

|  |  |
| --- | --- |
| Status | Zur Verifizierung freigegeben |
| Name des Projektes | Rocket |
| Projektleitung | J. Eckerle |
| Projektauftraggeber | J. Eckerle |
| Autoren (alphabetisch) | M. Käser, F. Schwab, M. Tschanz |
| Initialen | kasem5, schwf5, tschm23 |
| Genehmigung durch | M. Gasenzer |

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 | Überarbeitung gemäss Auftrag 2 | kasem5, schwf5, tschm23 |

[1 Einleitung 9](#_Toc401767777)

[1.1 Ziel des Dokuments 9](#_Toc401767778)

[1.2 Leserkreis des Dokuments 9](#_Toc401767779)

[2 Projektbeschrieb 9](#_Toc401767780)

[2.1 Stakeholderliste 10](#_Toc401767781)

[2.2 Nutzer- und Zielgruppen 10](#_Toc401767782)

[2.3 Projektmethode 10](#_Toc401767783)

[2.4 Technische Ressourcen 10](#_Toc401767784)

[2.5 Dokumente 10](#_Toc401767785)

[3 Projektziele 11](#_Toc401767786)

[3.1 Name und Hauptziel 11](#_Toc401767787)

[3.2 Teilziele 11](#_Toc401767788)

[4 Scoping 12](#_Toc401767789)

[4.1 Systemkontext und Systemgrenzen 12](#_Toc401767790)

[4.2 Scope of the product 12](#_Toc401767791)

[4.3 Out of scope 13](#_Toc401767792)

[5 Anforderungen 14](#_Toc401767793)

[5.1 Quellen und Herkunft 14](#_Toc401767794)

[5.2 Anforderungsliste 14](#_Toc401767795)

[5.3 Detailbeschreibung der Anforderungen 18](#_Toc401767796)

[5.4 Qualitätsanforderungen (NFR) 20](#_Toc401767797)

[6 Chancen und Risiken des Projekts 21](#_Toc401767798)

[7 Glossar 21](#_Toc401767799)

[8 Referenzen 22](#_Toc401767800)

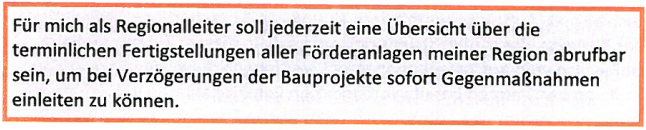
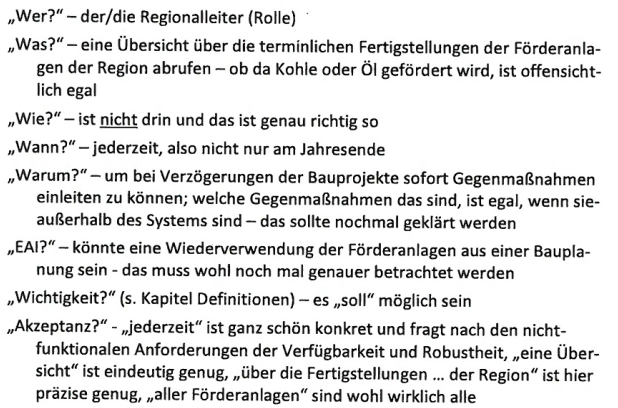
**Auftrag 2:**Die folgenden Punkte (verschiedene Anforderungen) werden in einer Anforderungsliste präsentiert.

1. **Funktionalen Anforderungen 6+ - natürlichsprachig formulieren**
   1. **4 auf Detailstufe hoch, also Komplexität schon aufgebrochen und**
   2. **2 eher grob, noch mit hohem Komplexitätslevel.  
        
      Darstellen in einer Anforderungsübersicht und einer Detailbeschreibung (wie bereits angefangen)  
      Risikoformel: Noch fakultativ**

Funktionale Anforderungen = Funktionen der Software, aber aus Kundensicht und nicht aus Software-Engineerer-Sicht (nocht nicht auf technische Umsetzung eingehen):  
Was soll sie leisten, wie soll sie es leisten und wann soll sie es leisten?  
Auch das warum es das leisten soll, ist hilfreich um das Requirement zu verstehen.

