

Anhang

Inhaltsverzeichnis

1.PROTOKOLLE	46
1.1 PROTOKOLL 01 – 30.03.2012	46
1.2 PROTOKOLL 02 – 03.04.2012	47
1.3 PROTOKOLL 03 – 10.04.2012	48
1.4 PROTOKOLL 04 – 12.04.2012	49
1.5 PROTOKOLL 05 – 17.04.2012	50
1.6 PROTOKOLL 06 – 20.04.2012	51
1.7 PROTOKOLL 07 – 27.04.2012	52
1.8 PROTOKOLL 08 – 02.05.2012	53
1.9 PROTOKOLL 10 – 11.05.2012	54
1.10 PROTOKOLL 11 – 22.05.2012	55
1.11 PROTOKOLL 12 – 06.06.2012	56
2. LASTENHEFT	57
3. PFLICHTENHEFT	59
ZIELBESTIMMUNG	59
11. SPEZIELLE ANFORDERUNGEN AN DIE ENTWICKLUNGSUMGEBUNG	62
11.1 SOFTWARE	62
11.2 HARDWARE	62
11.3 ORGWARE	62
11.4 ENTWICKLUNGSSCHNITTSTELLEN	62

1. Protokolle

1.1 Protokoll 01 – 30.03.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	Freitag, 30.03.2012 / 13:00 Uhr
Ort:	Postbankgebäude, Raum 4020
Verfasser:	Sebastian Adam
Tagesordnung:	keine spezielle, erstes Vorgespräch mit Prof. Hopf und Prof. Röttger
Protokoll:	<p>I. Aufgabenfelder:</p> <ul style="list-style-type: none">- Sounds capturen (evtl. auch einfache Umwandlung vorhandener SPL-Dateien von Prof. Hopf möglich – wird von Sergej bis zum nächsten Treffen überprüft)- Grafiken capturen- Online-Codeverwaltung (svn, github?)- Namenskonventionen für Ordnerstruktur, Aktionen, etc.- Festlegen der Sprite-Engine: Diese sollte möglichst performant sein, es wurde angeraten erste Testbenches in der originalen Spielfeldgröße durchzuführen → wenn ruckelfrei dann Engine möglicherweise gut geeignet! (wird von Pavlina bis zum nächsten Treffen durchgeführt mit Akihabara und Crafty – alle anderen Teammitglieder sollten sich bis Dienstag noch einmal mit den möglichen Engines befassen, Dokumentationen durchlesen und darauf achten welche Engines besonders gut dokumentiert sind und evtl. auf Aktualität der Engines achten → https://github.com/bebraw/jswiki/wiki/Game-Engines Auch darauf achten, dass die Engine als „isometrisch“ oder 2,5D Engine bezeichnet wird, das bedeutet, dass hier verschiedene Ebenen übereinander gelegt werden können)- Timing der einzelnen Aktionen (Bewegungen, Bombenexplosionen, Rundendauer) erforschen, stoppen bzw. reverse engineerieren <p>Nächstes Treffen mit Prof. Hopf und Prof. Röttger: Dienstag, 03.04.2012 im Raum E205</p> <p>Wer bereits in der Früh Zeit hat sollte gegen 10 Uhr in der FH sein um Vorbereitungen zu treffen (Volere Template, Lastenheft, etc.) – Pavlina ist bereits entschuldigt.</p> <p>Nach dem offiziellen Treffen würde ich gerne noch eine interne Besprechung anschließen, wo einzelne Aufgaben verteilt werden und ein wöchentlicher Besprechungstermin ausgemacht werden sollte weil es ja dann „richtig“ mit dem Projekt losgeht.</p> <p>Die kommenden Termine wurden in den gemeinsamen Google Kalender eingetragen, also in Zukunft darauf achten was ansteht.</p> <p>In den Dropbox Ordner wurden alle Teammitglieder eingeladen, bitte beitreten wenn noch nicht geschehen.</p> <p>Ansonsten sollten wir das schon irgendwie geschaukelt bekommen ;-) Bis zum nächsten Treffen am Dienstag!</p>

