* Leaderboard für erfolgreiches Verlassen des Dungeons

## Abgrenzung

* Accessibility: Die Applikation muss nicht Barierefrei aufgebaut werden (WCAG 2.1 Richtilinen müssen nicht erfüllt sein.)
* Keine umfangreiche Storyline: Um den Umfang und die Komplexität zu begrenzen, wird das Spiel keine tiefgreifende Geschichte haben.
* Keine Multiplayer-Funktionen: Das Spiel ist als Single-Player-Erlebnis konzipiert.
* Responsivnes: Das Spiel wird Mobile-First Designed.
* Mehsprachigkeit: Das Spiel ist nur in Englisch Verfügbar.
* Spielschwierigkeit: Es wird kein Spielbalancing gemacht.

# Technologie Stack

|  |  |
| --- | --- |
| Technologie | Beschreibung |
| HTML5 | Markup-Sprache zur Strukturierung und Darstellung von Inhalten im Web. |
| CSS3 | Stylesheet-Sprache zur Beschreibung des Aussehens und der Formatierung eines in HTML geschriebenen Dokuments. |
| React JS | JavaScript-Library zur Erstellung von Benutzeroberflächen. |
| Redux | JavaScript-Library für State Management im Frontend. |
| Jest | JavaScript-Library für Unit Tests. |
| Cypress | Library für automatisierte E2E Tests. |
| Supabase | Backend Komponente für die Speicherung von persistenten Daten. |
| RPGUI | Nur das CSS der UI Library für Retro RPG UI Elemente. <https://github.com/RonenNess/RPGUI/tree/master?tab=readme-ov-file> |

Tabelle 1: Technologie Stack

# Anforderungen

## Funktionale Anforderungen

|  |  |
| --- | --- |
| Name | Charaktererstellung |
| **ID** | FA-001 |
| **Ziel** | Ermöglicht dem Spieler, einen Charakter zu erstellen und zu personalisieren. |
| **Ereignis** | Der Spieler wählt «New Game» am Startbildschirm. |
| **Vorbedingung** | Das Spiel ist gestartet. |
| **Standardablauf** | 1. Der Spieler gibt einen Namen an. 2. Der Spieler kann zu seinen Basisattribute 2 Punkte hinzufügen. 3. Der Spieler wählt «Enter Dungeon». |
| **Alternativablauf** | - |
| **Nachbedingung Erfolg** | Charaktererstellung abgeschlossen, Spielbeginn im ersten Dungeon-Level. |
| **Nachbedingung Fehler** | - |
| **Klassifizierung** | MUSS |
| **Aufwand** | KLEIN |

Tabelle 2: FA-001 Charaktererstellung

|  |  |
| --- | --- |
| Name | Kampfsystem |
| **ID** | FA-002 |
| **Ziel** | Ermöglicht rundenbasierte Kämpfe gegen Gegner. |
| **Ereignis** | Spieler steht einem Gegner gegebnüber. |
| **Vorbedingung** | Der Spieler drückt beim Gegner auf «Fight» |
| **Standardablauf** | 1. Kampfmodus wird aktiviert 2. Startangreifer wird anhand der Schnelligkeit bestimmt. 3. Spieler wählt eine Attacke aus. 4. Gegner führt eine Attacke aus. 5. Rundenbasierte Logik bis zum Ende des Kampfes. |
| **Alternativablauf** | Spieler oder Gegner kann keine Attacke auswählen: Die andere Partei ist wieder am Zug. |
| **Nachbedingung Erfolg** | Gegner besiegt: Es erscheint eine Belohnungsmaske. |
| **Nachbedingung Fehler** | Spieler stirbt: Es erscheint eine Game Over-Maske. |
| **Klassifizierung** | MUSS |
| **Aufwand** | GROSS |

Tabelle 3: FA-002 Kampfsystem

|  |  |
| --- | --- |
| Name | Belohnungsauswahl nach Kampf |
| **ID** | FA-003 |
| **Ziel** | Dem Spieler wird nach jedem Kampf die Wahl zwischen HP-Regeneration oder einem zufälligen Gegenstand gegeben. |
| **Ereignis** | Ein Kampf wird gewonnen. |
| **Vorbedingung** | Der Spieler hat einen Kampf erfolgreich beendet und sieht die Belohnungsmaske. |
| **Standardablauf** | 1. Dem Spieler werden zwei Optionen angezegit: HP-Regneration oder zufälliger Gegenstand. 2. Der Spieler wählt eine der Optionen aus. |
| **Alternativablauf** | - |
| **Nachbedingung Erfolg** | Spieler wählt HP-Regneration: Der Spieler erhält die angegebene Lebenspunkte zurück.  Spieler wählt Gegenstand: Der Spieler kann den Gegenstand direkt ausrüsten oder ersetzen. |
| **Nachbedingung Fehler** | - |
| **Klassifizierung** | MUSS |
| **Aufwand** | MITTEL |

