|  |
| --- |
| platzhalter |
| Anforderungsspezifikation  Arcade Space Shooter  **Ueli Bühler**  **Marco Müller**  **Kevin Riesen**  **Version und Datum** |
| **Berner Fachhochschule**  Departement für Technik und Informatik |

Inhaltsverzeichnis

1 Zweck des Dokuments 3

2 Vision 3

3 Projektzielsetzung 3

4 Systemabgrenzung 4

5 Anforderungen 5

6 Glossar 6

7 Literaturverzeichnis 6

8 Anhang 6

9 Versionskontrolle 6

1. Zweck des Dokuments

Dieses Dokument beschreibt die Ziele und Anforderungen für das Projekt Arcade Space Shooter.

Das Projekt wird im Rahmen des Moduls «BTI7082 Projekt 1» im Herbstsemester 2019/2020 an der Berner Fachhochschule durchgeführt und dabei fachlich und methodisch von Peter Lange begleitet.

1. Vision

Das Projekt besteht aus der Umsetzung eines Videospiels, dessen Inhalt es ist, mit einem Raumschiff herumzufliegen, feindliche Raumschiffe zu bekämpfen und ein definiertes Ziel zu erreichen.

Ziel ist es, ein unterhaltsames Spiel zu entwickeln, welches durch Arcade-Spielelemente wie die Oben-herab-Sicht oder einfache Steuerung Erinnerungen an ältere Videospiele weckt.

Der Spieler des Spiels soll sich mit unserer Applikation in die Vergangenheit zurückversetzt fühlen und dabei den Spielspass geniessen können.

1. Projektzielsetzung
   1. Ausgangslage

Früher war alles besser. Wer hat diesen Spruch nicht schon gehört und sich dabei so seine Gedanken gemacht? Gemäss diesem Ansatz wird in diesem Projekt ein Space Shooter nach Arcade Vorbild erstellt. Zum Arcade-Genre gehören Spiele, welche sich durch einfache Spielmechaniken und eine intuitive Steuerung auszeichnen. Massgebend für dieses Genre sind die heute nicht mehr häufig anzutreffenden Arcade-Automaten. Das Spiel siedelt sich in diesem Umfeld an um eine Lücke zu füllen. Arcadespiele sind nicht mehr so verbreitet und wir möchten mit unserem Projekt diesem unterhaltsamen Genre wieder zu mehr Bekanntheit verhelfen. Damit sich nicht jeder Spieler einen Spielautomaten für zuhause beschaffen muss, wird das Spiel auf dem Computer gespielt werden können.

Das Projekt startet ohne bestehende Applikation, womit es sich um eine Neuentwicklung handelt.

* 1. Stakeholder

Die relevanten Stakeholder sind:

* Peter Lange – Betreuer
* BFH – begutachtende Institution
* K. Riesen, M. Müller, U. Bühler – Durchführung und Entwicklung
* Personen, welche das Spiel spielen werden
  1. Projektziele
     1. Ziele der Stakeholder
* Personen, welche das Spiel spielen werden
  + Spielen eines lauffähigen Spieles.
  + Ein Arcadespiel wie früher an einem Spielautomaten spielen.
  + Der Spieler erfreut sich an einem unterhaltsamen Spiel.
* Entwickler
  + Durchführung eines erfolgreichen Projektes.
  + Entwicklung eines lauffähigen Spieles.
  + Erfahrungen in der Spieleentwicklung sammeln und anwenden.
  + Sich mit den Herausforderungen moderner Spieleentwicklungs-Engines auseinandersetzen.
* Betreuer / BFH
  + Betreuung, Unterstützung und Abnahme einer erfolgreichen Projektdurchführung.
    1. Priorisierung

Da es sich bei einem Computerspiel um eine aufwendige und komplexe Applikation handelt, werden die Prioritäten wie folgt gesetzt. Das Projektvorgehen wird iterativ sein, somit werden Ziele erster Priorität zuerst abgeschlossen, bevor ein neues Ziel angefangen wird.