1.2 Protokoll 02 – 03.04.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	Dienstag, 03.04.2012 / 13:00 Uhr
Ort:	E205, Wassertorstr. 10
Verfasser:	Sebastian Adam
Tagesordnung:	keine spezielle, Besprechung der Spielelogik und Organisatorisches
Protokoll:	<p>I. SHOP – Goodies:</p> <p>Extra Bomb – man erhält 1 extra Bombe in der Spielrunde</p> <p>Power Up – der Bombenradius erhöht sich um ein Feld</p> <p>Superman – man kann Mauernblöcke verschieben</p> <p>Ghost – man ist nur 1x1px groß und kann durch Wände gehen</p> <p>Controler – man kann die Bombe statt der Spielfigur steuern, bei Loslassen der Taste Explosion</p> <p>Protection – man ist für eine Aktion geschützt (Spielfigur wird weiß)</p> <p>Timebomb – Bombe ist zeitgesteuert, bei Loslassen der Taste Explosion</p> <p>Speed Up – man wird schneller (max. 4x möglich)</p> <p>STOP Schild – alle anderen Spieler bleiben auf der Stelle stehen und rotieren und können sich für kurze Zeit nicht mehr bewegen, man selbst ist davon nicht betroffen</p> <p>Fragezeichen – hier kann sich ein STOP-Schild, Protection, Ghost oder „die Krankheit“ dahinter verbergen, bei Krankheit explodieren eigene Bomben bereits nach 0,5 statt 2s</p> <p>2. Developmentblog:</p> <p>Hier wurde von Prof. Hopf empfohlen die Einträge auf Englisch zu verfassen, da die AMIGA Community hauptsächlich im englischsprachigen Raum vorhanden ist. Falls das Projekt Wellen über Deutschland hinaus schlagen sollte.</p> <p>3. Nächster Termin</p> <p>Es wird ein nächster gemeinsamer Termin mit Prof. Hopf und Prof. Röttger nach den Osterfeiertagen angestrebt. Hierfür werden einige Terminvorschläge an die beiden Prof. geschickt.</p> <p>4. Codeverwaltung</p> <p>Also Codeverwaltung wird nach Vorschlag von Prof. Hopf wohl github zum Einsatz kommen. Nach Abschluss des Projekts wird der gesamte Code dann auf einen GIT-Server der Fakultät EFI gepusht.</p> <p>5. Aufgaben</p> <p>Bis zum nächsten mal befasst sich jeder mit der ausgewählten Crafty-Engine und arbeitet das Tutorial durch um Verständnis dafür zu bekommen wie die Engine im Zusammenspiel mit JavaScript funktioniert. Pavlina versucht sich schon einmal an einer ersten „Testwelt“ und Sergej versucht weiterhin die Sounds zu encodieren. Micha wird ein Logo erstellen und Sebastian das Volere Template durcharbeiten und eine erste Fassung des Lastenhefts erstellen, sowie den Developmentblog aufsetzen.</p>

1.3 Protokoll 03 – 10.04.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	10.04.2012 / 16:30 Uhr
Ort:	Lerncontainer
Verfasser:	Sergej Bjakow
<p>Tagesordnung:</p> <ol style="list-style-type: none"> 1. Organisatorisches 2. Planung weiteres Vorgehen 3. Festlegung der Meilensteine 4. Aufgabenverteilung <p>1. Alle Teammitglieder haben die restlichen offenen Fragen des Lastenheftes nach ihrem Ermessen beantwortet und dem Projektleiter Sebastian für die Zusammenstellung ausgehändigt. In git hub hochgeladene Codeblöcke sollten per Kommentar mit Namen versehen werden. Außerdem wurde für ein generelles regelmäßiges Teammeeting festgehalten: Dienstags 15 Uhr und Freitags ab 13 Uhr Nächster Vereinbarter Termin mit Herrn Hopf Donnerstag 11:30</p> <p>2. Einigung auf „Google Chrome“ als Testbrowser. Diskussion über die Anzahl an Spielern für das Spiel. Zunächst beschränken wir uns auf 2 Spieler (wegen Platzmangel an der Tastatur) und je nach Zeit werden Controller Plugins für bis zu 5 Spieler implementiert.</p> <p>3.</p> <p>Meilenstein 1 (->24.April 2012): Spielwelt mit allen Originalgrafiken sowie zwei steuerbaren Charaktere (mit allen für die Lauf Animation nötigen Sprites des Originals) erstellen Größe der Wandblöcke: 32x32px Größe der Spielfiguren: 32x44px</p> <p>Meilenstein 2(-> 8.Mai 2012): Realisierung von Eventfunktionen, wie das legen von Bomben, deren Explosion (samt Animation) und zerstörbaren bzw. bröckelnden Steinblöcken.</p> <p>Meilenstein 3(-> 30.Juni 2012): Realisierung von erscheinenden Items mit all ihren Funktionen und der originalen Auftrittswahrscheinlichkeit aus dem Spiel (□ erfordert intensive Analyse des Originals)</p> <p>Meilenstein 4 (Optional je nach Fortschritt) Menü- und Shop Screen mit logischem Bezug zum Spiel</p> <p>4.</p> <p>Sergej: Extrahiert sämtliche Graphiken/Sprites aus dem Spiel in der festgelegten Größe (siehe MSI) und ordnet sie in einer png. Datei systematisch an Pavlina: Realisierung des Übergangs von Menü- Screen zum Spiel und anschließend zum Shop Micha: Erstellt die Map mit 2 steuerbaren Figuren und den Original Sprites Sebastian: Organisation, Fertigstellung des Lastenheftes, Pflege der Webpräsenz Max: tba</p>	