Funktionale Anforderungen stehen in engem Zusammenhang zu den Anwendungsfällen

Anforderung auf dieses Schema hin prüfen:

1. **Nichtfunktionale Anforderungen (Qualitätsanforderungen): Beschreiebn sie mind. 2 nicht-funktionale Anforderungen bei ihrer Themenstellung.**Hier auch Forderungen an das Vorgehensmodell aufschreiben, was gefordert wird. Auch Anforderungen bezüglich Entscheidungen und wer was einbringt und so kann man hier reinnehmen.
2. **Eventuell: Technische Anforderungen**

Hardwareanforderungen, SW Anforderungen, hier auch Performance und Leistungsrequirements rein und nicht wie im SED in die NFR.

1. **Randbedingungen  
   Beschreiben sie die Randbedingungen**
   * **Charakterisieren der Aktivitäten des Systems**
   * **(analog Schablone Teil2, S.19 PDF)**

**Überarbeiten der bestehenden Inhalte:**

* Anforderungen beschreiben Funktionen mit denen die Ziele erreicht werden können.   
  WER stellt WARUM die Anforderung? Beim WER die Rolle erwähnen und nicht die Person
* Mit muss, soll, kann/darf schon die Wichtigkeit andeuten.
* Anforderungen messbar formulieren, wenn nötig schreiben wie sie das Gewünschte erreichen können.
* Ergänzen zu den FR/NFR können technische Requirements aufgenommen werden

PDF Requirement Engineering (Anf. dok)

Tasks daraus offen:

* Traceability der Anforderungen: Von wem kamen Sie?
* Nominalisierung OK
* Substantive ohne Bezugsindex – wurden sie erklärt, genau definiert?
* Universalquantoren NOK - Mengen und Häufigkeiten quantifizieren OK
* Unvollständig spezifizierte Bedingungen (if-then nicht ausgearbeitet, nicht auf  
  die Alternativen und anderen Möglichkeiten eingegangen)
* Unvollständig spezifizierte Prozesswörter – wenn wir „übertragen“ verwenden,  
  müssen wir spezifizieren „was“, „wohin“ und „wie“ etwas übertragen wird +

Konstruktion von Anforderungen mittels Satzschablone (Requirements Template)

-> welche Punkte man bei einer Anforderung beachten muss. Syntaktisch eindeutig:

* muss, sollte, wird – Einsatz der Wörter je nachdem, wie die Anforderung ist
* kern der anforderung detailliert beschreiben: bsp. „drucken“
* Funktionale Req.: Wie wird die Aktivität durchgeführt? Durch:  
   Usersteuerung -> Das System muss die Möglichkeit bieten <wem?> ...dies und das zu machen

System(autonom) -> Das System sollte/muss/wird: ...  
Fremde Systeme -> Das System wird fähig sein / wird durch...

Pohl Buch Kapitel 6, Seiten 69 – 100

Anforderungen modellbasiert dokumentieren. Das Modell ist eine Abstraktion der Realität.  
So bsp. werden dann die UML Diagramme verwendet (UML=Modellierungssprache).

Aus diesen Kapiteln nur das rausnehmen, was wichtig für die Aufgabenstellung ist.

* Ziele beschreiben – die Ziele des Stakeholders:  
  Dies sind vom Stakeholder charakteristische Merkmale des zu entwickelnden Systems.  
  *Wenn Zeit, noch ein Zielmodell machen.*
* S. 75 Use Cases und Szenarien dokumentieren, wie das System genutzt wird  
  (Bei uns eher die Bedienung des Games)
  + Ein Diagramm machen
  + Use Cases in Tabelle
  + Einzelne Use Cases ausführlich mit Alternativen etc.  
    (beide Punkte wie schon aus SED bekannt)
* S. 76 Systemanforderungen = Functional Requirements
* S. 81 Perspektiven auf die Anforderungen:  
  1. Struktur 2. Funktion 3. Verhalten (ebenfalls bekannt aus SED)
  + Strukturperspektive:   
    System im Systemkontext, Struktur von Input und Output
  + Funktionsperspektive:  
    Welche Informationen aus dem Systemkontext durch das System manipuliert werden und welche Systemteile den Kontext wie beeinflussen.
  + Verhaltensperspektive:  
    Zustandsorientierte Dokumentation, wie sich das System in den Kontext zu einem gegebenen Zeitpunkt einbettet. Reaktion auf Ereignisse, Effekte etc.
* Ab S. 82 die Modellierung dieser Perspektiven erklärt

ZieleBuerger - SSWE Anf - Kap 3-4 (Scan)

Weiteres (on going):

* Glossar überarbeiten

# 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.

## 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.

## 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.

# Projektbeschrieb

**Zielsetzung** Ein 3D-Adventure-Spiel mit der Unity-Game-Engine entwickeln

**Technologien**  Unity Game Engine, Blender, Java-Script / UnityScript, Automaten

**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 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 dem Projektleitung bzw. mit dem Projektauftraggeber garantiert, dass die Spielidee und das Konzept den Erwartungen und Vorstellung gerecht werden.

