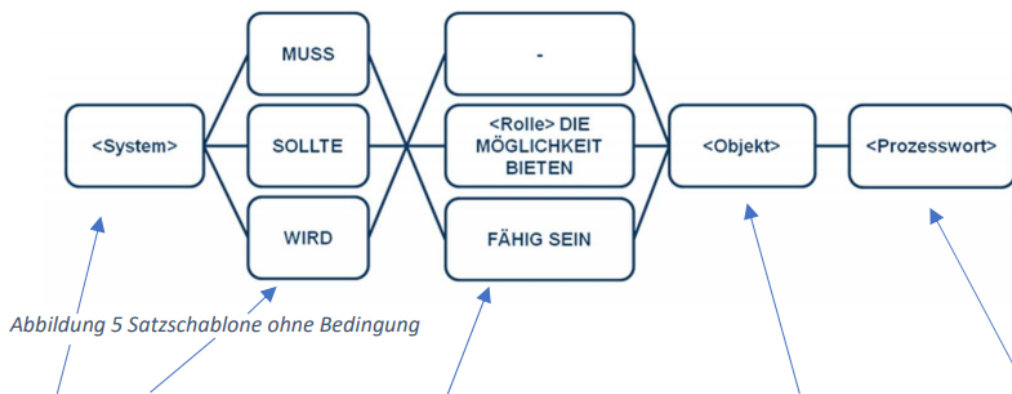
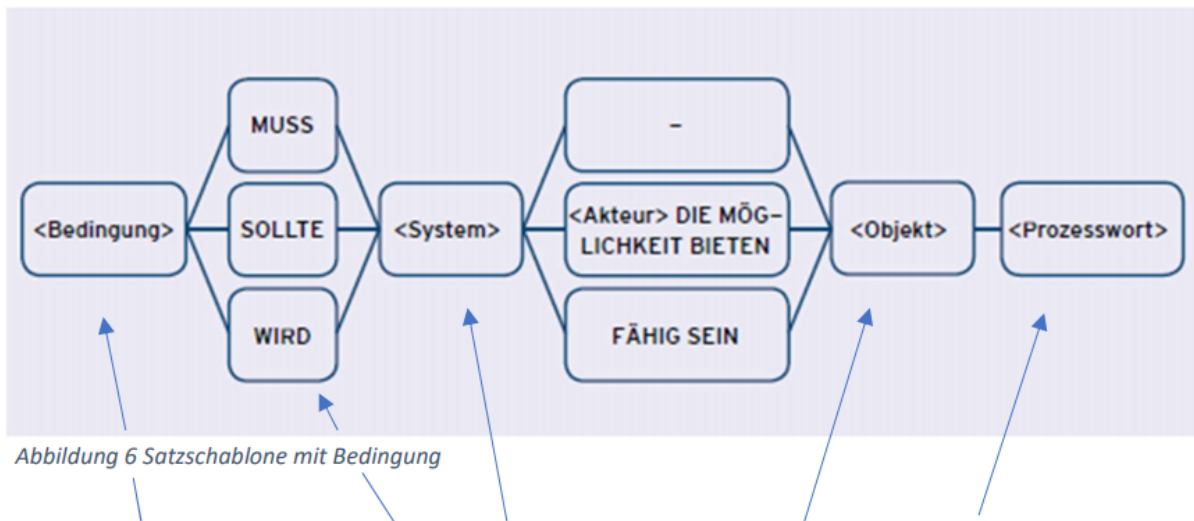


Requirements

Dieses Kapitel verfeinert die Feature-Liste, die in der Produktvision deklariert wird. Strukturieren Sie dieses evtl. lange Kapitel so, dass Zusammengehöriges in das gleiche Unterkapitel gelegt wird. Falls möglich, erstellen Sie ein Unterkapitel pro Feature. Sortieren Sie die Unterkapitel gemäss ihrer Priorität für die Entwicklung (oben das zuerst zu implementierende Unterkapitel). In diesem Kapitel wird erwartet, dass Sie sinnvolle Notationen selbständig auswählen. Machen Sie einen aktiven Transfer der REQ-Kursinhalte in dieses Dokument. Tipp: passen Sie die Strukturierung der Unterkapitel an, so dass Sie mit möglichst wenig Text und Grafiken die notwendige Inhalte präsentieren können. Vermeiden Sie Unnötiges und Redundanzen so weit wie möglich. Tipp: jeder Inhalt (Feature, Use Case, Anforderung, Klasse) soll einen einmaligen Identifikator erhalten, so dass aus anderer Dokumentation eindeutig darauf verwiesen werden kann. Tipp: je nach Applikation muss die Kapitelstruktur geändert werden. Diskutieren Sie dies mit den Dozierenden.