1.4 Protokoll 04 – 12.04.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	12.04.2012 / 11:30 Uhr
Ort:	E205 / Wassertorstr. 10
Verfasser:	Sebastian Adam
<p>1. Darstellung der Mauernblöcke Die Darstellung der Mauernblöcke sollte auf Hinweis von Herrn Hopf über ein Array erfolgen, welches am Anfang befüllt wird. Die Kollisionsabfrage sollte ebenfalls über dieses Array erfolgen und nicht über die in der Sprite-Engine bereits enthaltene Kollisionsabfrage. So soll also bei jeder Bewegung die der Spieler macht das Array abgefragt werden ob das Feld in das man sich bewegen möchte frei oder mit einem Mauernblock belegt ist.</p> <p>2. Screen Capturing Das Spiel sollte mit 50fps gecaptured werden. Wenn das Capturen nicht schnell genug geht (weil Rechner zu schwach?!) dann sollte nur jeder 3. Frame aufgenommen werden. Im E-UAE sollte bereits eine Screen-Capture Funktion eingebaut sein. Für das Spiel benötigen wir folgende Sprites: Bewegung nach links/rechts/oben/unten sowie die Sterbeanimation.</p> <p>3. Garbage Collection kann Spielperformance beeinträchtigen Herr Hopf merkte noch an, dass im Spielverlauf der (unkontrollierbare) Garbage Collector unter JavaScript aufgerufen werden könnte, was einen Ruckler verursachen kann. Mit dieser Tatsache müssen wir leben da wir wie schon erwähnt keinen Einfluss darauf haben wann dieser aufgerufen wird.</p> <p>4. Konfigurationsmöglichkeiten Die Konfigurationsmöglichkeiten könnten am Ende über ein JSON-Objekt eingebunden werden. Desweiteren könnte man auch default Werte (einmal vom Benutzer eingestellt) als Cookie im Browser speichern. Dies wird jedoch erst ganz am Schluss des Projekts behandelt.</p> <p>5. Spielsteuerung / Gameplay Ein ganz wichtiges Thema ist die Steuerung der Spielfigur, damit der Spielfluss erhalten bleibt. Die Spielfigur ist immer exakt einem Feld zugeordnet. Zuordnung erfolgt evtl. über exakt einen Pixel. (Aktuelle Position + halbes Feld) / ganzes Feld = zugeordnetes Feld. Die Spielfigur bewegt sich zudem nach einem Blockwechsel automatisch auf eine Mittelachse.</p> <p>6. Startposition der Spielfiguren Die Startpositionen der Spielfiguren sind von vornherein festgelegt. In den 4 Ecken sind um die Spielfigur herum 5 Felder ohne Mauernblöcke in L-Form. In der Mitte sind um die Spielfigur 3 freie Felder ohne Mauernblöcke, ebenfalls in L-Form.</p> <p>7. Nächstes Treffen</p> <p>intern: Dienstag, 17.04.2012, 15:00 Uhr vor der Mensateria mit Prof. Hopf / Prof. Röttger: Freitag, 20.04.2012, 10:00 Uhr... Ort tba</p>	

1.5 Protokoll 05 – 17.04.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	17.04.2012 / 15:00 Uhr
Ort:	Lerncontainer
Verfasser:	Maximilian Seyfert
Tagesordnung:	<ol style="list-style-type: none"> 1. Verifizierung der Anforderungsdefinition 2. Aktualisieren der Projektroadmap 3. Überlegungen zum Kollisionsmodell 4. Einbindung der Json Objekte 5. Aktualisierte Aufgabenverteilung
<p>Protokoll zu den einzelnen Tagesordnungspunkten:</p> <p>1. Verifizierung der Anforderungsdefinition In gemeinsamer Arbeit wurde die Anforderungsdefinition von allen Teammitgliedern nochmalig durchgegangen. Nach kleinen Änderungen gaben wir sie frei.</p> <p>2. Aktualisieren der Projektroadmap Besprochen wurde die Aktualisierung der Projektroadmap, weil die Erstellung der Spielwelt entgegen unseren Erwartungen mehr Zeit in Anspruch nimmt. Dies ist auf die Komplexität des Kollisionsmodells von Master Blaster zurück zu führen.</p> <p>3. Überlegungen zum Kollisionsmodell Pavlina, Michael und ich überlegten uns verschiedene Ansätze, das Kollisionsmodell von Crafty zu überschreiben, um dem von Master Blaster am nächsten zu kommen. Ein Ansatz wäre beispielsweise die Physik von der Darstellung zu trennen d.h. die Sprites auf einfache geometrische Objekte zu Mappen und nur mit ihnen die Kollision zu berechnen.</p> <p>4. Einbindung der Json Objekte Json Objekte werden für die UI-Konfiguration benutzt, damit die UI veränderbar bleibt. So ist es nicht nötig das alles Hardgecoded wird, sondern man packt es in eine Json Datei und kann diese so verwenden wie Strukturen in C.</p> <p>5. Aktualisierte Aufgabenverteilung</p> <p>Maximilian Seyfert - Kollisionsmodell Michael Koa - Kollisionsmodell Pavlina Pavlova - Json Einbindung Sergej Biakov - Re-Engineering der Sprites Sebastian Adam - Funktionalität der Bomben</p>	

