

Softwareprojekt I

Visionsdokument

Projektmitglieder-eMail: *mariowars@link2wall.de*

Vogel, Thomas	s740986	cody.vogel@googlemail.com
Palmer, Michael	s741050	mitch@palmomedia.de
Oeser, Florian	s744400	florian.oeser@web.de

Aufgabenbereiche/Verantwortlichkeiten

Die Aufgabenbereiche wurden sinngemäß nach den einzelnen Fähigkeiten und Vorlieben der Teammitglieder verteilt. Diese Aufteilung kann sich natürlich noch geringfügig je nach Fortschritt des Projekts ändern, da man jetzt noch schlecht abschätzen kann welche Probleme oder neue Aufgabenbereiche erschlossen werden müssen. Selbstredend werden manche Bereiche auch ineinander verschmelzen, wo eine genaue Abgrenzung nicht mehr gewährt werden kann und auch nicht mehr sinnvoll ist. Jedoch lässt sich jetzt schon ein grober Überblick erstellen, wobei auch hier die Aufgabenbereiche weiträumig gefasst sind.

OOA und OOD

Dieser Bereich wird logischerweise von allen Teammitgliedern bearbeiten und entwickelt. Die OOA und OOD gehört zu den wichtigsten und bedeutendsten Aufgaben eines jeden Softwareprojekts und entscheidet oft über Erfolg oder Misserfolg. Ein weiterer Punkt weshalb sich alle Mitglieder damit auseinander setzen müssen ist, dass jeder die Anforderungen an sich und das Projekt kennt und definiert.

Qualitätsmanagement

Auch dieser Aufgabenbereich betrifft alle Teammitglieder gleichermaßen und wird folglich auch von allen bearbeitet.

Programmierung

Der programmiertechnische Teil wird primär durch Michael Palmer und Florian Oeser umgesetzt. Beide verfügen über umgehende Erfahrung im Bereich Softwareentwicklung und versuchen diese in dem für beide neu erschlossenen Themengebiet einzusetzen.

Grafik/Sound und Spiellogik

Die Programmierung des Grafikworkarounds und die eigentlich Spielimplementierung / Spiellogik wird durch Herrn Oeser durchgeführt. Unterstützung durch Herrn Palmer ist beim 2. Punkt angesiedelt. Die Implementierung der Soundbibliotheken wird Aufgabe von beiden werden.

Netzwerkprogrammierung

Die Netzwerk-Programmierung (Client/Server) wird von Herrn Palmer bearbeitet. Hier zu werden mini Prototypen erstellt um diesen Themenbereich möglichst früh zu durchleuchten.

Datenbankprogrammierung

Die Datenbankprogrammierung– und Anbindung einschließlich der Datenhaltung wird Aufgabe von Herrn Palmer und Herrn Oeser sein.

Physik

Die Implementierung aller physikalischen Gegebenheit wird Herr Palmer übernehmen. Auf welche Techniken wir dabei zurückgreifen werden, wird sich noch herausstellen.

Einbindung 'sekundärer Ressourcen'

Dieser Bereich wird hauptsächlich von Herrn Vogel übernommen.

Modelling und Design(auch Models für etwaige GUI etc.)

Dieser Bereich wird ausschließlich von Herrn Vogel bearbeitet. Dies betrifft die einzelnen Charaktere sowie das komplette Environment.

Animation

Auch dieser Bereich wird, was die Erstellung der Animationen betrifft, primär von Herrn Vogel übernommen. Zur Einbindung dieser Ressourcen werden Herr Palmer und Herr Oeser Unterstützungsarbeit leisten.

Introgestaltung (Szenen allgemein)

Diesem Arbeitsfeld widmet sich wieder hauptsächlich Herr Vogel.

Allgemeine Aufgabenfelder

Diese Aufgabenbereiche werden wieder von allen bearbeitet, da es sich hier um allgemeine und nicht fachspezifische Aufgaben handelt. Hier diese aufzuführen wäre die Arbeit an Design und Stil welches unser Projekt verfolgt. Aufbau einer schönen und sinnvollen GUI. Ideen Sammlungen zum Spiel an sich. Alle Denkprozesse die vorher und auch während der Entwicklung anfallen. Eventuell wäre hier auch das Testing und Bugfixing in Bezug auf das oben erwähnte Qualitätsmanagement zu nennen.