Das System muss dem Unternehmen die Möglichkeit bieten die Rendite eines Projektes zu berechnen



Nach der Stempelung muss das System, die Projekt-Lohnkosten berechnen

Anforderungsanalyse

Basisanforderungen

ID	Use Case	Beschreibung der Anforderung	Verbindlichkeit	Akteure	Projektziel
BA01		Die Hardware muss dem User die Möglichkeit bieten, das System ohne weitere Hilfsmittel anzuschalten/zu aktivieren.	MUSS	User	

BA02		Das System muss dem User die Möglichkeit bieten, mittels Eingabe, sich als Spieler für das System zu registrieren.	MUSS	User	
BA03		Das System muss dem User die Möglichkeit bieten, die Spiellogik und den Spielaufbau mittels angezeigtem Informationsfeld zu verstehen.	MUSS	User	
BA04		Das System muss dem User die Möglichkeit bieten, die Verbindung zwischen Hardware und Spielelementen mittels Erklärungen im Interface, zu verstehen.	MUSS	System	
BA05		Das System muss fähig sein, Inputs des Hardwareelements Joystick zu verarbeiten.	MUSS	System	
BA06		Das System muss fähig sein, Inputs des Hardwareelements Stromzähler numerisch anzuzeigen.	MUSS	System	
BA07		Das System muss fähig sein, Strominput des Hardwareelements Kurbel an das Hardwareelement Stromzähler weiterzuleiten.	MUSS	Hardware/System	
BA08		Das System muss fähig sein, dem User ein logisches Spielfeld anzuzeigen, welches alle weiteren Spielelemente beherbergt.	MUSS	System	

BA09		Das System muss fähig sein, dem User Informationen zu seinen physisch gekurbelten Energie zu vermitteln.	MUSS	System/User	
BA10		Das System muss fähig sein, die physisch gekurbelten Energie ins Spiel einzubetten.	MUSS	System	
BA11		Das System muss fähig sein, eine Verbindung zum Energieverbrauch im Alltag herzustellen, welche spielerisch einen Lerneffekt erzeugt.	MUSS	System	
BA12		Das System muss fähig sein, dem User eine Beendigungsmöglichkeit des Spiels in Form eines Buttons zu geben.	MUSS	User	

Leistungsanforderungen

ID	Use Case	Beschreibung der Anforderung	Verbindlichkeit	Akteure	Projektziel
LE01		Das System sollte fähig sein, Spielleben des Spielobjekts dem User visualisiert auf dem Spielfeld anzuzeigen.	SOLL	User	
LE02		Das System muss fähig sein, die Schnelligkeit des Kurbelns im Spielfeld anzuzeigen	MUSS	System	

LE03		Das System sollte fähig sein, verschiedene Checkpoints dem User als Spielelemente visualisiert auf dem Interface anzuzeigen.	SOLL	User	
LE04		Das System sollte fähig sein, die erspielte Leistung mittels einer Highscore-Anzeige dem User anzuzeigen.	SOLL	User	
LE04		Das System sollte fähig sein, eine Herausforderung in Form eines Gegners dem User anzuzeigen.	SOLL	User	
LE05		Das System sollte fähig sein, dem User seinen aktuellen Spielstand/-leistung in Form einer Anzeige zu vermitteln.	SOLL	User	

Begeisterungsanforderungen

ID	Use Case	Beschreibung der Anforderung	Verbindlichkeit	Akteure	Projektziel
----	----------	------------------------------	-----------------	---------	-------------

BE01		Das System soll fähig sein, eine Statistik zur gekurbelten Energie des Tages darzustellen.	SOLL	System	
BE02		Das System soll fähig sein, eine Statistik zur gekurbelten Energie der Woche darzustellen.	SOLL	System	
BE03		Das System soll fähig sein, eine Statistik zur gekurbelten Energie der ganzen Zeit darzustellen.	SOLL	System	
BE04		Das System soll fähig sein, Nutzer Highscores über einen längeren Zeitraum abzuspeichern.	SOLL	System	

(Beschreibung Features)

Detailanforderungen:

- Kurzbeschreibung der Features (copy/paste des Texts aus Kapitel in der Produktvision).
- Übersicht auf Nutzungsszenarien (UML Use Case-Diagramm; Für die Diagramme steht das DrawIO-Makro zur Verfügung).
- Katalog von Nutzungsszenarien (Use Cases als Tabellen mit Use Case-Attributen und strukturiertem Text; der strukturierte Text kann auch als UML Aktivitätsdiagramm modelliert werden).
- Wo relevant: verweisen Sie auf die graphischen Nutzerschnittstellen/Wireframes, welche Sie zusammen mit dem Usability-Ingenieur entwickeln.
- Katalog von detaillierten Anforderungen für die in den Use Cases erwähnten Systemfunktionalitäten (zB spezifiziert als User Stories). Diese detaillierten Anforderungen