1. **Technische Umsetzung**  
   Der Fokus des Projekts liegt auch dem Programmieren des Spiels. Bevor neue Elemente hinzugefügt werden, muss die technische Umsetzung der zugrundeliegenden Ziele erfolgt sein.
2. **Gameplay**  
   Neue Elemente die das Spielerlebnis verbessern, werden hinzugefügt, ohne ein kompliziertes Design vorauszusetzen.
3. **Design**  
   Die Elemente des Spiels werden graphisch aufgewertet um dem Spieler ein besseres Erlebnis zu bieten.
4. Systemabgrenzung
   1. Prozessumfeld

Als Spiel hat das Resultat des Projekts keine Geschäftsprozesse zu unterstützen. Das Prozessumfeld wird somit in dieser Anforderungsspezifikation nicht weiter erläutert.

* 1. Systemumfeld

In diesem Projekt existieren nur Spieler und System.

Das Spiel ist dabei in sich selbst gekapselt und benötigt deshalb keine Kommunikation mit anderen Systemen. Das Spiel läuft lokal auf dem Rechner des Spielers und benötigt keine Verbindung zum Internet.

ToDo: Bild

* 1. Nicht unterstützte Projektziele
* Das Spiel soll keine Mehrspielerfunktionalitäten besitzen.
* Das Spiel muss keine Vielfältigkeit in der Levelauswahl aufweisen.
* Das Projekt verfolgt keine kommerziellen Zwecke und wird daher nicht Veröffentlicht oder Vermarktet.

1. Anforderungen
   1. Quellen und Vorgehen

Die verwendeten Quellen und genutzten Techniken (Interviews, Literaturstudium etc.) zur Ermittlung der Anforderungen werden aufgeführt

* 1. Funktionale Anforderungen
     1. Use Cases

Priorität: M = Muss / O = Optional / W = Wunsch (Nice to have)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **UC1** | Entwurf | M | Der Spieler kann das Raumschiff kontrollieren. |
| **UC1.1** | Entwurf | M | Der Spieler bewegt das Raumschiff mit Tastaturinput. |
| **UC1.2** | Entwurf | M | Der Spieler kann per Mausbewegung mit den Waffen zielen. |
| **UC1.3** | Entwurf | M | Der Spieler kann per Mausklick mit den Waffen feuern. |
| **UC1.4** | Entwurf | W | Der Spieler kann per Nummerntasten Fähigkeiten Einsetzen |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **UC2** | Entwurf | M | Der Spieler spielt ein Level. |
| **UC2.1** | Entwurf | M | Der Spieler kann in der Spielwelt zu einem Ziel gelangen, um das Spiel zu gewinnen. |
| **UC2.2** | Entwurf | M | Der Spieler muss Hindernissen wie Asteroiden ausweichen. |
| **UC2.3** | Entwurf | M | Der Spieler muss Feindeinheiten bekämpfen. |
| **UC2.4** | Entwurf | M | Der Spieler kann durch Schaden am Raumschiff zerstört werden. |
| **UC2.5** | Entwurf | O | Spielerwelt dynamisch generieren. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **UC3** | Entwurf | M | Der Spieler kann Informationen über das Spiel dem Userinterface entnehmen. |
| **UC3.1** | Entwurf | M | Der Spieler kann seine aktuellen Lebenspunkte einsehen. |
| **UC3.2** | Entwurf | O | Der Spieler kann die aktuellen Lebenspunkte der Feinde einsehen. |
| **UC3.3** | Entwurf | O | Der Spieler kann die Richtung und Distanz zum Ziel einsehen. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **UC4** | Entwurf | W | Spielerschiff mit Upgrades ausstatten können. |
| **UC4.1** | Entwurf | W | Der Spieler erhält für besiegte Gegner eine Menge einer Währung. |
| **UC4.2** | Entwurf | W | Der Spieler kann im Austausch einer Währung Verbesserungen für sein Schiff beziehen. |