Im Moment (Stand KW: 42) sind noch einige Zielvorstellungen und Anforderungen in Abklärung und müssen noch genauer mit dem Stakeholder geklärt werden.

## Stakeholderliste

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

## Nutzer- und Zielgruppen

* Testpersonen
* Spieler von Computerspielen und Interessierte an unserem Projekt als operative Anwender

## Projektmethode

* Das Projekt wird anlehnend an SCRUM agil entwickelt.

## Technische Ressourcen

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

## Dokumente

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

# Projektziele

Dieses Projekt hat zum Ziel, das Team anhand eines konkreten Auftrags mit der Praxis der Projektführung vertraut zu machen. Zu diesem Zweck entwickelt die Gruppe weitgehend selbständig ein spielbares 3-D-Spiel.

## Name und Hauptziel

**Projekt Rocket:** Eine an der Berner Fachhochschule entwickelte Basis eines spielbaren First-Person-Adventures, in welchem die Spielintelligenz der Akteure mit hierarchischen Automatenmodellen modelliert wird und dem Entwicklungsteam durch das Erlernen der Werkzeuge Unity3D und UnityScript einen Einblick in die Thematik „Virtual Reality“ ermöglicht.

## Teilziele

Dem Spiel liegt eine kleine Geschichte zu Grunde, die darüber Auskunft gibt, weshalb die Spielfigur sich in der momentanen Situation befindet. Zu Beginn des Spiels wird dies dem Spieler mitgeteilt.

Im Spiel sollten sich mindestens zwei verschiedene *hierarchical finite state machine* (zu deutsch: Hierarchisch aufgebaute, endliche Zustandsautomaten) gegenseitig beeinflussen. Dies HFSM einzusetzen hat höchste Priorität.

Die einzelnen Levelabschnitte sollen ebenfalls Abhängigkeiten zu einander haben und nur der Besuch eines jeden Levelabschnitts ermöglicht einen Spielerfolg. Ein Abschnitt verbleibt primär im gleichen Status, wie ihn der Spieler hinterliess (bsp.: Schalter ON) – bis er den gleichen Raum wieder aufsucht. Durch die Interaktion der Räume miteinander ist ein Statuswechsel aber trotzdem möglich.

Das Spiel soll die Elemente „Rätsel“ (in einer frei wählbaren Form), ein Inventar, sowie mindestens eine gegnerische Figur enthalten.

Elemente und Akteure innerhalb des Spiels müssen flexibel genug modelliert werden, dass auf Wunsch der Stakeholder Funktionen/Features wie:

* Neue Räume
* Andere Texturen
* Andere Bewaffnung
* Neue Bewegungsmöglichkeiten
* Zusätzliche Items

und schlussendlich ein neues Level einfach implementiert werden können. Dabei ist auf die Verwendung von Design Patterns zu achten.

# Scoping

## Systemkontext und Systemgrenzen

Grobe Architektur des 3-D-Adventures Rocket



## Scope of the product

Das Projekt1 umfasst die Umsetzung einer konkreten Spielidee und bringt dem Team einen Einblick in die Spieleentwicklung mit dem Entwicklungsfront-end Unity3D. Dazu gehören die praktische Anwendung der Automatentheorie und konkret für dieses Projekts Scripting-Aufgaben mit der an JavaScript angelehnten Entwicklungssprache UnityScript. Verschiedene Spielobjekte werden dadurch mit unterschiedlichem Verhalten und Reaktionsmöglichkeiten ausgestattet. Verknüpft mit Projekt1 ist auch die Anwendung der Theorie aus dem Fach Projektmanagement. Das Spiel soll eine Basis für Weiterentwicklungen sein und kann in seinen Funktionen und Levels erweitert werden.

## 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-Clients, auf dem das Spiel gespielt wird werden keine Änderungen vorgenommen.

