

Projektskizze

Cube Circle Tower Defense (CCTD)

Version 1.0
25. Oktober 2010

Rolf Koch, Lukas Spirig, Benjamin Felder, Fabian Eriksson, Nathanael Koch

Zusammenfassung

- Idee
- Hauptanwendungsfall
- Weitere Anforderungen
- Ressourcen
- Risiken
- Grobplanung
- Kundennutzen
- Wirtschaftlichkeit

Inhaltsverzeichnis

1 Idee	1
1.1 Begriffserklärung Tower Defense	1
2 Hauptanwendungsfall	2
3 Weitere Anforderungen	3
4 Ressourcen	4
4.1 Personelles	4
4.2 Materielles	4
5 Risiken	5
5.1 Verwertungsrisiken	5
5.1.1 Nachfrage	5
5.2 Projektrisiken	5
5.2.1 Kosten	5
5.2.2 Projektumfang	5
5.3 Allgemeine Risiken	5
5.3.1 Personalausfall	5
5.3.2 Hardwareausfall	5
6 Grobplanung	6
7 Kundennutzen	7
8 Wirtschaftlichkeit	7

1 Idee

Das Genre des Tower Defense Spiele ist sehr verbreitet. Die meisten dieser Spiele sind webbasiert und mit Flash realisiert. Ziel dieses Projektes ist es ein Javabasiertes multiplayer Tower Defense zu realisieren.

Das Spiel wird im Internet zum Download angeboten (gratis / kostenpflichtig), sodass es sich auf dem Markt verbreiten kann.

Der grosse Unterschied zu anderen Tower Defense Spielen ist die Wichtigkeit der Zusammenarbeit der beteiligten Spieler:

- Einzelne Türme werden zu teuer sein, sodass man diese nicht ohne Hilfe der anderen Spieler kaufen kann.
- Die Spieler haben die Möglichkeit, einen Tower von einem anderen Spieler zu upgraden. Diese Upgrades stehen für die eigenen Towers nicht zur Verfügung.
- Es besteht die Möglichkeit, dass die Spieler sich Geld zur Unterstützung einander überweisen.