* + 1. Detaillierte Anforderungen

|  |  |
| --- | --- |
| **Name** | UC1.1 |
| **Ziel im Kontext** | Der Spieler bewegt das Raumschiff mit Tastaturinput. |
| **Akteure** | Spieler und System |
| **Trigger** | Der Spieler befindet sich im Level und drückt eine zur Steuerung definierte Taste auf der Tastatur. |
| **Essenzielle Schritte** | 1. Der Spieler betätigt eine Steuertaste zum Steuern des Raumschiffes. 2. Das Raumschiff des Spielers wird entsprechend des Inputs bewegt. |
| **Erweiterungen** | 2a. Vorwärtstaste  2a1. Das Raumschiff bewegt sich vorwärts  2b. Rückwärtstaste  2b1. Das Raumschiff bewegt sich rückwärts  2c. Richtungstaste rechts  2c1. Das Raumschiff dreht sich nach rechts  2d. Richtungstaste links  2d1. Das Raumschiff dreht sich nach links |
|  |  |

|  |  |
| --- | --- |
| **Name** | UC1.2 |
| **Ziel im Kontext** | Der Spieler kann per Mausbewegung mit den Waffen zielen. |
| **Akteure** | Spieler und System |
| **Trigger** | Der Spieler befindet sich im Level und bewegt die Maus. |
| **Essenzielle Schritte** | 1. Der Spieler bewegt die Maus zum Zielen. 2. Ein Symbol zur Darstellung der Zielrichtung erschein an der Position der Maus. 3. Die Waffen des Spielerschiffes richten sich in die Richtung, in die gezielt wird. |
| **Erweiterungen** | - |
|  |  |

|  |  |
| --- | --- |
| **Name** | UC1.3 |
| **Ziel im Kontext** | Der Spieler kann per Mausklick mit den Waffen feuern. |
| **Akteure** | Spieler und System |
| **Trigger** | Der Spieler befindet sich im Level und klickt mit der Maus. |
| **Essenzielle Schritte** | 1. Der Spieler klickt mit der Maus um zu feuern. 2. Die Waffen des Spielerschiffes feuern Geschosse in die Richtung, in die gezielt wurde. 3. Wird ein Feind getroffen, wird diesem Schaden zugefügt. |
| **Erweiterungen** | 3a. Der zugefügte Schaden übersteigt die verbleibenden Lebenspunkte des Feindes  3a1. Der Feind wird zerstört.  3b. Das Geschoss trifft auf ein Hindernis  3b1. Das Geschoss verschwindet.  3c. Das Geschoss trifft auf kein Objekt.  3c1. Das Geschoss fliegt weiter bis es auf ein Hindernis trifft oder eine fixe Zeitspanne vergangen ist und verschwindet anschliessend. |
|  |  |

|  |  |
| --- | --- |
| **Name** | UC1.4 |
| **Ziel im Kontext** | Der Spieler kann per Nummerntasten Fähigkeiten Einsetzen |
| **Akteure** | Spieler und System |
| **Trigger** | Der Spieler befindet sich im Level und drückt eine für Fähigkeiten definierte Taste auf der Tastatur. |
| **Essenzielle Schritte** | 1. Der Spieler drückt eine Fähigkeitstaste auf der Tastatur, um diese einzusetzen. 2. Die Fähigkeit tritt in Kraft. 3. Eine Zeitspanne wird gesetzt, bis die Fähigkeit erneut eingesetzt werden kann (Cooldown genannt). |
| **Erweiterungen** | 2a. Die Zeit für den Cooldown ist noch nicht vergangen.  2a1. Der weitere Vorgang wird abgebrochen. |
|  |  |

