



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Projekt Rocket

Anforderungsdokument

Status	Zur Verifizierung freigegeben
Name des Projektes	Rocket
Projektleitung	Jürgen Eckerle
Projektauftraggeber	Jürgen Eckerle
Autoren (alphabetisch)	Martin Käser, Fabian Schwab, Marcel Tschanz
Initialen	kasem5, schwf5, tschm23
Genehmigung durch	Rolf Gasenzer

Anforderungsdokument Projekt „Rocket“

Versionen

Version	Datum	Beschrieb	Autoren
1.0	13.10.2014	Initialdokument fertig gestellt	kasem5, schwf5, tschm23
1.1	16.10.2014	Änderungen im Bereich der Anforderungen	kasem5, schwf5, tschm23
1.2	20.10.2014	Erweiterung durch Aufgabe 2	kasem5, schwf5, tschm23
1.21	29.10.2014	Überarbeitung, div. Ergänzungen	kasem5, schwf5, tschm23
1.3	05.11.2014	Erw. von: Haupt-/Nebenzielen, Use-Cases	kasem5, schwf5, tschm23
1.4	12.11.2014	Überarbeitung gemäss Meeting mit Rolf Gasenzer	kasem5, schwf5, tschm23
1.4.1	20.11.2014	Überarbeitung der Anforderungen	kasem5, schwf5, tschm23
1.4.2	23.11.2014	Überarbeitung der Anforderungen	kasem5, schwf5, tschm23
1.4.3	24.11.2014	Überarbeitung der Anforderungen	kasem5, schwf5, tschm23
1.4.4	25.11.2014	Überarbeitung der Anforderungen	kasem5, schwf5, tschm23
1.5	11.12.2014	Satzschablone für Anforderungen bearbeitet	kasem5, schwf5, tschm23
1.5.1	14.12.2014	Satzschablone für Anforderungen bearbeitet, Satzstellung der Teilziele umformuliert, Passivsätze eliminiert	kasem5, schwf5, tschm23
1.5.2	15.12.2014	Nichtfunktionale Anforderungen überarbeitet	kasem5, schwf5, tschm23
1.5.3	17.12.2014	Finalizing	kasem5, schwf5, tschm23

Table of Contents

1	Einleitung.....	4
1.1	Ziel des Dokuments.....	4
1.2	Leserkreis des Dokuments	4
2	Projektbeschreibung	4
2.1	Stakeholderliste	5
2.2	Nutzer- und Zielgruppen	5
2.3	Projektmethode	5
2.4	Technische Ressourcen	5
2.5	Dokumente	5
3	Projektziele	6
3.1	Name des Projekts und Hauptziel (HZ1).....	6
3.2	Teilziele (TZ1-9)	7
4	Scoping.....	9
4.1	Rahmenbedingungen (RB).....	9
4.1.1	Technische Rahmenbedingungen RB1-6.....	9
4.2	Systemkontext und Systemgrenzen.....	10
4.3	Out of scope	10
5	Anforderungen	11
5.1	Quellen und Herkunft.....	11
5.2	Anforderungslisten.....	11
5.2.1	Legende und ergänzende Hinweise.....	11
5.3	Funktionale Anforderungen.....	12
5.3.1	Detailbeschreibung der funktionalen Anforderungen.....	13
5.4	Qualitätsanforderungen (Nichtfunktionale Anforderungen)	16
5.4.1	Detailbeschreibung der nicht-funktionalen Anforderungen	16
6	Rollenkonzept.....	17
7	Glossar.....	18
7.1	Erklärungen und Übersetzungen	18
7.2	GUI.....	19
7.3	Synonyme	19
8	Referenzen.....	20

1 Einleitung

Das zu realisierende Spiel mit dem Namen „Rocket“ und die in diesem Zusammenhang entstehenden Artefakte müssen den Anforderungen des Projekts gerecht werden.

Die Mitarbeiter am Projekt kommunizieren deshalb fortgehend mit dem Projektverantwortlichen, um die notwendigen Informationen und Anforderungen elaborieren zu können. Auf der Gegenseite sind die Mitarbeiter verpflichtet, die Vorgaben originalgetreu umzusetzen.

1.1 Ziel des Dokuments

Die Ziele und Anforderungen an das Projekt, an die einzusetzenden Technologie und die Hilfsmittel müssen vorgängig dokumentiert werden. Im Verlaufe des Projektes wird man so erkennen können, ob man auf dem richtigen Weg ist und kann abschliessend auch eine sinnvolle Evaluierung durchführen.

1.2 Leserkreis des Dokuments

Das vorliegende Dokument steht der Öffentlichkeit zur freien Einsicht zur Verfügung. Primär richtet es sich an die Dozierenden der Module BTI7082q und BTI7301p, im Rahmen des fünften Semesters der q-Klasse an der BFH in Bern.