Tabelle 4: FA-003 Belohnungsauswahl nach Kampf

|  |  |
| --- | --- |
| Name | Belohnungsauswahl nach Level Up |
| **ID** | FA-004 |
| **Ziel** | Spieler können nache einem Level Up ihre Attribute erhöhen oder eine neue Attacke erlernen. |
| **Ereignis** | Spieler hat den XP-Schwellenwert zum nächsten Level überschritten. |
| **Vorbedingung** | Spieler gewinnt einen Kampf und erhält genügend XP. Die Belohnung vom Kampf wurde bereits abgehandelt. |
| **Standardablauf** | 1. Der Spieler sieht eine Level Up Maske. 2. Der Spieler erhält die Auswahl zwischen Attibutwerte erhöhen oder neue Attacke erlernen. |
| **Alternativablauf** | Der Spieler wählt neue Attacke und hat bereits 3 ausgerüstet: FA-005 |
| **Nachbedingung Erfolg** | Der Spieler wählt Attribute erhöhen: Der Spieler kann einen Attributwert um eins eröhen.  Der Spieler wählt neue Attacke: die neue Attacke wird ausgerüstet.  Generell: Die HP des Charakters wird komplett regeneriert. |
| **Nachbedingung Fehler** | Ausrüstungsslot ist bereits besetzt: Der Spieler wählt «Replace» und ersetzt die aktuelle Ausrüstung mit der neuen.  Es sind bereits 3 Attacken ausgerüstet: Der Spieler wählt eine bestehende Attacke aus und drückt «Replace». Die Attacke wird mit der neuen Attacke ersetzt. |
| **Klassifizierung** | MUSS |
| **Aufwand** | MITTEL |

Tabelle 5: FA-004 Belohnungsauswahl nach Level-Up

|  |  |
| --- | --- |
| Name | Verwalten von Attacken |
| **ID** | FA-005 |
| **Ziel** | Spieler können bis zu drei Attacken ausrüsten und bei Bedarf ersetzen. |
| **Ereignis** | Spieler erhält neue Attacke bei Level Up. |
| **Vorbedingung** | Spieler gewinnt Kampf und wählt beim Level Up eine neue Attacke. |
| **Standardablauf** | 1. Neue Attacke wird dem Spieler präsentiert. 2. Spieler wählt Attacke aus. |
| **Alternativablauf** | Es sind bereits 3 Attacken ausgerüstet: Es erscheint eine Maske mit der Auswahl welche Attacke ersetzt werden soll. |
| **Nachbedingung Erfolg** | Die Attacke ist am Charakter angelegt. |
| **Nachbedingung Fehler** | Es sind bereits 3 Attacken ausgerüstet: Der Spieler wählt eine bestehende Attacke aus und drückt «Replace». Die Attacke wird mit der neuen Attacke ersetzt. |
| **Klassifizierung** | MUSS |
| **Aufwand** | MITTEL |

Tabelle 6: FA-005 Verwaltung von Attacken

|  |  |
| --- | --- |
| Name | Verwalten von Ausrüstung |
| **ID** | FA-006 |
| **Ziel** | Spieler können Ausrüstung für Kopf, Körper, Hände, Füsse, Waffen oder Schilder ausrüsten und bei Bedarf ersetzen. |
| **Ereignis** | Spieler erhält neuen Ausrüstungsgegenstand. |
| **Vorbedingung** | Spieler gewinnt Kampf und wählt Ausrüstung bei der Belohnung. |
| **Standardablauf** | 1. Neue Attacke oder Ausrüstungsgegenstand wird dem Spieler präsentiert. 2. Spieler wählt Attacke oder Ausrüstung aus. |
| **Alternativablauf** | Ausrüstungsslot ist bereits besetzt: Es erscheint eine Maske, welche die beiden Ausrüsungsgegestände gegenüber stellt. |
| **Nachbedingung Erfolg** | Der Ausrüstungsgegenstand ist am Charakter angelegt. |
| **Nachbedingung Fehler** | Ausrüstungsslot ist bereits besetzt: Der Spieler wählt «Replace» und ersetzt die aktuelle Ausrüstung mit der neuen. |
| **Klassifizierung** | MUSS |
| **Aufwand** | MITTEL |

Tabelle 7: FA-006 Verwaltung von Ausrüstung

|  |  |
| --- | --- |
| Name | Charakter Information |
| **ID** | FA-007 |
| **Ziel** | Spieler können die Informationen des Charakters einsehen. |
| **Ereignis** | Spieler wählt das Charaktersymbol |
| **Vorbedingung** | Spieler hat einen Charakter erstellt und befindet sich im Dungeon. |
| **Standardablauf** | 1. Es erscheint eine gesammtübersicht des Charakters mit dessen Attribute, Ausrüstung und Attacken. 2. Der Spieler kann eine Ausrüstung oder Attacke selektieren. |
| **Alternativablauf** | - |
| **Nachbedingung Erfolg** | Es erscheinen die Details zur ausgewählten Ausrüstung oder Attacke. |
| **Nachbedingung Fehler** | - |
| **Klassifizierung** | KANN |
| **Aufwand** | KLEIN |

