

Eternal Depths

Mobile Web RPG Game

FFHS
Fernfachhochschule
Schweiz
Mitglied der SUPSI

Autoren: Sandro Gerber
Dozent: Christian Klauenbösch
Veröffentlichung: 13.03.2024

Inhaltsverzeichnis

1	Projektidee und Geschäftsregeln.....	3
1.1	Spielregeln.....	3
1.2	Abgrenzung	3
2	Technologie Stack	4
3	Anforderungen	5
3.1	Funktionale Anforderungen	5
3.2	Nicht-funktionale Anforderungen	10
4	Struktur und Planung	11
4.1	Wireframe.....	11
4.2	Spielzyklus	13
4.3	Modularisierung	14
4.3.1	Atomic Design.....	14
4.3.2	Store	15
4.4	Route Guard.....	15
5	Weitere Dokumentations Ressourcen	16

1 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.

1.1 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

1.2 Abgrenzung

- Accessibility: Die Applikation muss nicht Barrierefrei aufgebaut werden (WCAG 2.1 Richtlinien 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.
- Mehrsprachigkeit: Das Spiel ist nur in Englisch Verfügbar.
- Spielschwierigkeit: Es wird kein Spielbalancing gemacht.

2 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.
Firebase	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

3 Anforderungen

3.1 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	<ol style="list-style-type: none"> 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 gegenüber.
Vorbedingung	Der Spieler drückt beim Gegner auf «Fight»
Standardablauf	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 1. Dem Spieler werden zwei Optionen angezeigt: HP-Regeneration oder zufälliger Gegenstand. 2. Der Spieler wählt eine der Optionen aus.
Alternativablauf	-
Nachbedingung Erfolg	<p>Spieler wählt HP-Regeneration: Der Spieler erhält die angegebene Lebenspunkte zurück.</p> <p>Spieler wählt Gegenstand: Der Spieler kann den Gegenstand direkt ausrüsten oder ersetzen.</p>
Nachbedingung Fehler	-
Klassifizierung	MUSS
Aufwand	MITTEL

Tabelle 4: FA-003 Belohnungsauswahl nach Kampf

Name	Belohnungsauswahl nach Level Up
ID	FA-004
Ziel	Spieler können nach 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	<ol style="list-style-type: none"> 1. Der Spieler sieht eine Level Up Maske. 2. Der Spieler erhält die Auswahl zwischen Attributwerte erhöhen oder neue Attacke erlernen.
Alternativablauf	Der Spieler wählt neue Attacke und hat bereits 3 ausgerüstet: FA-005
Nachbedingung Erfolg	<p>Der Spieler wählt Attribute erhöhen: Der Spieler kann einen Attributwert um eins erhöhen.</p> <p>Der Spieler wählt neue Attacke: die neue Attacke wird ausgerüstet.</p> <p>Generell: Die HP des Charakters wird komplett regeneriert.</p>
Nachbedingung Fehler	<p>Ausrüstungsslot ist bereits besetzt: Der Spieler wählt «Replace» und ersetzt die aktuelle Ausrüstung mit der neuen.</p> <p>Es sind bereits 3 Attacken ausgerüstet: Der Spieler wählt eine bestehende Attacke aus und drückt «Replace». Die Attacke wird mit der neuen Attacke ersetzt.</p>

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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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 Ausrüstungsgegenstände 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	<ol style="list-style-type: none"> 1. Es erscheint eine gesamtübersicht 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	<ol style="list-style-type: none"> 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 erhalten.
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	2. 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

3.2 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

4 Struktur und Planung

4.1 Wireframe

Ein detailliertere Ansicht des Wireframes kann über folgenden Link abgerufen werden:

<https://www.figma.com/file/2yvGrcJwC7dIGxLXg62qeq/Eternal-Depths-Wireframe?type=design&node-id=0%3A1&mode=design&t=0KaVQDq6JAXOYxNF-1>