1.6 Protokoll 06 – 20.04.2012

	Sitzungsprotokoll Projekt MasterBlaster (mit Prof. Dr. Hopf)
Datum / Uhrzeit:	20.04.2012 / 10:00 Uhr
Ort:	Postgebäude
Verfasser:	Michael Kao
Tagesordnung:	<ol style="list-style-type: none">1. Besprechung des Projektplans2. Besprechung Git3. Diverses
Protokoll zu den einzelnen Tagesordnungspunkten:	
<ol style="list-style-type: none">1. Teilschritte für bestimmte Tätigkeiten sollen detaillierter im Projektplan ausgearbeitet werden (Realisierung der Spieledetails)2. Git: Wir haben ausgiebig über das Versionierungstool git gesprochen. Prof. Dr. Hopf hat uns eine Einführung in git und dessen Kommandos, sowie einige hilfreiche Tipps gegeben. Auf github.com ist unser gesamter Code mit allen wichtigen Files, sowie die Dokumentation zu versioniert finden.3. Wir haben uns dazu entschlossen die GPL Lizenz zu verwenden. Die Software soll für 2 Spieler ausgelegt sein, jedoch eine Implementierung von mehr Spieler möglich sein. Die Dokumentation soll auf unserer Webseite veröffentlicht werden. Wir schreiben nun eine eigenen Bewegungssteuerung sowie Kollisionsabfrage und verzichten somit auf die Nachteile der Crafty Komponenten.	

1.7 Protokoll 07 – 27.04.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	27.04.2012 / 14:00 Uhr
Ort:	Lerncontainer Keßlerplatz
Verfasser:	Sebastian Adam
Tagesordnung:	<ol style="list-style-type: none">1. Aufgabenverteilung bisher2. Neue Aufgabenverteilung3. Weitere Vorgehensweise4. Anmerkungen
Protokoll zu den einzelnen Tagesordnungspunkten:	
I. Aufgabenverteilung bisher <ul style="list-style-type: none">• Micha, Max und Sebastian: Grid-System, Bombenfunktion• Pavlina: befasste sich mit JSON Objekten und AJAX• Sergej: Analyse der Spielgeschwindigkeit im Originalspiel	
2. Neue Aufgabenverteilung <ul style="list-style-type: none">• Pavlina: befasst sich weiter mit dem Game-Loop und prüft wie oft dieser ausgeführt wird• Sergej: beschäftigt sich mit der Einbindung der Sounds, analysiert die Spieldetails und prüft inwieweit ein Loop der Hintergrundmusik mit steigender Sampling-Rate möglich ist• Sebastian: beginnt mit der Implementierung der Bombenfunktion, verfasst das Pflichtenheft wie von Prof. Hopf und Prof. Röttger gefordert → bis Anfang nächste Woche finale Version!!• Max, Micha: Implementierung der Bombenfunktion	
3. Weitere Vorgehensweise <p>Am Ende des Projekts sollen zwei spielbare Charaktere auf dem Spielfeld zur Verfügung stehen. Hierfür wird es nötig sein ein Elternobjekt zu erstellen als Basis für alle Spielfiguren. Die Eigenschaften der einzelnen Spielfiguren sollen vom Elternobjekt geerbt werden. Alternativ könnte man auch über einen „Clone“ nachdenken.</p>	
4. Anmerkungen <p>Das nächste Treffen mit Prof. Hopf entfällt! Optional (falls notwendig) Treffen in der Sprechstunde am Donnerstag um 11:30 Uhr</p>	