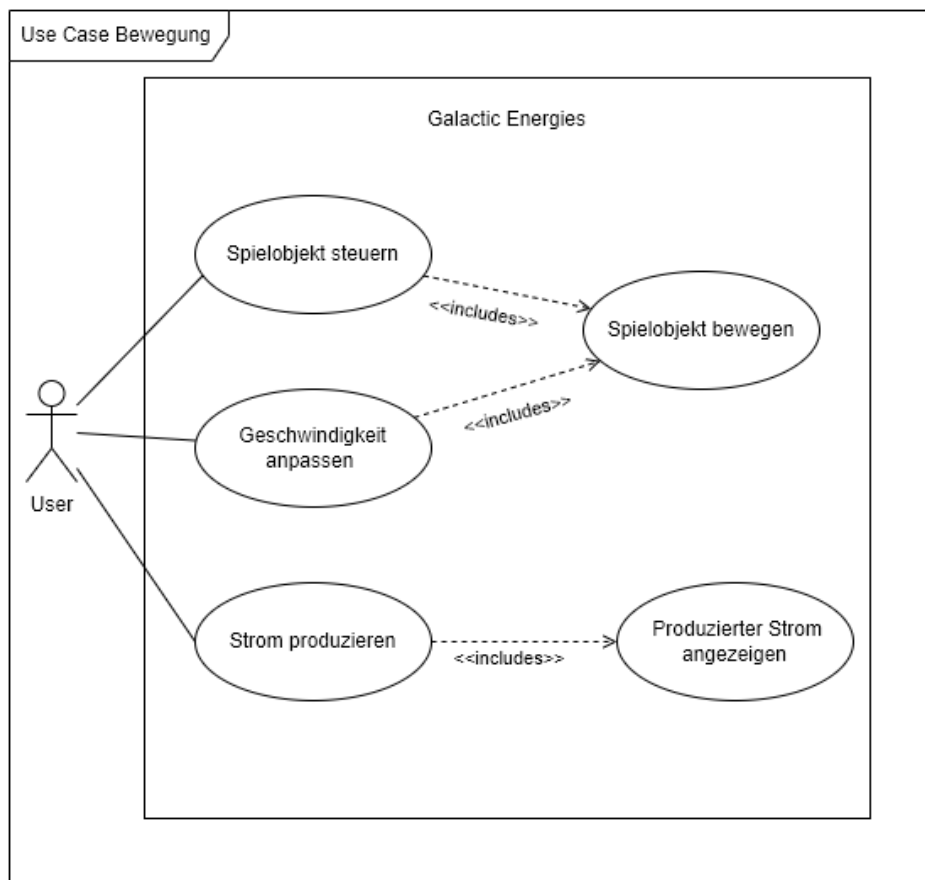
werden üblicherweise in Jira zur Planung und Überwachung der Implementierung übernommen.
- Feature-spezifische Qualitätsanforderungen, wo möglich quantifiziert in Zusammenarbeit mit dem Tester.

Feature 1: Bewegung

Kurzbeschreibung

Die Bewegung des Spielobjekts ist ein Muss und geschieht durch die Eingabe des Users. Die Eingabe wird durch Kurbeln und dem Joystick getätigt. Beim Kurbeln hat die Schnelligkeit/Geschwindigkeit einen Einfluss auf die Bewegung vom Spielobjekt. Die Geschwindigkeit des Spielobjekts passt sich dem Kurbelgeschwindigkeit an. Zudem wird die gekurbelte Geschwindigkeit abgemessen und stellt die produzierte Energie dar, welche gebraucht wird, um das Spielobjekt zu bewegen. Die momentane und gesamthafte erzeugte Energie soll auf dem Bildschirm, bzw. im Spiel angezeigt werden. Diese Anzeige dient als Vermittlung für die Information über die Stromproduktion. Das Spielobjekt lässt sich mit dem Joystick nach oben und unten bewegen - Kurbeln beeinflusst nur die Geschwindigkeit. Spielspass kann somit gefordert werden.

UML



Attribut	Wert
ID	F01
Name	Bewegung
Akteur	User
Voraussetzungen	User will spielen.
Hauptszenario	<ol style="list-style-type: none"> 1. User dreht an der Kurbel 2. Spielobjekt passt sich dem Kurbelgeschwindigkeit an 3. Mit der erzeugten Energie bewegt sich das Spielobjekt vorwärts (nach rechts) 4. Erzeugte Energie wird am Bildschirm angezeigt 5. User bedient den Joystick 6. Spielobjekt bewegt sich nach oben oder unten
Nachbedingungen	Spielobjekt wird bedient.
Alternative Szenarien	/
Ausnahmeszenario	<p>1a. Kurbel lässt sich nicht drehen.</p> <p>5a. Joystick lässt sich nicht bedienen.</p>

