



Ein Medieninformatik 3 Projekt von:

Jan Breier (243440)

Thomas Kuschmann (244724)

Ole Rößner (245930)

Hochschule Bremen | Medieninformatik B. Sc.

SS 2009

INHALT

The Spaceballs.....	3
Zielsetzung	3
Zielgruppe	3
Features	3
3D Spielfeldansicht.....	3
Realistische Ballphysik.....	4
XML Level	4
Wiimote Steuerung	4
Level Editor.....	4
Screen- und Spieldesign	4
Die Spielfelder	4
Menü.....	5
Steuerung.....	6
Maus.....	6
Wiimote.....	6
Mac Neigungssensor	6
Tastaturkommandos	7
Projektumsetzung	7
Das Team.....	7
Projektdurchführung.....	7
Produktionszeit	7
Feedback	8

THE SPACEBALLS

In naher Zukunft, Lichtjahre entfernt, könnte ein gewaltiger Krieg den Weltraum für immer verändern! Es ist eine Geschichte über Gut und Böse! Es ist die Geschichte der Spaceballs!

In diesem Krieg bekämpfen sich Planeten gegenseitig um die Vorherrschaft im Weltall. Die bösen Planeten, die Evilballs, versuchen die friedliebenden Planeten, die Spaceballs, zu zerstören! Die Spaceballs ihrerseits versuchen sich, an den Evilballs vorbei und durch knifflige Passagen, in eine einsame und friedliche Galaxie abzusetzen. Steht den Spaceballs bei ihrer gefährlichen Reise beiseite und stellt den Frieden im Weltall wieder her!

The Spaceballs ist eine online Variante der beliebten Holzlabirinthe in denen es darum ging durch Neigung des Spielfeldes eine Metallkugel durch ein Labyrinth zum Ziel zu führen.

Verpackt in eine abenteuerliche Story und mit den Vorteilen digitaler Spiele kombiniert entstand ein interessantes Spiel, welches durch seine ansprechende 3D Welt, eine realistische Physik und die Möglichkeit eigene Level zu kreieren und zu spielen im grauen allerlei der Flash Spiele im Internet hervorstechen versucht.

Das Spiel bietet neben eine intuitiven Maussteuerung auch die Möglichkeit, per Zusatzsoftware, das Spiel mit der Fernbedienung der Nintendo® Wii™¹ zu steuern und bietet dadurch ein ganz neues Spielerleben im Internet!

ZIELSETZUNG

Das Ziel ist es nicht einfach die Holzlabirinthe als Flash Spiel umzusetzen, sondern aus der Grundidee ein interessantes Spiel zu entwickeln. Dabei wird besonders darauf geachtet, wie sich das Spiel von anderen, bereits existierenden Spielen unterscheidet und wo unser Spiel einen Mehrwert bietet.

Das Spiel soll sowohl eine kleine Story, eine besondere Steuerung und die Möglichkeit eigene Levels zu erstellen bekommt. Außerdem wollten wir das Spiel in 3D und mit möglichst realistischer Physik erstellen. Details hierzu unter Features.

ZIELGRUPPE

Die Zielgruppe unseres Spiels sind Jugendliche und Erwachsene, die Spaß an gut designten und kniffligen Spielen haben oder sich gerne selbst kreativ durch den Bau eigener Levels beteiligen.

FEATURES

3D SPIELFELDANSICHT

Durch die externe Flash Library Papervision3D² kann das Spielfeld und die Spielbälle einfach in Flash umgesetzt werden. Die Library bietet eine gute Performance und einfache Anwendung und ist somit unsere erste Wahl.

¹ <http://de.wii.com>

² <http://www.papervision3d.org>

REALISTISCHE BALLPHYSIK

Um die Spielbälle möglichst realistisch wirken zu lassen und uns selbst keine Gedanken über die Kollisionsabfrage machen zu müssen werden sie mit Hilfe der Physikengine Box2DFlash³ ins Rollen gebracht. Die Steuerung ändert dabei nur noch die aktuell auf die Spielbälle wirkende Gravitation, was die Bewegungen sehr realistisch und die Steuerung sehr intuitiv macht. Da Box2DFlash, wie der Name schon vermuten lässt, nur eine 2D Physikengine ist, muss die zweidimensionale Physikwelt auf die 3D Welt gelegt werden um ein korrektes Verhalten der Objekte zu ermöglichen.

Es gibt noch andere Physikengines für Flash, auch welche, die direkt in 3D arbeiten, jedoch wurde uns Box2DFlash auf Grund der guten Performance und leichtem Debugging empfohlen.

