

Flappy Penguin - Projekttagebuch

Stefan Cimander

20. Juni 2016

Projektbeginn

Ein Teil der HCI-Vorlesung wurde von unserer Gruppe dazu genutzt, Ideen für das Abschlussprojekt zu generieren und daraus ein konkretes Konzept zu erarbeiten. Folgende Einschränkungen wurden dabei vorgegeben:

- Im Rahmen des Projekts soll ein Spiel entwickelt werden
- Dabei soll der Herzschlags- oder Atemsensor verwendet werden
- Ein Alltagsgegenstand (z.B. Kuscheltier) soll als Steuergerät dienen.
- Das System soll den Q-Learning-Algorithmus integrieren

Wir diskutieren über Möglichkeiten, bekannte Spieleklassiker anzupassen, oder ein eigenes Spielkonzept zu entwerfen. Im Gespräch waren ein vom Puls abhängiges *Tetris*, ein Jump-And-Run Spiel wie *Thomas was alone* sowie die Veränderung von *Breakout*. Unter anderem diskutierten wir allerdings auch darüber, *Flappy Bird* auf irgendeine Weise zu erweitern.

Projektidee

Über den Gedanken, den Spieler in ein Unterwasserszenario zu versetzen, in welchem er sein Ziel nur erreicht, wenn er regelmäßig ein wenig die Luft anhält, entstand die Idee von Flappy Penguin:

- Der Spieler spielt einen Pinguin, der unter der Wasseroberfläche verschiedenen Hindernissen wie zum Beispiel Eisschollen ausweichen muss. Analog zum bekannten Spiel *Flappy Bird* kann der Pinguin lediglich nach oben gesteuert werden und sinkt anschließend von alleine wieder ab. Hinzu kommt, dass dem Spieler nur eine begrenzte Anzahl an **Luftblasen** zur Verfügung steht um mit dem Pinguin unter Wasser zu tauchen. Sind die Luftblasen verbraucht, muss der Pinguin, damit das Spiel nicht verloren geht, an die Wasseroberfläche tauchen und neue Luftblasen einatmen.
- Die Idee der Integration eines **Atemsensors** ist, dass bei jedem Einatmen des Spielers eine Luftblase verbraucht wird, falls sich der Kopf des Pinguins unterhalb der Wasseroberfläche befindet. Ist der Kopf oberhalb der Wasseroberfläche, kann der Spieler durch Einatmen eine Luftblase zurückbekommen.
- Um die Steuerung intuitiver zu gestalten, wird ein **Pinguin-Kuscheltier** mit einem Drucksensor versehen. Drückt der Spieler das Kuscheltier gibt der Sensor das Signal zum Auftauchen
- Eine endgültige Idee für die Einbeziehung des Q-Learning-Algorithmus steht zu diesem Zeitpunkt noch nicht fest, in der Diskussion stehen das Platzieren von zusätzlichen Luftblasen, die der Spieler einsammeln kann, oder das Einfügen eines vom Computer gesteuerten Pinguins.