# Anforderungen

## Quellen und Herkunft

Quelle(n) woher die Anforderung stammt:  
JE: Projektauftraggeber Jürgen Eckerle

Team: Projektmitarbeiter im Plenum

Erarbeitet wurden die Anforderungen an Treffen mit J. Eckerle.  
Diese fanden an folgenden Daten statt:

17.09.2014

03.10.2014  
10.10.2014

17.10.2014

24.10.2014

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

## Anforderungsliste

Die Anforderungsliste ist ständigen Mutationen unterworfen. Dies aus dem Grund, da noch immer verschiedenste Ideen für die Umsetzung des Games plausibel sind und man sich auch noch nicht über alle zu verwendende Pattern im Klaren ist.

-> History machen, von da an, wo wir eine Erstfassung des Dokuments haben

**Legende und ergänzende Hinweise:**

P= Priorität (Skala: 1(low), 2(medium), 3(high))

V= Variabilität

K= Komplexität

R= Aus {P;V;K} "berechnetes" Risiko (Zahlenwert)

Zur Gewichtung der Attribute hinsichtlich der Risikoabschätzung:  
K am "höchsten", V am "zweithöchsten", P am "niedrigsten"

Datum:

Wann dieses Requirement aufgenommen wurde. Pro Gruppe häufig das gleiche Datum drin, da wir Anforderungen zum gleichen Thema oft am gleichen Tag elaboriert / in Erfahrung gebracht haben.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **Kurzbezeichnung** | **Status** | **P** | **V** | **K** | **R** | **Quelle** | **Datum** |
| 1 | Unity-3D-Engine |  | | | | | | |
|  | Die Gruppe arbeitet sich in die Laufzeit- und Entwicklungs-umgebung Unity3D ein. | erledigt | 3 | 1 | 2 |  | JE | 17.09.2014 |
|  |  |  |  |  |  |  |  |  |
| 2 | Verständnis für Automaten |  | | | | | | |
|  | Bearbeitung der Theorie „Artificial Intelligence for Games“ | geplant | 3 |  |  |  | JE | 10.10.2014 |
|  | Konsens über Möglichkeiten der Implementation und Wissensaustausch | geplant | 3 |  |  |  | Team | 13.10.2014 |
|  | Umsetzung der hierarchischen Automation | geplant | 3 |  |  |  | JE | 10.10.2014 |
| 3 | Spielkonzept |  | | | | | | |
|  | Entwurf einer Spielidee | erledigt | 3 |  |  |  | JE | 17.09.2014 |
|  |  |  |  |  |  |  |  |  |
| 4 | Level Design |  | | | | | | |
|  | Kennenlernen von 3-D-Modellierungssoftware Blender | erledigt | 2 |  |  |  | Team | 01.10.2014 |
|  | Protokollierung der Quellen der verwendeten Objekte |  | 2 |  |  |  | Team | 01.10.2014 |
|  | Updatefunktione: Export und Import der erstellten Levels in Unity realisieren und dokumentieren. |  | 2 |  |  |  | Team | 08.10.2014 |
| 5 | GUI / Menus |  | | | | | | |
|  | Startsequenz mit Loading / Welcome / Briefingscreens | geplant | 3 |  |  |  | Team | 01.10.2014 |
|  | Hauptmenu | geplant | 3 |  |  |  | Team | 01.10.2014 |
|  | Aktionslabels (-> Interaktion mit Gegenständen) während dem Spiel | geplant | 3 |  |  |  | Team | 01.10.2014 |
|  | Countdownanzeige während des Spiels | geplant | 3 |  |  |  | Team | 01.10.2014 |
|  | Pausenmenu | geplant | 2 |  |  |  | Team | 01.10.2014 |
|  | Endsequenz mit Statistik | offen | 2 |  |  |  | Team | 08.10.2014 |
| 6 | Spielfigur |  | | | | | | |
|  | Basisfunktion an Steuermöglichkeiten der Figur implementieren | geplant | 3 |  |  |  | Team | 01.10.2014 |
|  | Kollisionsmodell | geplant | 3 |  |  |  | Team | 03.10.2014 |
|  | Kamerasteuerung für „um-die-Ecke-Sicht“. | geplant | 2 |  |  |  | Team | 03.10.2014 |
|  | Interaktion mit Gegenständen | geplant | 3 |  |  |  | Team | 03.10.2014 |
|  | Erzeugen von Geräuschen durch die Spielfigur und Auswirkungen auf den Spielstatus | geplant | 3 |  |  |  | Team | 03.10.2014 |
| 7 | Gegner |  | | | | | | |
|  | Bewegungen der Gegner – Aktionsradius | erledigt | 3 |  |  |  | Team | 01.10.2014 |
|  | Patrouille - Pfadfindung | erledigt | 3 |  |  |  | Team | 03.10.2014 |
| 8 | Interaktionen Gegner <-> Spieler |  | | | | | | |
|  | Entdecken der Spielerfigur durch den Gegner | erledigt | 3 |  |  |  | Team | 03.10.2014 |
|  | Untersuchen einer Signalquelle durch den Gegner | geplant | 3 |  |  |  | Team | 03.10.2014 |
|  | Verfolgung der Spielfigur durch den Gegner | geplant | 3 |  |  |  | Team | 08.10.2014 |
|  | Zustammenstoss und Ausschaltung der Spielfigur durch den Gegner | zu planen | 3 | 3 |  |  | Team | 08.10.2014 |
| 9 | Testing |  | | | | | | |
|  | To Do – Erstellen der Testszenarien | geplant | 3 | 3 |  |  | Team/JE | 13.10.2014 |

