Projekt: JBomberman
Projektplan

Pascal Kistler
Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
23.02.15	1.00	Erstellung des Dokuments	Gruppe
24.02.15	1.01	Neue Texte eingefügt	Fabian Binna
25.02.15	1.02	Logo + Neue Texte	Silvan Adrian
27.02.15	1.03	Arbeitspakete	Fabian binna
27.02.15	1.04	Meilensteine + Phasen Iterationen	Silvan Adrian
27.02.15	1.06	Formatierungen + Design Anpassungen	Silvan Adrian
27.02.15	1.06	Korrekturen + Kostenvoranschlag	Fabian Binna
07.03.15	1.07	Migration auf LaTex und Verbesserungen	Silvan Adrian
17.03.15	1.08	Verbesserungen	Silvan Adrian

Inhaltsverzeichnis

1 Änderungshistorie **2**

2 Einführung **5**
2.1 Zweck . . . 5
2.2 Gültigkeitsbereich . . . 5
2.3 Referenzen . . . 5

3 Projektübersicht **5**
3.1 Zweck und Ziel . . . 5
3.2 Primäre Features . . . 5
3.3 Erweiterte Features . . . 5
3.4 Lieferumfang . . . 6
3.5 Annahmen und Einschränkungen . . . 6

4 Projektorganisation **6**
4.1 Organisationsstruktur . . . 6
4.2 Externe Schnittstellen . . . 7

5 Managment Abläufe **7**
5.1 Kostenvoranschlag . . . 7
5.2 Phasen / Iterationen . . . 7
5.3 Reviews . . . 8
5.4 Besprechungen . . . 8

6 Risikomanagement **8**
6.1 Risiken . . . 8
6.2 Umgang mit Risiken . . . 8

7 Arbeitspakete **8**

8 Infrastruktur **9**
8.1 Entwicklungsinfrastruktur . . . 9
8.2 Tools/Software . . . 9
8.3 Kommunikationsmittel . . . 9

9 Qualitätsmassnahmen **9**
9.1 Dokumentation . . . 9
9.2 Projektmanagement . . . 9
9.3 Entwicklung . . . 10
9.3.1 Vorgehen . . . 10
9.3.2 Unit Testing . . . 10
9.3.3 Code Reviews . . . 10

9.3.4 Code Style Guidelines 10
9.4 Testen 10
9.4.1 Komponententest 10
9.5 Systemtest 10

2 Einführung

2.1 Zweck

Dieses Dokument beschreibt die Planung des SE2 Projekts.

2.2 Gültigkeitsbereich

Dieses Dokument ist während des ganzen Projekts gültig und wird laufend aktualisiert.

2.3 Referenzen

Spielprinzip/Beschreibung: http://de.wikipedia.org/wiki/Bomberman
Konzepte für Games: Spiele entwickeln für iPad, iPhone und iPod touch, Thomas Lucka, 2012, ISBN 978-3-446-43085-3

3 Projektübersicht

JBomberman ist ein Geschicklichkeitsspiel, das die gleichen Regeln wie das klassische Bomberman umfasst. Es gibt nur einen Multiplayer Modus, in dem bis zu vier Spieler gegeneinander antreten können. Der Multiplayer Modus ist über ein Netzwerk spielbar.

3.1 Zweck und Ziel

Wir wollen gelerntes aus Prog1, Prog2, Uint1 und SE1 anwenden und erweiterte Programmierkonzepte erlernen. Hinzu kommen Module wie VSS und ParProg, die wir parallel besuchen und auch für die Durchführung des Projekts notwendig sind. Unser Ziel ist es ein Software Projekt in einem Team erfolgreich durchzuführen.

3.2 Primäre Features

- Spiellogik
- Netzwerkspiel
- Dedizierter Server
- Desktop Client

3.3 Erweiterte Features

- Verschiedene Powerups

3.4 Lieferumfang

- Source-Code
- Dokumentation (Projekt und Software)
- Ausführbare Applikation

3.5 Annahmen und Einschränkungen

Die Applikation wird lediglich für Desktops zur Verfügung gestellt, dabei sollen folgende Betriebssysteme (Mac OSX , Linux, Windows) unterstützt werden um eine möglichst grosse Userbasis anzusprechen. Das Projekt soll nach Abschluss Open Source für Interessenten zur Verfügung gestellt werden.

4 Projektorganisation

Jedes Teammitglied konzentriert sich auf seine eigenen Themengebiete, jedoch werden die Entscheidungen im Team besprochen und das Knowhow auf alle verteilt. Entscheidungen die das gesamte Projekt beeinflussen werden im Team besprochen und müssen von jedem akzeptiert werden.

4.1 Organisationsstruktur

Daniel Keller

- Projektbetreuer

Silvan Adrian

- Projektleiter
- Serveradministrator
- Build Managment
- Entiwcklung

Pascal Kistler

- Qulitätsmanagement
- Entwicklung

Fabian Binna

- Software-Architekt
- Entwicklung

4.2 Externe Schnittstellen

Projektbetreuer: Daniel Keller (d1keller@hsr.ch)

5 Managment Abläufe

5.1 Kostenvoranschlag

Wir rechnen mit 453 Stunden Gesamtaufwand. 93 Stunden davon sind Reserve aufgrund von potenziellen Risiken.

