



DONGER SoftwareEngineering
Professor-Heimer-Platz 1
Piltover
Chemnitz, 11.11.2016

Angebot Entwicklung des Spiels Bomberman

Sehr geehrter Herr Professor Dr. Becker,
sehr geehrter Herr Frank,

wir bedanken uns für Ihre Anfrage vom 14.10.2016 und Ihr Interesse an unserem jungen Entwicklerteam.

Auf Basis des vorliegenden Lastenheftes haben wir uns umfassend mit Ihren Wünschen, Anforderungen beschäftigt und ein entsprechendes Angebot erstellt.

Unser junges, flexibles Entwicklerteam ist hochmotiviert und sieht sich jeder Herausforderung gewachsen. Wir konnten schon einige Erfahrungen im Bereich der Spieleentwicklung und den Umgang mit Client-Server Anwendungen erlangen. Sie nehmen Im Laufe des Projektes begleitend Teil und werden stets über den Entwicklungsstand des Projektes informiert.

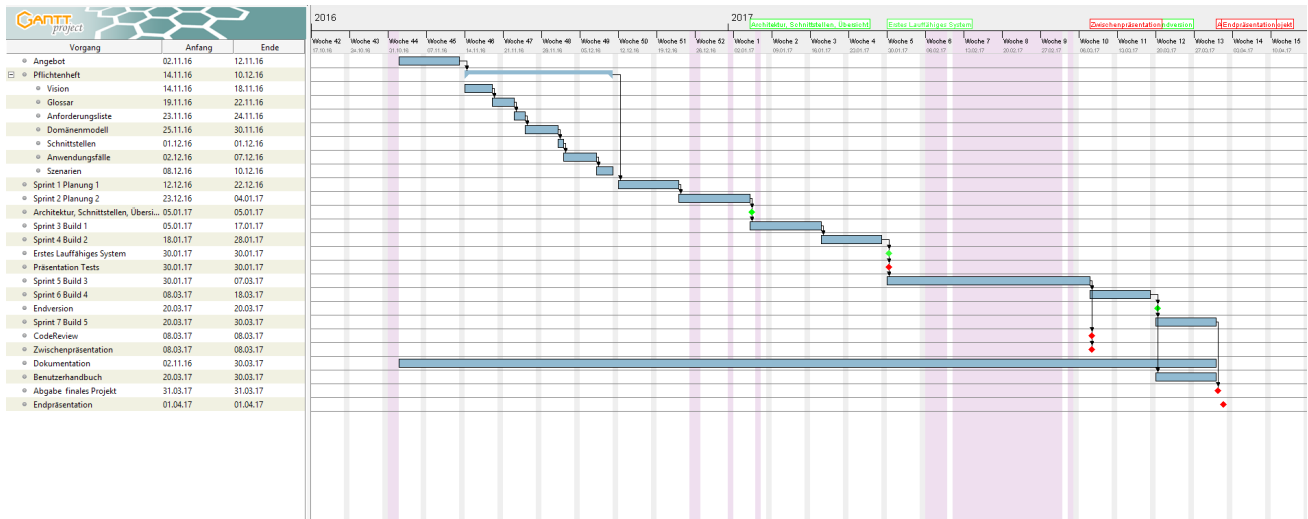
Wir freuen uns auf eine weitere Zusammenarbeit und stehen Ihnen gern für weitere Fragen zur Verfügung.

Mit freundlichen Grüßen

D. Hofmann

Danny Hofmann, Projektleitung

Projektplan:



Der Projektstart ist am 01.11.2016, wenn sie unser Angebot akzeptieren. Unser Team wird sich sofort an die Arbeit machen und wie von Ihnen gefordert ein Pflichtenheft konzipieren, in welchem wir die von Ihnen gestellten Ansprüche mit unserer Interpretation dieser vergleichen. Nach der Abgabe des Pflichtenhefts, werden wir mithilfe der agilen Softwareentwicklungsmethode SCRUM Ihr Projekt realisieren. Wie Sie im Projektplan sehen können, führen wir in regelmäßigen Abständen von jeweils 10 Tagen Abschnitte, sogenannte Sprints, durch. In diesen Sprints realisieren wir die gewünschten Produktfunktionalitäten. Aktuell ist uns noch nicht möglich, Ihnen mitzuteilen, zu welchem Zeitpunkt bestimmte Elemente vorhanden und funktionell sind. Wir versichern Ihnen aber, Ihr Projekt wird bis zur Abnahme vollständig funktional sein. Wir werden die ersten 2 Sprints dazu nutzen uns eine Architektur zu erarbeiten und die Schnittstellen aus dem Pflichtenheft einzuarbeiten. Nach der Planung werden wir in den fünf Build Sprints das Produkt implementieren und testen. Unser Team wird nicht über die Weihnachtsfeiertage arbeiten und auch nur an zwei Tagen der Prüfungsphase vom 06.02.17- 04.03.17 der TU-Chemnitz, da alle unsere Mitarbeiter noch studieren. Noch vor der Prüfungsphase werden Ihnen die ersten Tests vorgestellt. Nach der Prüfungsphase werden dann die Finalen Sprints durchgeführt, wo das Projekt am Ende abgeschlossen wird und mit einer Endpräsentation an Sie übergeben. Projektende ist der 31.03.2017.