1.8 Protokoll 08 – 02.05.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	02.05.2012 / 14:00 Uhr
Ort:	Lerncontainer Keßlerplatz
Verfasser:	Sebastian Adam
Tagesordnung:	<ol style="list-style-type: none">1. Aktueller Status2. Zeitplan, Präsentation & Pflichtenheft3. Szenen4. Anmerkungen
Protokoll zu den einzelnen Tagesordnungspunkten:	
<ol style="list-style-type: none">I. Aktueller Status<ul style="list-style-type: none">• Bombenfunktion: Eine Bombe kann mittlerweile gelegt werden, jedoch nur der Sprite, die Bombe selbst ist bisher noch ohne Funktion und verschwindet nach einer gewissen Zeit wieder• Steuerung: Die Steuerung bereitet aktuell noch Probleme, weil die Spielfigur stehen bleibt sobald man eine Bombe legt → Lösungsansatz: Speichern des letzten Tastendrucks der Bewegungsrichtung mittels Flags2. Zeitplan, Präsentation & Pflichtenheft<ul style="list-style-type: none">• Zeitplan: Verfeinerung der einzelnen Teilschritte in der Entwicklungsphase• Präsentation: Sebastian verfasst ein grobes Layout der Präsentation und jeder steuert seinen eigenen Beitrag zur Präsentation bei• Pflichtenheft: wird diese Woche in seiner finalen Version verfasst und Herrn Hopf sowie Herrn Röttger zukommen lassen3. Szenen<ul style="list-style-type: none">• Pavlina wird Versuchen das Wechseln der Szenen auf Tastendruck zu implementieren4. Anmerkungen<ul style="list-style-type: none">• Das nächste Treffen findet am 08.05. um 15 Uhr im Lerncontainer statt	

1.9 Protokoll 10 – 11.05.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	11.05.2012 / 14:00 Uhr
Ort:	Lerncontainer Keßlerplatz
Verfasser:	Sebastian Adam
Entschuldigt:	Maximilian Seyfert (Arbeit) Pavlina Pavlova (Lebensmittelvergiftung)
Tagesordnung:	1. Aktueller Status 2. Weitere Aufgabenverteilung
Protokoll zu den einzelnen Tagesordnungspunkten:	
I. Aktueller Status	
<ul style="list-style-type: none">• Implementierte Goodies:<ul style="list-style-type: none">○ SpeedUp +I○ Bomb +I○ PowerUp +I○ TimeBomb○ Disease○ Death Skull	
2. Weitere Aufgabenverteilung	
<ul style="list-style-type: none">• Pavlina: kümmert sich weiterhin um den Szenen Loop und die Konfiguration• Sergej: Einbindung der Hintergrundmusik• Micha: Optimierung der Feuer-Animation einer Bombenexplosion• Sebastian: betreibt Code Analyse	

1.10 Protokoll 11 – 22.05.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	22.05.2012 / 15:00 Uhr
Ort:	Lerncontainer
Verfasser:	Sergej Bjakow

Besprochene Themen:

6. Dynamische Menü Navigation
 7. Problemlösung für Soundwiedergabe
 8. Arbeitsverteilung
-
1. Bisher können Inhaltsveränderungen im Menübildschirm nur über Szenenwechsel erfolgen, was mit JS Canvas geändert werden soll. Um einen Loop aller Szenen, also vom Startbildschirm, über das Menü bis zum Spiel und wieder zurück zu realisieren wird die Crafty Engine aller Voraussicht nach nicht ausreichen und dafür eben JQuery genutzt werden müssen.
 2. Obwohl zu Beginn in kleinen Testprojekten erfolgreich angewandt, funktionierte im aktuellen Spielequellcode die „Crafty.audio.play“ unverständlicherweise nicht; tatsächlich blockt diese Codezeile den Spielverlauf an der Stelle an der sie aufgerufen wird. Die Lösung des Problems war es die Audio Funktionen von Crafty unberührt zu lassen und stattdessen mit reiner Java Script Methodik die Soundeffekte aufzurufen. Nun explodieren Bomben mit dem entsprechenden Explosions Geräusch.
 3. Max wird sämtliche Sounds in die Spielelogik implementieren und zusammen mit Sergej die Alarmfunktion ausbauen inklusive dem „Shrinking Event“. Bisher steht ein Funktionierender „Timeout“, der jedoch noch angepasst werden muss.