XML LEVEL

Um uns die Erstellung der Level zu erleichtern, aber auch um dem Spieler eine möglichst einfach verständliche Sprache an die Hand zu geben um selbst Level zu erstellen, haben wir uns für ein sehr simpel strukturiertes XML Format entschieden, aus dem die Level in die 3D Welt generiert werden.

WIIMOTE STEUERUNG

Durch Implementierung der WiiFlash⁴ Library und des WiiFlash Servers ist es möglich aus Flash heraus die Neigungssensoren und Knöpfe der Wiimote abzufragen. Dies bietet uns die Möglichkeit eine noch intuitivere Steuerung einzubauen, durch die das Spiel eine völlig neue Spielerfahrung bietet.

LEVEL EDITOR

Durch den eingebauten Level Editor hat der Spieler die Möglichkeit einfach und schnell eigene Level zusammen zu klicken, abzuspeichern oder direkt zu spielen. Er hat ebenfalls die Möglichkeit zuvor gespeicherte Level von seinem Computer zu laden und erneut zu spielen.

SCREEN- UND SPIELDESIGN

DIE SPIELFELDER



Normal

Normales Spielfeld

Verhalten: Kugel rollt einfach drauf herum



Wall

Wand

Verhalten: Kugel kommt nicht dran vorbei



End

Ziel

Verhalten: Level geschafft!



Gravitation

Gravitation out

Verhalten: Kugel wird wenn sie dem Feld zu nahe kommt leicht nach außen abgestoßen...

³ <http://box2dfash.sourceforge.net>

⁴ <http://wiiflash bytearray.org>



Empty

Leeres Feld

Verhalten: Ist nicht sichtbar! Gibt es eigentlich nur weil Leute im Level-editor Felder leer lassen können, wenn sie wollen, ohne Mauern herum zu bauen. Kugel soll, wenn es auf ein leeres Feld kommt, das selbe Verhalten zeigen wie beim Schwarzen Loch



Blackhole

Schwarzes Loch

Verhalten: Kugel wird kleiner bis sie verschwindet. Level ist gescheitert!



Hole

Loch

Verhalten: Gibt es nur wenn mehr als eine Kugel im Spiel. Wenn eine Kugel drüber geht wird sie wie beim schwarzen Loch kleiner bis sie verschwindet. Das Bild des Loches soll sich dann in das geschlossene Loch ändern, nächste Kugeln können nun drüber rollen ohne dass etwas passiert.



Wormhole

Wormloch

Verhalten: Kugel wird wenn sie auf das WormholeIn Feld kommt klein und verschwindet, und wird im fast selben Moment auf dem WormholeOut Feld von klein zu groß und rollt dort weiter...



Wormhole



Gravitation

Gravitation in

Verhalten: Kugel wird wenn sie dem Feld zu nahe kommt leicht an das Feld herangezogen...



Spaceball

Der Spaceball

Verhalten: Kugel eben!
Es kann mehrere geben.



Enemy

Evilball

Verhalten: Kugel eben!
Es kann mehrere geben.
Wenn die Kugel von einem Spaceball getroffen wird sollen beide Kugeln explodieren.

MENÜ



Abbildung 1: Hauptmenü

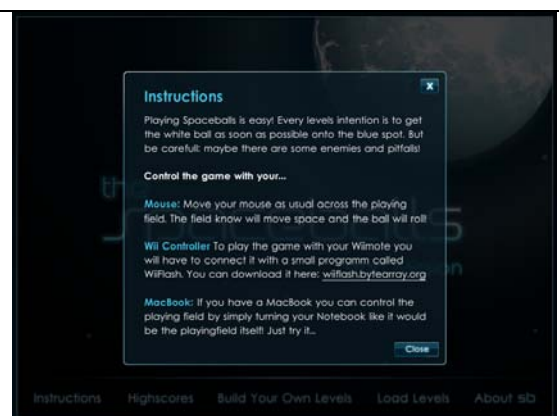


Abbildung 2: Instructions



Abbildung 3: Highscores



Abbildung 4: About Spaceballs



Abbildung 5: Levelbuilder

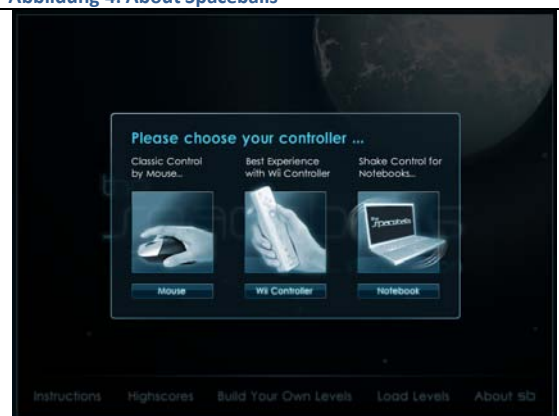


Abbildung 6: Controllerauswahl

STEUERUNG

MAUS

Mit der Maus kann die Neigung des Spielfeldes geändert werden, je nachdem, wie weit sich die Maus vom Mittelpunkt der Spielfläche befindet.