2 Projektbeschreibung

Zielsetzung	Ein 3D-Adventure-Spiel mit der Unity-Game-Engine entwickeln
Technologien	Unity Game Engine, Blender, C# / UnityScript, Automaten, MonoDevelop, Github als Versionisierungstool
Zeitressourcen	24 Stunden geführter Unterricht, 216 Stunden Selbststudium

Entwicklung einer 3D-Adventure-Game Basis (Minimum an Levels & Funktionen) mit Hilfe der Unity3D Engine. Der Spielzustand und der Zustand der Agenten soll durch Automaten modelliert werden. Das Verhalten der Agenten resp. deren Intelligenz wird über die JavaScript nahe Sprache UnityScript und C# realisiert. Die Agenten passen sich der jeweiligen Spielsituation an. Ein Aufbau von Wissen im Bereich der Automatentheorie ist dafür Voraussetzung.

Für die Spielidee und das Konzept wird dem Team freie Hand gelassen. Rücksprache mit der Projektleitung bzw. mit dem Projektauftraggeber garantiert, dass die Spielidee und das Konzept den Erwartungen und Vorstellung gerecht werden.

2.1 Stakeholderliste

- Jürgen Eckerle als leitender Dozent und somit Projektauftraggeber
- Mitarbeitende am Projekt Rocket, namentlich:
Martin Käser
Fabian Schwab
Marcel Tschanz

2.2 Nutzer- und Zielgruppen

- Testpersonen
- Spieler von Computerspielen und Interessierte an unserem Projekt als operative Anwender
- Entwickler, die unseren Prototypen weiterentwickeln

2.3 Projektmethode

- Das Projekt wird anlehnend an SCRUM agil entwickelt.

2.4 Technische Ressourcen

- Unity Editor (Entwicklungsumgebung, Unity-Tutorials, UnityScript API)
- UnityScript, Monodevelop
- Automatentheorie („Artificial Intelligence for Games“)
- Blender (Gestaltung 3D Modelle)
- GitHub (FileShare und Sicherheitskonzept für Projekt)
- Laptops der Projektmitglieder (Testen und Anwendung des Adventures)

2.5 Dokumente

- Projektdokumentation
- Anforderungsdokument
- Projektzeitplan
- Back- / Sprintlog mit den untereinander abhängigen Tasks
- Diary (Kurzbeschreibung der Arbeitsaufwände)
- Arbeitsjournal (Ausführlichere Informationen zu den Tasks)
- Code Repository

3 Projektziele

3.1 Name des Projekts und Hauptziel (HZ1)

Projekt: „ROCKET“

HZ1: Eine spielbare *Alpha-Version** eines Adventuregames, dessen *Basislevel** und die implementierte, künstliche Intelligenz der Gegner durch selbstgeschriebene Spielklassen offen für Erweiterungen in Komplexität und Umfang bleiben. Eine Kurzgeschichte begleitet den Spieler bei seinen Handlungen und sorgt für zusätzliche Unterhaltung.

*Glossareintrag

3.2 Teilziele (TZ1-9)

Kursive Wörter sind im Glossar näher beschrieben.

Termine: Präsentation Alpha-Release 09.01.2015
Projektabschluss 16.01.2015

Teilziel-code	Beschreibung der Teilziele	Termin
TZ1	Ziel von TZ1 ist es, mit der Alphaversion eine Klassenhierarchie zur Verfügung zu stellen, die einen einfachen Ausbau des Basislevels durch zusätzliche Spielkomponenten wie Räume und Gegenstände ermöglicht und die ein Aussenstehenden mit Erfahrung im Umgang mit Unity ohne zusätzliche Einführung erweitern kann.	16.01.2015
TZ2	Ziel von TZ2 ist es, mindestens einen nicht spielbaren Gegner ins Spielszenario einzubauen. Dieser computergesteuerte Gegner nimmt Einfluss auf den Schwierigkeitsgrad und den Ausgang des Spiels, indem er den Spieler aufsucht und am Verlassen des Levels hindert.	09.01.2015
TZ3	Ziel von TZ3 ist es, den Langzeitspielspass aufrecht zu erhalten. Bei jedem Level-Neustart sind die Zahlenkombinationen der Rätsel neu zu generieren. So bleibt das Basislevel auch für Spieler interessant, die das Level bereits einmal absolviert haben.	16.01.2015
TZ4	Ziel von TZ4 ist es, durch vom Spieler ausgeführte Aktionen in einzelnen Räumen des Levels andere Bereiche frei zu schalten. So entsteht eine logische Abhängigkeit zwischen den Einrichtungen. Um den Spielerfolg zu garantieren, ist der Besuch jeder dieser Räume notwendig.	09.01.2015
TZ5	Ziel von TZ5 ist es, die nicht sichtbare Spielfigur (<i>First-Person-Ansicht</i>) aufrecht gehend durch das Level steuern zu können. Ihr Sichtfeld entspricht damit dem Sichtwinkel einer aufrecht gehenden Person. Als Spieler besitzt man die Möglichkeit, seine Figur in zwei verschiedenen Geschwindigkeiten fortzubewegen.	09.01.2015
TZ6	Ziel von TZ6 ist es, dank Lade-und Speichermöglichkeiten ein Spiel zu verschiedenen Zeitpunkten unterbrechen und wieder aufnehmen zu können. Das ermöglicht auch eine komplexe Gestaltung des Levels, weil dadurch nicht die dem Spieler verfügbare Zeit kein mitbestimmender Faktor mehr ist.	16.01.2015
TZ7	Ziel von TZ7 ist es, dem Spieler die Möglichkeit zu geben, dafür vorgesehene Gegenstände in seiner Reichweite an sich zu nehmen und in einem Inventar abzulegen. Das Spielinventar kann jederzeit durchsucht werden. Ein bestimmter Gegenstand ist von zentraler Bedeutung und wird für das Verlassen des Levels benötigt.	09.01.2015