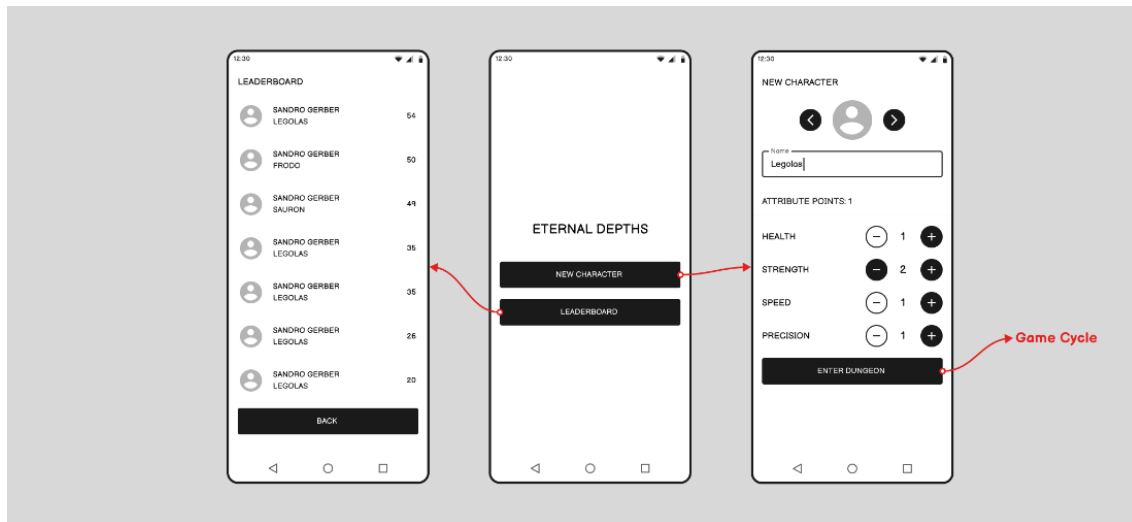


Abbildung 1: Start Screen

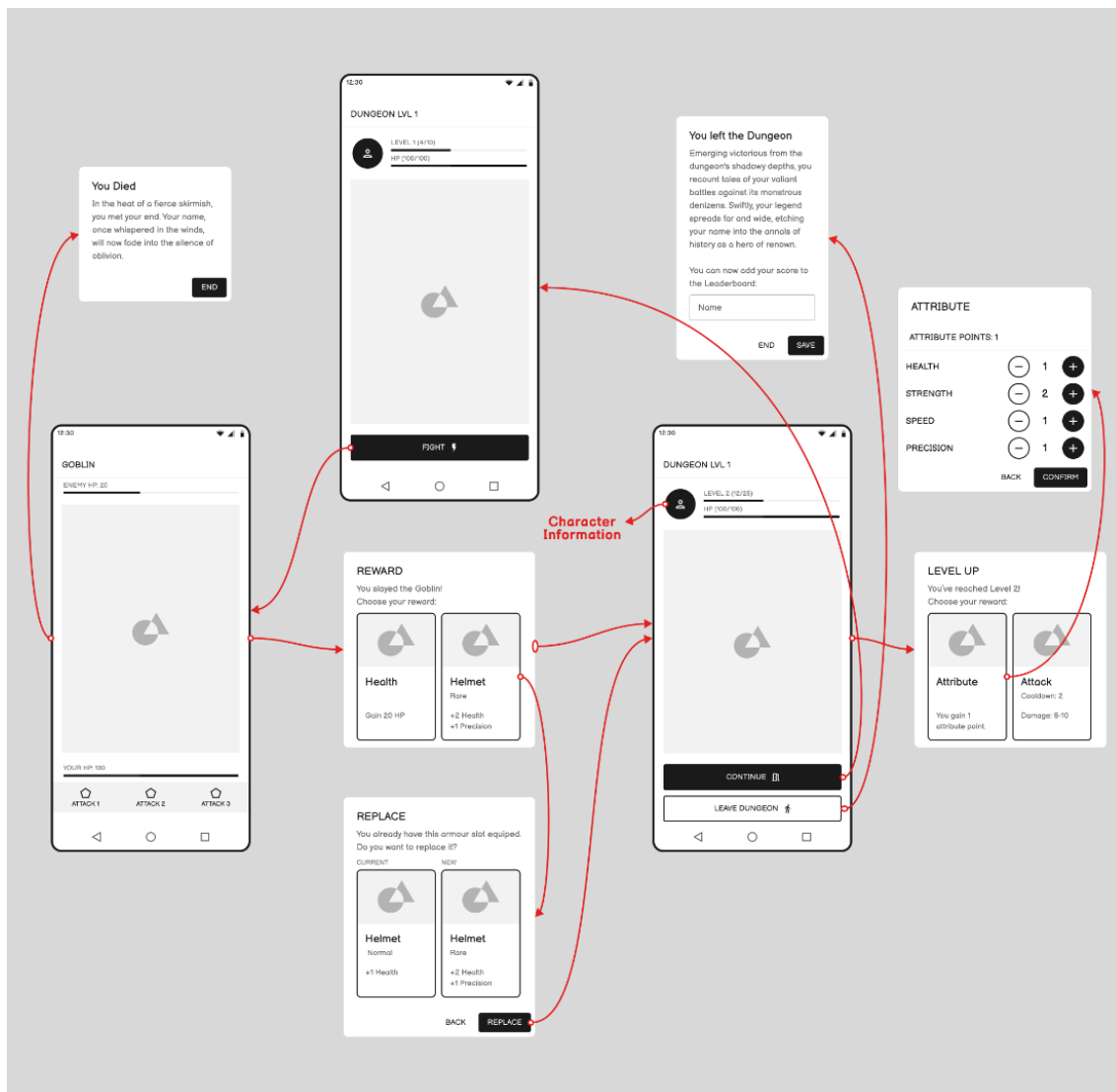


Abbildung 2: Game Cycle

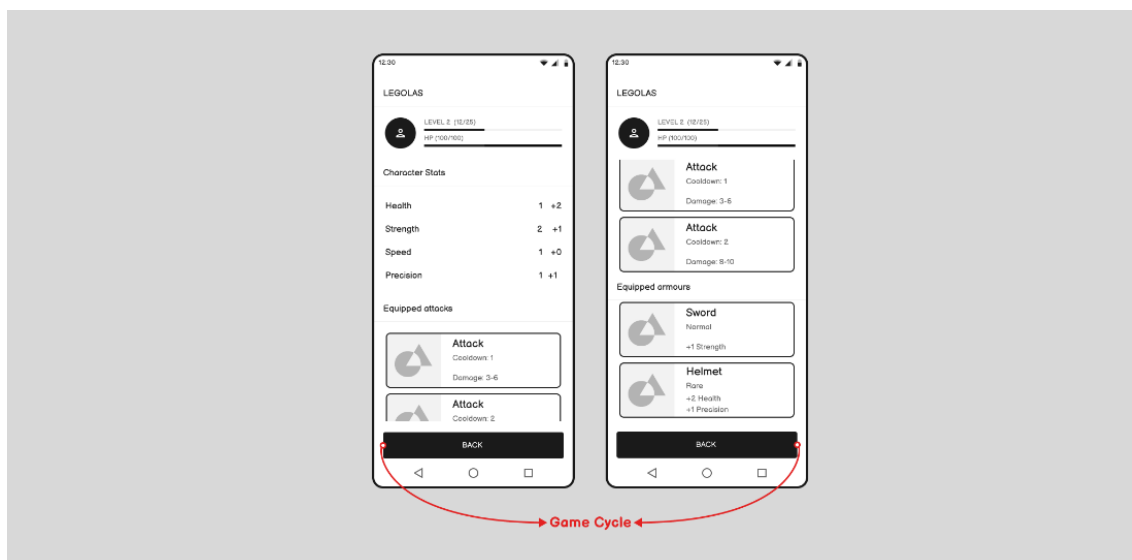


Abbildung 3: Character Information

4.2 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 und Level Up durchläuft. Der Battle Cycle wiederum besteht aus zwei Hauptphasen: dem Encounter Turn, in dem der Gegner angreift, und dem Player Turn, in dem der Spieler seine Aktionen durchführt.