Visionsdokument

In unserem Softwareprojekt wollen wir einen alten Nintendo-Gameklassiker wiederbeleben. Der Name des Originals lautet '*Super Mario Wars*'¹ und ist mittelmäßig bekannt. Es gibt sehr viele solcher kleinen Spiele die sich an den Nintendo-Originalen anlehnen, jedoch finden wir dieses Spiel am interessantesten da sie am meisten Spass bereiten. Hier erstmal zwei kleine Screenshots aus dem Spiel:



Der Sinn des Spiels ist sehr simpel und kurz zu erklären. Der Benutzer wählt einen Mario und hat das Ziel seine Mitspieler, auch in Form eines Marios, durch Sprung auf den Kopf zu eliminieren. Wer die meisten „Kills“ in einer gewissen Zeit schafft oder an das *Fraglimit*² gekommen ist, hat das Spiel gewonnen. Das Spiel dient somit zur Unterhaltung in Form von einem interaktiven Spielerlebnis.

In unserer Variante gibt es zwei große Unterschiede. Zum einen wird bis auf die GUI alles dreidimensional ablaufen. Sprich das 'Kampffeld' wird eine dreidimensionale Fläche sein auf der sich die Spieler bekriegen. Umgesetzt wird dies mit einer 3rd *Personcamera*³, die direkt hinter dem Spieler platziert sein wird. Das Umschalten in einen '*Free Camera Modus*'⁴ wird aber ebenfalls möglich sein. Ein anderer wichtiger Unterschied ist, dass wir das Spiel LAN sowie Internet fähig gestalten möchten, damit sich Spieler rund um den Globus herausfordern können. In der LAN Variante wird ein beliebiger Client auch den Server stellen, im Internet hingegen soll ein immer verfügbarer Server vorhanden sein. Eine KI (Künstliche Intelligenz) in Form von Bots wird nicht implementiert werden. Der Benutzer (Spieler) wird aber wie im Original den entsprechend bunten, comicartigen Grafikstil wiederfinden. Sprich die „Maps“ werden der gewohnten Super Mario Umgebung nachempfunden. Selbiges betrifft die Models (Figuren) jeglicher Art. Auch unsere GUI werden wir in diesem Design halten und erschaffen. Eine klare Anlehnung an die

Nintendo-Arts wird also gewährleistet sein und ist für diese Art des Spiels auch zwingend erforderlich. Auch die Sounds die der Benutzer im Menü und auch während des Spielens hören wird, sind eher kitschig und lustig.

Den Benutzer erwartet außerdem ein viel spannenderes und aktiongeladeneres Spielvergnügen als beim Original. Dies erreichen wir in dem wir mehrere *Modi's*⁵ einbringen werden. Ein einfaches '*Deathmatch*'⁶ ist auf Dauer ziemlich eintönig. Deshalb möchten wir unter anderem einen '*Capture the Flag*'⁷ und einen '*Last Man Standing*'⁸ – Modus implementieren. Ein weiterer Punkt der viel Spass bereiten wird, sind die 'Superwaffen' die der Spieler zur Verfügung gestellt bekommt. Einer davon wäre zum Beispiel das man doppeltes Leben bekommt. Sprich der Gegner muss ihm zweimal auf den Kopf springen damit er stirbt.

Des weiteren wird es für den Benutzer möglich sein verschiedene „Highscores“ und Statistiken nach dem Spiel einzusehen. Das fördert zum einen den Leistungsdruck besser zu werden und zum Anderen ist es immer schön zu sehen was man erreicht hat.

Als Eingabemöglichkeiten soll der Benutzer zwischen der Tastatur und dem Gamepad wählen können.

Die Zielgruppe richtet sich nicht direkt an junge Menschen, auch ältere Menschen werden problemlos unser Spiel spielen können. Eine gewisse Reaktionszeit ist allerdings vonnöten um das Spielgeschehen spannend und abwechslungsreich zu halten.