## Detailbeschreibung der Anforderungen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Kurzbezeichnung** | **P** | **V** | **K** | **R** |
|  | Unity-3D-Engine |
|  | Um abschätzen zu können, wie viel Aufwand für die Implementation der einzelnen funktionalen Anforderungen an das Spiel entsteht, muss genügend Zeit eingerechnet werden, um die Entwicklungsumgebung kennen zu lernen. Dies beinhaltet auch das Bearbeiten von Tutorials. Die Thematik ist für die Gruppe vollkommenes Neuland und der Projektauftraggeber stellt deshalb genügend zeitliche Ressourcen bereit, sich mit den Werkzeugen vertraut zu machen. | 3 | 1 | 2 |  |
|  |  |  |  |  |  |
|  | Verständnis für Automaten |
|  | Eine weitere Grundlage für den Erfolg des Projektes bildet das Verständnis der Automatentheorie. Speziell der Einsatz von Automaten innerhalb von Computerspielen. Die Bearbeitung der Theorie „Artificial Intelligence for Games“ wurde vom Auftraggeber als Aufgabe gegeben und die Umsetzung der Theorie stellt ebenfalls eines der Hauptziele des Projektes dar. |  |  |  |  |
|  | Wichtig ist ein Konsens über die Möglichkeiten der Implementation und ein fortlaufender Wissensaustausch zwischen den Gruppenmitgliedern. |  |  |  |  |
|  | Das Verhalten der Gegner soll trotz dem hierarchischen Aufbau der Automaten vom Spieler nicht voraussehbar sein. Realisiert werden die Automaten gemäss bekannten Algorithmen. Eine eigene Entwicklung „from scratch“ ist nicht das Ziel. Für einzelne Zustände kommen State Patterns zum Einsatz. |  |  |  |  |
|  | Spielkonzept |
|  | Entwurf einer Spielidee resp. eines Konzeptes wie der Spieler ein Basislevel abschliessen kann. Eine lineare Beschreibung zum Ablauf des Spiels als Veranschaulichung wird hilfreich sein. Ein Willkommensbildschirm teilt dem Spieler die Spielstory und die Ziele mit. |  |  |  |  |
|  | Das Konzept soll so aufgebaut sein, dass der Spieler bereits nach kurzer Zeit die generelle Aufgabe erkennt. Die Aufgabe zu erreichen soll aber nach und nach schwerer werden und mehr und mehr Spielelemente oder Spielerinteraktionen sollen gleichzeitig zum Einsatz kommen. |  |  |  |  |
|  | Level Design |
|  | Eigene Levels oder zumindest eigene Elemente des Levels mit ihren Texturen können nur schwerlich mit Unity erstellt werden. Mit der Animationssoftware Blender wird eine Umgebung geschaffen, die als Basislevel dienen wird. |  |  |  |  |
|  | Protokollierung der Quellen der verwendeten Objekte  Sofern nicht selber erstellt, ist die Verwendung der Texturen und Modelle zu dokumentieren, damit dies bei Bedarf eingesehen werden kann. |  |  |  |  |
|  | Updatefunktion: Damit dem Team die erstellten Levels in Unity zur Verfügung stehen, muss der Import/Export Mechanismus aus der Animationssoftware in die Entwicklungsumgebung Unity schnell möglich sein. So stehen den anderen Mitgliedern des Teams laufend die neusten Änderungen an den Levels zur Verfügung. |  |  |  |  |
|  | Das Level soll im Grundkonzept einheitlich gehalten werden, aber einzelne Passagen oder Räume sollen die verschiedenen Bereich klar abgrenzen. Das wird mit Beleuchtung oder anderen Texturen erreicht. |  |  |  |  |
|  | GUI / Menus |
|  | Eine Startsequenz bringt den Spieler in einen Willkommensbildschirm, wo er über die Ziele des Spiels und das Gameplay informiert wird. Ausserdem wird dem Spieler die Situation erläutert, in welcher sich seine Figur zu Beginn des Levels befindet. |  |  |  |  |
|  | Im Hauptmenu hat der Spieler die Möglichkeit Einstellungen zur grafischen Auflösung, zu Soundeffekten und zu seinem Spielernamen vorzunehmen. Speicher- und Ladeoption sind noch Gegenstand der Diskussion mit dem Stakeholder. |  |  |  |  |