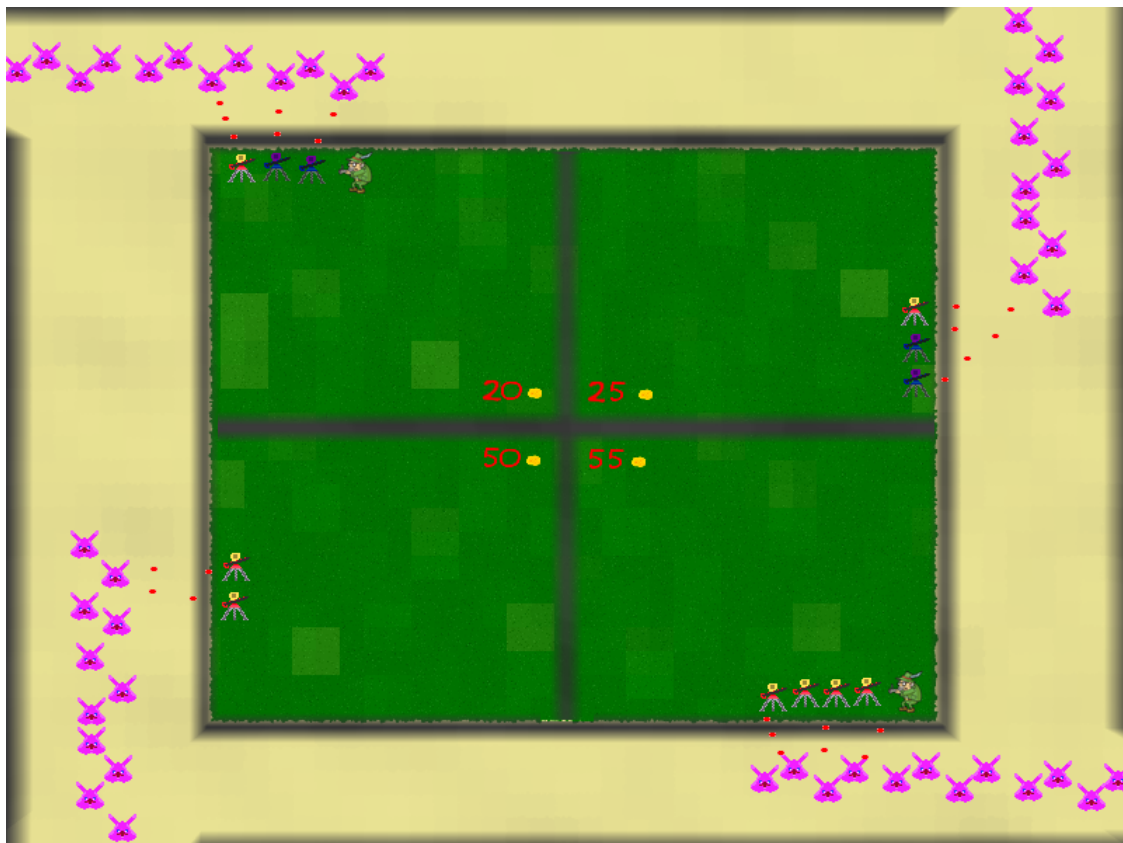


Abbildung 1: Skizze des Spielfelds

1.1 Begriffserklärung Tower Defense

Tower Defense Spiele basieren immer auf dem selben Grundprinzip: Es existieren 4 Hauptelemente. Die Karte (Bauplatz für Spieler), Der Weg, die Gegner und die Türme. Die Gegner laufen dem vordefinierten Weg entlang und der Spieler kann Türme bauen (meistens verschiedene Typen), welche autonom auf die Gegner schießen, worauf die Gegner Lebenspunkte verlieren.

2 Hauptanwendungsfall

Das Spiel muss mit mindestens einem Spieler gespielt werden. Maximal können am Spiel 4 Personen teilnehmen. Der Spielablauf funktioniert folgendermassen:

- Ein Spieler erstellt ein Spiel und startet es bei Erreichen der gewünschten Anzahl Spieler.
- Haben alle Spieler das Spiel betreten, beginnt der eigentliche Spielverlauf.
- Alle Spieler sehen einen Kartenausschnitt, der 1/4 der gesamten Karte ausmacht.
- Vor Beginn der ersten Runde haben alle Spieler Zeit um die ersten Türme mit ihrem Startkapital zu bauen.
- Nach Ablauf dieser Schonzeit erscheinen die Gegner an allen Seiten die von Spieler belegt sind und laufen auf dem Pfad entlang. Pro Gegner, der das Spielfeld betritt wird ein Zähler um 1 inkrementiert.
- Erreicht der Zähler eine vordefinierte Grenze ist das Spiel verloren.
- Die von den Spielern erstellten Türme beginnen autonom auf die Gegner in Reichweite zu schießen.
- Gegner die getroffen werden verlieren Lebenspunkte und verschwinden vom Spielfeld, wenn diese 0 Lebenspunkte erreichen. Der Gegner-Zähler wird dadurch um 1 dekrementiert.
- Durch das zerstören von Gegnern bekommen die Spieler Geld, welches sie brauchen um neue Türme zu bauen.
- Mit Geld kann man auch bestehende Türme erweitern(updaten) und so die Kerneigenschaften stärken (Von Typ zu Typ verschieden).
- Nach einer gewissen Zeit erscheinen wieder neue Gegner(Neue Runde/Wave), welche wieder von den Türmen angegriffen werden.
- Damit die Spieler gewinnen können, müssen diese alle Runden/Creep-Wellen überstehen und alle übrig gebliebenen Creeps besiegen. Die Anzahl Runden werden vom Server bestimmt.

3 Weitere Anforderungen

Folgende Funktionen sollten in der Grundauführung ebenfalls vorhanden sein:

- Das Spiel soll sowohl alleine als auch im Multiplayer Modus gespielt werden können. Im Einzelspieler-Modus erscheint also nur ein Feld, auf welchem der Spieler seine Türme platzieren kann, anstatt deren vier.
- Im Multiplayer Modus sollen bis zu vier Spieler miteinander gegen die Creeps kämpfen können.
- In diesem Spiel sind alle Spieler aufeinander angewiesen. Ist einer von ihnen mit den Gegnern überfordert, weil er zum Beispiel eine Fehlinvestition getätigt hat, so kann ihm ein Kollege Geld überweisen, damit dieser sein Viertel wieder besser verteidigen kann.
- Die Spieler sehen zwar nur ein Viertel des Spielfeldes, können diesen Kartenausschnitt jedoch verschieben, um zu sehen, wie es den Mitspielern ergeht, damit er ihnen unter Umständen helfen kann.
- Des weiteren soll das Spiel eine einfache Chat-Funktion enthalten, damit die Spieler sich auch austauschen oder Taktiken besprechen können.
- Falls nur 2 oder 3 Spieler mitspielen, bleiben die restlichen Spieler-Felder unbelegt und es spawnen auch keine Creeps an diesen jeweiligen Spawnpunkten.

