

---

# Projektdokumentation „Bahnnenspiel“

---

**Studienarbeit**

Alex Schmitt und Simon Weiler

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Elektrotechnik  
und Informationstechnik  
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation  
Prof. Dr.-Ing. Ralf Steinmetz

---

---

## **Ehrenwörtliche Erklärung**

---

Hiermit versichere ich, die vorliegende Studienarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in dieser oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Die schriftliche Fassung stimmt mit der elektronischen Fassung überein.

Darmstadt, den 16. September 2015

Alex Schmitt und Simon Weiler

---

## Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Projektdefinition</b>	<b>3</b>
2.1	Zielsetzung und Anforderungen . . . . .	3
2.2	Features . . . . .	3
2.3	Priorisierung . . . . .	3
<b>3</b>	<b>Konzept</b>	<b>4</b>
3.1	Architektur . . . . .	4
3.2	Game Design Document . . . . .	4
3.2.1	Spielobjekte . . . . .	4
3.2.2	User Interface . . . . .	5
3.2.3	Sounds . . . . .	5
3.2.4	Steuerung . . . . .	5
3.2.5	Levels . . . . .	5
3.3	Hardware . . . . .	6
3.3.1	Smovetec . . . . .	6
3.3.2	Texas Instruments SensorTag . . . . .	6
<b>4</b>	<b>Recherche</b>	<b>8</b>
4.1	Ähnliche Spielideen . . . . .	8
4.1.1	Skyroads . . . . .	8
4.1.2	Audiosurf . . . . .	8
4.2	Spiele mit Fahrradergometern . . . . .	9
4.2.1	Atari Puffer . . . . .	9
4.2.2	Thera Trainer . . . . .	10
4.2.3	Jeff Sinclair - Edith Cowan University . . . . .	10
4.2.4	Ergo Active . . . . .	11
4.3	Fazit der Recherche . . . . .	12
<b>5</b>	<b>Methoden und Vorgehen</b>	<b>13</b>
<b>6</b>	<b>Planung</b>	<b>14</b>
6.1	Projektplanung . . . . .	14
6.2	Zeitplanung . . . . .	15
6.3	Meilensteine . . . . .	16
6.4	Arbeitspakete . . . . .	16
6.4.1	Koordination und Organisation . . . . .	16
6.4.2	Konzeption und Findung der Spielidee . . . . .	17
6.4.3	Recherche . . . . .	17
6.4.4	Entwicklung des Spielgrundgerüsts . . . . .	18
6.4.5	Leveldesign . . . . .	19
6.4.6	Steuerung . . . . .	19
6.4.7	Audiovisuelle Gestaltung . . . . .	20
6.4.8	GUI . . . . .	21

---

6.4.9 Benutzerdokumentation . . . . .	21
6.4.10 Entwicklerdokumentation . . . . .	22
<b>7 Umsetzung</b>	<b>23</b>
7.1 Steuerung . . . . .	23
7.2 Leveleditor . . . . .	24
7.3 Szenen . . . . .	24
<b>8 Anleitung</b>	<b>26</b>
8.1 Konfiguration . . . . .	26
8.2 Steuerung . . . . .	27
8.2.1 Tastatursteuerung . . . . .	27
8.2.2 Ergometersteuerung . . . . .	27
8.3 Spiel . . . . .	27
<b>9 Zusammenfassung und Ausblick</b>	<b>29</b>
<b>Literaturverzeichnis</b>	<b>29</b>

---

## 1 Einleitung

---

Seit geraumer Zeit findet man auf dem Spielmarkt immer mehr Spiele die durch den eigenen Körper gesteuert werden können. Bekannte Beispiele dafür umfassen die Spielekonsole Wii der Firma Nintendo, sowie der Bewegungssensor Kinect von Microsoft für die Konsole Xbox. Viele dieser Spiele bedürfen entweder den Einsatz des Oberkörpers, beziehungsweise der Arme, wenige andere werden durch Fahrradergometer gesteuert, wie zum Beispiel ErgoActive der TU Darmstadt [vgl. Kapitel 4.2]. Bei beiden Arten wird jedoch stets nur ein Teil des Körpers zur Steuerung genutzt. Eine Kombination aus beiden Steuerungen bietet *Smovetec*. Smovetec ist eine Plattform auf die Fahrradergometer montiert werden können, um dadurch zusätzlich zur Trittfrequenz auch die Neigung beim Fahrradfahren zu simulieren. Allerdings gibt es noch keine Spiele, die diese Möglichkeit als zusätzliche Steuerung nutzen.

In diesem Projekt soll ein solches Spiel realisiert werden. Beim „Bahnsenspiel“ steuert der Spieler ein Raumschiff in einer fiktionalen Weltraum-Umgebung über enge Bahnen, die mit Hindernissen gespickt sind und immer im Kampf gegen die Uhr, dargestellt über den verbleibenden „Treibstoff“ des Raumschiffs. Das Besondere des Spiels ist die ausgefallene Steuerung, die mit dem Smovetec System ermöglicht wird. Statt mit Maus und Tastatur am Schreibtisch, gibt man die Lenkimpulse und Geschwindigkeit für das Raumschiff mit dem Fahrradergometer vor.

Ziel des Spiels ist es aber nicht, möglichst schnell durch die Strecken zu fahren, sondern vielmehr die immer schwieriger und komplizierter werdenden Levels überhaupt in der vorgegebenen Zeit zu meistern. Dazu ist weniger die körperliche Fitness gefordert, sondern vor allem die Geschicklichkeit, das Gleichgewicht zu halten und gleichzeitig gerade genug in die richtige Richtung zu steuern um den Hindernissen auszuweichen. Zusätzlich zu der Steuerung mittels Neigung und Trittfrequenz kommt später im Spiel auch noch das drücken einer Taste am Lenker hinzu, womit ein Sprung ausgelöst wird, der an manchen Stellen notwendig ist, um einem großen Hindernis zu entgehen.

Dieses Dokument beschreibt zunächst das Projekt mit allen Anforderungen und Zielgruppen, anschließend wird das Konzept und die Architektur des Projektes, sowie die daraus resultierende Planung in Bezug auf Zeitplanung, Meilensteine und Arbeitspakete erläutert. In einem weiteren Kapitel werden ähnliche Spielideen und Eingabemethoden betrachtet. Ebenfalls behandelt das Dokument die eigentliche Umsetzung der Projektidee. Bevor ein Fazit des Projektes gezogen und auf mögliche Erweiterungen des Spieles eingegangen wird, findet man eine Beschreibung zur Steuerung des Spiels, sowie nötige Hinweise um eine eigene Version des Spiels mit eigenen Levels erstellen zu können.

---

## 2 Projektdefinition

---

### 2.1 Zielsetzung und Anforderungen

---

Das Ziel des Projekts ist es, ein Spiel zu entwickeln, das die Möglichkeiten des Smovetec-Fahrradergometers im Bereich Videospiele demonstriert. Insbesondere liegt also der Fokus des Projekts darin, eine Spielidee zu finden, welche die Steuerungsmechanismen Trittfrequenz und Neigung mit einem spaßigen Konzept verbindet.

Die wichtigste Anforderung an das Projekt ist also eine sinnvolle Nutzung des Fahrradergometers als Eingabegerät. Da das eingesetzte Fahrradergometer nicht unbedingt immer zur Verfügung steht, ist eine Ersatzsteuerung nur mit Maus und Tastatur sinnvoll.

Eine weitere wichtige Anforderung ist außerdem, dass das entstandene Spiel Spaß machen soll. Da das Spiel jedoch vor allem zu Demonstrationszwecken dienen soll, ist es insbesondere wichtig, dass das Spiel nicht kompliziert zu erlernen ist und auch für alle Altersstufen geeignet ist, zumindest im Rahmen der vom Fahrradergometer vorgegebenen Grenzen. Für Kinder unter 6 Jahren wird das Spiel somit vermutlich nicht geeignet sein, da für das Ergometer eine gewisse Körpergröße erforderlich ist.

Aus dem Einsatzzweck heraus ergibt sich auch, dass eine längere Bindung von Spielern, wie beispielsweise durch eine Story, ein Highscore-System oder durch einen Mehrspielermodus nicht notwendig ist. Die Zielgruppe ist somit jeder, insbesondere auch ohne Videospiel-Erfahrung, im Alter ab 6 Jahren.

---

### 2.2 Features

---

- Steuerung durch Trittfrequenz und Neigung
- Ersatzsteuerung mit Maus und Tastatur
- Ansprechende optische Darstellung
- Langsam steigender Schwierigkeitsgrad
- Untermalung mit Musik und Geräuschen
- Verschiedene Designs der Spielumgebung

---

### 2.3 Priorisierung

---

Oberste Priorität hat die Steuerung mittels Fahrradergometer. Direkt danach kommen eine ansprechende Präsentation und eine gewisse Zahl an Levels, welche einen gemächlich steigenden Schwierigkeitsgrad abbilden.

Die Ersatzsteuerung, Musik und verschiedene Designs gehören eher zu den Features mit niedriger Priorität.

---

### 3 Konzept

---

Bei dem Spiel handelt es sich um ein Geschicklichkeitsspiel, bei dem der Spieler ein Raumschiff über einen Parcours im Weltall steuern muss. Die Level bestehen aus einer geraden Bahn auf der der Spieler das Raumschiff nach Links und Rechts bewegen kann indem er sich auf dem Ergometer in die jeweilige Richtung lehnt. Zusätzlich muss der Spieler durch Springen Hindernissen auf der Bahn ausweichen. Durch die Veränderung seiner Trittfrequenz erhöht, bzw. verringert der Spieler die Geschwindigkeit des Raumschiffes. Für die Bewältigung eines Levels gibt es eine vorgegebene Zeitspanne innerhalb der das Ziel erreicht werden muss. Die Zeitspanne wird durch den verbleibenden Kraftstoff des Raumschiffs symbolisiert. Auf diese Art und Weise nimmt der Spieler die Motivation daraus, das Level überhaupt zu bestehen und nicht aus einer Highscore-Liste und dem Wettbewerb mit anderen Spielern. Dies ist für den Verwendungszweck des Spiels als Demonstrationsspiel besser geeignet.