Sub-Features

ID	Sub-Feature	Wichtigkeit	Zusätzliche Funktionalität
F01.1	Geschwindigkeitsanpassung	Muss	Das Spielobjekt passt seine Geschwindigkeit der Kurbel-Geschwindigkeit an. Wenn nicht gekurbelt wird, kann sich das Spielobjekt nicht bewegen.
F01.2	Steuerung	Muss	Das Spielobjekt bewegt sich entsprechend der Eingabe des User nach oben oder unten.
F01.3	Spielfeld	Muss	Ein logisches Spielfeld wird angezeigt. Alle Elemente, welche Teil des laufenden Spiels sind, befinden sich auf dem Spielfeld.
F01.4	Stromanzeige	Soll	Der momentan produzierte Strom wird vom Strommesser ausgelesen und auf dem Spielfeld angezeigt.

F01.5	Lenkverhalten	Kann	Das Lenkverhalten passt sich der Geschwindigkeit an. Wenn das Spielobjekt langsam fliegt, wird die Lenkung träge.
-------	---------------	------	---

Detaillierte Anforderungen

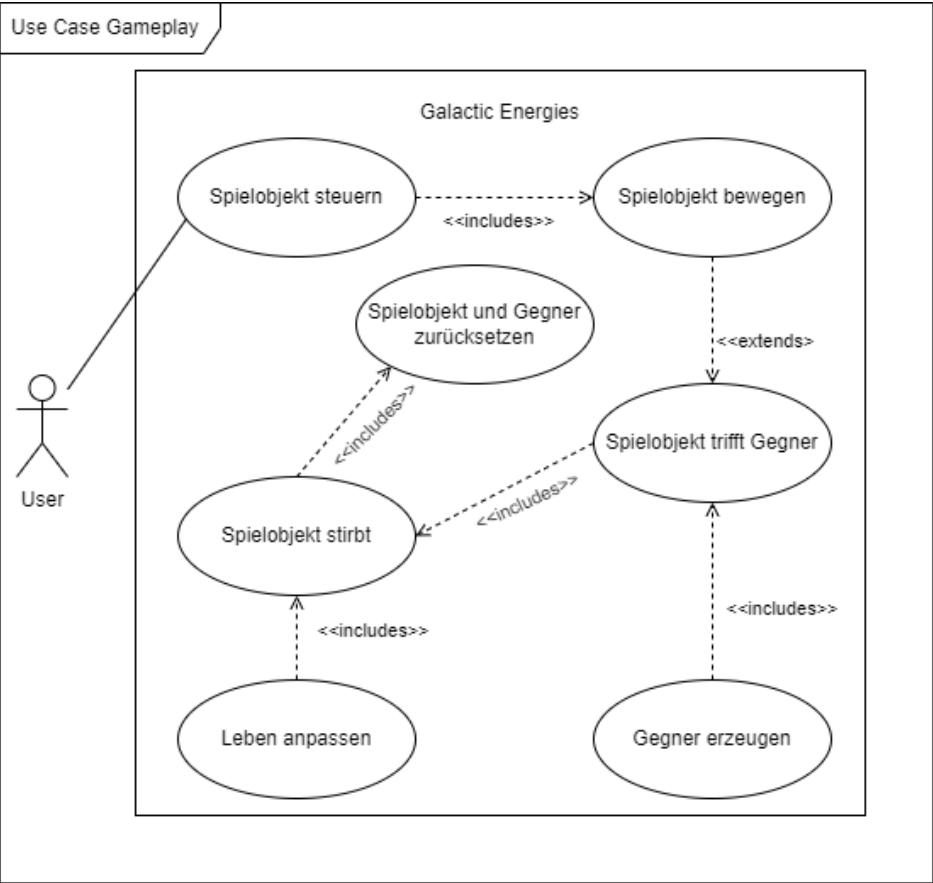
Detaillierte Anforderungen - F01 Bewegung			
ID	Feature ID	Beschreibung der Anforderung	Verbindlichkeit
F01A01	F01.1	Die Kurbel-Geschwindigkeit muss eingelesen werden.	MUSS
F01A02	F01.1	Das Spielobjekt muss sich der eingelesenen Geschwindigkeit anpassen können.	MUSS
F01A03	F01.1	Das Spielobjekt kann sich nicht bewegen, wenn nicht gekurbel wird.	KANN
F01A04	F01.2	Das System muss fähig sein, Inputs das Hardwareelement Joystick zu verarbeiten.	MUSS
F01A05	F01.2	Das Spielobjekt muss sich bis zum Rand des Spielfelds nach oben oder unten bewegen.	MUSS
F01A06	F01.2	Das Spielobjekt muss sich entsprechend des Joysticks nach oben oder unten bewegen.	MUSS
F01A07	F01.3	Das System muss fähig sein, ein Spielfeld anzuzeigen.	MUSS
F01A08	F01.3	Das System muss alle Elemente auf dem Spielfeld anzeigen können.	MUSS
F01A09	F01.3	Das System muss fähig sein, das Spielfeld passend an den Bewegungen der Elemente zu aktualisieren.	MUSS
F01A10	F01.3	Das System muss die aktuelle Anzeige erzeugen können.	MUSS
F01A11	F01.3	Das System muss die korrekte Anzeige erzeugen können.	MUSS
F01A12	F01.4	Die produzierte Energie soll vom Strommesser eingelesen werden.	SOLL
F01A13	F01.4	Die produzierte Energie soll im Spielfeld angezeigt werden.	SOLL
F01A14	F01.4	Das System soll den produzierten Strom anzeigen können.	SOLL
F01A15	F01.4	Das System soll die korrekten Angaben anzeigen können.	SOLL
F01A16	F01.5	Das Lenkverhalten vom Spielobjekt kann sich der Geschwindigkeit anpassen.	KANN
F01A17	F01.5	Das Spielobjekt kann sich langsam steuern lassen, wenn die Geschwindigkeit niedrig ist.	KANN
F01A18	F01.5	Das Spielobjekt kann sich schneller steuern lassen, wenn die Geschwindigkeit hoch ist.	KANN