Teilziel-code	Beschreibung der Teilziele	Termin
TZ8	Ziel von TZ8 ist es, bereits mit dem Basislevel eine einfache Spielstory einzuführen. Über eine Startsequenz erhält der Spieler erste Informationen zu Spielkonzept und Spielinhalt. Während des Spiels, wird die Spielgeschichte zu weiter ausgeführt, um sie gleichzeitig mit dem erfolgreichen Verlassen des Levels durch den Spieler in sich abzuschliessen.	16.01.2015
TZ9	Ziel von TZ9 ist es, die Spielstatistik (Dauer des Spieldurchgangs und erreichte Punkte) nach dem Spielabschluss an eine zentrale Stelle zu senden. Eine Gesamtstatistik soll verfügbar sein, damit der Spieler sich mit anderen Spielern indirekt messen und seine erreichte Punktzahl mit fremden Werten vergleichen kann.	09.01.2015

4 Scoping

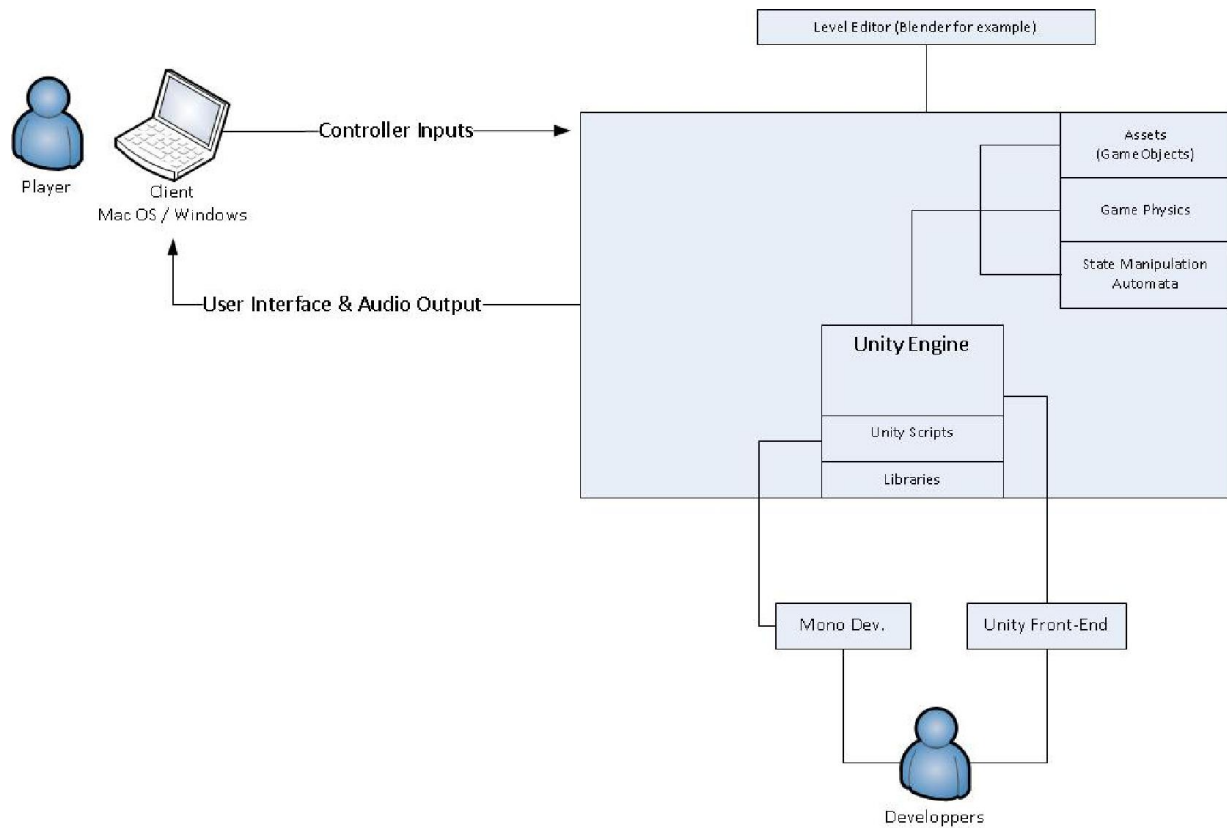
4.1 Rahmenbedingungen (RB)