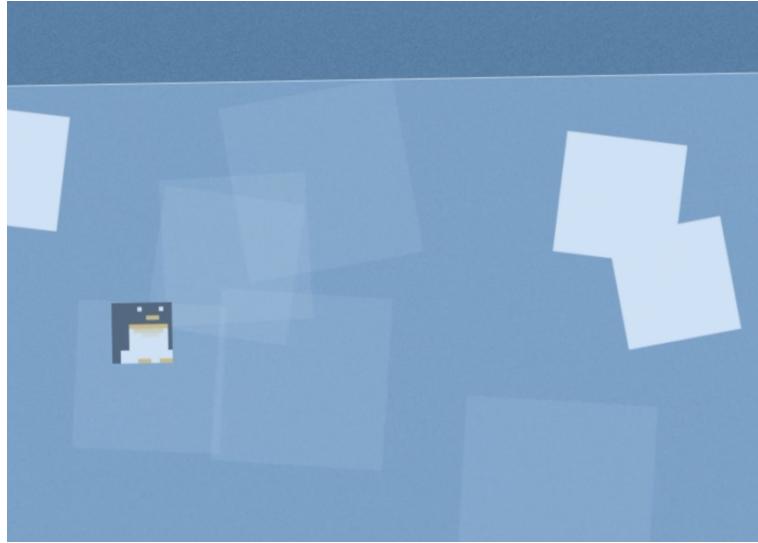


Figure 1: Prototyp der graphischen Spieloberfläche

Aufgabenverteilung

Nachdem wir uns auf ein grobes Konzept für unsere Projektidee geeinigt haben, diskutieren wir das weitere Vorgehen. Gernot bietet an, sich um den Brustgurt mit dem Atemsensor zu kümmern, Thomas schlägt das Framework *create.js* für die Entwicklung der Spieloberfläche vor, woraufhin wir beschließen, dass Thomas und Martin sich in das Framework einarbeiten und ich mich mit der Projektorganisation und der Dokumentation unserer Projektidee befasse, sowie einen kurzen Projektplan erstelle.

23. Juni 2016

Spiel-Prototyp

Zwischen zwei Vorlesungen präsentiert Thomas der Gruppe einen ersten frühen Prototyp der graphischen Oberfläche des Spiels. Auch Springen und Erkennen von Kollisionen mit Eisblöcken sind bereits implementiert, was es uns ermöglicht, erste Spielversuche zu starten. Das Design des Pinguins und die Animation der Umgebung kommt bei der Gruppe sehr gut an. Wir entscheiden, dass diese Projektidee in dem festgelegten Zeitraum umsetzbar ist und dass es deswegen nicht notwendig ist, die Idee zu überarbeiten.

Ein Ausschnitt des Prototyps der graphischen Oberfläche ist in Abbildung 1 zu sehen. Für die Pinguin-Spielfigur existiert auch eine weibliche Variante.

25. Juni 2016

Dokumentation und Projektplan

Zu Hause formuliere ich unserer Projektidee und fasse die wesentlichen Eigenarten des zu entwickelnden Spiels zusammen. Außerdem stelle ich einen Projektplan auf, der die folgenden drei Abschnitte enthält:

Aufgabenverteilung dokumentiert die Verantwortlichkeiten der Teammitglieder,
Projektmeilesteine definieren feste Zeitpunkte zur Fortschrittskontrolle,
Risiken erläutern und bewerten potenzielle Gefährdungen für das Projekt.

Da zum jetzigen Zeitpunkt noch nicht für jede Projektanforderung eine Umsetzung in unser Spiel gefunden ist, werden die Dokumentation und der Projektplan während dem weiteren Projektverlauf durchgehend angepasst. So ist beispielsweise noch die endgültige Entscheidung zu treffen, wie der Q-Learning-Algorithmus in das Spiel integriert werden soll. Auch sind die weiteren Verantwortlichkeiten der Teammitglieder noch nicht endgültig festgelegt.

27. Juni 2016

Abstimmung des Projektplans

Zu Beginn unseres Treffens im Rahmen der Vorlesung stimme ich den von mir erstellten Projektplan mit der Gruppe ab. Wir beschließen, uns spätestens auf eine Möglichkeit festzulegen, den Q-Learning-Algorithmus in unser Spiel zu integrieren. Außerdem wollen wir an der bisherigen Rollenverteilung festhalten und uns falls nötig in unseren Aufgaben gegenseitig unterstützen.

Demonstration des Atemsensors

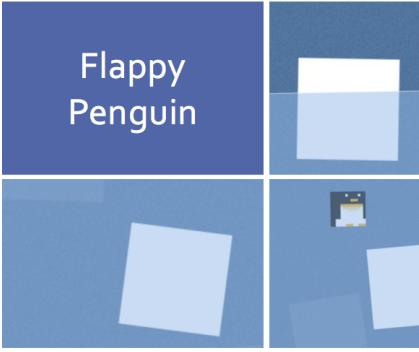
Des Weiteren präsentiert uns Gernot den Atemsensor, den er auf einem Brustgurt befestigt hat. Die anschließende Integration über den Arduino und Node.js in das Spiel sieht nach einer ersten Testphase sehr vielversprechend aus.

Dokumentation der technischen Umsetzung

Den Rest der Vorlesung nutze ich zur Dokumentation der von uns verwendeten Technologien und Frameworks. Dazu gehören ein *Arduino* Mikrocontroller, ein *Node.js* Webserver, das *Create.js* Games Framework sowie ein Dehnungssensor zur Atmungserkennung und ein Biegesensor zur Steuerung mit dem Kuscheltier.