1.11 Protokoll 12 – 06.06.2012

	Sitzungsprotokoll Projekt MasterBlaster
Datum / Uhrzeit:	06.06.2012 / 14:00 Uhr
Ort:	Lernbunker
Verfasser:	Sebastian Adam
Tagesordnung:	<ol style="list-style-type: none">1. Konfigurationsmenü2. Loop-Szenenabfolge3. Wahrscheinlichkeiten der Goodies4. Invincible Goodie5. sonstiges
Protokoll zu den einzelnen Tagesordnungspunkten:	
<p>1. Pavlina stellte der Projektgruppe das fertige Konfigurationsmenü vor. Hierbei stellte Micha Kao jedoch fest, dass Pavlina eine andere Version von Crafty nutzt als er für die Spielelogik. Dieses Problem muss auf jeden Fall gelöst werden! Pavlina versucht nun die Funktionen auf die Crafty Version von Micha anzupassen und die gleiche Crafty Version wie bei der Spielelogik einzusetzen.</p>	
<p>2. Bezüglich der Szenenabfolge ist geplant, dass wenn die Spieleranzahl kleiner als 2 ist, dass dies die Maßgabe ist, damit die Hall of Fame aufgerufen wird.</p>	
<p>3. Die Wahrscheinlichkeiten der Goodies sollen möglichst nahe an denen des Originalspiels sein. Da wir es in unserem Projekt jedoch nicht schaffen werden ALLE Goodies zu implementieren, sollen die Wahrscheinlichkeiten der nicht implementierten Goodies auf die vorhandenen aufgeteilt werden. Das Verhältnis sollte danach dennoch dem originalen entsprechen.</p>	
<p>4. Der momentane Stand sieht wie folgt aus: Der Spielersprite wird weiß und der Charakter ist für eine Aktion unverwundbar, danach erscheint wieder der normale Player-Sprite. Im Original blinkt der Sprite jedoch dann abwechselnd weiß und normal, dies wird von Micha überprüft und umgesetzt.</p>	
<p>5. Sergej kümmert sich um die Alarm- Funktion mit rot blinkendem Hintergrund. Für das Menü soll eine zusätzliche Sprite-Auswahl der Spielcharaktere eingebaut werden. Hierfür soll Sergej vorsorglich schon einmal alle Spielersprites aus dem Original extrahieren.</p>	

2. Lastenheft

Einführung

Das Projekt „ReMasterBlaster“ und die Projektgruppe bestehend aus Sebastian Adam, Sergej Bjakow, Michael Kao, Pavlina Pavlova und Maximilian Seyfert entstanden durch die im 6. Semester des Bachelorstudiengangs Media Engineering zu erbringende Projektarbeit. Diese Projektarbeit ist durch den zeitlichen Rahmen von 26.03.2012 bis einschließlich 20.07.2012 begrenzt. Als Auftraggeber treten Prof. Dr. Matthias Hopf und Prof. Dr. Stefan Röttger in Erscheinung. Die Idee den AMIGA-Klassiker MasterBlaster in einer im Webbrowser spielbaren Version neu aufzulegen stammte von Herr Hopf, auf Nachfrage für eine Projektidee von Sebastian Adam. Die Projektgruppe setzte sich daraufhin aufgrund des gemeinsamen Interesses an einer Umsetzung des Spiels mittels neuen HTML5 – Webtechnologien zusammen.

Beschreibung des Ist-Zustandes

Der Ist – Zustand äußert sich wie folgt: Der AMIGA-Klassiker MasterBlaster liegt in einer ursprünglichen Fassung für den AMIGA vor. Dieser stammt aus dem Jahr 1994 und wurde von Alexander Ivanof entwickelt. Das Hauptziel im Spiel MasterBlaster ist es der letzte Überlebende in einer Runde zu sein. Um seine Gegenspieler „loszuwerden“ kann man Bomben legen und versuchen sie in die Luft zu sprengen oder sie durch „Wände verschieben“ töten. Die Wände können ebenfalls in die Luft gesprengt werden und manchmal verbirgen sich dahinter PowerUps oder Münzen um sich Ausrüstungsgegenstände im nach einer Spielrunde erscheinenden Shop zu kaufen. Das Originalspiel unterstützt eine Anzahl von 1-5 Spielern, die mittels Tastatur und Joysticks gesteuert werden können.

Beschreibung des Soll-Zustands

Der Soll – Zustand stellt sich folgendermaßen dar: Umgesetzt werden soll ein Remake des AMIGA Klassikers MasterBlaster. Dies soll mittels Reverse Engineering in Anlehnung an das Original erfolgen. Das Endprodukt soll mittels JavaScript und der 'Crafty JavaScript Game Engine' entwickelt werden. Das Spielverhalten sollte in höchstmöglichem Maß dem des Originals entsprechen. Es sollen am Ende des Projekts 2 Spielcharaktere gleichzeitig an einer Tastatur gesteuert werden können.

Funktionale Anforderungen

Der Remake soll sich in höchstmöglichem Maß am original orientieren. Siehe Punkt 3. Beschreibung des Soll-Zustands.