Tabelle 8: FA-007 Charakter Information

|  |  |
| --- | --- |
| Name | Leaderboard für Dungeon Erfolge |
| **ID** | FA-008 |
| **Ziel** | Spieler können ihre Erfolge mit denen anderer Spieler vergleichen. |
| **Ereignis** | Spieler verlässt erfolgreich einen Dungeon. |
| **Vorbedingung** | Spieler hat einen Kampf gewonnen und geht nicht zum nächsten Dungeon-Level. |
| **Standardablauf** | 1. Es erscheint eine Erfolgsmeldung. 2. Spieler kann einen Usernamen angeben. 3. Spieler kann auf «Save» klicken. |
| **Alternativablauf** | Der Spieler möchte seine Werte nicht ins Leaderboard speichern. |
| **Nachbedingung Erfolg** | Es gibt einen Eintrag ins Leaderboard mit Charaktername und Username. Das Leaderboard ist sortiert nach erreichtem Dungeonlevel. |
| **Nachbedingung Fehler** | Es gibt keinen Leaderboardeintrag und der Spieler wird auf die Startseite navigiert. |
| **Klassifizierung** | MUSS |
| **Aufwand** | MITTEL |

Tabelle 9: FA-008 Leaderboard für Dungeon Erfolg

|  |  |
| --- | --- |
| Name | Nächstes Dungeon-Level besuchen |
| **ID** | FA-009 |
| **Ziel** | Spieler kann zum nächsten Dungeon Level übergehen. |
| **Ereignis** | Spieler klickt auf «Delve Deeper». |
| **Vorbedingung** | Spieler hat einen Kampf gewonnen und die Belohnung erhelten. |
| **Standardablauf** | 1. Der nächste Gegner wird angezeigt. |
| **Alternativablauf** | Dungeon verlassen |
| **Nachbedingung Erfolg** | Der Spieler kann nun wieder FA-002 ausführen. |
| **Nachbedingung Fehler** | Wenn das Dungeon verlassen wird, hat der Spieler das Spiel gewonnen mit dem Dungeon Level als Highscore. |
| **Klassifizierung** | MUSS |
| **Aufwand** | KLEIN |

Tabelle 10: FA-009 Nächstes Dungeon-Level besuchen

|  |  |
| --- | --- |
| Name | Spielanleitung |
| **ID** | FA-010 |
| **Ziel** | Spieler werden am Anfang instruiert wie das Spiel funktioniert. |
| **Ereignis** | Spieler möchte wissen, wie das Spiel funktioniert. |
| **Vorbedingung** | Spieler öffnet die Help Seite. |
| **Standardablauf** | 1. Es erscheint ein User Manual |
| **Alternativablauf** | - |
| **Nachbedingung Erfolg** | Der Spieler kann nachlesen wie das Spiel funktioniert. |
| **Nachbedingung Fehler** | - |
| **Klassifizierung** | KANN |
| **Aufwand** | KLEIN |

Tabelle 11: FA-010 Spielanleitung

## Nicht-funktionale Anforderungen

|  |  |
| --- | --- |
| Name | Schnelle Ladezeiten |
| **ID** | NFA-001 |
| **Ziel** | Das Spiel soll innerhalb von 3 Sekunden nach dem Starten bereit sein. |
| **Klassifizierung** | MUSS |

Tabelle 12: NFA-001 Schnelle Ladezeiten

|  |  |
| --- | --- |
| Name | Intuitive Benutzeroberfläche |
| **ID** | NFA-002 |
| **Ziel** | Spieler sollen ohne Anleitung navigieren können. |
| **Klassifizierung** | MUSS |

Tabelle 13: NFA-002 Intuitive benutzeroberfläche

|  |  |
| --- | --- |
| Name | Progressive Web App (PWA) |
| **ID** | NFA-003 |
| **Ziel** | Das Spiel soll als PWA installierbar sein. |
| **Klassifizierung** | KANN |

Tabelle 14: NFA-003 PWA

# Struktur und Planung

## Wireframe

Ein detailiertere Ansicht des Wireframes kann über folgenden Link abgerufen werden: <https://www.figma.com/file/2yvGrcJwC7dlGxLXg62qeq/Eternal-Depths-Wireframe?type=design&node-id=0%3A1&mode=design&t=0KaVQDq6JAXOYxNF-1>