Das Level gilt als erfolgreich beendet, wenn der Spieler es schafft mit dem Raumschiff innerhalb der vorgegebenen Zeit das Ziel zu erreichen. Läuft die Zeit ab, kollidiert der Spieler mit einem Hindernis oder kommt von der Bahn ab, so hat der Spieler das Level nicht geschafft und muss es wiederholen. Hat der Spieler das Level geschafft, erreicht er automatisch das nächste Level. Es ist nicht vorgesehen, den Spielfortschritt zu speichern, beim nächsten Start des Spiels beginnt er wieder beim ersten Level. Letzteres dient dem Zweck, dass das Spiel vor allem zu Demonstrationszwecken gedacht ist und nicht als langfristige Unterhaltung eines Spielers. Dementsprechend ist auch der Umfang so angelegt, dass man alle Levels in einem Durchgang ohne Pausen schaffen kann.

---

#### 3.1 Architektur

---

Die Architektur der Software wird grob schon durch die Spieleengine Unity vorgegeben, deren Nutzung Teil der Aufgabenstellung ist. Das Verhalten einzelner Spielobjekte wird von an diese angehängte Skripte gesteuert, wobei vermieden wird, unterschiedliche Kompetenzen im selben Skript zu behandeln, auch wenn es sich um das selbe Objekt handelt.

Um den gesamten Zustand des Spiels zu halten, gibt es ein GameManager-Objekt mit einem gleichnamigen Skript, welches in der ersten Szene erzeugt wird und ab da an auch beim Laden einer neuen Szene nicht zerstört wird.

Die Eingaben vom Fahrradergometer, also Neigung und Trittfrequenz, werden über ein Smartphone beziehungsweise einen daran angeschlossenen Bluetooth-Sensor verarbeitet und dann über WLAN an den Rechner, auf dem das Spiel läuft, übertragen.

Eine detailliertere Beschreibung der Architektur des fertigen Spiels wird im Kapitel 7 – Umsetzung gegeben.

---

#### 3.2 Game Design Document

---

---

##### 3.2.1 Spielobjekte

---

Die Spielfigur ist ein Raumschiff, welches sich auf der Bahn bewegen kann. Auf der Bahn befinden sich Hindernisse in Form von Tunnel, Barrikaden und Abgründen. Außerhalb der Bahn befindet sich ebenfalls ein Abgrund.

Die Bahn kann auf zwei unterschiedlichen Ebenen verlaufen, wobei der Spieler entweder per Sprung oder Rampe die Ebene wechseln kann.



**Abbildung 3.1:** Handskizze der Spielidee

---

### 3.2.2 User Interface

---

Das User Interface zeigt eine Anzeige für die aktuelle Geschwindigkeit und den verbleibenden Kraftstoff in Form von Rundinstrumenten. Der Fortschritt im aktuellen Level wird als horizontaler Balken am oberen Bildschirmrand dargestellt, der sich mit zunehmender erfolgreich absolvierter Strecke füllt. Außerdem wird die Nummer des Levels, in dem sich der Nutzer aktuellen befindet, angezeigt.

---

### 3.2.3 Sounds

---

Bestimmte Spielsituationen, wie zum Beispiel das Beenden eines Levels, eine Kollision mit einem Hindernis und Springen, werden von entsprechenden und zum Thema des Spiels passenden Geräuscheffekten untermalt. Außerdem wird das Klicken eines Knopfes im Menü auch durch einen Ton an den Spieler rückgemeldet.

---

### 3.2.4 Steuerung

---

Das Spiel wird mit dem *Smovetec* Fahrradergometer gesteuert. Durch die Trittfrequenz wird die Geschwindigkeit des Raumschiffs beeinflusst. Die Neigung des Ergometers hingegen lässt das Raumschiff nach Links oder Rechts lenken. Um mit dem Raumschiff springen zu können, muss der Spieler eine Taste am Griff des Ergometers betätigen. Eine weitere Taste lässt den Spieler das aktuelle Level unmittelbar neu starten.

---

### 3.2.5 Levels

---

Es gibt eine feste Anzahl von Levels. Mit dem Erreichen eines höheren Levels steigt auch der Schwierigkeitsgrad und neue Elemente werden schrittweise eingeführt. Die Level befinden sich in verschiedenen Umgebungen, was man anhand der veränderten Hintergründe und Texturen erkennt, sich aber nicht auf das Verhalten des Spiels auswirkt.



---

## 3.3 Hardware

---

### 3.3.1 Smovetec

---

Bei dem Smovetec-System handelt es sich um eine Plattform, durch welche man Fahrradergometer horizontal beweglich machen kann. Dadurch kann man das Kurvenfahren mit einem echten Fahrrad sowie den Wiegetritt simulieren. Es eignet sich somit nicht nur für das übliche Konditionstraining, sondern auch zum Training von Balance und Geschicklichkeit.

Es können verschiedene Ergometer auf der Plattform festgeschraubt werden, wodurch man in der Wahl des Ergometers freigestellt ist.



**(a)** Das Smovetec-System im Einsatz



**(b)** Das Smovetec-System ohne Ergometer

**Abbildung 3.2:** Das Smovetec-System<sup>1</sup>

---

### 3.3.2 Texas Instruments SensorTag

---

Das Smovetec-System selbst ist nicht mit Sensoren ausgestattet um Trittfrequenz oder Neigung zu messen. Um die Trittfrequenz zu messen, kommt in diesem Projekt ein *SensorTag* der Firma Texas Instruments zum Einsatz. Hierbei handelt es sich um ein Bluetooth-fähiges Gerät, welches mit verschiedenen Sensoren ausgestattet ist. Diese umfassen:

- Temperatursensor
- Feuchtigkeitssensor
- Barometer
- Beschleunigungssensor

---

<sup>1</sup> Quelle: <http://www.smovetec.de/smovetec-produkt.htm>

- 
- Gyroskop
  - Magnetometer

Für unser Projekt wird davon ausschließlich der Beschleunigungssensor genutzt.

Um die Daten auszulesen, kommt eine erweiterte Version der Beispiellapp von Texas Instruments zum Einsatz, welche insbesondere auch noch die Beschleunigungssensoren des Smartphones nutzt, um die Neigung zu erfassen. Die Daten werden vom Smartphone per WLAN weiter an den Rechner übermittelt, der das Spiel ausführt.



**Abbildung 3.3:** Texas Instruments SensorTag

---

## 4 Recherche

---

### 4.1 Ähnliche Spielideen

---

#### 4.1.1 Skyroads

---

Das Spiel *Skyroads* von *Bluemoon* ist eine Art Jump and Run Spiel aus dem Jahr 1993. Ziel des Spiels ist es ein Raumschiff durch einen Parcour zu manövrieren. Der Spieler muss hierbei über Hindernisse oder Abgründe springen, beziehungsweise diesen ausweichen. Schafft er das Level nicht, kann er es beliebig oft neu starten. Das Prinzip des Spiels stammt aus dem Spiel *Kosmonauts*, welches von den gleichen Entwicklern bereits im Jahre 1989 veröffentlicht wurde. Zu Beginn des Spiels stehen dem Nutzer bereits alle Levels zur Verfügung. Er kann in einer Übersichtsseite das gewünschte Level auswählen. Mit *Skyroads XMAS-Special* wurde 1994 ein Nachfolger veröffentlicht. Außerdem existieren mittlerweile sehr viele andere Spiele, die nach einer ähnlichen Spielmechanik funktionieren.



Abbildung 4.1: Skyroads

---

#### 4.1.2 Audiosurf

---

Das Geschicklichkeitsspiel *Audiosurf* des Entwicklers *Invisible Handlebar* aus dem Jahr 2008 ist ebenfalls ein Beispiel für eine vergleichbare Spielidee. Auch hier steuert der Spieler ein raumschiffartiges Gefährt über Bahnen und muss dabei Hindernissen ausweichen. Die Besonderheit dieses Spiels ist jedoch, wie der Name andeutet, dass die Strecken automatisch passend zu einem Musikstück generiert werden, das während des Spiels passend dazu abgespielt wird.

Je nach Spielmodus muss der Spieler farbige Blöcke auf den Bahnen einsammeln oder grauen Blöcken ausweichen. Die Blöcke werden dann als Rechtecke in einem Raster am unteren Bildschirmrand gesammelt, jeweils auf der Bahn, auf der sie eingesammelt wurden. Sobald drei oder mehr Blöcke gleicher Farbe aneinander angrenzen, blinken diese für wenige Sekunden und lösen sich dann auf, wobei der Spieler dafür Punkte erhält.

Ziel des Spiels ist es, auf diese Weise auf jeder Strecke bzw. jedem Lied möglichst viele Punkte zu sammeln. Diese Punkte werden dann auch in einer lokalen und globalen Highscore-Liste eingetragen. Eine weitere Motivation bieten verschiedene Bonusregelungen und Herausforderungen, wie beispielsweise, das Abschließen einer Strecke ohne dass sich am Ende noch Blöcke im Raster befinden oder die Strecke zu beenden ohne einen einzigen grauen Block berührt zu haben.



**Abbildung 4.2:** Screenshot einer typischen Szene in Audiosurf<sup>1</sup>

Ähnlich wie bei unserem Spiel handelt es sich also bei *Audiosurf* um ein Geschicklichkeitsspiel, bei dem man ein Gefährt auf Bahnen hin- und herbewegt um Hindernissen auszuweichen. Anders ist jedoch vor allem, dass es bei *Audiosurf* nicht darum geht, ein Level überhaupt zu schaffen, sondern die Motivation für den Spieler über ein Punkte- und Highscore-System generiert wird. Das wird auch dadurch deutlich, dass es in *Audiosurf* nicht möglich ist, ein Level gar nicht zu schaffen. Kollisionen mit Hindernissen oder ein Überlaufen des Rasters führen hingegen nur zu einem Punktabzug, stören aber sonst nicht den Spielablauf.

## 4.2 Spiele mit Fahrradergometern

### 4.2.1 Atari Puffer

Der 1982 von Atari entwickelte *Atari Puffer* (Abb 4.3) [3] war das erste System, welches die Möglichkeit bieten sollte Videospiele mit dem Körper zu steuern. Plan war es eine Art Fahrradergometer an den *Atari* anzuschließen um somit die Spiele zu steuern. Eine erhöhte Umdrehungszahl der Pedale sollte dafür sorgen, dass zum Beispiel das Auto des Spielers im Spiel *Pole Position* schneller fuhr oder der Spieler im Spiel *Dig-Dug* schneller gräbt. Zusätzlich konnte ein Gamepad am Lenker montiert werden um zusätzliche Eingaben zu ermöglichen.

Auf Grund der Videospielekrise im Jahr 1984 kam es jedoch nie zur Markteinführung.