4.1.1 Technische Rahmenbedingungen RB1-6

RB1	Als Programmiersprache ist die objektorientierte Sprache C# zu verwenden
RB2	Zielplattformen sind Windows (ab Windows 7) und MacOS (ab OSX)
RB3	Als Laufzeit- und Entwicklungsumgebung ist das Produkt „Unity 3D“ in der Version 4.5.4.f1 oder höher zu verwenden.
RB4	Als Entwicklungsumgebung ist der Editor „Mono-Develop“ zu verwenden. „Mono-Develop ist an Unity 3D gekoppelt und somit ebenfalls erforderlich.
RB5	Zustände der Agenten müssen mit Zustandsmaschinen (hierarchisch oder einfach endlich) modelliert werden und müssen sich gegenseitig beeinflussen
RB6	Die Spielkomponenten kommunizieren untereinander über ein Messaging System, wie es für Spiele üblich ist und in <i>Buckland (Referenz, ID 2)</i> ab S. 69 beschrieben wird

4.2 Systemkontext und Systemgrenzen

Grobe Architektur des 3-D-Adventures Rocket



4.3 Out of scope

Der Fokus liegt nicht auf der Entwicklung eigener Grafiken, 3D-Modellen oder Audiodateien. Das Spiel soll ausserdem keine Rennsimulation werden. Auch ein Shooterspiel ist nicht Ziel der Entwicklung. Das Spiel enthält kein Mehrspielermodus und kann nicht über das Netzwerk oder über das Internet gespielt werden.

Die Skripte zu den Grundfunktionalitäten was Bewegungen der Spielfigur betreffen werden nicht direkt verändert. An der Konfiguration des Windows- oder Mac-Clients, auf dem das Spiel gespielt wird werden keine Änderungen vorgenommen.

5 Anforderungen

5.1 Quellen und Herkunft

JE: Projektauftraggeber Jürgen Eckerle
Team: Projektmitarbeiter Martin Käser, Fabian Schwab, Marcel Tschanz im Plenum

Wo das Team als Quelle angegeben wird, ist dies als Artefakt der Gruppenarbeit zu interpretieren und wurde jeweils von allen Mitgliedern und dem Projektauftraggeber validiert.

5.2 Anforderungslisten

5.2.1 Legende und ergänzende Hinweise

Nr.

Die Kennung der Anforderung, die im restlichen Dokument als Referenzcode dient.

Kurzbezeichnung

Ein grober Beschrieb der Anforderung. Wird in der dazugehörigen Detailbeschreibung jeweils genauer erläutert.

Status

In welchem Zustand sich die Anforderung befindet.

Mögliche Zustände sind: offen, geplant, in Arbeit, erledigt

P - Priorität Bewertung: 1(low), 2(medium), 3(high)

V= Variabilität Bewertung: 1(low), 2(medium), 3(high)

K= Komplexität Bewertung: 1(low), 2(medium), 3(high)

R= Risiko Bewertung : <8(low), 8-10(medium), >10(high)

Das aus {P;V;K} berechnete Risiko (Zahlenwert), ergibt sich durch Addition der gewichteten Eigenschaften Priorität (P), Variabilität (V) und Komplexität (K).

Zur Gewichtung der Eigenschaften hinsichtlich der Risikoabschätzung:

Priorität *2, Komplexität *2, Variabilität *1

Priorität und Komplexität werden also doppelt gewichtet.

Quelle

Aus welcher Quelle (Punkt 5.1) die Anforderung entstammt.

Datum

Wann diese Anforderung aufgenommen wurde.

Kursive Formatierung

Fachausdrücke, die im Glossar näher ausgeführt wurden, sind kursiv formatiert.

5.3 Funktionale Anforderungen

Nr.	Kurzbezeichnung	Status	P	V	K	R	Quelle	Datum
FR1	Allgemeines	Bezug zu Teilziel TZ8						
FR1.1	Startsequenz bei Spielstart	erledigt	3	1	1	med	Team	01.10.2014
FR1.2	Briefing des Spielers	erledigt	3	1	1	med	Team	01.10.2014
FR2	Level-Design	Bezug zu Teilziel TZ4						
FR2.1	Ein spielbares, durch Wände abgegrenztes Level mit fixem Start und fixem End-Punkt	erledigt	3	3	2	high	Team	24.09.2014
FR3	GUI / Menus	Bezüge zu Teilzielen TZ6, TZ9						
FR3.1	Spielsteuerung (Tastenbelegung) an eigene Bedürfnisse anpassen können	geplant	2	2	1	med	Team	09.01.2015
FR3.2	Speichern der Spielstände	erledigt	1	2	2	med	Team	24.10.2014
FR3.3	Laden der Spielstände	In Arbeit	1	2	2	med	Team	14.01.2015
FR3.4	Endsequenz mit anschliessender Statistik	geplant	2	3	2	high	Team	09.01.2015
FR4	Spielfigur	Bezug zu Teilziel TZ5						
FR4.1	Erweiterter Sichtradius durch Spezialkamera: „Um-die-Ecke-Sicht“	erledigt	3	2	2	high	Team	10.10.2014
FR4.2	Aufnahme von Gegenständen in das Inventar	erledigt	3	1	2	high	Team	10.10.2014
FR5	Gegner (Roboter)	Bezug zu Teilziel TZ2						
FR5.1	Der Robotergegner patrouilliert durch das Level	erledigt	3	2	3	high	Team	03.10.2014
FR5.2	Der Robotergegner löst einen Alarm aus, sobald er den Spieler entdeckt	erledigt	1	3	3	high	Team	03.10.2014
FR5.3	Roboter hat eine limitierte Energiereserve	erledigt	3	2	1	high	Team	15.10.2014
FR5.4	Roboter fängt den Spieler	In Arbeit	1	2	2	med	Team	05.01.2015