Nichtfunktionale Anforderungen

Das fertige Endprodukt soll im Webbrowser Google Chrome (ab v18.0) lauffähig sein und wird unter der MIT / GPL-Lizenz veröffentlicht. Der Source-Code ist seit Projektbeginn auf github einsehbar.

Benutzbarkeit

Das fertige Produkt soll im World Wide Web frei und somit für jeden zugänglich auf einer Website zur Verfügung gestellt werden. Es soll von vorerst zwei Spielern auf einer Tastatur gesteuert werden können.

Zuverlässigkeit

Der Spielfluss soll dem des Originals in höchstmöglichem Maße entsprechen. Es wird versucht einen möglichst reibungslosen Spielablauf ohne „Ruckler“ zu entwickeln. Einzig und allein durch die Garbage Collection von JavaScript kann es zu vereinzelt Rucklern kommen. Auf die Garbage Collection haben wir jedoch keinen Einfluss.

Lieferumfang

Geliefert wird nach Projektabschluss eine Website mit dem implementierten Spiel. Des weiteren soll eine kurze Dokumentation zur Einführung in das Spiel geschrieben werden, die ebenfalls auf der Webseite des Spiels angezeigt werden kann.

3. Pflichtenheft

Zielbestimmung

Musskriterien

Am Ende des Projekts muss ein spielbares Produkt von „ReMasterBlaster“ vorliegen, wobei zwei Spielfiguren auf einem Spielfeld mittels einer Tastatur gesteuert werden können. Es müssen Bomben gelegt und Wände gesprengt werden können. Des Weiteren wird die Spielfigur eines Gegenspielers vom Spielfeld entfernt, wenn diese innerhalb eines Bombenradius getroffen wird.

Sollkriterien

Es sollen weitere Spieldetails umgesetzt werden. Beim Wegsprengen einer Wand sollen verschiedene Goodies, die aus dem Original bekannt sind, erscheinen und der Spielfigur, die das jeweilige Goodie einsammelt, zusätzliche Eigenschaften übertragen. Hierbei sollen folgende Goodies berücksichtigt werden:

- Extra Bomb (man erhält eine extra Bombe)
- PowerUp (der Bombenradius erhöht sich um ein Feld)
- Superman (man kann Mauernblöcke verschieben)
- Ghost (man ist nur 1x1px groß und kann durch Wände gehen)
- Controller (man kann die Bombe fernsteuern)
- Protection (man ist für eine Aktion, die die Spielfigur des Spielers normalerweise vom Feld entfernen würde, geschützt → Spielfigur wird weiß)
- Timebomb (Bombenexplosion ist zeitgesteuert)
- SpeedUp (man wird schneller → max. 4x möglich)
- Stopschild (alle anderen Spieler drehen sich auf der Stelle)
- Fragezeichen (hier verbirgt sich eines der folgenden Goodies: Protection, Ghost, Stopschild oder Krankheit → Timer für Bombenexplosion verkürzt sich)
- Totenkopf (die Spielfigur des Spielers wird entfernt)

Weiterhin sollen verschiedene Szenen definiert werden, die in einer vordefinierten Reihenfolge ablaufen. Am Anfang soll ein Startscreen mit dem Logo erscheinen, auf den das Konfigurationsmenü folgt. Die Konfigurationsmöglichkeiten sollen mittels eines JSON-Objekts eingebunden werden. Nachdem man die Konfiguration abgeschlossen hat soll man in die eigentliche Spielszene gelangen wo man dann das Spiel spielen kann.

Kannkriterien

Im Optimalfall, vor allem aber falls am Ende der Muss- und Sollkriterien noch Zeit vorhanden sein sollte, kann überprüft werden ob mehr als 2 Spieler möglich sind. Des Weiteren kann eine Analyse stattfinden ob andere Eingabegeräte eingebunden werden können. Ebenfalls kann überprüft werden, ob das Spiel auch in Firefox, IE (v9 oder grösser) und Safari läuft bzw. lauffähig gemacht werden kann.

Produkteinsatz

Anwendungsbereiche

Das Endprodukt wird dem Nutzer auf einer Website zur Verfügung gestellt. Diese wird unter <http://www.remasterblaster.com> zu erreichen sein.

Zielgruppen

Primäre Zielgruppe des Projekts „ReMasterBlaster“ sind ambitionierte AMIGA-Fans, vor allem jedoch die Fans des originalen MasterBlaster-Spiels im Speziellen. Sekundäre Zielgruppe sind interessierte und spielfreudige Internetuser.