Abbildung 2 zeigt eine Übersicht der fertigen Dokumentation, einschließlich des Titelblatts, der Beschreibung der Projektidee und technischen Umsetzung, sowie des Projektplans.

Flappy Penguin



HC1-Abschlussprojekt
Gernot Brunner - Stefan Cimander - Martin Hartmond - Thomas Weber

Projektkonzept

Der Spieler spielt einen Pinguin, der unter der Wasseroberfläche verschiedenen Hindernissen wie zum Beispiel Eisschollen ausweichen muss. Analog zum bekannten Spiel „[Flappy Bird](#)“ kann der Pinguin lediglich nach oben gesteuert werden und sinkt anschließend von alleine wieder ab. Ziel des Spiels ist es, einen weiteren, vom Computer gesteuerten Pinguin einzuholen. Die Hindernisse bewegen sich dabei von der rechten Seite in den sichtbaren Bereich hinein, bei einem Zusammstoß eines Pinguins mit einem Hindernis wird für einen Moment lang aufgehoben. So ist es möglich, durch geschicktes Ausweichen den Abstand zum anderen Pinguin zu vergrößern, falls dieser einen Eisblock berührt und kurzzeitig gestoppt wird. Gelingt es hingegen dem verfolgten Pinguin den sichtbaren Bereich des Bildschirms zu verlassen, so hat der Spieler verloren.

Als zusätzliches Spielelement stehen dem Spieler nur eine begrenzte Anzahl an Luftblasen zur Verfügung, die mit der Zeit vom Pinguin verbraucht werden. Außerdem kann der Spieler den Vorrat an Luftblasen wieder auffüllen, indem er den Pinguin mit dem Kopf über die Wasseroberfläche manövriert. Sind alle Luftblasen aufgebraucht ist das Spiel ebenfalls verloren.

Erweiterungen

Um dem Spieler eine interessantere Herausforderung zu geben, die ihn auch körperlich etwas miteinbezieht, wird der Spieler mit einem Atemsensor mit dem Computer verbunden. Die Idee ist, dass bei jedem Einatmen des Spielers eine Luftblase verbraucht wird, falls sich der Kopf des Pinguins unterhalb der Wasseroberfläche befindet, ist der Kopf oberhalb der Wasseroberfläche, kann der Spieler durch Einatmen eine Luftblase zurückbekommen. Diese Integration der Atmung des Spielers erweitert das Spiel um eine Vielzahl an Strategien, beispielweise kann der Spieler versuchen, möglichst lange die Luft anzuhalten, um Luftblasen zu sparen. Eine andere Strategie wäre etwa ein häufiges Aufatmen.

Um die Steuerung intuitiver zu gestalten, wird ein Pinguinkuscheltier mit einem Drucksensor versehen. Drückt der Spieler das Kuscheltier gibt der Sensor das Signal zum Auftauchen, das im Prototyp noch mit der Computer tastatur ausgetauscht wird. Diese Art der Steuerung macht zum einen mehr Spaß als das Drücken einer Taste auf der Tastatur und passt zum anderen besser in den Kontext des Spieles. Für Kinder als Zielgruppe des Spieles bietet ein Kuscheltier eine wesentlich greifbarere Möglichkeit der Interaktion und haptischen Wahrnehmung.

Den Spieler zusätzlich zu motivieren ist das Ziel der nächsten Erweiterung. Mithilfe des Q-Learning Algorithmus soll der verfolgte Pinguin mit zunehmendem Spielerlauf lernen, wie er am besten vermeidet, mit Eisblöcken zusammenzustoßen. Dies erhöht nach ausreichendem Lernen den Schwierigkeitsgrad des Spieles und fordert somit den Spieler zunehmend. Das Spiel bietet dem Spieler somit neue Herausforderungen und wird nicht so schnell zu einfach und damit langweilig.

Technische Umsetzung

Grundlegende Implementierung als Browsergame

Das Spiel wird plattformunabhängig als Browsergame umgesetzt. Dazu werden die für die Web-Programmierung üblichen Technologien HTML, CSS und JavaScript verwendet. Das Framework Create.js stellt grundlegende Spielkomponenten zur Verfügung und wird unter anderem für das Laden und Animieren von Sprites verwendet.