5.3.1 Detailbeschreibung der funktionalen Anforderungen

Kursiv formatierte Wörter oder Wortfolgen sind im Glossar näher erklärt.

	Detaillierte Beschreibung der funktionalen Anforderungen
FR1	Allgemeines
FR1.1	<p>Startsequenz: Hat der Spieler die „game.exe“ Datei geöffnet, startet das Spiel „Rocket“, und das System initiiert die Startsequenz, welche dem Benutzer am Bildschirm präsentiert wird. Diese Startsequenz blendet während drei (3) Sekunden nebst dem Spielelogo auch den Titel des Spiels auf dem Bildschirm ein.</p> <p>Auf den Ablauf der Sequenz kann der Spieler keinen Einfluss nehmen.</p> <p>Nach Ablauf der drei (3) Sekunden wechselt das System ins Hauptmenu des Spiels.</p>
FR1.2	<p>Briefing des Spielers: Hat der Spieler die Option „New Game“ aus dem Hauptmenu ausgewählt, stellt das System einen Text auf dem Bildschirm dar, welcher den Spieler über das zu erreichende Ziel des <i>Basislevels</i> informiert (=Briefing). Dieses Text-Briefing darf in seiner Länge eine Bildschirmseite nicht überschreiten.</p> <p>Beendet wird das Briefing über einen Button, der den Einstieg in Level1 einleitet.</p>
FR2	Level-Design
FR2.1	<p>Spielbares, durch Wände abgegrenztes Level: Die innerhalb des Projektes entwickelte spielbare Basisversion des Spiels umfasst bei der Freigabe der Alphaversion ein einzelnes Level. Dieses durch Wände begrenzte Level soll sich über eine einzige Ebene erstrecken und muss dem Spieler die Möglichkeit bieten, in mindestens drei verschiedene Räume mit seiner Spielfigur einzutreten. Dabei sollten sich die Räume gegenseitig im Verhalten beeinflussen. Vom Stakeholder J. Eckerle ist bezüglich der Räume eine solche minimale Komplexität gefordert.</p> <p>Beispiel: Schalter 1 aus Raum A öffnet Türe zu Raum B.</p> <p>Der Spieler beginnt das Level bei jedem Neustart des Spiels vom selben Startpunkt aus. Auch der Ort, an dem die Spielfigur das Level verlässt soll einzigartig sein.</p>
FR3	GUI / Menus
FR3.1	<p>Spielsteuerung manuell zu konfigurieren: Will der Spieler die Spielsteuerung seinen Wünschen anpassen, soll er im <i>Hauptmenu des Spiels</i> die Möglichkeit haben, die Tastenbelegung für die Spielsteuerung zu verändern. Er kann bei Bedarf jeder Bewegungsmöglichkeit der Spielfigur eine beliebige Taste zuweisen. Eine Taste kann dabei aber nur eine Funktion erfüllen und wird nur einmal zu belegen sein. Das System weist den Spieler auf mehrfach eingesetzte Tasten und nicht zugeordnete Aktionen hin.</p> <p>Innerhalb des Spiels sind die Einstellungen zur Spielsteuerung nicht mehr zu ändern.</p>
FR3.2	<p>Speichern der Spielstände: Will der Spieler das Spiel pausieren, so soll er über das <i>Pausenmenu</i> des Spiels die Möglichkeit besitzen, das aktuelle Spiel zu speichern.</p> <p>Zur Speicherung stellt das Spiel 10 Speichereinheiten zur Verfügung. Dies bedeutet, dass pro Spieler höchstens 10 verschiedene Spielstände vorhanden sein können. Sind diese 10 Speicherplätze belegt, muss ein bestehender Eintrag überschrieben werden.</p> <p>Ein einzelner Speichervorgang soll nicht mehr als zwei Sekunden in Anspruch nehmen.</p>