Feature 2: Gameplay

Kurzbeschreibung

Die Gameplay ist für die Funktionalität des Spieles zuständig. Wenn ein Spiel keine Funktionalität hat, kann es nicht als ein Spiel angesehen werden und würde den Spielspass nicht anfeuern. Der Spielspass kann durch den Gegner und Leben vom Spielobjekt gefördert werden. Der Gegner fördert die Motivation des Users, um zu kurbeln und das Spielobjekt zu bewegen. Mit der zur Verfügung stehenden Leben soll die Motivation noch weiter angefeuert werden. Somit würde der User den Spiel ernst nehmen und dementsprechend "alles" geben, um am Leben zu bleiben. Der Tod vom Spielobjekt wird mit Zusammenstoß vom dem Gegner verwirklicht. Durch den Tod vom Spielobjekt wird der Spiel aber nicht beendet, sondern das Spielobjekt kehrt zum letzten Checkpoint zurück. Es sollte auch möglich sein, das Spiel zu pausieren und zu stoppen.

UML



Attribut	Wert
ID	F02
Name	Gameplay
Akteur	User
Voraussetzungen	Spiel besteht.
Hauptszenario	<ol style="list-style-type: none">1. Gegner taucht im Bild auf.2. Gegner fliegt Richtung Spielobjekt.3. Mittels Joystick muss der User ausweichen und versuchen den nächsten Checkpoint zu erreichen.4. Joystick lässt sich nach oben und unten bewegen.
Nachbedingungen	Spielobjekt wird bedient.
Alternative Szenarien	<ol style="list-style-type: none">5. Spielobjekt kann nicht ausweichen und stirbt.6. Spielobjekt verliert ein Leben und wird zum vorherigen Checkpoint zurückgesetzt.

Ausnahmeszenario	6a. Spielobjekt hat alle Leben verloren und das Spiel wird beendet.
------------------	---

Sub-Features

ID	Sub-Feature	Wichtigkeit	Zusätzliche Funktionalität
F02.1	Gegner	Muss	Der Gegner taucht im Bildschirm auf und bewegt sich zufällig. Das Spielobjekt muss ausweichen.
F02.2	Zusammenstoss	Muss	Beim Zusammenstoss von Gegner und Spielobjekt stirbt das Spielobjekt.
F02.3	Angriffsfunktion der Gegner	Kann	Das Spielobjekt kann durch den Gegner angegriffen werden. Das Spielobjekt muss nach oben oder unten ausweichen.
F02.4	Leben	Soll	Das Spielobjekt soll drei Leben haben. Beim Tod wird jeweils ein Leben abgezogen.
F02.5	Reset auf Checkpoint	Soll	Mit jedem Tod soll das Spielobjekt zurück zum letzten Checkpoint gesetzt werden. Die gesammelte Energie ab diesem Checkpoint wird ebenfalls zurückgesetzt.
F02.6	Pause & Stop	Nach Absprache mit Primeo - Entfernt	Das Spiel soll sich pausieren und stoppen lassen.
F02.7	Stillstand	Kann	Bei Stillstand kann das Spielobjekt sterben.