Wir haben uns für diese Software-Form entschieden weil wir selber leidenschaftliche Spieler sind und großes Interesse am Aufbau, beziehungsweise Software Engineering, von modernen Spieltiteln habe. Die Unterhaltungsbranche freut sich seit Jahren über eine steigende Nachfrage. Spiele integrieren sich immer mehr in die heutige Gesellschaft und gewinnen nach wie vor an Interesse bei einer breiten Nutzerschicht.

Die ingenieurmäßige Entwicklung vom Anfang bis zum Ende eines solchen Spieltitels erfordert ein sachgemäßen Umgang in vielen Bereichen. Wir werden die neusten Technologien verwenden um ein Spielerlebnis auf hohem Niveau zu bieten. Im Vordergrund steht bei unserem Spiel der Spielspaß; es handelt sich um keine E-learning Anwendung sondern um Software die zur Unterhaltung dient.

Technologien

Zur Umsetzung dieses Vorhabens werden viele verschiedene Technologien zum Einsatz kommen. Generell werden wir, was den programmiertechnischen Teil anbetrifft, mit C# arbeiten. Für die graphische Umsetzung (GUI & Spiel) werden wir auf das XNA-Framework zurückgreifen. Auch alle Effekte werden wir damit lösen. Für die Soundtransformierung– und Ausgabe werden wir auch mit einer API aus diesem Framework arbeiten oder eventuell auch mit OpenAL. Die Netzwerkprogrammierung wird mit Klassen aus dem .NET Framework umgesetzt. Für die im Spiel vorkommenden physikalischen Vorgänge (zum Beispiel die Sprünge oder Kollisionsabfragen) und deren Umsetzung werden wir uns der freien ODE-Engine bedienen. Als Datenbank werden wir wahrscheinlich MySQL oder HSQL einsetzen.

Für die Modellierung verwenden wir Maya 2008. Die Texturierung erfolgt in Photoshop.

Damit unsere Charaktere performant im Spiel implementiert werden können benötigen wir normal/displacementmaps⁹ diese werden wir in Zbrush generieren und auf das fertige Model dann mappen.

Für die Animationen verwenden wir das Programm Endorphin. Die fertig modellierten Models laden wir in Endorphin und animieren die Figuren.

Bitte haben Sie Verständnis dafür, das sich bis zum Pflichtenheft einige Technologien ändern oder neue hinzukommen. In dem aktuellen Entwicklungsstatus können wir eine Realisierung mit einer bestimmten API/Schnittstelle noch nicht festmachen (zum Beispiel OpenAL für den Sound). Gerne würden wir mit Ihnen diesbezüglich auch noch einmal Rücksprache halten.

Glossar

¹ <http://smw.72dpiarmy.com/>

² vorher definierter Zahlenwert; wie viel „Kills“ braucht man um zu gewinnen

³ eine Kamera die mit ein wenig Entfernung oberhalb auf den Spieler (das Model) gerichtet ist

⁴ eine komplett frei bewegliche Kamera die auf den Spieler gerichtet ist

⁵ Modifikationen

⁶ jeder tötet jeden; bekannt durch viele Multiplayer Egoshooter

⁷ in dieser Modifikationen gibt es zwei Teams mit jeweils einer Flagge. Ziel ist es die gegnerische Flagge zu stehlen und zur eigenen Basis zurückzubringen. Ist dies passiert erhält man einen Punkt und die gegnerische Flagge erscheint wieder an der Basis; bekannt durch viele Multiplayer Egoshooter

⁸ hier spielt wieder Jeder gegen Jeden. Mit dem Unterschied das jeder nur eine begrenzte Zahl an Leben hat. Ist man bei 0 angekommen fliegt man aus dem Spiel. Sieger ist, wer als letztes überlebt hat; bekannt durch viele Multiplayer Egoshooter

⁹Normalmaps stellen eine besondere Art der Texturierung dar. Es werden Höhen und Tiefen auf der Geometrie vorgetäuscht. Dies hat den Effekt das Models bzw Charaktere in Spielen wesentlich mehr Details besitzen. Es wird immens an Performance gewonnen und ist in heutigen Spieletiteln ein muss.