	Ein Spielstand wird immer mit der aktuellen Levelbezeichnung und dem aktuellen Datum und Uhrzeit versehen und ist so von allen anderen Spielständen eindeutig unterscheidbar.
FR3.3	<p>Laden der Spielstände: Will der Spieler gespeicherte Spielstände laden, muss ihm m Hauptmenu <u>und</u> dem Pausenmenu die Möglichkeit zum Laden eines Spielstandes zur Verfügung stehen. Wird eine Auswahl über „Load Game“ vom Spieler getroffen, lädt das System den gespeicherten Zustand.</p> <p>3.31) Aus dem Hauptmenu wird der Spielstand umgehend geladen und die Spielfigur befindet sich an der Position zum Zeitpunkt des Speichervorganges. Das Inventar und die Gegnerpositionen werden ebenfalls auf den gespeicherten Zustand gebracht.</p> <p>3.32) Aus dem Pausenmenu wird das System den Spieler fragen, ob er das aktuelle Spiel speichern will, bevor ein Spielstand geladen wird. Bei Bestätigung wird vor dem Ladevorgang nun ein neuer Eintrag in die Liste der gespeicherten Spielstände gemacht. Wird die Frage verneint, lädt das verhält sich das System analog 3.31</p>
FR3.4	<p>Endsequenz und Statistik: Hat der Spieler mit seiner Figur das Ende des Spiels erreicht, unterbricht der Spielfluss und es wird nicht weiter auf Eingaben des Spielers reagiert.</p> <p>Eine Nachricht auf dem Bildschirm informiert den Spieler darüber, dass er das Level erfolgreich abgeschlossen hat. Im direkten Anschluss erscheint eine kurze Film- oder Bildsequenz, die ihr Ende in einer tabellarisch dargestellten Statistik zur erreichten Punktzahl des Durchgangs findet. Sobald diese Statistik eingeblendet wird, muss der Spieler die Möglichkeit haben auszuwählen, ob er das Level neu starten oder das Spiel beenden möchte.</p> <p>Wählt er keine der Optionen, bleibt der Statistikbildschirm eingeblendet.</p>
FR4	Spielfigur
FR4.1	<p>Erweiterter Sichtradius „Sicht um die Ecke“: Will der Spieler mit seiner Spielfigur einen Blick „um-die-Ecke“ zu werfen, soll er dies mit einer eigens dafür zu belegenden Taste während des laufenden Spiels tun können. Diese Funktion dient dazu, Gegner auszumachen und/oder sich einen sicheren Überblick über den weiteren Verlauf der Spielewelt zu verschaffen. Diese Steuerungsfunktion bewirkt im eigentlichen Sinne ein Verschieben des Sichtfeldes der Spielfigur für 2-3 Sekunden nach links oder nach rechts (abhängig von der gedrückten Taste). In der Standardkonfiguration tätigt die Spielfigur den Blick nach rechts über die Taste R und den Blick nach links über die Taste Q. Wird also eine mit dieser Sicht-Funktion belegten Taste gedrückt, verschiebt sich das Sichtfeld automatisch um eine noch zu definierende Anzahl Einheiten nach links, respektive rechts und wieder zurück zur Ausgangsposition.</p> <p>Es spielt dabei für die Aktion keine Rolle, ob die Taste mehrmals, oder nur einmal gedrückt wird. Auch ein Halten der Taste hat nichts anderes als den beschriebenen Bewegungsablauf zur Folge.</p> <p>Diese fließende Bewegung, wird vom System selber durchgeführt, ohne Interaktionsmöglichkeit des Spielers während der Bewegung selbst.</p>

FR4.2	<p>Gegenstände in Inventar aufnehmen: Hat der Spieler mit seiner Spielfigur den <i>Aktionsradius eines interaktiven Elements</i> erreicht, muss das System ein Aktionsmenu mit entsprechenden Handlungsmöglichkeiten auf dem Bildschirm anzeigen. Aus diesen Möglichkeiten, wählt der Spieler „Gegenstand ins Inventar aufnehmen“. Das System fügt den Gegenstand in das Inventar des Spielers ein und der Gegenstand selbst verschwindet von seinem ursprünglichen Platz im Level.</p> <p>Die Interaktion durch die Spielfigur mit vordefinierten Gegenständen ist eine zentrale Funktion des Spiels und für den Spielerfolg unabdingbar.</p>
FR5	Gegner
FR5.1	<p>Robotergegner patrouilliert durch das Level: Die Rolle der gegnerischen Figur erhöht die Komplexität des Spielgeschehens und erschwert es dem Spieler, die Rätsel im Level zu lösen. Während dem ganzen Spiel muss sich der Robotergegner auf vordefinierten Pfaden fortbewegen. Er patrouilliert dabei am Boden und hält Ausschau nach dem Spieler. Die Bewegungen des Gegners und dessen Aktionsradius muss so modelliert sein, dass dem Spieler eine faire Chance bleibt, mit seiner Spielfigur dem Gegner auszuweichen oder sich vor ihm verstecken zu können. Dieser Gegner kann vom Spieler nicht kontrolliert oder zerstört werden.</p>
FR5.2	<p>Der Robotergegner löst einen Alarm aus, sobald er den Spieler entdeckt: Wenn sich ein Spieler mit seiner Spielfigur innerhalb des Sichtradius des Robotergegners befindet, wird ein akustischer Alarm ausgelöst, welcher die bis dahin gespielte Spielmusik ersetzt.</p> <p>Der Spieler weiss so genau, wenn er für die Gegner sichtbar ist. Der Alarm-Ton verstummt und wird durch die normale Spielmusik ersetzt, sobald die Spielfigur für 2 Sekunden nicht das Robotersichtfeld betreten hat.</p>
FR5.3	<p>Roboter hat eine limitierte Energiereserve: Innerhalb des Levels stehen Ladestationen an jeweils gegenüberliegenden Seiten des Levels. Der Roboter berechnet im Abstand von 10 Millisekunden, ob sein Batteriestand noch ausreichend ist, um die nächste Ladestation zu erreichen, bevor die Energiereserve auf 0 fällt. Der Roboter berechnet die nächste Ladestation und begibt sich zu dieser, um seine Batterie wieder auf 100 Einheiten aufzuladen.</p>
FR5.4	<p>Roboter berührt den Spieler: Kommt der Roboter während der Verfolgung des Spielers auf 2 m (<i>Unity Einheit</i>) an den Spieler heran, ist das Spiel frühzeitig beendet und die Schlusssequenz von FR3.4 erscheint.</p>