Die Dokumentation und alle Dokumente des Projekts werden im Laufe des Projekts im Sinne des V-Modells entworfen. In den Planungs-Sprints werden die ersten groben Entwürfe gefertigt, welche dann in den Buildsprints verfeinert werden. Noch vor der Prüfungsphase werden Ihnen die ersten Tests vorgestellt und Einblick in den Fortschrittschritt. Zum Ende eines Sprints werden die implementierten Funktionen getestet und ins Gesamtsystem integriert. In Sprint fünf und sechs werden dann Systemtests durchgeführt und dokumentiert. Schließlich werden im siebten Sprint zum Abschluss Abnahme Tests durchgeführt.

11.11.2016

Gruppe 7

Version 5.0

Kategorie	Anwendungsfall		Komplexität	Server/Client
Eingabedaten	Spielkonfiguration	festlegen der Spielkonfiguration	Einfach	Server
	Benutzereingaben	Benutzereingaben einlesen	Einfach	Client
Abfragen	Spielsimulation	Eingabedaten vom Client empfangen	Mittel	Server
		Spielberechnung	Komplex	
		KI-Berechnung	Komplex	
		Ranking berechnen	Mittel	
	Netzwerk	Verbindung erstellen	Mittel	Server und Client
		Verbindung überwachen	Einfach	
		Spieldaten senden und empfangen	Einfach	
	Verwaltungsserver	Spielserver verwalten	Einfach	Server
Ausgabedaten	Client-GUI	Spielfeld anzeigen	Mittel	Client
		Menü anzeigen	Einfach	
		Fehlermeldungen	Mittel	
		Ranking anzeigen	Einfach	
	Server-Interface	Spiel-Log	Mittel	Server
		Fehlermeldungen	Mittel	
		Verwaltungsanzeige	Einfach	
Datenbestände	Spielzugehörige Daten	Texturen anzeigen	Einfach	Client
		Audio abspielen	Einfach	
		Objektdaten	Mittel	
Referenzdaten	Spielereigenschaften	letzte Spielerposition, Spielerrichtung	Mittel	Server
		Momentane Eigenschaften	Mittel	

Kategorie	Anzahl	Klassifizierung	Gewichtung	Summe
Eingabedaten	2	Einfach	3	6
	-	Mittel	4	-
	-	Komplex	6	-
Abfragen	3	Einfach	3	9
	3	Mittel	4	12
	2	Komplex	6	12
Ausgabedaten	3	Einfach	4	12
	4	Mittel	5	20
	-	Komplex	7	-
Datenbestände	2	Einfach	7	14
	1	Mittel	10	10
	-	Komplex	15	-
Referenzdaten	-	Einfach	5	-
	2	Mittel	7	14
	-	Komplex	10	-
Summe	E1			109
Einflussfaktoren ändern den Function-Point- Wert um +/- 30%	1. Verflechtung mit anderen Anwendungssystemen (0-5) → Kompatibilität mit anderen Guppen			4
	2. Dezentrale Daten,dezentrale Verarbeitung (0-5)			0
	3. Transaktionsrate (0-5)			4
	4. Verarbeitungslogik a) Rechenoperationen (0-10) → KI, Spielberechnung b) Kontrollverfahren (0-5) → mogeln/cheaten soll nicht möglich sein c)Ausnahmeregelungen (0-10) → Netzwerkprobleme d) Logik (0-5)			10
				2
				3
				0
	5. Wiederverwendbarkeit (0-5) → ähnliches Spiel wird von unserer Gruppe wahrscheinlich nicht wieder programmiert			5
Summe der 7 Einflüsse	6. Datenbestandskonvertierungen (0-5)			0
	7. Anpassbarkeit (0-5) → Spielupdates-/veränderungen			3
Faktor Einflussbewertung $E2/100 + 0,7$	E3			1,01
Bewertete	AFP			110,09

Function Points E1 * E3		
Man Month nach IBM	MM	9
Man Hours nach IBM MM * 160	MH	1440
Arbeitsaufwand pro Mann (in Monaten) nach IBM MM / 6		1,5
Arbeitsaufwand pro Mann (in Stunden) nach IBM MH / 6		240
Arbeitsaufwand pro Mann (in Monaten) aufgrund der Auslastung (15h Zeit pro Woche)		4