Gesamtaufwand: 453h
Davon Risikoaufwand: 93h
Aufwand pro Woche: 30h
Aufwand pro Woche pro Teammitglied: 10h

5.2 Phasen / Iterationen

Iteration	Beschreibung	Datum
Inception 1	Projektantrag einreichen	20.02.15
Elaboration 1	Anforderungen definiert Technologien ausgewählt und Projektplan fertig gestellt. Erster Prototyp zum Testen mit Grundfunktionen (Befehle an Server senden).	05.03.15
Elaboration 2	Use Cases aufschreiben, dazu das Spielprinzip genauer beschreiben. Nicht Funktionale Anforderungen bestimmen.	22.03.15
Elaboration 3	Architektur Prototyp (Kommunikationsverfahren zwischen Client und Server), Performance Tests zum ausgewählten Verfahren.	05.04.15
Construction 1	Hauptfeatures implementieren (Kollisionsdetektion, Bomben Funktionieren, Powerups funktionieren)	19.04.15
Construction 2	Spielablauf und Matchmaking fertig. Falls genügend Zeit werden werden noch erweiterte Features eingebaut.	03.05.15
Construction 3	Suchablauf und Matchmaking fertig. Falls genügend Zeit werden noch erweiterte Features eingebaut.	23.05.15
Transition	Vorbereitung zur Schlusspräsentation und Abgabe.	30.05.15

5.3 Reviews

Datum	Reviews	Beschreibung
05.03.15	Review Projektplan mit Zeitplan und aktuellen Iterationsplänen	MS1-RV Projektplan
19.03.15	Review der Anforderungsspezifikation und der Domainanalyse	MS2-RV Anforderungen und Analyse
09.04.15	Zwischenpräsentation mit Demo eines Architekturprototypen, Review	MS3-RV Ende Elaboration
07.05.15	Review von Architektur + Design und Architekturdokumenten	MS4-RV Architektur/Design
28.05.15	Präsentation und Demo der Software	MS5-RV Schlusspräsentation und Schlussabgabe

5.4 Besprechungen

Es werden regelmässige Besprechungen am Donnerstag/Montag morgen eingeplant um den Stand der jeweiligen arbeiten zu erfahren und gegebenenfalls Hilfestellung anzubieten ,falls etwas nicht funktionieren sollte.

6 Risikomanagement

6.1 Risiken

Risiken werden im Dokument TechnischeRisiken-JBomberman.xslx beschrieben

6.2 Umgang mit Risiken

Für die Risiken werden Reserven eingeplant. Die Reserven werden direkt in die einzelnen Tickets eingerechnet. Falls Risiken eintreffen werden diese sofort kommuniziert. Jede Woche werden eingetroffene Risiken und potenzielle Risikogefahren diskutiert. Eingetroffene Risiken werden im Team besprochen und mögliche Lösungen evaluiert.

7 Arbeitspakete

Die Arbeitspakete werden in Redmine erstellt und gepflegt (siehe Screenshot). Lesender Zugriff ist anonym möglich, schreibender nur eingeloggt (Projekt ist öffentlich). Link zum Redmine: http://se2p.zonk.io/redmine

8 Infrastruktur

8.1 Entwicklungsinfrastruktur

Aus persönlicher Präferenz ist die Infrastruktur breit gesät. Beim IDE konnten wir uns auf einen gemeinsamen Nenner bringen.

Name	Hardware	Betriebssystem	IDE
Pascal Kistler	ASUS	Windows 8.1	Eclipse Luna
Silvan Adrian	MacBook Pro	OSX 10.9.5	Eclipse Luna
Fabian Binna	Lenovo T530s	Ubuntu 14.04.1	Eclipse Luna

8.2 Tools/Software

- BuildServer: Jenkins
- CodeReview GitLab
- Konfigurationsmanagement GIT

8.3 Kommunikationsmittel

Ausserhalb des Reviewtermins wird mit folgenden Mitteln kommuniziert:

- E-Mail
- Skype
- Redmine (Wiki Kommentare)
- GitLab (Kommentare)
- Whatsapp

9 Qualitätsmassnahmen

9.1 Dokumentation

Sämtliche Dokumente werden in Git gespeichert. So sind diese immer für alle verfügbar und können einfach versioniert und Änderungen nachvollzogen werden.

9.2 Projektmanagement

Als Projektmanagementtool wird Redmine verwendet. Hier werden alle ausstehenden, laufenden und erledigen Tickets verwaltet.

9.3 Entwicklung

Der Source Code befindet sich ebenfalls in einem Git Repository und wird durch regelmässige Code Reviews überprüft.

9.3.1 Vorgehen

Jedem Commit wird eine Beschreibung hinzugefügt, danach knnen die Commits im Redmine dem passenden Ticket zugewiesen werden

9.3.2 Unit Testing

Es sollten für jede Klasse / jedes Codemodul Junit Tests geschrieben und durchgeführt werden. Vor jedem push müssen alle Test erfolgreich durchlaufen werden.

9.3.3 Code Reviews

Durch Code Reviews wird sichergestellt, dass der Code den Style Guidelines entspricht. Durch die Kontrolle durch ein oder zwei weitere Teammitglieder wird sichergestellt, dass die Codequalität stets auf hohem Niveau ist.

9.3.4 Code Style Guidelines

Es werden die Code Style Guidelines aus den Modulen Prog1 und SE1 verwendet. Namen im Code (wie Variablen, Methoden, etc.) werden in Englisch benannt.

9.4 Testen

9.4.1 Komponententest

Zu jeder Komponente werden mehrere Test geschrieben. Diese stellen sicher, dass alle Komponenten ordnungsgemäss funktionieren und kein nicht funktionierender Code gepusht wird.

9.5 Systemtest

Systemtest werden gegen Ende des Projekts erstellt. Sie garantieren, dass das gesamte System ordnungsgemäss funktioniert und alle Komponenten richtig zusammen arbeiten.