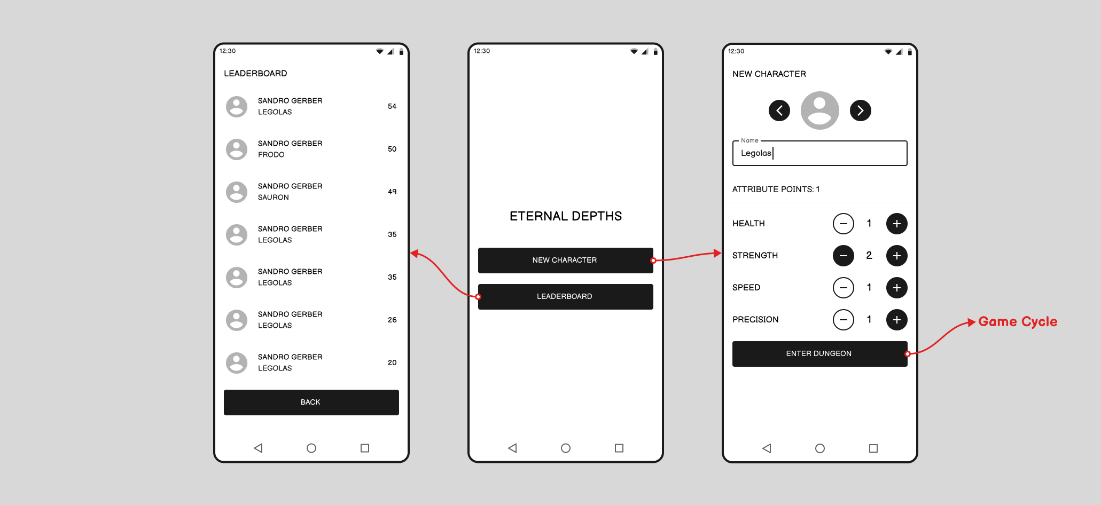


Abbildung 1: Start Screen

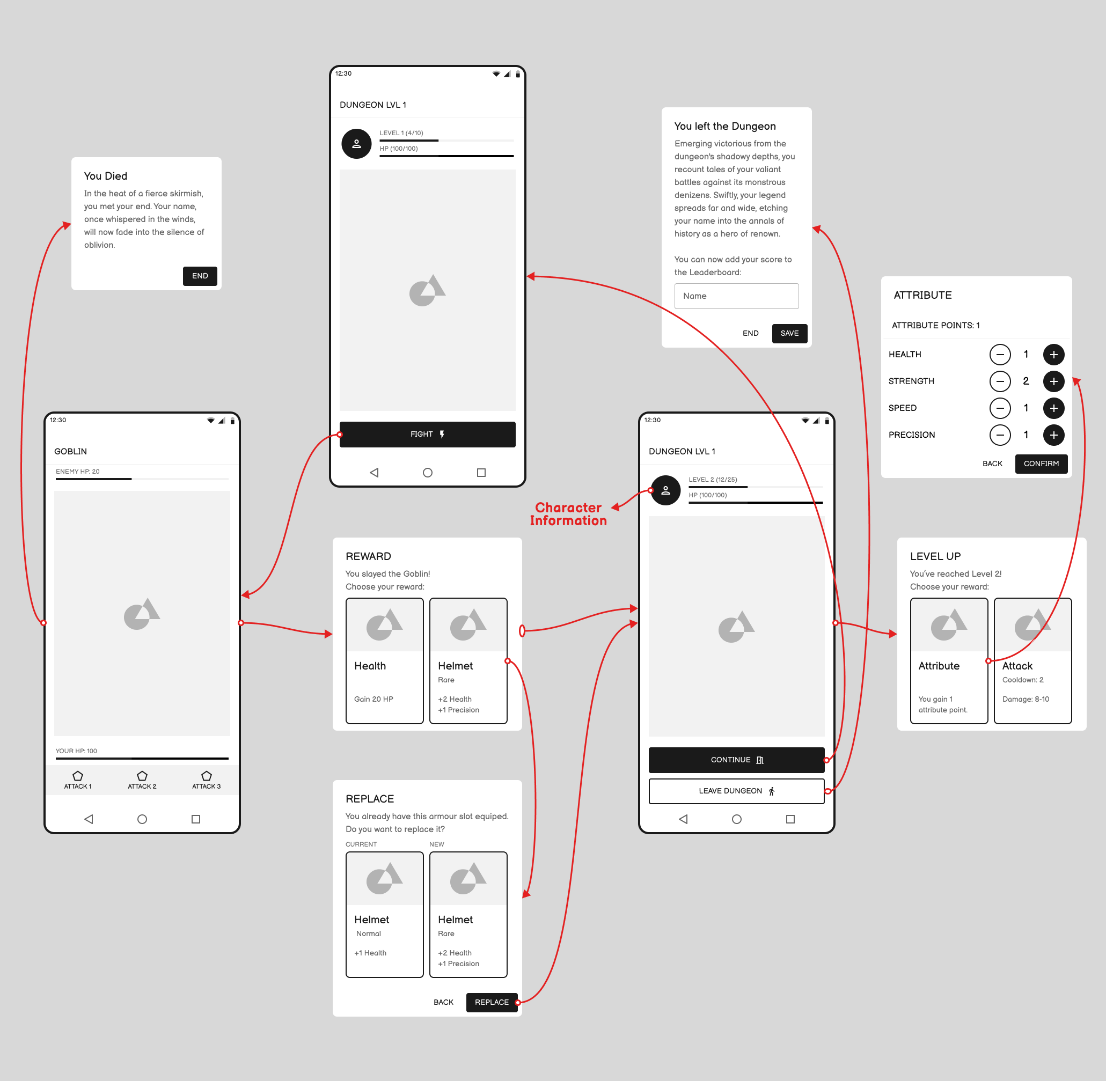


Abbildung 2: Game Cycle

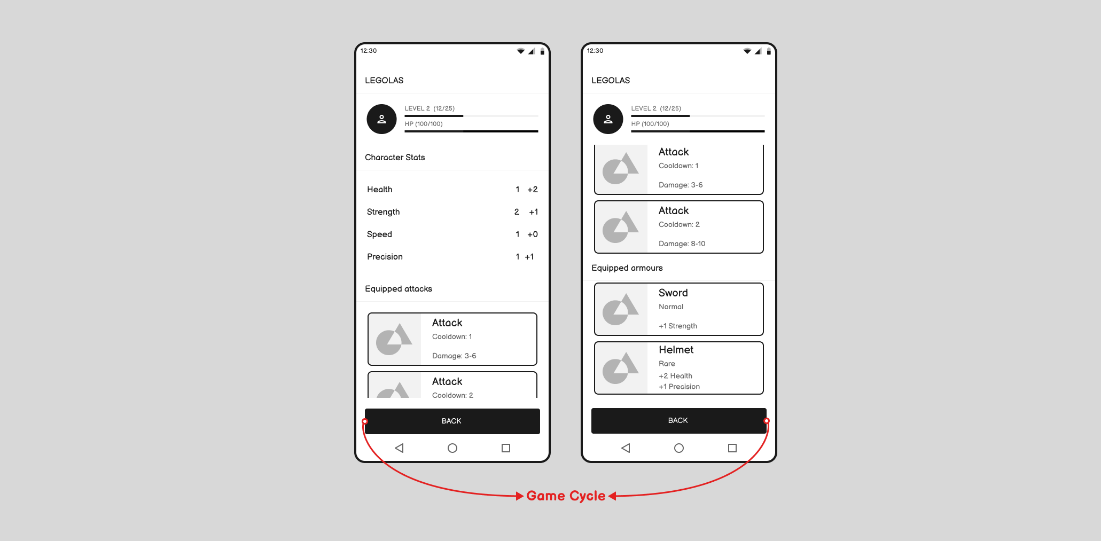


Abbildung 3: Character Information

## Spielzyklus

Der Spielzyklus besteht aus verschiedenen States und Cycles, die den Ablauf des Spiels definieren. Der Game State umfasst vier Phasen: Idle, Playing, Over und Won, die den allgemeinen Zustand des Spiels repräsentieren. Innerhalb des "Playing"-Zustands wird der Game Cycle aktiv, der spezifische Phasen wie Encounter, Battle (mit einem eigenen Battle Cycle), Battle Victory, Battle Defeat, Level Up und Reward durchläuft. Der Battle Cycle wiederum besteht aus zwei Hauptphasen: dem Encounter Turn, in dem der Gegner angreifft, und dem Player Turn, in dem der Spieler seinen Angriff wählt.