|  | Aktionslabels signalisieren im Spiel die Möglichkeit der Interaktion mit Gegenständen. Befindet sich der Spieler an einem entsprechenden Ort im Level, erscheint eine Art Menu, welches von ihm die gewünschte Aktion erfragt. |  |  |  |  |
|  | Eine Countdownanzeige in einer Ecke des Bildschirms ist während des ganzen Spiels eingeblendet. Sie startet zu Beginn des Levels bei X Minuten. Diese rückläufige Zeitanzeige gibt die Anzahl Minuten und Sekunden an, die dem Spieler noch zur Verfügung stehen, um das Level zu absolvieren.  Ob die laufende Zeit am Ende für eine Rangliste oder Bestzeiten verwendet wird, wird noch diskutiert. |  |  |  |  |
|  | Im Pausenmenu, in welches man mit der ESC Taste gelangt, kann man das Spiel speichern (-> Anforderung in Abklärung) oder zu einem früheren Speicherpunkt zurück kehren. |  |  |  |  |
|  | In der Endsequenz wird dem Spieler eine Statistik zur erreichten Punktzahl präsentiert. Am Ende der Sequenz d.h. nach dem Verlassend es Statistikbildschirms soll der Spieler die Möglichkeit haben, das Level erneut zu meistern. |  |  |  |  |
|  | Spielfigur |
|  | Die grundlegenden Steuermöglichkeiten einer Figur sollen implementieren werden (WASD). Die Nutzung des CharacterMotor Scripts resp. die Erweiterung des FPSInputController Scripts ist für diese Zwecke sinnvoll. |  |  |  |  |
|  | Damit der Spieler sich realitätsnah bewegt und so auch mit der Umgebung interagiert muss ein Kollisionsmodell erstellt werden. Dies soll mit anderen Kollisionsmodellen entsprechend agieren. |  |  |  |  |
|  | Der Spieler muss die Möglichkeit haben via Kamerasteuerung einen Blick „um-die-Ecke“ zu werfen, um Gegner auszumachen, und/oder sich einen sicheren Überblick über den weiteren Verlauf der Spielewelt zu verschaffen. |  |  |  |  |
|  | Die Interaktion mit Gegenständen ist eine zentrale Funktion des Spiels und für den Spielerfolg unabdingbar. Über ein Aktionsmenu -> siehe Nummer: X soll der Spieler zwischen verschiedenen Aktionen auswählen können, |  |  |  |  |
|  | Die Spielfigur soll während der Fortbewegung oder dem Nutzen von Gegenständen Akkustik erzeugen, welche Auswirkungen auf den Spielstatus und den Verlauf haben. Gegner sollen von Geräuschen angelockt werden. |  |  |  |  |
|  | Gegner |
|  | Die Bewegunen der Gegner resp. deren Aktionsradius soll so modelliert sein, dass dem Spieler eine faire Chance bleibt, den Gegnern auszuweichen oder sich vor Ihnen verstecken zu können. |  |  |  |  |
|  | Ein Teil der Gegnerintelligenz macht deren Patrouillenverhalten aus. Dabei soll die Pfadfindung über Scripts ermöglicht werden. |  |  |  |  |
|  | Das Verhalten der Gegner soll mit einem Hierarchical State Maschine mit verschiedenen State Patterns umgesetzt werden. Das Gelernte wird später wiederverwendet und bildet eine solide Grundlage um ein bestehendes Verhaltenskonzept der Gegner zu erweitern. |  |  |  |  |
|  | Interaktionen Gegner <-> Spieler |
|  | Wird die Spielerfigur durch den Gegner entdeckt, wird der Gegner die Spielfigur verfolgen. Der Gegner reagiert auf visuelle und akkustische Signale in einem vordefinierten Radius. |  |  |  |  |
|  | Ist der Gegner durch ein Signal alarmiert worden, soll er die Signalquelle ausmachen und die Umgebung in einem bestimmten Radius genauer untersuchen. |  |  |  |  |
|  | Die Verfolgung der Spielfigur durch den Gegner ist so zu gestalten, dass dem Spieler eine faire Chance bleibt, dem Gegner zu entkommen. |  |  |  |  |
|  | Ein Zustammenstoss der Spielfigur mit dem Gegner kommt der Ausschaltung der Spielfigur gleich. Dies bedeuted das vorzeitige Ende des Spiels. |  |  |  |  |
|  | Um das Gameplay interessanter zu gestalten und taktische oder story-bedingtes Vorgehen abbilden zu können, ist eine Projektilwaffe für den Gegner vorgesehen. |  |  |  |  |
|  |  |  |  |  |  |