Der Prototyp des Spieles wird zunächst über die Computer tastatur gesteuert. Zur Integration des Atemsensors und eventuell eines alternativen Gegenstands zur Spieldsteuerung wird ein [Node.js](#) Mikrocontroller verwendet. Dieser kommuniziert wiederum mit einem Node.js Webservice der das entsprechend erkannte Ein- und Ausatmen an den Game Client im Browser gesendet. Auf diese Weise ist es möglich, dem Spieler eine Luftblase abzuziehen, falls er unter Wasser einatmet und ihm eine Luftblase zurückzugeben, falls er über der Wasseroberfläche einatmet.

Brustgurt mit Atemsensor

Der Atemsensor ist ein vom Lehrstuhl zur Verfügung gestellter Stretch-Sensor und wird auf einen elastischen Brustgurt befestigt. Atmet der Spieler ein, vergibt sich sein Brustgurt und der Brustgurt wird fester gespannt. Der Sensor misst und signalisiert den Kraftaufschwung. Der Spieler krägt während dem Spiel den Brustgurt, der an den [Arduino](#)-angeschlossen ist. Mithilfe des Node.js Webservices werden das entsprechend erkannte Ein- und Ausatmen an den Game Client im Browser gesendet. Auf diese Weise ist es möglich, dem Spieler eine Luftblase abzuziehen, falls er unter Wasser einatmet und ihm eine Luftblase zurückzugeben, falls er über der Wasseroberfläche einatmet.

Alltagsobjekt zur Steuerung des Pinguins

Zur Steuerung der Spielfigur soll ein Kuscheltier, passend in Form eines Pinguins, verwendet werden. Dazu wird in das Kuscheltier ein Flex-Sensor eingebracht, der beim Zusammendrücken seinen Widerstand verändert. Dieser ist ebenfalls mit dem [Arduino](#) Mikrocontroller verbunden.

Reinforcement Learning

Um den verfolgten Pinguin mit fortgeschrittenem Spielerleuf ein intelligenteres Ausweichen zu beibringen, wird der in der Vorlesung behandelte Q-Learning Algorithmus verwendet. Dabei werden dem Algorithmus die aktuelle Position des Pinguins auf einer diskreten Skala und festgelegte Hindernis-Abstandspunkte vor dem Pinguin als Spielstatus übermittelt. Belohnt wird der Algorithmus bei erfolgreichem Ausweichen, bestraft bei einem Zusammstoß mit einem Hindernis.

Projektplan

Aufgabenverteilung

Gernot Brunner	Konstruktion und Integration des Atemsensors, Plüschtier-Controller
Stefan Cimander	Projektplan, Dokumentation, Videodreh und -schnitt
Martin Hartmond	Implementierung, Spielogik
Thomas Weber	Implementierung, graphische Oberfläche, Spiellogik

Projektmilesteine

23. Juni 2016	Demonstration eines ersten Prototyps
27. Juni 2016	Integration des Atemsensors und Anpassung der Spielogik
06. Juli 2016	Drücken eines Plüschtiers als alternative Steuerung
11. Juli 2016	Implementierung des Q-Learning-Algorithmus
14. Juli 2016	Abschlusspräsentation und Abgabe der Dokumentation und Projekttagebücher

Risiken

Zeitrahmen	Die genannten Meilensteine erlauben eine einfache Fortschrittskontrolle. Auf diese Weise sollen Zeitprobleme vermieden und das Projekt zum rechtzeitigen Abschluss gebracht werden.
Atmungserkennung	Es existiert ein nicht unwesentliches Risiko, dass die Erkennung der Atmung des Spielers technisch nicht umgesetzt werden kann, wie es für das Spiel notwendig wäre. Aus diesem Grund wird neben der Spieloberfläche als zweiter wichtiger Schwerpunkt zu Beginn des Projekts mit der Konstruktion des Brustgurts und dem Atemsensor beginnen. Dem Risiko wird somit frühzeitig Begegnung, mögliche Probleme werden durch regelmäßiges manuelles Testen sofort erkannt.
Projekteam	Der Ausfall eines oder mehrerer Teammitglieder, beispielsweise aufgrund von Krankheit, gefährdet prinzipiell den Projekterfolg. Erfahrungen aus vergangenen Projektarbeiten und Präsentationen während dem Semester haben allerdings gezeigt, dass unser Team flexibel auf solche Ausfälle reagieren konnte, weshalb dieses Risiko als nicht zu hoch bewertet wird.