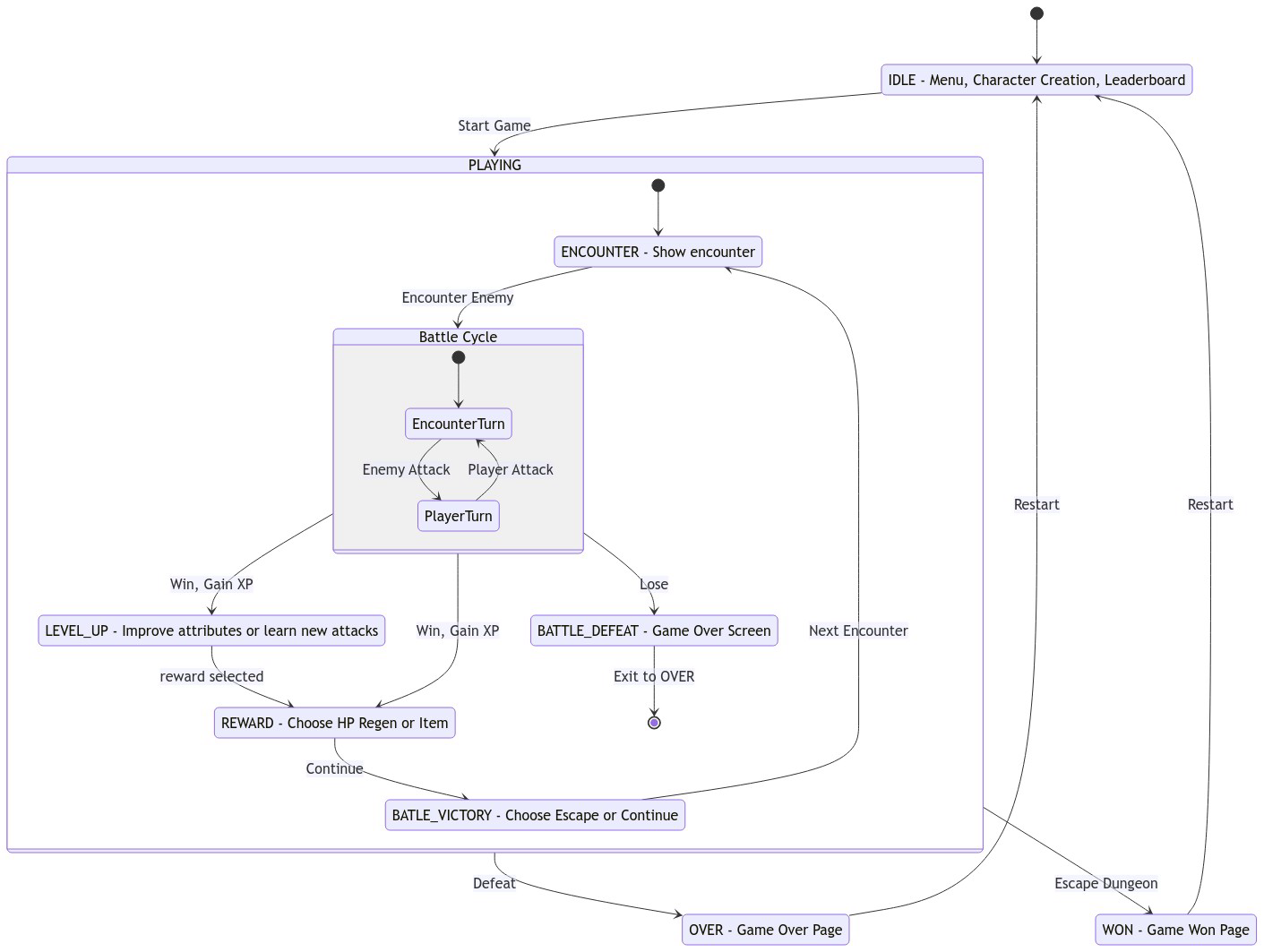


Abbildung 4: Game Cycle

## Modularisierung

### Atomic Design

Atomic Design ist eine Methodologie, die von Brad Frost[[1]](#footnote-2) entwickelt wurde, um Benutzeroberflächen systematisch zu gestalten. Sie basiert auf der Analogie zur Chemie, wobei die Benutzeroberfläche in fundamentale Bausteine unterteilt wird, die sich zu komplexen Strukturen zusammensetzen lassen. Die fünf Ebenen des Atomic Designs sind Atome, Moleküle, Organismen, Templates und Seiten.