Zusammenfassung des geschätzten Gesamtaufwands		
Phase		Aufwand
	Arbeitsstunden	Kosten (in €)
Vorarbeiten		2400
Machbarkeitsstudie	48	2400
Lastenheft und Aufwandsschätzung	48	
Nachstehend		
Anforderungsdefinition	70	
Pflichtenheft	70	
Analyse und Entwurf	450	
Übersicht- und Komponentendiagramm	90	
Sequenzdiagramm und Statecharts	150	
Analysetabellen	30	
Architektur und Analyse-Klassendiagramm	50	
Entwurfs-Klassendiagramm	50	
Administration und Verifikation	50	
Schnittstellendefinition	30	
Implementierung	450	
Prototyp für		
- Client	90	
- Spielserver	120	
- Verwaltungserver	40	
KI-Implementierung	100	
Client-GUI	80	
Server-Interface	30	
Tests	300	
Erarbeiten der Testwerte / Szenarien (Netzwerktest, Spieltest, GUI Test, KI Test, ...)	100	
Implementierung der Szenarien	150	
Tests durchführen	50	
Präsentation	30	
Dokumentation	140	
Entwicklerhandbuch/Entwurfs-dokument	100	
Benutzerhandbuch	40	
Gesamter Aufwand	1440	72000

Im folgenden Abschnitt sollen die einzelnen Funktionen und Kosten des Bomberman-Projektes erklärt werden, um unsere Kostenabschätzung zu erläutern.

Der Aufwand für die Eingabedaten ist relativ gering, da Steuerung und Konfiguration des Spiels mit Tastatureingaben zu realisieren sind. Benutzereingabe umfassen dabei alle Aktionen, die vom Nutzer des Clienten übernommen werden, also beispielsweise Steuerung der Spielfigur, Einschalten der KI-Funktion, etc.

Abfragen des Projektes gliedern sich in Spielsimulation, Netzwerk und Verwaltungsserver. Die Spielsimulation ist ein sehr aufwendiger Teil, welches den zentralen Punkt des Programms darstellen wird. Der Empfang der Aktionen des Clienten darf nicht verzögert und unzuverlässig funktionieren, damit die eigentliche Berechnung des Spiels flüssig läuft und ein verzögerungsfreies Spielerlebnis bietet. Das „Herz“ des Programms ist die Berechnung des Spieles, welche von uns als aufwendig kategorisiert wurde. Diese soll möglichst zuverlässig sein und sofort stattfinden. Das Erstellen der Künstlichen Intelligenz, die den Spieler bei Bedarf ersetzen soll, ist hierbei ebenfalls aufwendig, da sie nicht zufällig agieren soll, sondern nach Möglichkeit sehr Spieler-ähnlich, was wiederum den Großteil des Aufwandes in diesem Punkt einnehmen wird. Das Ranking, welches am Ende des Spiels angezeigt werden soll, ist im Vergleich zu den vorherigen Punkten weniger aufwendig anzusetzen. Schließlich die Netzwerkverbindung muss eine stetige Verbindung zur Verfügung stellen, die die Spieldaten sendet und empfängt. Weiterhin stellt der Verwaltungsserver eine Liste der Spielserver zur Verfügung und verwaltet die Server.

Zu den Ausgabedaten gehören die Client-GUI und das Serverinterface. Die Client-GUI soll dem Spieler das Spielfeld anzeigen, was beinhaltet, dass die Spieldaten des Servers in ein darstellbares Bild umgewandelt werden. Außerdem ist die Anzeige von Menü und Ranking relativ simpel zu realisieren. Zusätzlich werden aussagekräftige Fehlermeldungen benötigt, damit die User bei Problemen nach Möglichkeit selbst Lösungen finden können. Das Server-Interface soll den aktuellen Spiel-Log bereitstellen, der während des Spielens erstellt wird und alle protokollierten Aktionen der Spieler auflisten. Dies ist wiederum etwas aufwendiger, da dieser Log sehr umfangreich sein muss. Serverseitige Fehlermeldungen müssen außerdem genau sein, damit der Administrator Probleme schnell und effizient beheben kann, was sonst zu Unmut bei den Spielern führen könnte. Darüber hinaus muss die Verwaltungsanzeige des Servers den Status der Spielserver und Optionen für den Administrator zum Auswählen bereitstellen, was ebenfalls gering vom Aufwand anzusetzen ist.

Zu den Datenbeständen bzw. den Spiel-zugehörigen Daten zählen unter anderem die Texturen, die vom Clienten angezeigt werden. Texturen für jedes Objekt müssen eigens erstellt und bearbeitet werden, damit diese gut erkenn- und unterscheidbar sind. Für die Audiodaten trifft dies ebenfalls zu. Diese Daten muss der Client jeweils bereithalten und bei Bedarf anzeigen oder abspielen können. Außerdem müssen bestimmte Eigenschaften der Spielobjekte gespeichert werden, wie beispielsweise Position der Spieler, Zerstörbarkeit von Mauern, etc.

Bei den Referenzdaten handelt es sich um Daten, die sich innerhalb des Spielverlaufs ändern können bzw. ändern. Diese werden vom Server gespeichert und bei Bedarf an den Spieler übermittelt. Zum Beispiel sollten Spielerposition und Blickrichtung möglichst exakt bestimmt werden. Insgesamt soll dem Spieler ein möglichst flüssiges Erlebnis geboten werden, was sicherlich etwas Feinschliff benötigen wird.

Bei den momentanen Eigenschaften handelt es sich um Werte, die zum Beispiel die einzelnen Bombermanfiguren zu bestimmten Zeiten annehmen können. Dazu gehören unter anderem die Aufnahme von Boni (Rüstungen, etc.).