5.4 Qualitätsanforderungen (Nichtfunktionale Anforderungen)

Es gilt die Legende aus Punkt 5.2.1

Nr.	Kurzbezeichnung	Status	P	V	K	R	Quelle	Datum
NF1	Konzeptionelles	Bezug zu Teilziel TZ1, TZ7						
NF 1.1	Basis für Weiterentwicklungen	in Arbeit	3	2	3	high	JE	16.01.2015
NF 2.1	Beschreibungen, Fundorte und Einsatzmöglichkeiten zu allen Gegenständen sind schriftlich festzuhalten (Prozessanforderung)	erledigt	2	3	1	med	Team	01.11.2014
NF2	Rechtliches	Bezug zu Teilziel TZ7						
NF 2.1	Urheberrechte & Gesetze sind bei verwendeten Assets zu berücksichtigen	in Arbeit	3	1	2	high	Team	16.01.2015

5.4.1 Detailbeschreibung der nicht-funktionalen Anforderungen

	Kurzbezeichnung
NF1	Konzeptionelles
NF 1.1	Basis für Weiterentwicklungen: Beim Bedarf eines zusätzlichen spielbaren Levels, soll es das Klassenkonzept einem aussenstehenden Programmierer ermöglichen, innerhalb eines Arbeitstages ein weiteres Level implementieren zu können. Dies setzt voraus, dass der Programmierer Erfahrung mit der Entwicklungsumgebung Unity mitbringt. Sauber formatierter und kommentierter Programmcod soll die Einführung eines fremden Entwicklers in die Programmlogik erleichtern.
NF 1.2	Beschreibungen der Gegenstände: Um sich einen Überblick über die Artefakte im Spiel verschaffen zu können, sind alle Gegenstände die im Spiel zu finden sind, in einem Kapitel der Projektdokumentation zu beschreiben. Ihr Zweck, der Fundort und die Einsatzmöglichkeiten des Gegenstandes sind festzuhalten. Innerhalb des Teams einigte man sich am 29.10.14 auf die Darstellung in einer Matrix.
NF2	Rechtliches
NF 2.1	Urheberrechte(Lizenzen) sind zu berücksichtigen: Vor dem Herunterladen und der anschliessenden Verwendung von durch Dritte entwickelte Assets, muss das Team in Erfahrung bringen, in welchem lizenztechnischen Restriktionen diese Komponenten unterliegen. Eingekaufte kommerzielle Assets sind als solche in der Projektdokumentation zu deklarieren. Auch ist darauf zu achten, keine eigenen Grafiken zu erstellen, die Urheberrechte von Marken verletzen, oder durch ihren Inhalt gegen andere Gesetze verstossen.

6 Rollenkonzept

Rollenname	Funktion
Spieler	Die Person, welche die Spielfigur im Spiel steuert
Spielfigur	Die virtuelle Figur, die vom Spieler innerhalb des Systems gesteuert wird
Gegner, gegnerische Figur	Nicht spielbare Figur im Spiel, die den Spieler am erfolgreichen Abschluss des Levels hindern soll. Im Basislevel modelliert als ein Roboter.
Entwickler	Das Projektteam bestehend aus Martin Käser, Fabian Schwab und Marcel Tschanz