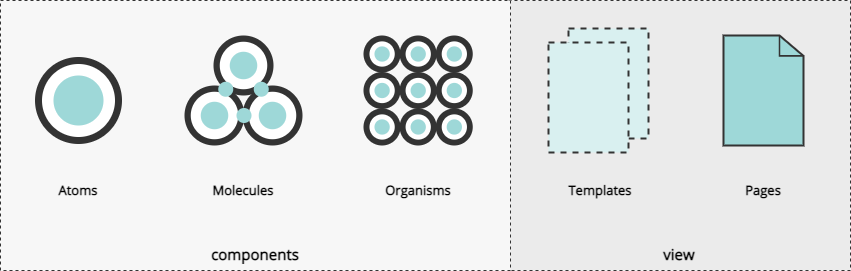


Abbildung 5: Atomic Design System

|  |  |
| --- | --- |
| **Atoms** | Diese sind die grundlegendsten Bausteine und stellen UI-Elemente dar, wie Buttons, Inputs oder Labels. Atome sind hochgradig wiederverwendbar und dienen als Basis für komplexere Komponenten. In deinem Projekt sind Atome vollständig von Properties abhängig und vermeiden direkten Zugriff auf den Redux Store. |
| **Molecules** | Moleküle sind Gruppierungen von Atomen, die zusammen eine funktionale Einheit bilden. Wie Atome sind Moleküle ausschliesslich von Properties getrieben und interagieren nicht direkt mit dem Redux Store. |
| **Organisms** | Organismen sind relativ komplexe UI-Komponenten, die aus mehreren Molekülen und möglicherweise weiteren Atomen zusammengesetzt sind. Auch Organismen folgen dem Prinzip, nur über Properties gesteuert zu werden und nicht direkt auf den Redux Store zuzugreifen. |
| **Templates** | Templates sind in diesem Web Projekt strukturierungselemente, die entwededer die Struktur der Seite Vorgeben oder einfach Elemente Wie Trennlinie oder Box sind. Templates haben keine interaktion mit dem Redux Store. |
| **Pages** | Auf der höchsten Ebene stehen die Seiten, gefüllt mit realen Inhalten und Daten. Im Gegensatz zu den anderen Ebenen dürfen Seiten auf den Redux Store zugreifen, um globale Zustände abzurufen oder zu manipulieren. Dies ermöglicht eine klare Trennung der Zuständigkeiten, wobei die Verwaltung des Anwendungszustands auf der höchsten Ebene zentralisiert ist.  Pages können zur Übersicht in weitere Subkomponente unterteilt werden. |

Tabelle 15: Atomic Design System

## Route Guard

Die Routes werden durch den Game State gesteuert. Folgende Seiten sind im dedizierten State zugänglich:

* **IDLE:** Menu (Home), Character Creation, Leaderboard
* **PLAYING:** Dungeon
* **WON:** Einschreiben in Leaderboard
* **OVER:** Game Over

## Backend Konfiguration

Um die Applikation lokal zu betreiben, ist es notwendig, eine Verbindung zum Supabase-Backend herzustellen. Dies wird über Umgebungsvariablen geregelt, die in einer Datei namens .env.local im Root-Verzeichnis des Projekts definiert werden. Die Einrichtung dieser Datei ermöglicht es dem Frontend, sicher auf das Backend zuzugreifen, ohne sensible Informationen im Code hart zu kodieren.

Für diese Projektarbeit habe ich bereits eine Test Umgebung aufgesetzt. Es ist nur notwendig das File .env.local anzulegen mit follgendem Inhalt:

REACT\_APP\_SUPABASE\_URL='https://ccsgujvmudczzpadmxmh.supabase.co'

REACT\_APP\_SUPABASE\_ANON\_KEY='eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImNjc2d1anZtdWRjenpwYWRteG1oIiwicm9sZSI6ImFub24iLCJpYXQiOjE3MTIyNzE1NTYsImV4cCI6MjAyNzg0NzU1Nn0.3Ndzp5\_siyjaHJn0igKn7z7MW5znFAtQ6zKGIcrBYOY'

# Abschlussbericht

## Erfüllte Anforderungen

Es konnten alle MUSS Anforderungen erfüllt werden. Zwei funktionale KANN Anforderungen wurden nicht umgesetzt.

Die Anforderung NFA-001 ist nicht messbar, jedoch hat jede Person die das Spiel das erste Mal gespielt hat die Applikation ohne Anleitung bedienen und hat das Spielkonzept verstanden.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Titel | Klassifizierung | Erfüllt |
| FA-001 | Charaktererstellung | MUSS | ✅ |
| FA-002 | Kampfsystem | MUSS | ✅ |
| FA-003 | Belohnungsauswahl nach Kampf | MUSS | ✅ |
| FA-004 | Belohnungsauswahl nach Level Up | MUSS | ✅ |
| FA-005 | Verwalten von Attacken | MUSS | ✅ |
| FA-006 | Verwalten von Ausrüstung | MUSS | ✅ |
| FA-007 | Charakter Information | KANN |  |
| FA-008 | Leaderboard für Dungeon Erfolge | MUSS | ✅ |
| FA-009 | Nächstes Dungeon-Level besuchen | MUSS | ✅ |
| FA-010 | Spielanleitung | KANN |  |
| NFA-001 | Schnelle Ladezeiten | MUSS | Nicht Messbar |
| NFA-002 | Intuitive Benutzeroberfläche | MUSS | ✅ |
| NFA-003 | Progressive Web App (PWA) | KANN | ✅ |

Tabelle 16: Erfüllte Anforderungen

## Zukunftsblick

Als nächstes veröffentliche ich das Spiel als Open-Source-Projekt auf GitHub. Anschliessend setze ich die zwei nicht umgesetzten Anforderungen um. Das Rebalancing der Attributwerte im Spiel ist ebenfalls notwendig und ein fortlaufender Prozess. Hier sind noch weitere Ideen für die Weiterentwicklung:

* Items
* Gold Belohnung
* Shop Integration für Items
* Spezial Attacken für Gegner
* Element Typen (Waffen und Verteidigung)
* Einmalige Wiederbelebung durch Werbevideo

## Reflexion

Bei dieser Projektarbeit habe ich mein erstes Spiel entwickelt. React war für mich neu, ich hatte jedoch bereits viel Erfahrung in VueJS und Angular, was mir auch bei der Erarbeitung der Webapplikation einen Vorteil erbrachte.