WIIMOTE

Mit der Wiimote kann man das Spielfeld analog zur Wiimote bewegen, wenn man diese so hält, dass sie auf den Monitor zeigt.

Neigung nach vorne und hinten sowie Rollen der Wiimote nach links und rechts, neigt das Spielfeld dementsprechend.

Des Weiteren sind folgende Tasten der Wiimote belegt:

- A Pause
- „Minus“ Rumble Funktion an/aus
- „Home“ Zum Menü zurück

MAC NEIGUNGSSENSOR

Die Steuerung über den Neigungssensor funktioniert analog zur Wiimote Steuerung, dass man hier jedoch das ganze Macbook bewegt.

TASTATURKOMMANDOS

Während des Spiels sind folgende Tastaturkommandos dauerhaft und egal bei welchem Controller aktiv:

- F1 Instructions Dialog anzeigen
- Esc Zum Menü zurück
- P Pause
- S Sound on/off

PROJEKTUMSETZUNG

DAS TEAM

- **Jan Breier**
ActionScript 3, Artwork, MXML
Webseite: www.areablue.de
- **Thomas Kuschmann**
ActionScript3, XML, Leveldesign
- **Ole Rößner**
ActionScript3, Papervision3D, Box2D
Webseite: www.basster.de

Besonderer Dank geht an **Jan Rigerl** von DEVM-Games⁵, der uns Einblick in den Quellcode seiner Version von „Tilt“⁶ gewährte und uns somit einen guten Start für unser Projekt ermöglichte.

PROJEKTDURCHFÜHRUNG

Nach der gemeinsamen Ideenfindung wurde das Projekt im Team in klare Aufgaben aufgeteilt. Artwork und Leveldesign wurden bereits vor Beginn der Programmierung durchgeführt, so dass klare Anforderungen an die Software gestellt werden konnten.

Daraufhin wurden parallel die Menüs und der Levelbuilder in MXML Flex sowie der 3D Spielbereich in AS3 entwickelt. Nachdem beide Teilprojekte zu etwa 80% fertig waren wurde der 3D Spielbereich in die vorhandene MXML Umgebung implementiert und von nun an gemeinsam weiterentwickelt.

Um an einem gemeinsamen Datenbestand zu arbeiten wurde das Quellcodehosting Programm von Google⁷ wahrgenommen, welches einen kostenlosen SVN Server, sowie ein simples Bugtracking System zur Verfügung stellt.

Die Programmierarbeiten wurden mit FlexBuilder 3 erstellt. Adobe stellt für Studenten eine kostenlose Educational Lizenz zur Verfügung!

PRODUKTIONSZEIT

- Artwork ca. 40 Std.
- MXML Gerüst ca. 40 Std.
- Levelbuilder ca. 30 Std.
- 3D ca. 30 Std.

⁵ www.devm-games.com

⁶ www.devm-games.com/#tilt

⁷ <http://code.google.com/p/camlabyrinth>

- Physik ca. 35 Std.
- Spiellogik ca. 35 Std.
- Leveldesign ca. 30 Std.
- Dokumentation ca. 10 Std.

Alle Zeiten grob geschätzt ($\pm 10\%$) und inklusive Einarbeitung in den jeweiligen Themenbereich

FEEDBACK

Das Projekt wurde erfolgreich abgeschlossen. Das Feature der Neigungssteuerung für Macs konnte auf Grund von Inkompatibilitäten mit Web swfs vorerst nicht realisiert werden. Alle weiteren Features wurden, mit teilweise kleinen Einschränkungen in Funktion und Design, umgesetzt.

Die Zusammenarbeit über Google Code Hosting war, wie bereits in früheren Projekten erprobt, erneut sehr erfolgreich und hat die Projektfertigstellung enorm unterstützt.

Die Arbeit mit Papervision3D und vor allem Box2D war oft auf Grund von unvollständigen oder gar fehlenden Dokumentationen eine wahre Strapaze.

ANHANG

Angefügt sind ein Klassendiagramm über alle Packages, sowie ein Programmablaufplan.