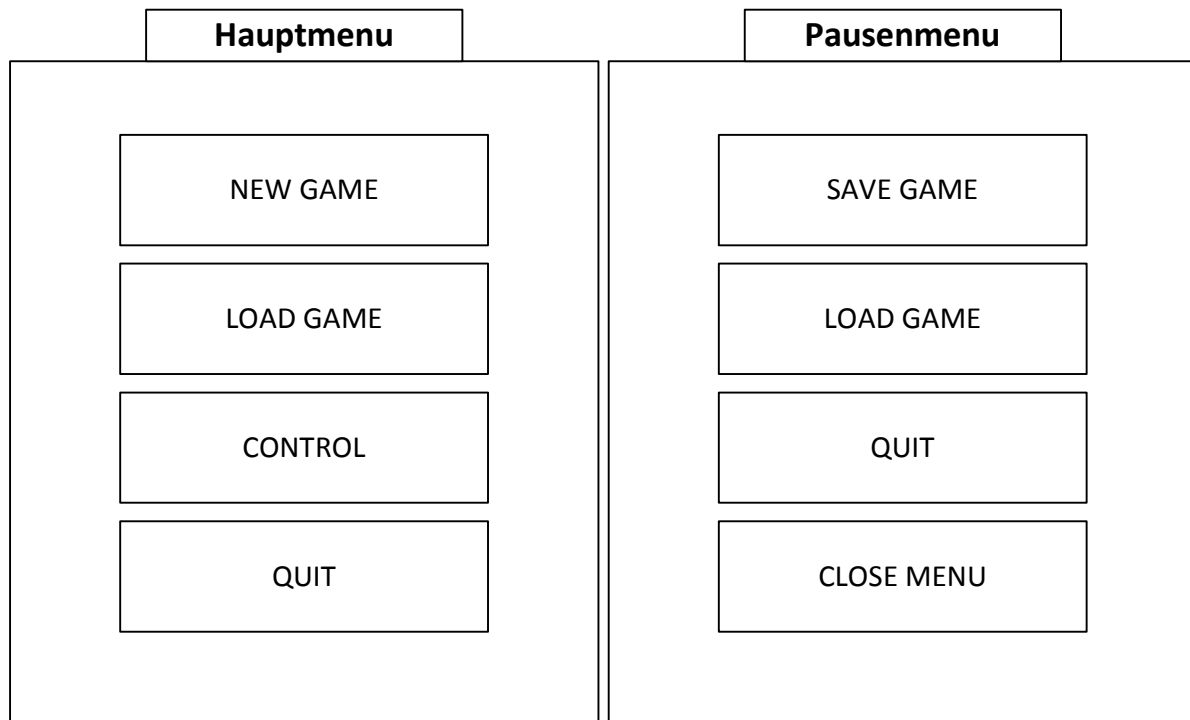
7 Glossar

Verantwortlich für das Glossar ist Fabian Schwab.

7.1 Erklärungen und Übersetzungen

Wort / Abkürzung	Bedeutung
HZ1 Alpha-Version	Entwicklungsstand einer Software, die schon die grundlegenden Funktionen enthält. Bereit zum Testen, aber auf keinen Fall auslieferbar.
HZ1 Basislevels	Als Basislevel wird, der erste Level des Spiels Rocket bezeichnet. Dieser Level 1 stellt als solcher auch das spielbare Endprodukt des Projektes dar.
FR3.3 Pausenmenu	Das Pausenmenu steht dem Spieler zu jedem Zeitpunkt im laufenden Spiel zur Verfügung. Der Spieler erreicht es durch Drücken der Taste ESC. Das Drücken der Taste ESC veranlasst das System dazu, das Spiel zu pausieren und das Pausenmenu mit den Optionen „Resume“, „Save game“, „Load game“ und „Quit Game“ und „Close Menu“ darzustellen.
FR4.2 Aktionsradius eines interaktiven Elements	Gegenstände, mit denen der Spieler interagieren kann, werden im Spiel durch eine farbige Kennung als solche ausgewiesen. Nähert sich die Spielfigur dem Gegenstand, so werden dem Spieler entsprechende Aktionsmöglichkeiten zur Auswahl gestellt, wie mit diesem Item umgegangen werden soll.
NF2.1 Assets	Als Assets werden Spielkomponenten bezeichnet, welche in anderen Programmen (bsp. 3D Editoren) erstellt wurden und in Unity importiert werden können. Dies können einfache Texturen sein, oder aber komplexe Modelle von Gegenständen und Figuren.
TZ5 First Person Ansicht	Die First-Person-Ansicht zeigt das Blickfeld aus der Sicht einer anderen Person. In unserem Fall ist dies das Blickfeld der Spielfigur.
FR5.4 Unity Einheit	Längeneinheit im Unity, entspricht einem Meter in der Spielwelt.

7.2 GUI



7.3 Synonyme

Wort / Abkürzung	
Rocket, System, Spiel	Als Synonyme verwendet
Gruppe, Team,	Als Synonyme verwendet
Gegner, gegnerische Figur	Roboter, Robotergegner.

8 Referenzen

ID	Titel	Autor	Format
1	Artificial Intelligence for Games, Second Edition	Ian Millington John Funge	E-Book
2	Programming Game A.I. by Example	Mat Buckland	E-Book