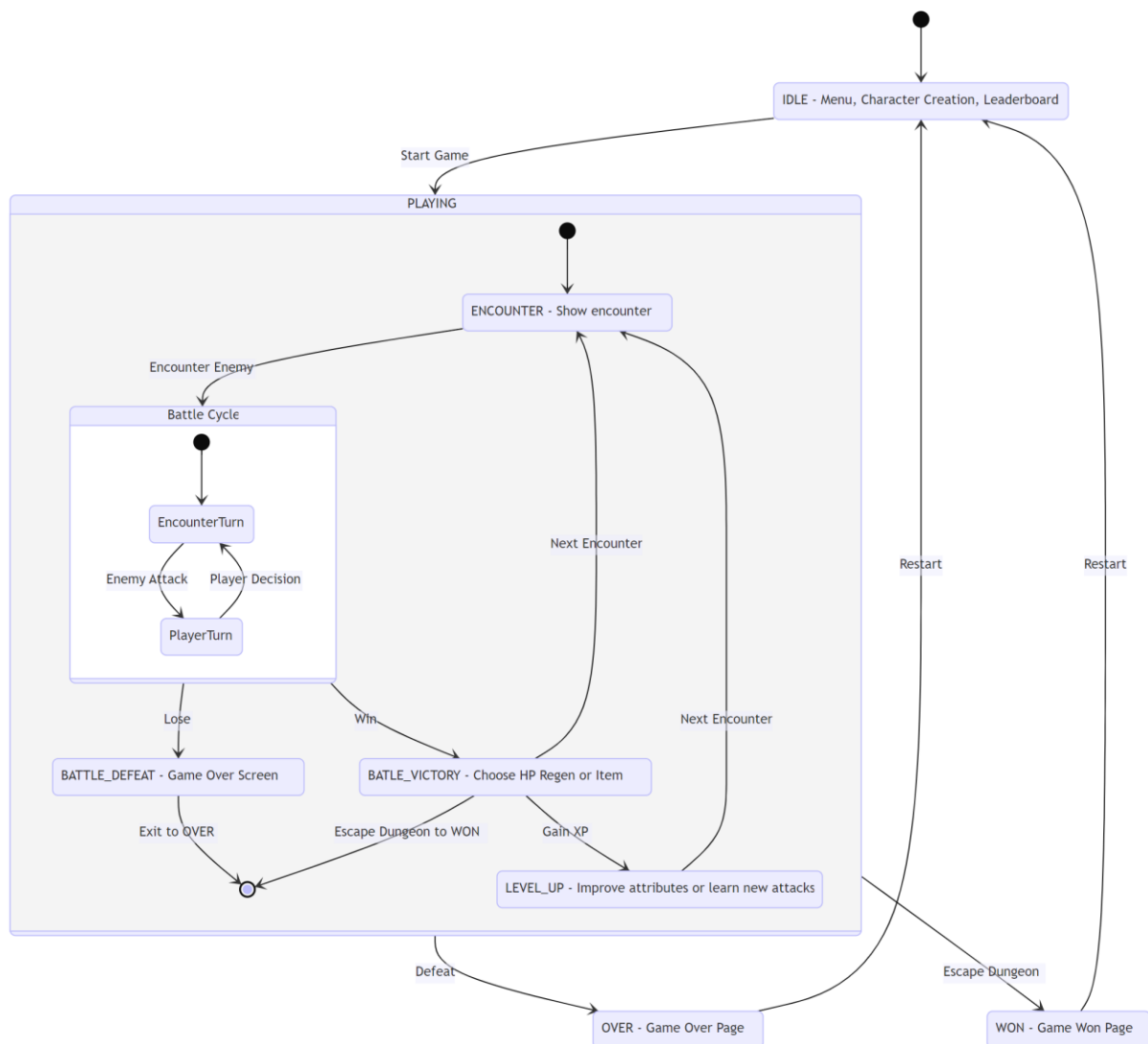


Abbildung 4: Game Cycle

4.3 Modularisierung

4.3.1 Atomic Design

Atomic Design ist eine Methodologie, die von Brad Frost¹ 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.