**Abbildung 4.3:** Atari Puffer

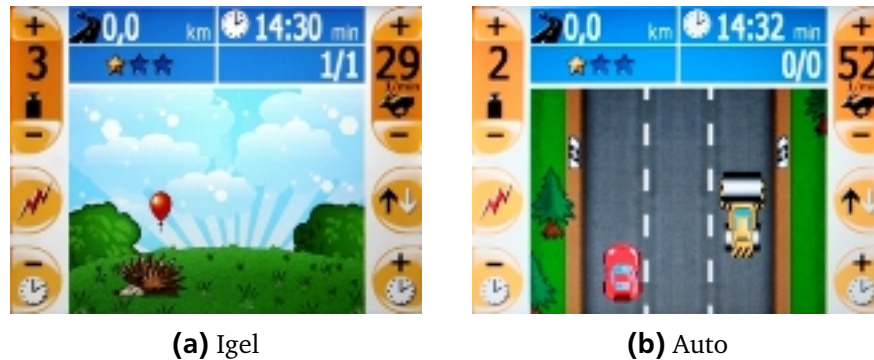
<sup>1</sup> Quelle: <https://market.myo.com/app/5474d1c6e4b081c4011c77b4/audiosurf-connector>

---

## 4.2.2 Thera Trainer

---

Die Firma *medica Medizintechnik GmbH* entwickelt, neben dem *Balance-Trainer*, verschiedene Geräte und Programme zur Unterstützung der Bewegungstherapie. Mögliche Anwendungszwecke sind beispielsweise die Therapie von Schlaganfällen, sowie von Muskel- und rheumatischen Erkrankungen. Es findet jedoch auch Anwendung auf dem Gebiet der Orthopädie und Kardiologie. Hierzu bietet der Hersteller verschiedene Trainingsgeräte in Form von Ergometern und Balance Plattformen.



**Abbildung 4.4:** Thera Trainer

Auf dem am Gerät montierten Bildschirm können verschiedene optische Feedbacksysteme angezeigt werden:

### **Igel:**

Durch unterschiedliche Kraftverteilung des linken und rechten Beins bewegt sich der Igel in die entsprechende Richtung. Herab fallende Gegenstände sollen dadurch zum platzen gebracht werden. (Abb. 4.4a)

### **Auto:**

Der Patient sieht eine Fahrbahn, sowie sein eigenes Fahrzeug. Ziel ist es möglichst viele andere Fahrzeuge zu überholen. Hierzu muss er schneller Treten als eine voreingestellte Drehzahl. Durch Veränderung der Kraftverteilung (siehe *Igel*) kann das Auto nach links und rechts bewegt werden. (Abb. 4.4b)

---

## 4.2.3 Jeff Sinclair - Edith Cowan University

---



**Abbildung 4.5:** CatEye GameBike



**Abbildung 4.6:** Side Scroller

Bei dem Exergaming Projekt der *Edith Cowan University Perth* [4] handelt es sich um einen einfachen Side-Scroller, der mit einem Fahrradergometer gesteuert wird (Abb. 4.6). Ziel des Spiels ist es einen



Hubschrauber in der richtigen Höhe zu halten und dabei Gegenstände einzusammeln, Hindernissen auszuweichen und Gegner abzuschießen. Je schneller der Spieler tritt, desto höher fliegt der Helikopter. Verringert der Spieler die Geschwindigkeit verliert der Helikopter an Höhe.

Als Ergometer kommt hierbei das *GameBike* der Firma *CatEye* (Abb. 4.5) zum Einsatz. Das Ergometer kann an die gängigsten, auf dem Markt erhältlichen Spielekonsolen angeschlossen werden und funktioniert mit sämtlichen Spielen, die auf Geschwindigkeit basieren. Für das Projekt der Universität Perth wurde eine Modifikation vorgenommen um es auch an einen PC anschließen zu können.

---

#### 4.2.4 Ergo Active

---

Entwickelt wurde *Ergo Active* an der *Technischen Universität Darmstadt* [2] zur Förderung von Serious Games for Health mit wissenschaftlichem Hintergrund, in Bezug auf Sport und Gesundheit. Es umfasst drei Spiele *Taubenjagd*, *Film* und *Balance* zum Ausdauer-, beziehungsweise Herz-Kreislauftraining. Gesteuert werden alle drei Spiele mittels eines Fahrradergometers.

##### **Taubenjagd:**

Hierbei handelt es sich um einen einfachen Sidescroller. Der Spieler steuert eine Brieftaube mit Hilfe des Fahrradergometers. Ziel ist es vorbei fliegende Briefe einzusammeln um Punkte zu bekommen. Um so schneller er tritt, desto höher fliegt die Taube. Der Spieler tritt dabei innerhalb eines vorher festgelegten Geschwindigkeit Intervalls, wodurch einerseits verhindert wird, dass der jeweilige Spieler ausserhalb seiner körperlichen Möglichkeiten trainiert und andererseits ein Adaptionsmechanismus für verschiedene Spieler gegeben ist.

##### **Film:**

*Film* ermöglicht dem Spieler zum Beispiel Etappen der *Tour de France* nachzufahren. Hierfür wird ein Film der jeweiligen Strecke gezeigt. Je nach dem wie schnell der Spieler Tritt um so schneller wird der Film der jeweiligen Etappe abgespielt. Ähnlich wie bei Taubenjagd findet auch hier eine Überwachung des Spielers statt. Über-, oder unterschreitet er seine Leistungsgrenzen gibt das Spiel ein Signal um den Spieler daraufhin zuweisen wieder innerhalb seines persönlichen Intervalls zu trainieren.

##### **Balance:**

Bei *Balance* hat der Spieler zwei Aufgaben. Er balanciert einen Clown auf einem Ball. Hierzu muss er mittels Ergometer eine vorgegebene Geschwindigkeit halten, beziehungsweise darf maximal nur um einen bestimmten Wert davon abweichen. Während er den Clown in der Balance hält fallen Luftballons herab, die er mittels Maus oder Wiimote abschießen muss, um Punkte zu sammeln. Die voreingestellte Geschwindigkeit und die maximale Abweichung dienen hierbei zur Überwachung, damit sich der Spieler innerhalb seiner Leistungsgrenzen bewegt.



**Abbildung 4.7:** Ergo Active - Balance



**Abbildung 4.8:** Taubenjagd

---

### 4.3 Fazit der Recherche

---

In Bezug auf die benutzte Hardware von anderen Systemen, erkennt man, dass Fahrradergometer sehr oft als das bevorzugte Mittel zur Eingabe genutzt werden. Fahrradergometer bieten jedoch nur eine sehr limitierte Bandbreite von möglichen Eingaben. Streng genommen findet man in den betrachteten Systemen ausschließlich zwei Aktionen, die Erhöhung der Trittfrequenz, sowie die Verringerung, beziehungsweise Abbremsen um eine Spielfigur zu steuern und Aktionen einzuleiten.

Auch in Bezug auf die beanspruchten Körperpartien werden stets nur der untere Teil des Körpers, speziell die Beine, belastet. Einzige Ausnahme sind die Produkte der Firma *medica Medizintechnik GmbH*. Sie bieten ebenfalls Ergometer die mit der Hand und den Armen bedient werden können und sind somit auch, zum Beispiel, für Rollstuhlfahrer geeignet. Alle Systeme jedoch benutzen jeweils nur einen Teil des Körpers zur Eingabe - entweder Beine oder Oberkörper. Keines der betrachteten Systeme bietet die Möglichkeit sowohl Ober- als auch Unterkörper zu aktivieren.

Kaum ein System bietet neben der Trittfrequenz, beziehungsweise Geschwindigkeit des Ergometers weitere Eingabeoptionen. Das in unserem Projekt genutzte Ergometer bietet diese Möglichkeit jedoch. Durch die Gewichtsverlagerung mit Hilfe des Oberkörpers können wir in unserem Spiel eine weitere Ebene der Kontrollmöglichkeiten erreichen. Hierdurch wird ermöglicht, dass sich die Spielfigur in diesem Projekt in zwei, statt wie in den anderen Publikationen, nur in einer Dimension bewegen kann. Darüber hinaus soll in diesem Projekt eine weitere, dritte Dimension, der Bewegungsfreiheit, das Springen, eingebaut werden. Durch die verschiedenen Aspekte der Steuerung soll die Motivation des Spielers möglichst hoch gehalten werden, da das Spiel vielfältiger ist. Andere Projekte beschränken sich lediglich auf eine mögliche Eingabe, wodurch die Gefahr entsteht, dass das Spiel schnell eintönig und langweilig wird.

Trotz der Erweiterten Eingabemöglichkeit dieses Systems im Vergleich zu anderen Projekten, besitzt sie dennoch nicht die volle Bandbreite der möglichen Kommandos, wie es zum Beispiel Spiele bieten, die an einem Stationären Computer mit Tastatur und Maus gesteuert werden. Aus diesem Grund muss das entwickelte Spiel ebenfalls eine einfache, aber interessante, Spielmechanik mit hohem Motivationsfaktor haben. Die beiden betrachteten Spiele *Skyroads* und *Audiosurf* sind hierfür gute Beispiele. Sie kommen jeweils mit verhältnismäßig wenigen Eingaben aus, bieten jedoch trotzdem eine attraktive und motivierende Spielmechanik. Aus diesem Grund sollen diese beiden Spiele eine Art Vorbild für das im Rahmen dieses Praktikums entwickelten Bewegungsspiels darstellen. Ziel wird es sein interessante Konzepte zu adaptieren, beziehungsweise zu erweitern um die Funktionen des Fahrradergometers möglichst optimal nutzen zu können.

---

## 5 Methoden und Vorgehen

---

Bei der Durchführung des Projekts hat sich das Team an einem agilen Entwicklungsmodell orientiert. Das bedeutet, dass man nicht zu Beginn eine ausführliche Planung aller Features bis ins Detail durchgeführt hat, sondern, dass vielmehr jede Idee und jeden Ansatz direkt als Mock-up im Spiel ausprobiert wurde, um schnelles und exaktes Feedback über die Machbarkeit und Sinnhaftigkeit des Features zu erlangen. Dieses Vorgehen hat den Effekt, dass manche Features begonnen, aber dann nachdem man festgestellt hat, dass sie so nicht machbar oder nicht zielführend sind, wieder verworfen wurden. Auf der anderen Seite ermöglicht dieses Vorgehen aber auch ein sehr schnelles Reagieren auf neue Begebenheiten. Insbesondere wird so sichergestellt, dass keine überflüssigen Features erst aufwändig geplant werden und sich dann in der Realität als obsolet herausstellen. Aber insbesondere ermöglicht diese Methode ein sehr schnelles Lernen aus Feedback um so ein für den Nutzer optimales Spielerlebnis zu schaffen.