Detaillierte Anforderungen

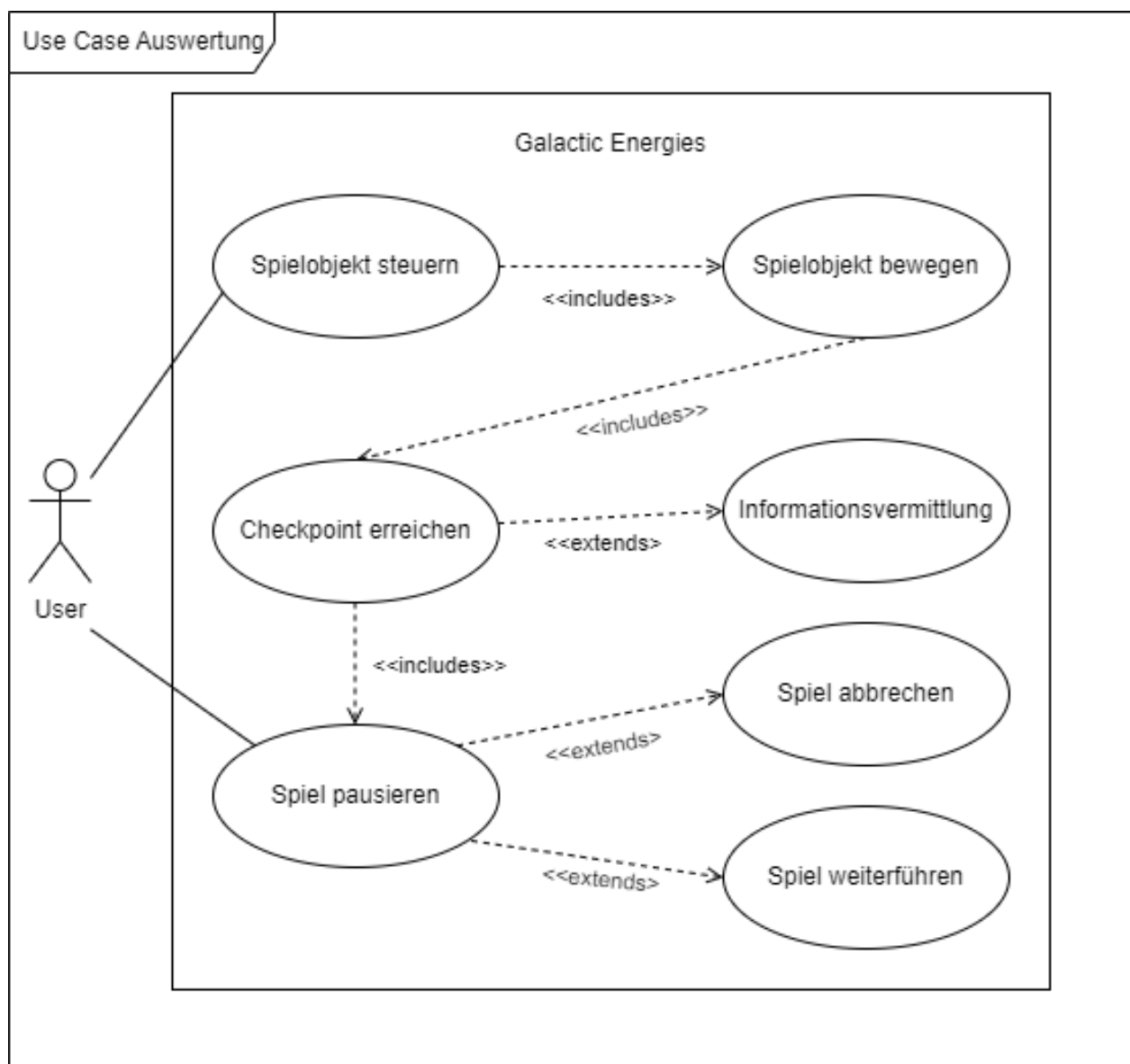
Detaillierte Anforderungen - F02 Gameplay		
ID	Feature ID	Beschreibung der Anforderung
F02A01	F02.1	Der Gegner muss generiert werden.
F02A02	F02.1	Der Gegner muss angezeigt werden.
F02A03	F02.1	Der Gegner hat eine zugällige Flugbahn.
F02A04	F02.2	Der Zusammenstoss zwischen Gegner und Spielobjekt muss verarbeitet werden können.
F02A05	F02.3	Der Gegner kann das Spielobjekt angreifen.
F02A06	F02.4	Das Spielobjekt soll drei Leben haben.
F02A07	F02.4	Ein Leben vom Spielobjekt soll beim Tod abgezogen werden.
F02A08	F02.5	Nach dem Tod soll das Spielobjekt zum letzten Checkpoint zurückgesetzt werden.
F02A09	F02.5	Die gesammelte Energie soll zurückgesetzt werden.
F02A10	F02.5	Alle Gegner sollen entfernt werden.
F02A11	F02.6	Das Spiel soll über Buttons verfügen.
F02A12	F02.6	Das Spiel kann pausiert werden.
F02A13	F02.6	Das Spiel kann gestoppt (abgebrochen) werden.
F02A14	F02.6	Bei Pause sollen alle Objekte eingefroren werden.
F02A15	F02.7	Bei Stillstand soll das Spielobjekt nach einer gewissen Zeit sterben.