|  |  |
| --- | --- |
| **Name** | UC2.1 |
| **Ziel im Kontext** | Der Spieler kann in der Spielwelt zu einem Ziel gelangen, um das Spiel zu gewinnen. |
| **Akteure** | Spieler, System |
| **Trigger** | Der Spieler berührt den Endpunkt des Levels |
| **Essenzielle Schritte** | 1. Der Spieler bewegt sich auf den Endpunkt 2. Das System beendet das Level |
| **Erweiterungen** | - |
|  |  |

|  |  |
| --- | --- |
| **Name** | UC2.2 |
| **Ziel im Kontext** | Der Spieler muss Hindernissen wie Asteroiden ausweichen. |
| **Akteure** | Spieler, System |
| **Trigger** | Asteoriden erscheinen im sichtbaren Spielraum |
| **Essenzielle Schritte** | 1. Das System lässt Asteroiden auf dem Spielraumherumfliegen 2. Der Spieler nutzt die Steuerungstasten um den Asteroiden auszuweichen |
| **Erweiterungen** | * 2.2a. Der Spieler fliegt in einen Asteroiden hinein. * 2.2a1. Der Spieler nimmt Schaden. |
|  |  |

|  |  |
| --- | --- |
| **Name** | UC2.3 |
| **Ziel im Kontext** | Der Spieler muss Feindeinheiten bekämpfen. |
| **Akteure** | Spieler, System |
| **Trigger** | Feinde erscheinen im sichtbaren Spielraum |
| **Essenzielle Schritte** | 1. Das System lässt Feinde auf dem Spielraum herumfliegen 2. Der Spieler schiesst auf die sichtbaren Feinde   Die Feinde nehmen Schaden falls sie getroffen werden |
| **Erweiterungen** | * 2.3a. Der Spieler wird von Feinden beschossen * 2.3a1. Der Spieler nimmt Schaden, falls er getroffen wird. |

|  |  |
| --- | --- |
| **Name** | UC2.4 |
| **Ziel im Kontext** | Der Spieler kann durch Schaden am Raumschiff zerstört werden. |
| **Akteure** | System |
| **Trigger** | Der Spieler wird auf 0 Lebenspunkte reduziert |
| **Essenzielle Schritte** | 1. Der Spieler wird auf 0 Lebenspunkte reduziert 2. Der Spieler wird ausgelöscht 3. Das System zeigt «Game Over» an und beendet das Level |
| **Erweiterungen** |  |

|  |  |
| --- | --- |
| **Name** | UC2.5 |
| **Ziel im Kontext** | Spielerwelt dynamisch generieren. |
| **Akteure** | Spieler und System |
| **Trigger** | Der Spieler startet den Level. |
| **Essenzielle Schritte** | 1. Der Spieler startet den Level um das Spiel zu beginnen. 2. Das System kreiert einen temporäten Level. 3. Das System platziert den Spieler im Level. 4. Das System platziert ein Ziel an einem zufälligen Ort im Level. 5. Das System platziert Hindernisse an zufälligen Orten im Level. 6. Das System platziert Feindeinheiten an zufälligen Orten im Level. 7. Der Spieler erhält die Möglichkeit den temporären Level zu spielen. |
| **Erweiterungen** | * 2a. Die Zeit für den Cooldown ist noch nicht vergangen. * 2a1. Der weitere Vorgang wird abgebrochen. |

|  |  |
| --- | --- |
| **Name** | UC3.1 |
| **Ziel im Kontext** | Der Spieler kann seine aktuellen Lebenspunkte einsehen. |
| **Akteure** | Spieler, System |
| **Trigger** | Das Level wird gestartet |
| **Essenzielle Schritte** | 1. Das System Stellt die Aktuellen Lebenspunte des Spielers dar. 2. Der Spieler sieht seine aktuellen Lebenspunkte auf dem Bildschirm. |
| **Erweiterungen** | - |