Figure 2: Übersicht unserer Projektdokumentation

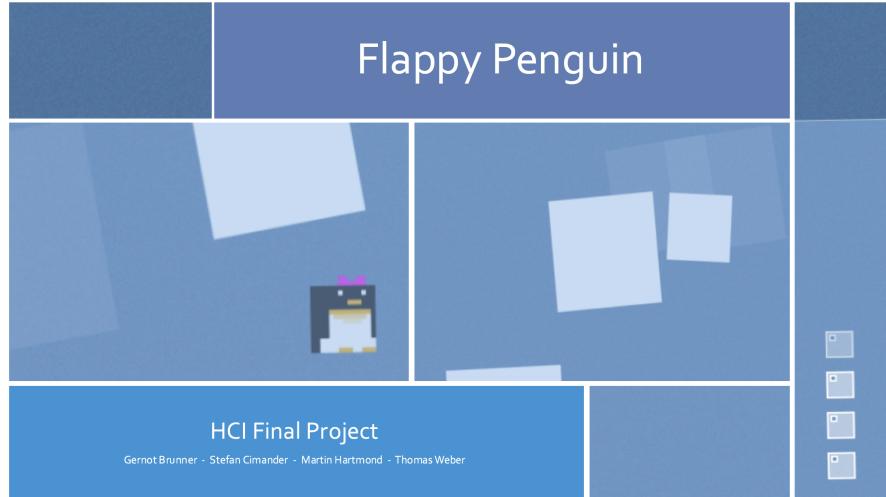


Figure 3: Titelfolie unserer Präsentation

4. Juli 2016

Präsentationsfolien

Das heutige Treffen nutzte ich zur Erstellung erster Präsentationsfolien. Zwar sind es noch zehn Tage bis zur Präsentation, dennoch möchte ich rechtzeitig damit beginnen, um genug Zeit für nachträgliche Änderungen und die Erstellung unseres Videos übrig zu haben. Das gekachelte Design der Folien orientiert sich an der Graphik des Spiels und ist, genau wie die Projektdokumentation farblich an das eisige Unterwasserszenario angelehnt. Nach Absprache des Folien-Layouts mit dem Team erstelle ich die Vorlagen für verschiedene Abschnitte der Präsentation. Die Titelfolie der Präsentation ist in Abbildung 3 zu sehen.

Drehbuch für Projektvideo

Gemeinsam mit Gernot mache ich mir anschließend Gedanken über das zu erstellende Projektvideo. Wir erstellen ein erstes Drehbuch, das die folgenden wichtigsten Kategorien von Szenen umfasst:

- Demonstration der Hardware (Brustgurt & Pinguin-Kuscheltier)
- Bildschirmaufnahmen des Spiels zum Erklären des Spielkonzepts
- Aufnahme des Spielers und seinen (Re-)Aktionen während dem Spielen

Ich biete an, mich um die Aufnahme der Videos sowie um die Bearbeitung und Schnitt zu kümmern.

9. Juli 2016

Vorbereitungen für Videoaufnahmen

Als Vorbereitung für das Drehen der Videos experimentiere ich zu Hause mit der Möglichkeit gleichzeitig einen Ausschnitt des Bildschirms und ein Video mit der Webcam meines Laptops aufzunehmen. Auf diese Weise ist es möglich die Aufnahmen des Spielgeschehens mit denen der Reaktionen des Spielers zu synchronisieren. Zu meiner positiven Überraschung ist das mit dem *QuickTime Player* ohne Probleme möglich.

Außerdem mache ich mich mit dem Bearbeitungsprogramm *iMovie* vertraut und suche nach einem geeigneten Template und passender Musik. Als abschließende Vorbereitung installiere ich *Node.js* und die für unser Spiel notwendigen *npm*-Packages. Somit ist für die Videoaufnahmen am Montag alles bereit.

11. Juli 2016

Abschlussbesprechung

Abschließend vor der Präsentation am Donnerstag erklärt Thomas uns außerdem, wie er den Q-Learning-Algorithmus implementiert hat. Wir haben uns letzte Woche, dafür entschieden die Variante mit dem vom Computer gesteuerten Pinguin auszuwählen. Das angepasste Ziel des Spiels ist daher den anderen Pinguin einzuholen und zu fangen. Auch ist das Spiel nicht mehr verloren, sobald ein Eisblock berührt wird, sondern der kollidierende Pinguin wird lediglich kurz abgebremst. Allerdings ist es nun möglich das Spiel zu verlieren, falls der zu fangende Pinguin aus dem für den Spieler sichtbaren Bereich entkommt.