Feature 3: Auswertung

Kurzbeschreibung

Die Auswertung eines Spieles ist wichtig, da sie die Motivation der User zeigt. Durch die Auswertung ist es möglich zu erkennen, welche User wie weit kommen konnten und strebt somit die Wettbewerbsstrategie zwischen den Usern. Mittels Checkpoint ermöglicht es den Usern an Wissen zu eignen. Kurz gefasst, ermöglicht die Auswertung den Projektziel zu erreichen, d.h. dass die Zielgruppe Energie-/Strombewusster ist. Beim Erreichen einer Checkpoint werden Information von Alltagsgegenständen aufgezeigt.

UML



Attribut	Wert
ID	F03
Name	Auswertung
Akteur	User
Voraussetzungen	Spiel wurde gespielt.
Hauptszenario	<ol style="list-style-type: none"> 1. Checkpoint wird erreicht. 2. Informationen werden angezeigt. 3. Spiel wird beendet. 4. Highscores werden angezeigt.
Nachbedingungen	Checkpoint wurde erreicht, Highscore wurde geknackt.
Alternative Szenarien	/
Ausnahmeszenario	1a. Spielobjekt konnte keinen Checkpoint erreichen.

Sub-Features

ID	Sub-Feature	Wichtigkeit	Zusätzliche Funktionalität
F03.1	Checkpoint	Muss	Das Spielobjekt muss Checkpoints erreichen. Beim Erreichen wird das Spiel gestoppt und eine Information über ein Gegenstand ausgegeben.
F03.2	Mehrere Gegenstände	Kann	Die Checkpoint-Information kann Informationen über mehrere Gegenstände ausgeben.
F03.3	Highscores	Soll	Ende des Spiels soll eine Liste mit Highscores ausgegeben werden.
F03.4	Gesamtauswertung	Soll	Das Spiel soll eine Gesamtauswertung der gesammelten Energie im Vergleich zu Alltagsgegenständen ausgeben.
F03.5	Namenseingabe	Nach Absprache mit Primeo - Entfernt	Beim Erreichen eines Highscores soll es möglich sein, den Usernamen einzugeben.

Detaillierte Anforderungen

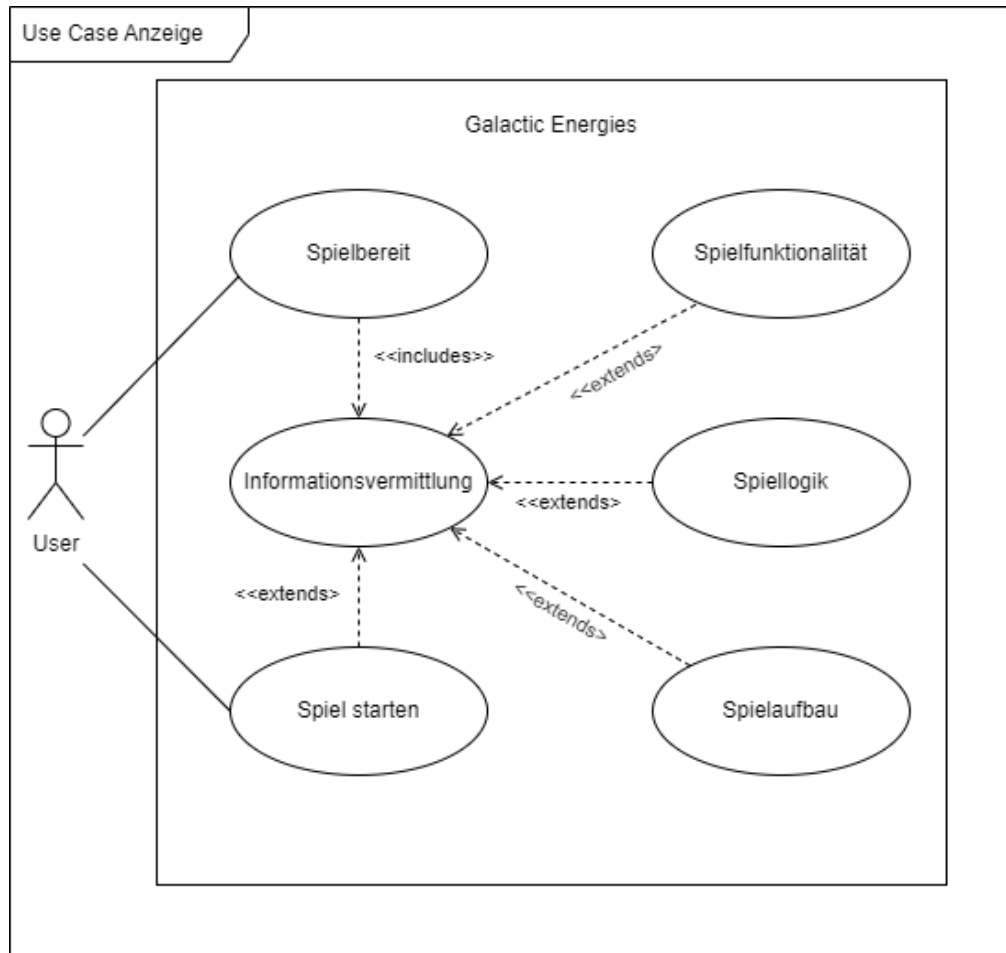
Detaillierte Anforderungen - F03 Auswertung		
ID	Feature ID	Beschreibung der Anforderung
F03A01	F03.1	Die Checkpoints müssen existieren.
F03A02	F03.1	Bei einer erreichten Checkpoint muss das Spiel gestoppt werden.
F03A03	F03.1	Nach Erreichen muss eine Information ausgegeben werden.
F03A04	F03.2	Bei den Checkpoints können auch Informationen über mehrere Gegenstände ausgegeben werden.
F03A05	F03.2	Es können 2 - 3 Geräte aus dem Alltagsverbrauch ausgegeben werden.
F03A06	F03.3	Am Ende des Spiels sollen Highscores ausgegeben werden.
F03A07	F03.3	Die Highscores solle in einer Liste angezeigt werden.
F03A08	F03.4	Die Ausgabe von einer Gesamtauswertung soll möglich sein.
F03A09	F03.4	Die Gesamtauswertung soll den Energievergleich von einer Alltagsgegenstand aufzeigen.
F03A10	F03.4	Die Gesamtauswertung soll den Energievergleich von mehreren Alltagsgegenständen aufzeigen.
F03A11	F03.4	Die Gesamtauswertung der gesammelten Energie soll in einer bestimmten Zeit aufgezeigt werden.
F03A12	F03.4	Die Gesamtauswertung soll Informationen über den Energieverbrauch ausgeben.
F03A13	F03.5	Die Namenseingabe soll möglich sein.
F03A14	F03.5	Die Namenseingabe soll bei einer neu erreichten Highscores möglich sein.
F03A15	F03.5	Für die Namenseingabe soll eine Tastatur angezeigt werden.
F03A16	F03.5	Die Eingabe soll über den Joystick ermöglicht werden.
F03A17	F03.5	Die Eingabe soll mit einem Button bestätigt werden.