## Qualitätsanforderungen (NFR)

* Das Spiel soll eine Basis für Weiterentwicklungen bieten und nicht ein sich abgeschlossenes Spiel sein
* Das Spiel soll sich flüssig auf aktuellen Rechnern spielen lassen
* Das Spiel verursacht während mehrer Stunde Spielzeit keine Abstürze, die auf Fehler / Bugs im Code zurückzuführen sind
* Vorerst wird das Spiel in englischer Sprache entwickelt
* Das Spielkonzept wird zu Beginn anhand eines kurzen Briefings erläutert, so dass der Spieler nach dem Durchlesen dieser Anleitung grundsätzlich befähigt ist, das erste Level erfolgreich zu absolvieren.
* Urheberrechte (Logos, Texturen etc.) sind zu berücksichtigen
* Ein Level soll übersichtlich gehalten und von erfahrenen Spielern in unter 5 Minuten abgeschlossen werden können.

# Chancen und Risiken des Projekts

Die Entwicklungsumgebung Unity3D, die an JavaScript angelehnte Unityscript Sprache, so wie die Entwicklung geeigneter Automaten sind für die Gruppe bisher unbekannte Werkzeuge der Spieleentwicklung. Anfänglich unbekannte Themengebiete stellen von Natur aus eine grosse Herausforderung an ein realistisches Zeitmanagement.

Nachdem der Projekt Scope geklärt wurde und in der Design-Thinking-Phase erste Protoypen entwickelt wurden, konnte die Einschätzung der zeitlichen Ressourcen neu bewertet werden. Eine erste Spieleidee, verschiedene Use Cases und die Modellierung des Levels konnten das Risiko einer Fehlkalkulation minimieren. Trotzdem sind noch immer Unsicherheiten bezüglich der Machbarkeit - insbesondere was das Umsetzen der Automatentheorie betrifft - vorhanden.

Ein regelmässiger Austausch mit den Stakeholder garantiert ein Verständnis für die Prioritäten im Projekt und mögliche Risiken können mit diesen besprochen und neu bewertet werden. Da in diesem Projekt Projektleitung und Auftraggeber durch den Dozenten verkörpert werden, wirkt sich ein Ausfall von Herr Eckerle wegen Krankheit oder andere Ereignissen stark auf unser Projekt aus.

Im Requirement Engineering besteht zudem das Risiko, nicht alle Wünsche und Anforderungen des Stakeholders erfüllen zu können. Deshalb wird diesem Teil in der ersten Phase des Projekts genug Ressourcen zugeteilt. Dennoch bietet sich auf Grund unserer gewählten agilen Entwicklungsmethode die Möglichkeit dynamisch auf Änderungswünsche und Problem-stellungen einzugehen resp. diese zu lösen.

# Glossar

Verantwortlichkeit: F. Schwab

| Wort / Abkürzung | Bedeutung |
| --- | --- |
| Rocket, Game, Spiel, Adventure | Als Synonyme verwendet |
| Gruppe, Team, | Als Synonyme verwendet |

# Referenzen

| Bezeichnung | Titel, Quelle |
| --- | --- |
| 1 | Artificial Intelligence for Games,  Second Edition by Ian Millington&John Funge, |