Betriebsbedingungen

Das Produkt wird auf einem gemieteten Webserver gehostet. Es wird 24h rund um die Uhr und 7 Tage die Woche zur Verfügung stehen sofern es keine Probleme mit dem Server von Seiten des Hosters gibt. Die Verfügbarkeit sollte jedoch zu 99,9% garantiert sein.

3. Produktübersicht

Das Produkt besteht aus einer Website auf der das Spiel „ReMasterBlaster“ eingebunden wird. Das Spiel selbst wird in JavaScript umgesetzt. Als Sprite-Engine wird die „Crafty JavaScript Game Engine“ zum Einsatz kommen.

4. Produktfunktionen bzw. Projektumsetzung

Die Produktfunktionen sind unter Punkt 1.1 Musskriterien und Punkt 1.2 Sollkriterien bereits genauer ausgeführt. Das Projekt wird in diversen Phasen durchgeführt. Am Anfang steht die Planungsphase, gefolgt von der Entwicklungsphase, wo die Hauptentwicklungsarbeit im Projekt stattfinden wird. Am Ende des Projekts steht die Dokumentationsphase an, in der die einzelnen Teammitglieder jeweils eine eigene Prüfungsstudienarbeit sowie die Gesamtdokumentation anfertigen werden. Zudem ist für den 09. Mai 2012 eine Zwischenpräsentation geplant, die ebenso der Dokumentationsphase zugeordnet ist. Die Projektumsetzung wird in einem offenen git-basierten Content-Management-System stattfinden.

5. Produktdaten

Im Projekt „ReMasterBlaster“ gibt es keine langfristig zu speichernden Daten aus Benutzersicht. Das Spiel wird langfristig auf einem Webserver zugänglich sein und kann vom Benutzer zu jeder Zeit über den Webbrowser aufgerufen werden.

6. Produktleistungen

Die zeitliche Abfolge der Aktionen im Spiel soll in höchstmöglichem Maße der im originalen Spiel MasterBlaster entsprechen. Hierfür wird eine genaue Zeitanalyse der Aktionen mittels Screencapturing und einem Videoschnittprogramm Frame für Frame durchgeführt.

7. Qualitätsanforderungen

Die eigentliche Qualitätsanforderung im Projekt „ReMasterBlaster“ orientiert sich daran, das Spiel möglichst originalgetreu umzusetzen. Dieser Punkt wurde jedoch schon mehrfach erwähnt, da es sich um einen Remake handelt. Eine weitere

Qualitätsanforderung ist, dass das Spiel im Browser Google Chrome ab v18.0 uneingeschränkt lauffähig sein soll.

8. Benutzungsoberfläche

Die Benutzeroberfläche wurde grundlegend schon in Punkt 1.2. Sollkriterien thematisiert. Es wird zu Beginn einen Eröffnungsscreen geben auf den dann das Konfigurationsmenü folgt. Nachdem die Spielkonfiguration durchgeführt wurde kann das Spiel gestartet werden. Dadurch dass das Spiel auf einer Website im World Wide Web zur Verfügung gestellt wird hat jeder Benutzer mit Zugang zum Internet automatisch auch Zugriff auf das Spiel, sofern ihm die URL bekannt ist.

9. Nichtfunktionale Anforderungen

Das Spiel wird unter der General Public License (GPL) entwickelt.

10. Technische Produktumgebung

Der Benutzer muss lediglich über einen Webbrowser verfügen um das Produkt abrufen zu können. Hierbei soll jedoch nur Google Chrome ab Version 18.0 vollständig unterstützt werden. Da das Spiel über den Browser gespielt werden kann stellt sich keine Plattformabhängigkeit im Sinne des Betriebssystems dar.

11. Spezielle Anforderungen an die Entwicklungsumgebung

11.1 Software

Die Entwicklungsumgebung besteht im Projekt „ReMasterBlaster“ aus einem Texteditor sowie dem Browser Google Chrome. Eine eigentlich eigenständige Entwicklungsumgebung wie z.B. Eclipse existiert somit nicht. Google Chrome verfügt über eine JavaScript Konsole in der Fehlermeldungen aus dem Quelltext angezeigt werden können.

11.2 Hardware

An die Hardware werden in unserem Projekt keine speziellen Anforderungen gestellt.

11.3 Orgware

Als Orgware wird GanttProject verwendet. Dies ist eine OpenSource Projektplanungssoftware mit der auch der Zeit- / Projektplan erstellt wird.

11.4 Entwicklungsschnittstellen

Entwicklungsschnittstelle ist im Projekt „ReMasterBlaster“ die Crafty JavaScript Game Engine auf der die gesamte Spielelogik sowie der Einsatz der verschiedenen Sprites aufbaut.