### Controlling

Um den Fortschritt im Projekt und auch mögliche Rückschläge immer möglichst gut im Blick zu behalten, bestreiten wir einen möglichst großen Teil der Entwicklung immer mit dem Team gemeinsam. Wenn dies zeitweise nicht möglich ist, so halten wir uns per E-Mail detailliert auf dem Laufenden. Dennoch ist ein gegenseitiges Update beim nächsten persönlichen Treffen unumgänglich.

So oft möglich erfolgt die Entwicklung in den Räumlichkeiten des Lehrstuhls, sodass bei Schwierigkeiten, aber auch für regelmäßiges Feedback der Betreuer unkompliziert hinzugezogen werden kann.

### „lessons-learned“

Ein genaues Durchführen und Einhalten eines vorab erstellten Plans ist in so einem Softwareprojekt nicht möglich. Gerade wenn man wenig Erfahrung mit den eingesetzten Technologien hat, ist ein Abschätzen der Aufwände nicht einfach und äußerst ungenau. Daher hat es sich am praktikabelsten erwiesen, von Tag zu Tag neu zu priorisieren und zu entscheiden. Trotzdem konnte man das große Ganze immer gut im Blick behalten und man wusste immer in etwa, wo man im Projekt stehen.

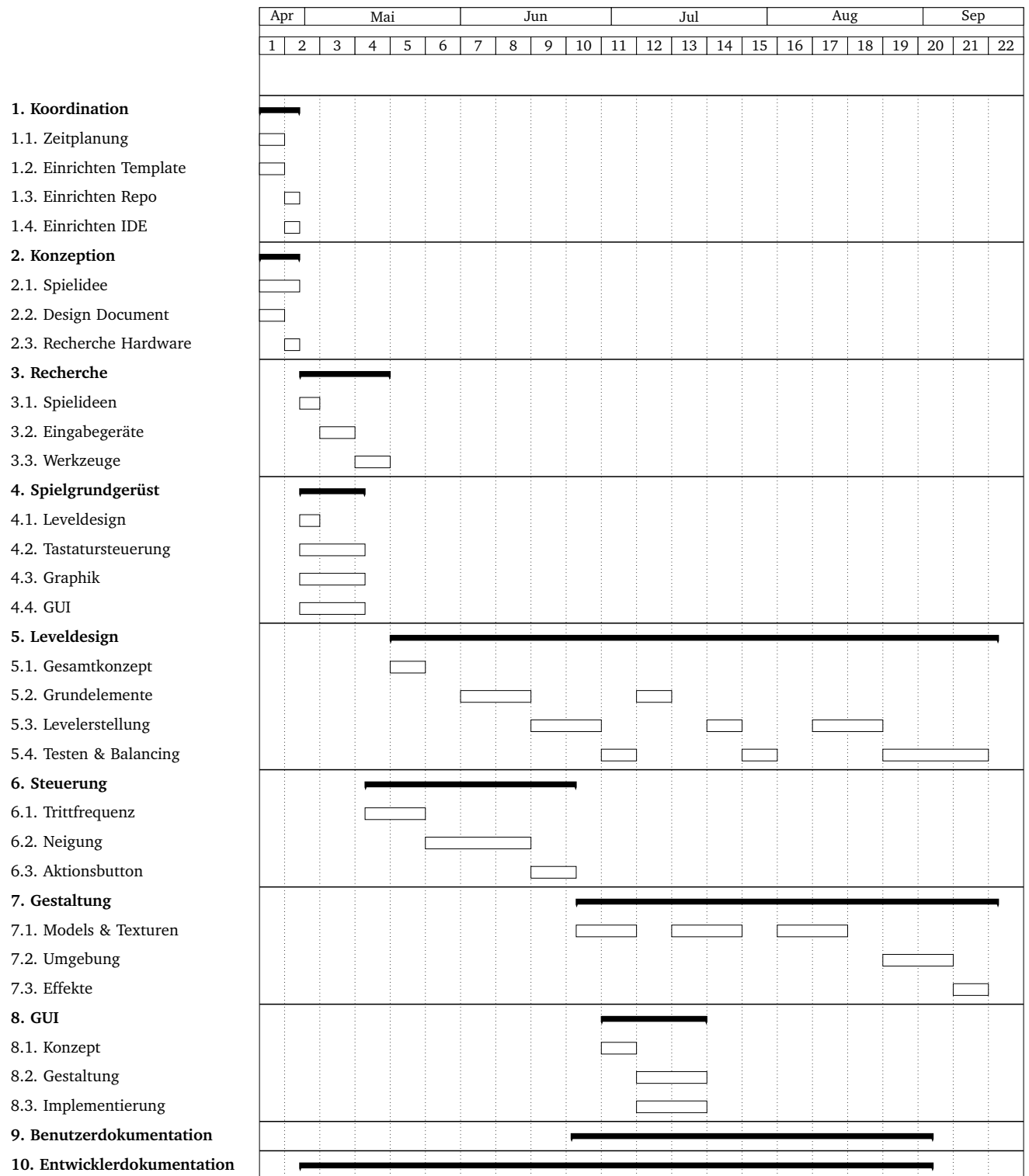


## 6 Planung

### 6.1 Projektplanung

Teilobjekte und Arbeitspakete	Summe (Personentage)	Wer?	
		Alex	Simon
<b>1. Koordination und Organisation</b>	7	3	4
1.1. Zeitplanung	2	1	1
1.2. Einrichten des Dokumentationstemplates	2	1	1
1.3. Einrichten der Repositories	1	0	1
1.4. Einrichten der Entwicklungsumgebung	2	1	1
<b>2. Konzeption und Findung der Spielidee</b>	7	5	2
2.1. Findung der Spielidee	3	2	1
2.2. Game Design Document	2	2	0
2.3. Recherche zur Hardware	2	1	1
<b>3. Recherche</b>	8	5	3
3.1. Ähnliche Spielideen recherchieren	2	1	1
3.2. Spiele mit ähnlichen Eingabegeräten recherchieren	4	3	1
3.3. Geeignete Werkzeuge und Technologien	2	1	1
<b>4. Entwicklung des Spielgrundgerüsts</b>	7	4	3
4.1. Rudimentäres Leveldesign	2	1	1
4.2. Einfache Steuerung per Tastatur	2	1	1
4.3. Einfache graphische Darstellung	2	1	1
4.4. Einfaches graphische Benutzerschnittstelle	1	1	0
<b>5. Leveldesign</b>	27	12	15
5.1. Entwicklung eines Levelgesamtkonzepts	7	3	4
5.2. Gestaltung von Grundelementen für die Levels	6	3	3
5.3. Erstellung einer größeren Anzahl Levels	8	3	5
5.4. Testen und Balancing	6	3	3
<b>6. Steuerung</b>	15	7	8
6.1. Anbindung der Trittfrequenz	3	1	2
6.2. Anbindung der Fahrradneigung	8	4	4
6.3. Anbindung eines Aktionsbuttons	4	2	2
<b>7. Audiovisuelle Gestaltung</b>	17	9	8
7.1. Gestaltung von Models und Texturen	10	5	5
7.2. Gestaltung der Spielumgebung	4	2	2
7.3. Gestaltung von Effekten	3	2	1
<b>8. GUI</b>	10	4	6
8.1. Entwicklung eines GUI Konzepts	2	1	1
8.2. Gestaltung der GUI-Elemente	4	1	3
8.3. Implementierung der GUI im Spiel	4	2	2
<b>9. Benutzerdokumentation</b>	7	3	4
<b>10. Entwicklungsdokumentation</b>	7	4	3
<b>Gesamt</b>	<b>98</b>	<b>49</b>	<b>49</b>

## 6.2 Zeitplanung



---

## 6.3 Meilensteine

---

### Projektdefinition – 29.4.

Die Projektdefinition enthält die *tabellarische Planung* (vgl. 6.4.1) des Projektes, das *Gantt-Diagramm* (vgl. 6.4.1) und das *Game Design Document* (vgl. 6.4.2). Ziel ist es, eine erste Übersicht über die Spielidee, den geschätzten Aufwand und die zeitliche Planung des Projektes zu gewinnen.

### Mock-up – 13.5.

Das Mock-up soll dem Projektteam und dem Betreuer eine bereits spielbare Veranschaulichung der Spielidee bieten. Daher enthält dieser Meilenstein das wesentliche *Spielgrundgerüst* (vgl. 6.4.4).

### Alpha-Version – 23.6.

Die Alpha-Version soll bereits erahnen lassen, wie das Spiel zu Projektende aussehen soll. Der Umfang ist aber noch sehr eingeschränkt. Zu diesem Meilenstein gehören ein *einfaches Levelpaket* (vgl. 6.4.5), die *Steuerung* (vgl. 6.4.6), das *Spielermodell* und ein *einfaches Texturenpaket* (vgl. 6.4.7).

### Beta-Version – 19.7.

In der Beta-Version soll das Spiel dann schon alle Elemente enthalten, die es auch in der fertigen Version hat, wenn auch in geringerem Umfang. Daher kommt in dieser Version noch eine *graphische Benutzeroberfläche* (vgl. 6.4.8) hinzu.

### Pre-finale Ausarbeitung – 2.9.

Die pre-finale Ausarbeitung enthält bereits alle Inhalte der finalen Ausarbeitung, muss aber noch gegebenenfalls überarbeitet werden. Außer den Inhalten der Projektdefinition enthält diese eine *Recherche* (vgl. 6.4.3), sowie die *Benutzer- und Entwicklerdokumentation* (vgl. 6.4.9, 6.4.10).

### Finale Version – 15.9.

Zur finalen Version werden alle noch für das fertige Spiel fehlenden Arbeiten erledigt. Dies beinhaltet ein *vollständiges Levelpaket* (vgl. 6.4.5) und *vollständige Texturen, Umgebungen und Effekte* (vgl. 6.4.7).