Das Atomic Design von Brad Frost ist ein sehr gutes Kodularisierungskonzept. Dadurch konnte ich eine Vielzahl wiederverwendbarer Komponenten entwickeln. Atoms, Molecules und Organisms dürfen keine Store-Zugriffe machen, damit sie unabhängig bleiben.

Das Projekt ist einfach erweiterbar, weil es nicht nur dieses Konzept gibt, sondern auch die Generatoren, die ich für Gegner, Items und Attacken geschrieben habe. Diese nutzen eine Basis-Konstante und transformieren eine Kopie davon zu einem tatsächlichen Gegner oder Item. Dadurch kann z. B. ein neuer Gegner ganz einfach durch das Erweitern durch eine neue Konstante erstellt werden.

Ein Game Balancing ist eine Arbeit für sich. Ich habe eine ganze Iteration damit verbracht, die Attributwerte im Spiel auszubalancieren. Sicherlich ist noch nicht alles perfekt, aber das Spiel funktioniert bereits sehr gut und jeder Attributwert hat eine signifikante Gewichtung, die spürbar ist.

Die Arbeit hat mich mehr Aufwand gekostet als im Modul definiert, aber ich habe jede Minute genossen, die ich mit der Entwicklung meines ersten Spiels verbracht habe. Es war schon immer ein Traum von mir, ein eigenes Spiel zu entwickeln. Nun konnte ich dies in diesem Modul verwirklichen und bin mit dem Resultat extrem zufrieden.

# Weitere Dokumentations Resourcen

|  |  |
| --- | --- |
| Dokumentation | Link |
| Logbuch / Journal | <https://moodle.ffhs.ch/mod/forum/view.php?id=4597530> |
| Task Planung | <https://git.ffhs.ch/web-technologien/fwebt/fs24/inf-w-af004-fwebt-ol-sa-1-pva-fs24/projektarbeiten/ol010-sandro-gerber-eternal-depths/-/issues> |

Tabelle 17: Weitere Dokumentations Resourcen

Abbildungsverzeichnis

[Abbildung 1: Start Screen 11](#_Toc169815586)

[Abbildung 2: Game Cycle 12](#_Toc169815587)

[Abbildung 3: Character Information 12](#_Toc169815588)

[Abbildung 4: Game Cycle 13](#_Toc169815589)

[Abbildung 5: Atomic Design System 14](#_Toc169815590)

Tabellenverzeichnis

[Tabelle 1: Technologie Stack 4](#_Toc169815591)

[Tabelle 2: FA-001 Charaktererstellung 5](#_Toc169815592)

[Tabelle 3: FA-002 Kampfsystem 5](#_Toc169815593)

[Tabelle 4: FA-003 Belohnungsauswahl nach Kampf 6](#_Toc169815594)

[Tabelle 5: FA-004 Belohnungsauswahl nach Level-Up 7](#_Toc169815595)

[Tabelle 6: FA-005 Verwaltung von Attacken 7](#_Toc169815596)

[Tabelle 7: FA-006 Verwaltung von Ausrüstung 7](#_Toc169815597)

[Tabelle 8: FA-007 Charakter Information 8](#_Toc169815598)

[Tabelle 9: FA-008 Leaderboard für Dungeon Erfolg 8](#_Toc169815599)

[Tabelle 10: FA-009 Nächstes Dungeon-Level besuchen 9](#_Toc169815600)

[Tabelle 11: FA-010 Spielanleitung 9](#_Toc169815601)

[Tabelle 12: NFA-001 Schnelle Ladezeiten 10](#_Toc169815602)

[Tabelle 13: NFA-002 Intuitive benutzeroberfläche 10](#_Toc169815603)

[Tabelle 14: NFA-003 PWA 10](#_Toc169815604)

[Tabelle 15: Atomic Design System 15](#_Toc169815605)

[Tabelle 16: Erfüllte Anforderungen 16](#_Toc169815606)

[Tabelle 17: Weitere Dokumentations Resourcen 18](#_Toc169815607)

Hilfsmittelverzeichnis

|  |  |  |
| --- | --- | --- |
| Hilfsmittel | Nutzung | Stellen |
| ChatGPT | Weiterführung meiner Grundidee | Kapitel 1 und 1.1 |
| ChatGPT | Schreiben des README | README.md |
| DeepL Write | Rechtschreibkorrektur, Formulierung | Gesammtes Dokument |
| GitHub Copilot | Code Assistenz | Gesammtes Projekt |

1. <https://bradfrost.com/blog/post/atomic-web-design/> [↑](#footnote-ref-2)