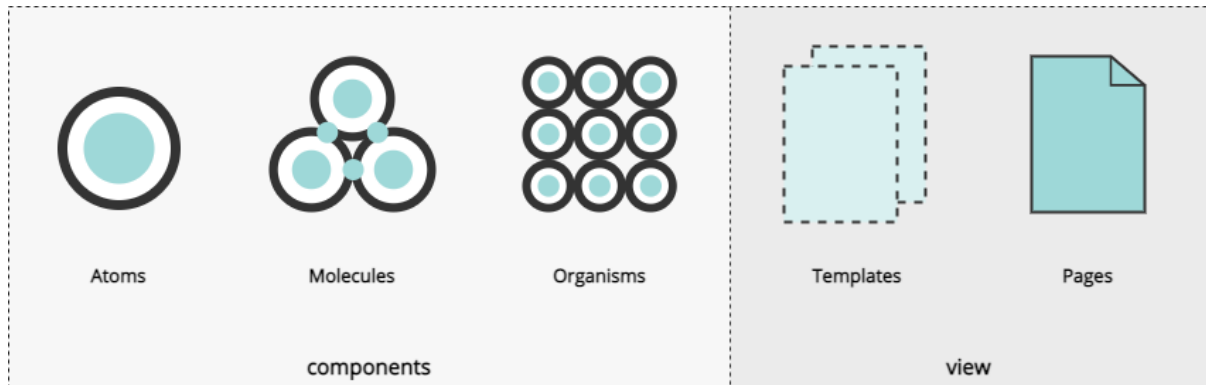


Abbildung 5: Atomic Design System

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 entweder die Struktur der Seite Vorgeben oder einfach Elemente Wie Trennlinie oder Box sind. Templates haben keine interaktion mit dem Redux Store.

¹ <https://bradfrost.com/blog/post/atomic-web-design/>

Pages	<p>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.</p> <p>Pages können zur Übersicht in weitere Subkomponente unterteilt werden.</p>
--------------	--

Tabelle 15: Atomic Design System

4.3.2 Store

Für das State Management der Webapplikation, wird Redux eingesetzt. Der Zustand der Anwendung ist in zwei Hauptbereiche unterteilt: game und character. Diese Modularisierung des Stores ermöglicht eine klare Trennung der Anwendungslogik und erleichtert die Verwaltung des Zustands, da sich die Zuständigkeiten spezifisch auf verschiedene Aspekte des Spiels beziehen.

Character	<p>Der character Slice beinhaltet alle Daten, die den Spielercharakter betreffen. Dazu zählen sowohl die Attribute des Charakters als auch die Ausrüstung und die Attacken. Die Verwaltung dieser Daten in einem dedizierten Slice ermöglicht es, Änderungen am Spielercharakter effizient zu handhaben und auf Aktionen wie Level-Up, das Erlernen neuer Attacken oder die Veränderung von Ausrüstungsgegenständen reagieren zu können.</p>
Game	<p>Der game Slice verwaltet alle Daten, die den aktuellen Spielstand betreffen, einschliesslich der Stage, der Gegner und des allgemeinen Spielzyklus.</p>

Tabelle 16: Store Modularisierung

4.4 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

5 Weitere Dokumentations Ressourcen

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 Ressourcen

Abbildungsverzeichnis

Abbildung 1: Start Screen	11
Abbildung 2: Game Cycle.....	12
Abbildung 3: Character Information	12
Abbildung 4: Game Cycle.....	13
Abbildung 5: Atomic Design System.....	14

Tabellenverzeichnis

Tabelle 1: Technologie Stack	4
Tabelle 2: FA-001 Charaktererstellung.....	5
Tabelle 3: FA-002 Kampfsystem.....	5
Tabelle 4: FA-003 Belohnungsauswahl nach Kampf.....	6
Tabelle 5: FA-004 Belohnungsauswahl nach Level-Up.....	7
Tabelle 6: FA-005 Verwaltung von Attacken	7
Tabelle 7: FA-006 Verwaltung von Ausrüstung	7
Tabelle 8: FA-007 Charakter Information	8
Tabelle 9: FA-008 Leaderboard für Dungeon Erfolg.....	8
Tabelle 10: FA-009 Nächstes Dungeon-Level besuchen	9
Tabelle 11: FA-010 Spielanleitung.....	9
Tabelle 12: NFA-001 Schnelle Ladezeiten	10
Tabelle 13: NFA-002 Intuitive benutzeroberfläche	10
Tabelle 14: NFA-003 PWA	10
Tabelle 15: Atomic Design System.....	15
Tabelle 16: Store Modularisierung	15
Tabelle 17: Weitere Dokumentations Ressourcen	16