---

## 6.4 Arbeitspakete

---

---

### 6.4.1 Koordination und Organisation

---

<b>Beginn:</b> 22.4.	<b>Ende:</b> 29.4.	<b>Lead:</b> alle	<b>Beteiligte:</b> alle	<b>Aufwand:</b> 7 Personentage
----------------------	--------------------	-------------------	-------------------------	--------------------------------

---

#### Zielstellung

Ziel des Arbeitspaketes *Koordination und Organisation* ist es, die Grundlagen für einen erfolgreichen Start des Projektes zu legen. Nach Abschluss des Arbeitspaketes Koordination und Organisation sollen die Gruppenmitglieder in der Lage sein, die Arbeit am eigentlichen Produkt aufnehmen zu können.

#### Beschreibung

Das Arbeitspaket *Koordination und Organisation* gliedert sich in vier Teilobjekte. Im ersten Teilobjekt wird die Zeitplanung des Projektablaufs vorgenommen. Hierfür wird neben einer tabellarischen Gliederung der Arbeitspakete ein Gantt-Diagramm des zeitlichen Ablaufs erstellt.

Anschließend werden Templates für das Projektmanagementdokument eingerichtet und angepasst. Dafür, und für den Quellcode des Projekts, werden im dritten Teilobjekt Versionsverwaltungsrepositories mit Git angelegt und eingerichtet. Abschließend beginnen die Projektbeteiligten damit, die nötigen Entwicklungsumgebungen und Tools einzurichten.

---

## Rolle der Beteiligten

Da das Arbeitspaket *Koordination und Organisation* für den gesamten Projektablauf entscheidend ist, sind alle Projektmitglieder hier gleichermaßen beteiligt. Dabei ist jedes Mitglied für die Einrichtung seiner Werkzeuge selbst verantwortlich.

## Deliverables

- Tabellarischer Projektplan, Dokument, bis 29.4.
- Gantt-Diagramm, Dokument, bis 29.4.

## Meilensteine

- Der *tabellarische Projektplan* und das *Gantt-Diagramm* sind Teil des Meilensteins *Projektdefinition*.

---

### 6.4.2 Konzeption und Findung der Spielidee

---

<b>Beginn:</b> 22.4.	<b>Ende:</b> 29.4.	<b>Lead:</b> Alex	<b>Beteiligte:</b> alle	<b>Aufwand:</b> 7 Personentage
----------------------	--------------------	-------------------	-------------------------	--------------------------------

---

## Zielstellung

Ziel des Arbeitspaketes *Konzeption und Findung der Spielidee* ist es, ein Konzept für ein Bewegungsspiel mit einem Fahrradergometer bedient werden soll, zu entwickeln. Außerdem soll ein *Game Design Document*, welches das Konzept genauer erläutert, verfasst werden.

## Beschreibung

Das Arbeitspaket *Konzeption und Findung der Spielidee* untergliedert sich in drei Teilobjekte. Zunächst erarbeiten die Projektmitglieder eine auf die Aufgabenstellung passende Spielidee. Das gefundene Konzept wird im zweiten Teilobjekt detailliert ausgearbeitet und als *Game Design Document* ausformuliert. Da die bereitgestellte Hardware in Form des Eingabegeräts Fahrradergometer für die Projektbeteiligten unbekannt ist, wird außerdem eine umfassende Recherche zu den technischen Möglichkeiten und Einschränkungen durchgeführt.

## Rolle der Beteiligten

Alle Mitglieder des Projektes erarbeiten gemeinsam das Konzept für die Spielidee, sowie die Recherche zur Hardware. Das *Game Design Document* wird überwiegend vom Hauptverantwortlichen des Arbeitspaketes verfasst.

## Deliverables

- *Game Design Document*, Dokument, bis 29.4.

## Meilensteine

- Das *Game Design Document* ist Teil des Meilensteins *Projektdefinition*.

---

### 6.4.3 Recherche

---

<b>Beginn:</b> 30.4.	<b>Ende:</b> 17.5.	<b>Lead:</b> Alex	<b>Beteiligte:</b> alle	<b>Aufwand:</b> 8 Personentage
----------------------	--------------------	-------------------	-------------------------	--------------------------------

---

## Zielstellung

Ziel des Arbeitspaketes *Recherche* ist es, eine Übersicht über einige, bereits existierende, Spiele mit vergleichbarer Steuerung beziehungsweise ähnlicher Spielidee zu gewinnen. Außerdem sollen für die Umsetzung des Projektes geeignete Technologien und Werkzeuge gesucht und bewertet werden.

---

## Beschreibung

Das Arbeitspaket *Recherche* gliedert sich in drei Teilobjekte, wobei sich die Pakete zur Recherche von ähnlichen Spielideen und Spielen mit ähnlichen Eingabegeräten in ihrer Ausgestaltung überschneiden. Der Fokus der Recherche liegt sowohl auf kommerziellen Produkten, als auch auf wissenschaftlichen Projekten.

Nach Abschluss des Arbeitspaketes *Konzeption und Findung der Spielidee* wird in diesem Arbeitspaket eine Recherche der für die Umsetzung der Spielidee passenden Werkzeuge, Spielengines und Technologien durchgeführt. Darauf aufbauend wird eine Auswahl geeigneter Werkzeuge getroffen.

## Rolle der Beteiligten

Der Leiter des Arbeitspaketes übernimmt die Recherche bestehender Spiele und Systeme. Der zweite Projektteilnehmer führt parallel die Recherche über Technologien und Werkzeuge durch.

## Deliverables

- State-of-the-art-Recherche zu Spielen und Systemen, Dokument, bis 2.9.

## Meilensteine

- Der Bericht zur Recherche ist Teil des Meilensteins *Pre-finale Ausarbeitung*.

---

### 6.4.4 Entwicklung des Spielgrundgerüsts

---

<b>Beginn:</b> 30.4.	<b>Ende:</b> 12.5.	<b>Lead:</b> alle	<b>Beteiligte:</b> alle	<b>Aufwand:</b> 7 Personentage
----------------------	--------------------	-------------------	-------------------------	--------------------------------

---

## Zielstellung

Ziel des Arbeitspaketes *Entwicklung des Spielgrundgerüsts* ist es, nach der Findung des Konzeptes eine prototypische Implementierung des Konzeptes zu erstellen und zu testen. Zweck des Prototypen ist es, als Machbarkeitsstudie des zuvor entworfenen Konzeptes zu dienen.

## Beschreibung

Das Arbeitspaket *Entwicklung des Spielgrundgerüsts* gliedert sich in die vier Teilaspekte Levelbau, Steuerung, Grafik und graphische Nutzerschnittstelle.

Dabei soll das Level nur minimal die Idee des Spiels widerspiegeln und die grundlegende Spielmechanik ermöglichen. Die Steuerung soll zunächst nur über die Tastatur erfolgen, um den Entwicklungsaufwand gering zu halten. Die eigentliche Steuerung mit Fahrradergometer wird in einem späteren Arbeitspaket realisiert. Im Zuge des Levelbaus werden einfache 3D-Modelle als Levelobjekte erstellt und mit einfachen Texturen versehen. Abschließend wird eine rudimentäre graphische Nutzerschnittstelle programmiert, die hauptsächlich zur Veranschaulichung der Funktionalität der späteren Nutzerschnittstelle dient.

## Rolle der Beteiligten

Damit alle Beteiligten einen ähnlichen Wissensstand bezüglich der verwendeten Technologien erreichen, erfolgt die Entwicklung in diesem Arbeitspaket ausschließlich gemeinsam nach dem Prinzip des *Pair Programming*[1, S. 42ff].

## Deliverables

- Spielgrundgerüst, Software, bis 13.5.

---

## Meilensteine

- Das Spielgrundgerüst ist Teil des Meilensteins *Mock-Up*.

---

### 6.4.5 Leveldesign

---

**Beginn:** 18.5.    **Ende:** 15.9.    **Lead:** Simon    **Beteiligte:** alle    **Aufwand:** 27 Personentage

---

#### Zielstellung

Ziel des Arbeitspaketes *Leveldesign* ist es, eine möglichst hohe Anzahl interessanter Levels zu erstellen, welche zunehmend schwerer und fordernder werden. Abschließend sollen die Levels getestet und balanciert werden.

#### Beschreibung

Das Arbeitspaket *Leveldesign* gliedert sich in vier Teilpakete. Im ersten Schritt wird ein grundlegendes Levelgesamtkonzept erarbeitet. Basierend auf diesem Konzept werden anschließend die einzelnen Levels und im Zuge dessen die Levelgrundobjekte, wie zum Beispiel Bahnen, Rampen und Tunnel erstellt. Parallel dazu wird jedes Level nach der Fertigstellung ausgiebig evaluiert und gegebenenfalls angepasst. Nach Abschluss der Erstellung aller Levels werden diese in Bezug auf Probleme des Schwierigkeitsgrades analysiert und überarbeitet. Hierdurch soll eine angenehme Lernkurve für den Spieler gewährleistet werden.

#### Rolle der Beteiligten

Alle Projektbeteiligten sind gleichermaßen an der Erstellung, Evaluierung und Balancing der Levels beteiligt. Die Erstellung des Gesamtkonzepts wird federführend vom Lead des Arbeitspakets durchgeführt. Die Erstellung der Spielobjekte und Levels erfolgt zu Beginn gemeinsam, um ein einheitliches Gesamtbild zu gewährleisten. Später wird die Levelerstellung getrennt durchgeführt, wodurch die Effizienz gesteigert werden soll.

#### Deliverables

- Einfaches Levelpaket, Software, bis 23.6.
- Vollständiges Levelpaket, Software, bis 15.9.

## Meilensteine

- Das einfache Levelpaket ist Teil des Meilensteins *Alphaversion*.
- Das vollständige Levelpaket ist Teil des Meilensteins *Finale Version*.

---

### 6.4.6 Steuerung

---

**Beginn:** 13.5.    **Ende:** 23.6.    **Lead:** Simon    **Beteiligte:** alle    **Aufwand:** 15 Personentage

---

#### Zielstellung

Ziel des Arbeitspaketes *Steuerung* ist es, alle für das Spiel notwendigen Steuerungsmechanismen zu implementieren und zu testen. Dies sind die Trittfrequenz und Neigung des Fahrradergometers und eine am Lenker angebrachte Aktionstaste.

---

## Beschreibung

Das Arbeitspaket *Steuerung* gliedert sich entsprechend der Eingabemöglichkeiten in drei Teilobjekte, die zeitlich parallel bearbeitet werden können. Durch die Trittfrequenz, die der Spieler auf dem Fahrradergometer leistet, wird die Geschwindigkeit des Raumschiffes im Spiel gesteuert. Die Neigung des Fahrradergometers wird gemessen, um Richtungswechsel des Raumschiffes im Level zu vollziehen. Zusätzlich wird eine Aktionstaste am Lenker des Fahrradergometers dazu verwendet um springen zu können. Hierdurch kann der Spieler Hindernissen ausweichen oder die Fahrbahn wechseln.