Feature 4: Anzeige

Kurzbeschreibung

Die Anzeige des Spieles spiegelt die geleistete Arbeit, investierte Zeit und die Motivation der User. Die Anzeige ermöglicht den Informationsfluss, wie die Funktionalität des Spieles, den Spiellogik und Spielaufbau. Somit erhalten die User ebenfalls die Einführung ins Spiel. Auf der Anzeige ist der Spielstart ersichtlich.

UML



Attribut	Wert
ID	F04
Name	Anzeige
Akteur	User
Voraussetzungen	Spiel ist vorhanden.
Hauptszenario	<ol style="list-style-type: none"> 1. Anzeige startet. 2. Informationen werden vermittelt. 3. Spiel startet.
Nachbedingungen	Hardware funktioniert.
Alternative Szenarien	/

Ausnahmeszenario	1a. Keine Anzeige vorhanden. / Anzeige startet nicht.
------------------	---

Sub-Features

ID	Sub-Feature	Wichtigkeit	Zusätzliche Funktionalität
F04.1	Information	Muss	Die Spielfunktionalität, Spiellogik und Spielaufbau werden angezeigt/vermittelt.
F04.2	Statistik des Tages	Kann	Die gekurbelte Energie während eines Tages kann angezeigt werden.
F04.3	Statistik der Woche	Kann	Die gekurbelte Energie während einer Woche kann angezeigt werden.
F04.4	Statistik der gesamten Zeit	Kann	Die gekurbelte Energie im Total des Spiels kann angezeigt werden.
F04.5	Spiel starten	Muss	Das Spiel muss gestartet werden können.

Detaillierte Anforderungen

Konzeptuelles Modell

UML Klassendiagramm zur Beschreibung der für die Nutzenden wichtigen Konzepte mit Beschreibung der einzelnen Klassen.

Für die wichtigen Klassen: Verhalten (UML Zustandsdiagramm).

Qualitätsanforderungen

Liste von global gültigen Qualitätsanforderungen, wo möglich quantifiziert in Zusammenarbeit mit dem Tester.

Anhänge

Zukünftige Anforderungen

Hier dokumentieren Sie eine Liste der gewünschten Verbesserungen oder Veränderungen am Produkt, welche nach Ihrem Projekt realisiert werden sollen.

Alles Andere

Verweisen Sie hier auf Dokumente, Modelle, Photos, Videos, etc., welche zum Verstehen der Anforderungen oder als Evidenz zur Korrektheit der Anforderungen genutzt werden können.