Zudem ist letzte Woche der Controller in Form eines Pinguin-Kuscheltier fertig geworden. Gernot nähte dafür einen Biegesensor in das Plüschtier ein. Der Sensor wird beim Drücken des Pinguins verbogen, ändert dadurch seinen Widerstand und signalisiert diese Aktion über den *Arduino* an das Spiel.

Videoaufnahmen

Heute nehmen wir außerdem das für die Abschlusspräsentation benötigte Videomaterial auf. Gernot ist damit einverstanden, die Rolle des Spielers zu übernehmen und die Hardware zu präsentieren. Für das Drehen der Spielszenen nehmen wir wie zuvor beschrieben gleichzeitig den Bildschirm mit der graphischen Oberfläche des Spiels und den Spieler auf. Das Erklären des Brustgurts mit dem Atemsensor und des Plüschtier-Controllers nehme ich mit der Kamera eines Handys auf.

Abbildung 4 zeigt den Teil einer Videoaufnahme, bei der Gernot den Brustgurt mit dem Atemsensor vorführt und sich diesen anschließend anlegt.



Figure 4: Gernot beim Vorführen des Atemsensors

12. Juli 2016

Videoschnitt

Ich übernehme das Auswählen, Nachbearbeiten und Zuschneiden der ausgewählten Videosequenzen. Dazu orientiere ich mich an einem *iMovie* Template, das für das einfache Erstellen von Filmtrailern gedacht ist. Zunächst suche ich in unserem aufgenommenen Videomaterial nach passenden Szenen, schneide diese passend zu und füge sie nacheinander in die Vorlagesequenz ein. In der nachfolgenden Abbildung ist ein Screenshot von *iMovie* zu sehen, der einen Teil der fertigen Sequenz mit der grün unterlegten Audiospur zeigt. Spielszenen, die für das Video schön wären aber am Montag nicht von uns aufgenommen wurden, nehme ich nachträglich auf. Am Ende des Schneidens passe ich die Übergänge noch auf die Musik an und exportiere das fertige Video.

Probleme mit dem Controller

Gernot teilt der Gruppe außerdem mit, dass beim Videodreh am Montag der Plüschtier-Controller kaputt gegangen ist. Im innenliegenden Biegesensor ist durch zu festes Drücken des Plüschtiers eine Lötstelle abgebrochen. Zwar versucht er noch dies zu reparieren und den Lehrstuhl um einen Ersatzsensor zu bitten, wir sollten uns allerdings eine Alternative für die zur Präsentation gehörige Demo überlegen. Ich schlage vor, auf die ursprünglich für das Springen verwendete Leertaste zurückzugreifen.



Figure 5: Schneiden des Videos mit iMovie

14. Juli 2016

Controller Hotfix

Um dem Problem des defekten Plüschtier-Controllers zu begegnen, entscheiden wir, den Presenter als Ersatz zu verwenden. Thomas passt dazu kurzfristig den Programmcode an. Auf diese Weise ist es möglich, eine vergleichbarere Spielsteuerung zu vermitteln, als es die Verwendung der Leertaste zulassen würde.

Abschlusspräsentation

Das Ziel ist erreicht, heute ist die Abschlusspräsentation unserer Projekte am Lehrstuhl in Augsburg. Leider vergesse ich beim Erklären des Spielprinzips näher auf die Bedeutung des Atemsensors einzugehen, weswegen die Demo etwas chaotisch abläuft. Dies liegt allerdings auch daran, dass der kurzfristige Ersatz unseres Controllers durch den Presenter nicht ganz zu Ende gedacht war. Die Tastenbelegung auf dem Presenter ist nämlich so angeordnet, dass die Taste zum Springen genau links neben einer Taste liegt, die zum Neuladen der Seite im Browser führt. Dies führt bei der Demo dazu, dass das versehentliche Drücken der falschen Taste das Spiel unterbricht und neu startet.

Ansonsten verläuft die Präsentation aber ungefähr so, wie wir sie uns vorgestellt haben. Rückblickend auf das Projekt können wir meiner Meinung nach sehr zufrieden mit unserer Leistung sein.