Alle drei Steuerungsmechanismen werden ausgiebig getestet, um einen angenehmen und flüssigen Spielablauf zu garantieren.

## Rolle der Beteiligten

Da mit einigen Schwierigkeiten bei der Anbindung der externen Steuerungshardware gerechnet wird, erfolgt die Umsetzung dieses Arbeitspaketes gemeinsam und nach dem Prinzip des *Pair Programming* [1, S. 42ff].

## Deliverables

- Implementierung der Steuerung, Software, bis 23.6.

## Meilensteine

- Die Steuerung ist Teil des Meilensteins *Alphaversion*.

---

### 6.4.7 Audiovisuelle Gestaltung

---

<b>Beginn:</b> 24.6.	<b>Ende:</b> 15.9.	<b>Lead:</b> Alex	<b>Beteiligte:</b> alle	<b>Aufwand:</b> 17 Personentage
----------------------	--------------------	-------------------	-------------------------	---------------------------------

---

## Zielstellung

Ziel des Arbeitspaketes *Audiovisuelle Gestaltung* ist es, das Spiel graphisch zu gestalten. Dies umfasst das Spielermodell, Texturen, die Spielumgebung und Effekte. Nach Abschluss des Paketes sollte das Spiel einen stimmigen und ansprechenden graphischen Gesamteindruck vermitteln.

## Beschreibung

Das Arbeitspaket *Audiovisuelle Gestaltung* gliedert sich drei Teilpakete: *Gestaltung von Models und Texturen*, *Gestaltung der Spielumgebung* und *Gestaltung von Effekten*. Das erste Teilpaket umfasst vor allem die graphische Gesamtgestaltung, das zweite Teilpaket die Erstellung einer dazu passenden Spielumgebung. Abschließend wird das Spielerlebnis durch ansprechende audiovisuelle Effekte, wie zum Beispiel Explosionen, angereichert.

## Rolle der Beteiligten

Alle Projektbeteiligten werden bei der Gestaltung der Spielelemente einbezogen, da diese sehr wichtig für den Gesamteindruck des Spiels ist. Die Modellierung von 3D-Modellen wird dabei insbesondere vom Hauptverantwortlichen des Arbeitspaketes durchgeführt.

## Deliverables

- Spielermodell, Software, bis 23.6.
- Einfaches Texturenpaket, Software, bis 23.6.
- Vollständige Texturammlung, Software, bis 15.9.
- Spielumgebung, Software, bis 15.9.
- Effekte, Software, bis 15.9.

---

## Meilensteine

- Das Spielermodell und das einfache Texturenpaket sind Bestandteile des Meilensteins *Alphaversion*.
- Das vollständige Texturenpaket, die Spielumgebung und die Effekte sind Teil des Meilensteins *Finale Version*

---

### 6.4.8 GUI

---

**Beginn:** 29.6.    **Ende:** 19.7.    **Lead:** Simon    **Beteiligte:** alle    **Aufwand:** 10 Personentage

---

#### Zielstellung

Ziel des Arbeitspaketes *GUI* ist es, eine Benutzeroberfläche zu dem eigentlichen Spiel zu erstellen, welche dem Spieler Informationen zum aktuellen Spielzustand darstellt. Die GUI sollte gut ablesbar sein und sich thematisch an der Gesamtgestaltung des Spiels orientieren.

#### Beschreibung

Das Arbeitspaket *GUI* gliedert sich drei Teilpakete: Zuerst wird ein Konzept entworfen, wie die Benutzeroberfläche aussehen soll. Im zweiten Teilpaket werden die einzelnen Elemente gestaltet und die erforderlichen Grafiken erstellt. Parallel dazu wird auch an der Implementierung und Verknüpfung mit dem Spiel gearbeitet. Dabei wird auch getestet, ob sich die Oberflächenelemente gut ablesen lassen und notfalls noch korrigierend eingegriffen.

#### Rolle der Beteiligten

Alle Projektbeteiligten erstellen gemeinsam ein GUI-Konzept. Der Hauptverantwortliche des Arbeitspaketes *GUI* erstellt anschließend die vom Konzept vorgegebenen Grafiken. Die fertigen Grafiken werden vom anderen Projektbeteiligten in die Software integriert und mit Funktionalität versehen.

#### Deliverables

- Graphische Benutzeroberfläche, Software, bis 19.7.

## Meilensteine

- Die graphische Benutzeroberfläche ist Teil des Meilensteins *Beta-Version*.

---

### 6.4.9 Benutzerdokumentation

---

**Beginn:** 23.6.    **Ende:** 2.9.    **Lead:** Alex    **Beteiligte:** alle    **Aufwand:** 7 Personentage

---

#### Zielstellung

Ziel des Arbeitspaketes *Benutzerdokumentation* ist es, eine verständliche Dokumentation des Spieles für den Endanwender bereit zu stellen. Sie soll vollständig, aber dennoch einen für den Benutzer annehmbaren Umfang haben, sowie leicht verständlich sein.

#### Beschreibung

Das Arbeitspaket *Benutzerdokumentation* ist nicht weiter unterteilt, stattdessen wird nach Fertigstellung jedes, für den Benutzer relevanten, Arbeitspaketes die dazugehörige Benutzerdokumentation erstellt. Abschließend werden die einzelnen Kapitel in eine ansprechende Gesamtdokumentation zusammengefasst und Korrektur gelesen.



---

## Rolle der Beteiligten

Die Verantwortlichen der jeweiligen Arbeitspakete erstellen die entsprechenden Kapitel der Dokumentation. Sollte ein Arbeitspaket keinen Hauptverantwortlichen haben, so wird diese Aufgabe von allen Projektbeteiligten gleichermaßen bearbeitet.

Der Verantwortliche des Arbeitspaketes *Benutzerdokumentation* fasst die jeweiligen Abschnitte zusammen in eine Gesamtdokumentation und beseitigt eventuelle Rechtschreibfehler.

## Deliverables

- Benutzerdokumentation, Dokument, bis 2.9.

## Meilensteine

- Die Benutzerdokumentation ist Bestandteil des Meilensteins *Pre-finale Ausarbeitung*.

---

### 6.4.10 Entwicklerdokumentation

---

<b>Beginn:</b> 30.4.	<b>Ende:</b> 2.9.	<b>Lead:</b> Simon	<b>Beteiligte:</b> alle	<b>Aufwand:</b> 7 Personentage
----------------------	-------------------	--------------------	-------------------------	--------------------------------

---

## Zielstellung

Ziel des Arbeitspaketes *Entwicklerdokumentation* ist es, eine umfassende Dokumentation des Projektes für nachfolgende Entwicklungsschritte bereit zu stellen. Vor allem Entwickler, denen das Projekt unbekannt ist, sollen sich schnell im Konzept und der Architektur zurechtfinden. Darüber hinaus soll die Entwicklerdokumentation einen Überblick über das Projektmanagement während der gesamten Laufzeit bieten.

## Beschreibung

Das Arbeitspaket *Entwicklerdokumentation* ist nicht weiter unterteilt, stattdessen wird nach Fertigstellung jedes Arbeitspaketes die dazugehörige Entwicklerdokumentation erstellt. Abschließend werden die einzelnen Kapitel in eine ansprechende Gesamtdokumentation zusammengefasst und Korrektur gelesen.

## Rolle der Beteiligten

Die Verantwortlichen der jeweiligen Arbeitspakete erstellen die entsprechenden Kapitel der Dokumentation. Sollte ein Arbeitspaket keinen Hauptverantwortlichen haben, so wird diese Aufgabe von allen Projektbeteiligten gleichermaßen bearbeitet.

Der Verantwortliche des Arbeitspaketes *Entwicklerdokumentation* fasst die jeweiligen Abschnitte zusammen in eine Gesamtdokumentation und beseitigt eventuelle Rechtschreibfehler.

## Deliverables

- Entwicklerdokumentation, Dokument, bis 2.9.

## Meilensteine

- Die Entwicklerdokumentation ist Bestandteil des Meilensteins *Pre-finale Ausarbeitung*.

---

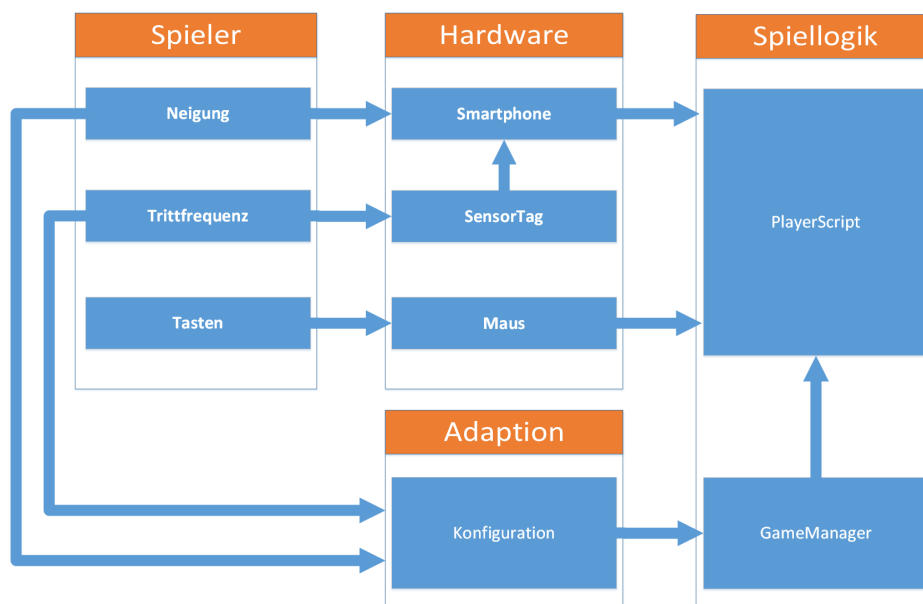
## 7 Umsetzung

---

Vorgabe aus der Aufgabenstellung war, das Spiel, das im Rahmen dieses Projekts entwickelt wird, mit der Spieleengine Unity zu entwickeln. Außerdem sollte die Trittfrequenz des Fahrradergometers mittels *Texas Instruments SensorTag* via Bluetooth zur Steuerung des Spiels erfasst werden. Damit war der technologische Rahmen für das Projekt bereits gesetzt.