|  |  |
| --- | --- |
| **Name** | UC3.2 |
| **Ziel im Kontext** | Der Spieler kann die aktuellen Lebenspunkte der Sichtbaren Feinde einsehen. |
| **Akteure** | Spieler, System |
| **Trigger** | Feinde Betreten den Sichtbaren Spielraum |
| **Essenzielle Schritte** | 1. Feinde Betreten den Sichtbaren Spielraum 2. Das System Stellt die Aktuellen Lebenspunkte des Spielers dar. 3. Der Spieler sieht seine aktuellen Lebenspunkte auf dem Bildschirm. |
| **Erweiterungen** | - |

|  |  |
| --- | --- |
| **Name** | UC3.3 |
| **Ziel im Kontext** | Der Spieler kann die Richtung und Distanz zum Ziel einsehen. |
| **Akteure** | Spieler, System |
| **Trigger** | Der Spieler startet das Spiel |
| **Essenzielle Schritte** | 1. Der Spieler bewegt seine Maus 2. Das System zeigt dem spieler die Schussrichtung und Distanz an , die er im moment hat. |
| **Erweiterungen** | UC3.3 |

|  |  |
| --- | --- |
| **Name** | UC4.1 |
| **Ziel im Kontext** | Der Spieler erhält für besiegte Gegner eine Menge einer Währung. |
| **Akteure** | Spieler, System |
| **Trigger** | Ein Feind wurde vom Spieler zerstört |
| **Essenzielle Schritte** | 1. Der Spieler zerstört einen Feind 2. Das System honoriert dem Spieler eine Menge einer Währung. |
| **Erweiterungen** | - |

p

|  |  |
| --- | --- |
| **Name** | UC4.2 |
| **Ziel im Kontext** | Der Spieler kann im Austausch einer Währung Verbesserungen für sein Schiff beziehen. |
| **Akteure** | Spieler, System |
| **Trigger** | Der Spieler öffnet das Verbesserungsmenu |
| **Essenzielle Schritte** | 1. Der Spieler öffnet das Verbesserungsmenu. 2. Der Spieler wählt eine verfügbare Verbesserung aus 3. Der Spieler klickt auf den Kaufen Knopf 4. Das System gibt dem Spieler die Verbesserung und zieht die nötige Menge an Währung ab. |
| **Erweiterungen** | - |

Attribute:

ID: eindeutige Identifikation

Status: Entwurf / geprüft / freigegeben

Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

Die Beschreibung erfolgt als **User Stories** (halbformal)

* 1. Qualitätsanforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **Q1.1** | Entwurf | M |  |
|  |  |  |  |

* 1. Randbedingungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **R1.1** | Entwurf | M |  |
|  |  |  |  |

Welche Randbedingungen sind zu beachten?

* Technische Rahmenbedingungen
* Organisatorische Vorgaben
* Standards
* Anforderungen sollten zur besseren Übersichtlichkeit zu Themen zusammengefasst werden
* Jeweils auch mit NICHT-Anforderungen
* Projektziel(e) benennen, zu deren Erreichen die Anforderungen beitragen
  1. Datenmodell

Modell der wichtigsten Entitäten, soweit es für das Verständnis der Anforderungen notwendig ist.

1. Glossar
2. Literaturverzeichnis

**Literatureintrag**

*Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr* 7

**Literatureintrag**

*Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr* 9

**Literatureintrag**

*Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr* 11

1. Anhang
   1. Abstimmung der Anforderungen

Eventuell aufgetretene Konflikte und die gewählte Lösung dafür werden kurz dokumentiert

* 1. Definition of Ready – Checklist

Spezifische Kriterien, die die Anforderungen in diesem Projekt erfüllen müssen, um reif für die Umsetzung zu sein

1. Versionskontrolle

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Beschreibung** | **Autor** |
| X0.1 | 23.09.2019 | Initiale Version kopiert und Angepasst | Riesen |
| 1.0 | 26.09.2019 | Vision und Zielsetzung | Bühler, Müller, Riesen |
|  |  |  |  |
|  |  |  |  |