Unity gibt eine relativ klare Struktur für die Umsetzung vor: Es gibt Szenen, in diesen Szenen mehrere Spielobjekte und an diesen dann ein oder mehrere Skripte, die das Verhalten dieses Objektes beeinflussen. Auf die wichtigsten dieser Skripte wird später detailliert eingegangen.

Für die Anbindung des Bluetooth-SensorTags erhielten wir von unserem Betreuer eine Android-App, welche die Bluetoothverbindung des Handys nutzt um mit dem Sensor zu kommunizieren, welcher an einem der Pedale befestigt wird. Zusätzlich erfasst das Smartphone die Neigung des Rades. Beide Informationen werden von der App gemeinsam über eine UDP-Verbindung an den Rechner geschickt, auf dem das Spiel läuft. Dieses Zusammenspiel von SensorTag, Smartphone und Spiel wird in Abbildung 7.1 dargestellt.



**Abbildung 7.1:** Zusammenspiel der verschiedenen Steuerungskomponenten

---

### 7.1 Steuerung

---

Zu den wichtigsten Skripten gehört das *PlayerScript*. Dieses verarbeitet die verschiedenen Eingaben des Spielers und wandelt diese in Bewegungen um. Je nach Konfiguration wird entweder die Steuerung per Fahrradergometer oder zu Testzwecken die Steuerung per Tastatur verwendet.

Die Daten für die Steuerung per Fahrradergometer werden vom *SensorDataReceiver*-Skript erfasst, indem dieses ein UDP-Socket erstellt, mit dem sich dann das Smartphone verbindet. Das Parsing der Daten erfolgt ebenfalls in diesem Skript.

---

## 7.2 Leveleditor

---

Von Anfang an war klar, dass eine große und abwechslungsreiche Anzahl an Levels für ein gut funktionierendes Spiel wichtig ist. Zuerst war die Idee, diese Levels direkt in Unity als einzelne Szenen zu bauen, doch das erschien uns zu umständlich. Aber selber einen vollständigen Leveleditor zu bauen, hätte wohl den Umfang des Projektes gesprengt.

Die Lösung war, einfache Textdateien zu nutzen, und somit einen Texteditor zum Leveleditor zu machen. Die Idee ist dabei, dass jeweils ein Zeichen eine Zelle des Levels füllt. Je nachdem, welches Zeichen dann dort steht, erscheint an dieser Stelle der ein oder andere Baustein, oder es bleibt eine Lücke. Ein Beispiel eines solchen Levels ist in Abbildung 7.2 zu sehen.

Die Bedeutung der verschiedenen Buchstaben wird im Inspektor des Unity-Editors konfiguriert. Hierbei wird jedem Buchstaben ein Prefab zugewiesen, also ein vorgefertigtes Spielobjekt inklusive eventueller Skripte, die das Verhalten beeinflussen.

Ein Buchstabe ohne Zuordnung verursacht dabei keinen Fehler, sondern wird als Lücke interpretiert. Die Konvention ist jedoch, dass nur Punkte für Lücken benutzt werden. Die von uns im Spiel gewählte Zuordnung ist die Folgende:

- 1: Einfacher Streckenabschnitt
- s: Spieler-Startposition
- g: Ziel des Levels (Goal)
- t: Strecke mit Tunnel
- r: Rampe
- b: Quadratischer Block in der Mitte des Abschnitts
- f: Block der über die ganze Breite und Länge des Abschnitts geht

25
...g...
...l...
...l...
...l...
...L...
...L...
...r...
..ftf..
...ll..
....l..
....ll.
.....s.

Großbuchstaben werden genau wie der entsprechende Kleinbuchstabe behandelt, jedoch wird das Prefab einige Einheiten weiter oben instanziiert, sodass man mittels Klein- und Großbuchstaben Level mit zwei Höhenebenen bauen kann. Die Zahl in der ersten Zeile wird gesondert gelesen und gibt die maximale Leveldauer in Sekunden an.

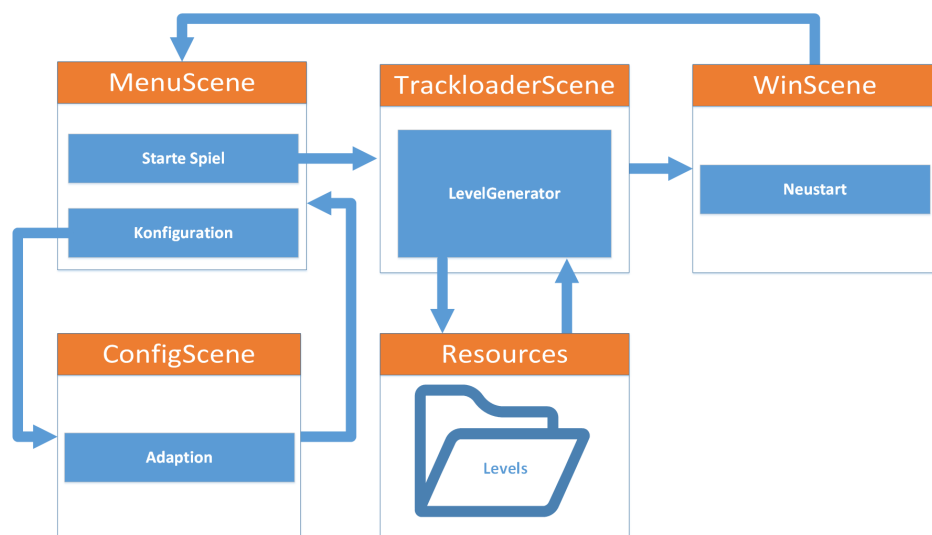
**Abbildung 7.2:** Beispiel für eine Leveldatei

---

## 7.3 Szenen

---

Auch wenn die einzelnen Level nicht als Szenen realisiert sind, so gibt es dennoch mehrere Szenen im Spiel. Diese sind mit den Szenenübergängen in Abbildung 7.3 visualisiert.



**Abbildung 7.3:** Szenen und Übergänge

---

## 8 Anleitung

---

### 8.1 Konfiguration

---

In der Konfiguration kann man verschiedene Einstellungen für das Verhalten des Spiels vornehmen.

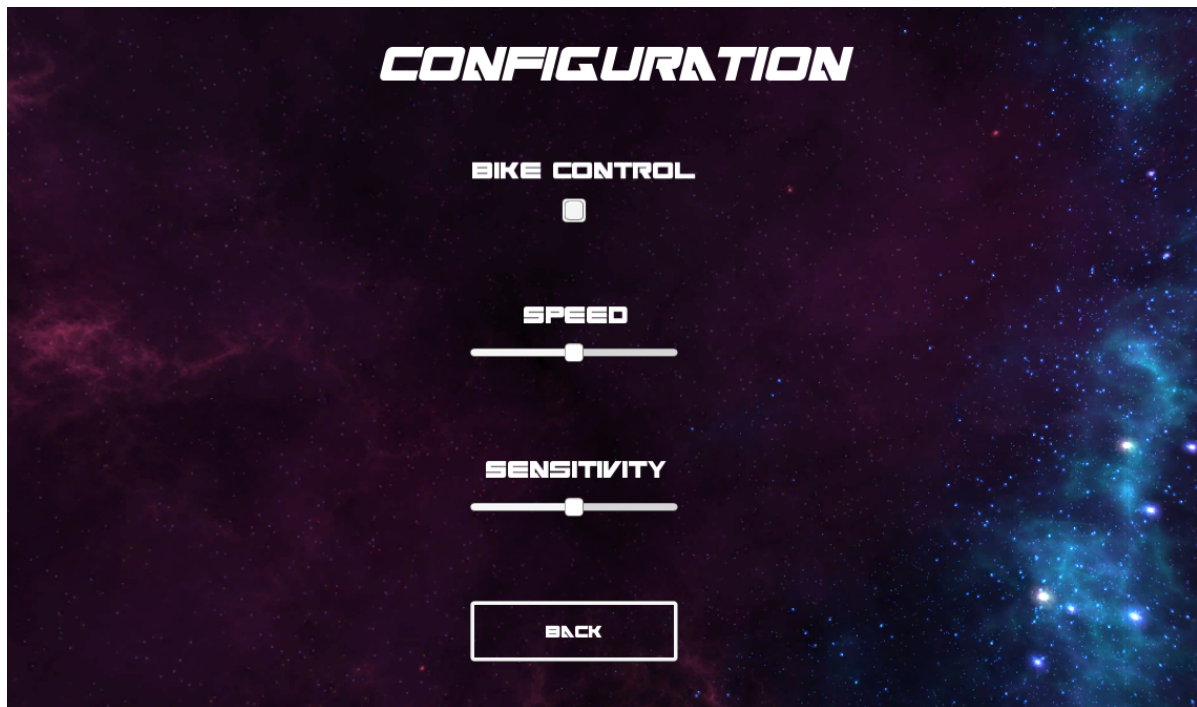


Abbildung 8.1: Konfiguration

- **Bike Controls**  
Ist diese Option ausgewählt, kann das Spiel mit Hilfe des Ergometers gesteuert werden, anderenfalls steuert man das Raumschiff mit den Pfeiltasten auf der Tastatur, sowie der Leertaste.
- **Speed**  
Diese Option bietet die Möglichkeit die Geschwindigkeit des Ergometers anzupassen. Standardmäßig befindet sich der Geschwindigkeitsmodifikator auf einem mittleren Wert. Um die Geschwindigkeit anzupassen kann der Wert entweder verringert oder erhöht werden. Ein höherer Wert des Reglers bedeutet eine schnellere Geschwindigkeit, ein niedriger Wert entsprechend eine langsamere Geschwindigkeit.
- **Sensitivity**  
Diese Option beeinflusst, ähnlich wie die Geschwindigkeit, die Sensitivität des Ergometers. Je nach Einstellung des Reglers reagiert das Fahrrad leichter oder schwerer. Wird der Regler auf einen hohen Wert eingestellt reagiert das Ergometer mehr auf Veränderungen der Neigung. Ein niedriger Wert senkt die Sensitivität entsprechend.
- **Back**  
Durch drücken des Buttons *Back* werden die Einstellungen gespeichert und man gelangt ins Hauptmenü zurück.

---

## 8.2 Steuerung

---

Das Spiel besitzt zwei Möglichkeiten der Steuerung: Tastatur- und Ergometersteuerung. Bei beiden Varianten kann jeweils zusätzlich die Maus genutzt werden.

---

### 8.2.1 Tastatursteuerung

---

Ist in der Konfiguration nicht die Steuerung mittels Ergometer aktiviert gilt folgende Tastenbelegung.

- **Pfeiltaste Hoch:** Erhöht die Geschwindigkeit des Raumschiffes.
- **Pfeiltaste Runter:** Verringert die Geschwindigkeit des Raumschiffes.
- **Pfeiltasten Links/Rechts:** Steuerung des Raumschiffes nach Links oder Rechts.
- **Leertaste:** Lässt das Raumschiff springen.
- **Rechte Maustaste:** Neustart des aktuellen Levels
- **Escape:** Zurück zum Hauptmenü.

---

### 8.2.2 Ergometersteuerung

---

Ist die Ergometersteuerung eingeschaltet gibt es folgende Eingabemöglichkeiten.

- **Trittfrequenz erhöhen:** Erhöht die Geschwindigkeit des Raumschiffs.
- **Trittfrequenz verringern:** Verringert die Geschwindigkeit des Raumschiffes.
- **Neigung des Ergometers nach Links/Rechts:** Steuerung des Raumschiffes nach Links oder Rechts.
- **Linke Maustaste:** Lässt das Raumschiff springen.
- **Rechte Maustaste:** Neustart des aktuellen Levels
- **Escape:** Zurück zum Hauptmenü.

---

## 8.3 Spiel

---

Hat der Benutzer das eigentliche Spiel gestartet findet er diverse Informationen vor, die ihm ein Feedback über den aktuellen Status des Spiels vermitteln sollen. Die einzelnen Objekte und Anzeigen werden nachfolgend erläutert.

### 1. Levelanzeige

Diese Anzeige zeigt die Nummer des aktuellen Levels an.

### 2. Fortschrittsanzeige

Hier wird der Fortschritt im aktuellen Level angezeigt. Je voller der Balken ist, desto weiter fortgeschritten ist der Spieler.

### 3. Spielfigur

Die durch den Spieler gesteuerte Figur. Die Düsenantriebe geben zusätzlich ein Feedback über die aktuelle Geschwindigkeit.

### 4. Kraftstoffanzeige

Hier erkennt der Spieler den verbleibenden Kraftstoff. Verbleibender Kraftstoff symbolisiert die Zeit, die dem Spieler bleibt das Level zu beenden. Je voller der Kreis ist, desto mehr Kraftstoff verbleibt dem Spieler. Ist der Kraftstoff aufgebraucht gilt das Level als verloren und der Spieler muss das Level von Vorne beginnen. Das Raumschiff beginnt Kraftstoff zu verbrauchen, sobald es zum ersten mal beschleunigt wird.



**Abbildung 8.2:** Spielsicht

#### 5. Geschwindigkeitsanzeige

Zeigt die aktuelle Geschwindigkeit an. Befindet sich das Raumschiff im Stillstand ist die Anzeige leer, hat der Spieler die maximale Geschwindigkeit erreicht ist die Anzeige komplett gefüllt.

#### 6. Bahn

Die Bahn ist das eigentliche Level. Der Spieler darf nicht von ihr abkommen, sonst muss er das Level neu beginnen. Am Ende der Bahn befindet sich ein Zielbereich. Erreicht er das Ziel bevor ihm der Treibstoff ausgeht hat er das Level erfolgreich beendet. Die Bahn kann neben der normalen Fahrbahn aus verschiedenen Objekten und Hindernissen bestehen. Beispiele hierfür sind Tunnel, Rampen oder verschiedene Blöcke, die in späteren Leveln nach und nach eingeführt werden.

---

## 9 Zusammenfassung und Ausblick

---

Das Ziel des Projektes war es, ein Spiel zur Förderung des Trainings des Oberkörpers auf einem Fahrradergometer zu konzipieren und umzusetzen. Hierfür wurde zunächst eine umfassende Recherche möglicher Spielideen und der Hardware durchgeführt. Anhand dieser Recherche konnte ein Spielkonzept inklusive Spielbeschreibung erstellt werden. Die entsprechende Umsetzung wurde in weiteren Schritten geplant und dokumentiert, bevor die eigentliche Implementierung begonnen wurde.

Bei der Umsetzung konnten sämtliche geplanten Features realisiert und getestet werden, dennoch bietet das in diesem Praktikum abgeschlossene Projekt einige Möglichkeiten zur Erweiterung der Funktionalität. Es wäre zum Beispiel denkbar dem Spieler zusätzliche Anreize innerhalb der Level zu bieten. So könnte man bestimmte Objekte auf den Bahnen platzieren durch welche der Spieler Vorteile innerhalb des aktuellen, beziehungsweise der nächsten Levels hat oder wodurch der Spieler neue Fähigkeiten und/oder Objekte freischalten kann.

Eine weitere Möglichkeit das Spiel stetig zu erweitern bietet der realisierte Leveleditor. Durch die einfache Integration in das Projekt können zukünftige Entwickler mit sehr geringem Aufwand neue Level-Objekte erstellen und dadurch den Umfang des Spieles schnell erheblich vergrößern. Ebenfalls mit Hilfe des Leveleditors können schnell neue Levels erstellt werden, die anschließend durch die verschiedenen Texturen und Hintergründe nach bestimmten Themen sortiert werden können. Es wäre zum Beispiel denkbar ein Thema mit stark erhöhter Gravitation zu erstellen, welches sich über eine gewisse Anzahl von Levels erstreckt. Dadurch können, ohne dass neue Spielobjekte oder -mechaniken erstellt werden müssten neue Anreize gesetzt werden.

Um den kompetitiven Gedanken der Spieler zu fördern wäre eine Highscoreliste dienlich. Statt dem einzigen Ziel das Spiel erfolgreich zu beenden, könnte man zusätzlich Faktoren wie zum Beispiel benötigte Zeit, benötigte Versuche, verbleibende Energie, oder andere in einer Liste speichern. Hierfür sind jedoch noch weitere Anpassungen notwendig, um die genannten Faktoren erfassen und speichern zu können.

Die wohl weitreichendsten Änderungen könnten jedoch mit der Personalisierung des Spieles an den Spieler erreicht werden. Zwar bietet das Projekt zum jetzigen Stand bereits die Möglichkeit die gemessene Geschwindigkeit und die Sensitivität des Ergometers an den jeweiligen Spieler anzupassen, allerdings werden diese Einstellungen noch nicht gespeichert. Denkbar hierfür wäre eine Art Benutzerverwaltung, in der verschiedene Spieler ein Profil hinterlegen können, welches ihre Einstellungen speichert. Im Zuge dessen könnte ebenfalls der Spielfortschritt gespeichert werden.

Zusammenfassend kann man festhalten, dass im Rahmen dieses Praktikums ein Bewegungsspiel konzipiert und implementiert wurde, dass alle relevanten Punkte der Aufgabenstellung erfüllt und darüber hinaus zusätzlich noch sehr viel Raum für eventuelle weitere Entwicklungsschritte bietet. Durch die realisierten Adaptionenmechanismen eignet sich das Spiel für eine große Bandbreite von Zielgruppen und besitzt dabei das Potenzial spielerisch zur Gesundheitsförderung beitragen. Dieser Aspekt sollte in naher Zukunft durch eine Benutzerstudie evaluiert werden.



---

## Literaturverzeichnis

---

- [1] K. Beck. *Extreme Programming Explained: Embrace Change*. An Alan R. Apt Book Series. Addison-Wesley, 2000. ISBN 9780201616415.
- [2] S. Göbel, S. Hardy, V. Wendel, F. Mehm, and R. Steinmetz. Serious games for health: personalized exergames. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1663–1666, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874316. URL <http://doi.acm.org/10.1145/1873951.1874316>.
- [3] J. Sinclair, P. Hingston, and M. Masek. Considerations for the design of exergames. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 289–295, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-912-8. doi: 10.1145/1321261.1321313. URL <http://doi.acm.org/10.1145/1321261.1321313>.
- [4] J. Sinclair, P. Hingston, M. Masek, and K. Nosaka. Testing an exergame for effectiveness and attractiveness. In *Games Innovations Conference (ICE-GIC), 2010 International IEEE Consumer Electronics Society's*, pages 1 –8, dec. 2010. doi: 10.1109/ICEGIC.2010.5716909.

# **Protokoll vom 20. April 2014**

**Anwesend:** Stefan, Simon, Alex

## **Inhalt**

### **Aufgabenstellung**

- Besprechen der Aufgabenstellung
- Es soll ein Bewegungsspiel entwickelt werden, das mit einem Fahrradergometer gesteuert wird und zusätzlich Bewegung des Oberkörpers fördert.
- Rahmenbedingungen des Spiels sind offen.

### **Nächste Schritte**

- Gedanken über Spielidee machen
- 2-3 Konzepte Skizzieren
- Termin für Hardware Besichtigung und Treffen ausmachen

# **Protokoll, 27. April 2014**

**Anwesend:** Stefan, Simon, Alex

## **Inhalt:**

### **Hardware**

- Stefan zeigt und erklärt uns die Hardware
- SensorTag zum Testen ausgeliehen

### **Spielideen**

- Vorstellung der beiden Spielideen *PacMan* und *Bahnenspiel*
- Machbarkeit der Spielideen in Bezug auf Hardware besprochen

### **Nächste Schritte**

- Auf eines der Konzepte festlegen
- Game Design Document anfertigen
- Mockup anfertigen
- Neuen Termin für Treffen ausmachen

## **Protokoll, 6. Juli 2015**

**Anwesend:** Stefan, Simon, Alex

### **Inhalt:**

#### **Beta Version**

- Vorstellung des aktuellen Fortschritts
- Stefan testet vorläufige Beta Version

#### **Kritik und Anregungen**

- Adaptionenmechanismen wären wünschenswert
- Anfangen Spiel grafisch besser zu gestalten
- Angefangene Teile zusammenfügen
- Am Dokument weiterarbeiten

#### **Nächste Schritte**

- Dokument bearbeiten
- Grafik überarbeiten
- Beta-Version hochladen
- Neuen Termin für Treffen ausmachen

## **Protokoll, 7. September 2015**

**Anwesend:** Stefan, Simon, Alex

### **Inhalt:**

#### **Finale Version**

- Vorstellung der Finalen Version
- Stefan testet fertige Version

#### **Kritik und Anregungen**

- Kleinere Bugs beheben, sonst sehr gut

#### **Nächste Schritte**

- Fertige Version anfertigen
- Dokument schicken
- Folien anfertigen und schicken