4 Ressourcen

4.1 Personelles

Unser Team besteht aus 5 Personen, welche alle über ein solides Wissen zu Java verfügen. Einzelne Teammitglieder sind des weiteren spezialisiert zu elementaren Themen wie Server / Client - Kommunikation und 2D Spiele mit Grafiken und Sound in Java.

Insgesamt sind für dieses Projekt 600 Mannstunden eingerechnet. Dies entspricht pro Person 120 Stunden in einem Zeitrahmen von 12 Wochen.

4.2 Materielles

Jedes Teammitglied verfügt über einen Laptop mit folgender Konfiguration:

- Eclipse mit Maven Plugin (zur einheitlichen Konfiguration)
- SVN Client mit Anbindung zum ZHAW-Repository (*rom.zhaw.ch*)
- L^AT_EX- Editor / Übersetzer

5 Risiken

5.1 Verwertungsrisiken

5.1.1 Nachfrage

Nach einer kurzen Suche mit Google ergeben sich etliche Treffer für Tower Defense Spiele in etlichen Formen. Zum Beispiel towerdefence.net listet 196 unterschiedliche Tower Defense Implementationen (Stand: 01.10.2010). Ausserdem unterstützen mehrere Multiplayer Strategy Spiele einen Tower Defense Modus (Warcraft 3, Starcraft 2, ...).

Ein weiteres Problem stellt die Gewinngenerierung dar. Die meisten Tower Defense Spiele die sind entweder eine Erweiterung zu einem bereits existierenden Spiel oder können Gratis heruntergeladen werden.

5.2 Projektrisiken

5.2.1 Kosten

Für dieses Projekt werden pro Person 120 Arbeitsstunden geplant. Unsere Gruppe besteht aus 5 Personen, dies ergeben 600 Arbeitsstunden. Bei einem Stundenansatz von 60 Franken pro Stunde ergibt dies erwartete Entwicklungskosten von 36'000 Franken. Um diese Kosten innerhalb von 3 Jahren zu decken, müsste durchschnittlich 12'000 Franken pro Jahr eingenommen werden.

Wie bereits im Abschnitt [Nachfrage](#) beschrieben, wird es in diesem Markt schwierig, ein neues Produkt zu präsentieren und die Entwicklungskosten zu kompensieren.

5.2.2 Projektumfang

Der Projektumfang ist für die kurze Zeit von einem Semester gross gehalten. Deswegen kann es schwierig werden das ganze Produkt mit allen vorgesehenen Features in der kurzen Zeit zu implementieren.

Stellt sich im Verlaufe des Projektes heraus, dass das Projekt nicht im gegebenen Zeitrahmen zum vollständigen Abschluss gebracht werden kann, werden gewisse Features weggelassen.

5.3 Allgemeine Risiken

5.3.1 Personalausfall

Es besteht das Risiko, dass ein oder mehrere Projektmitglieder schwer erkrankt oder durch Militäreinsätze für einen beträchtlichen Zeitraum während dem Projekt ausfallen.

In dieser Situation müsste der Projektumfang gekürzt werden, damit im gleichen Zeitraum das Projekt zu einem Abschluss gebracht werden kann.

5.3.2 Hardwareausfall

Während der ganzen Projektdauer werden mehrere Computer eingesetzt. Während der Entwicklung der Software besteht eine Möglichkeit, dass eine Harddisk ausfällt und mit dieser Daten, welche noch nicht ins SVN-Repository hochgeladen wurden.

Während dem Projekt müssen regelmässig Backups von dem SVN-Repository erstellt werden und jeder ist dafür verantwortlich, dass die persönlichen Änderungen regelmässig und in kurzen Abständen ins SVN-Repository hochgeladen werden.

6 Grobplanung

Die Gesamtlaufrzeit des Projekts CCTD wird durch externe Vorlagen festgelegt. Das Projekt startete am 24.09.2010 und endet gemäss Abgabetermin am 10.12.2010. Dies ergibt eine Gesamtlaufrzeit von 12 Wochen. Geplant ist nur ein Release mit der gesamten geplanten Funktionalität des Spiels. Die Planung erfolgt gemäss UP mit einer Iterationsdauer von 2 Wochen:

Phase	Woche	Iteration	Ziele
Inception	38	1	Projektidee gefunden, Projekt Ausrichtung definiert, Gruppenbildung, Projektleader definiert
	39		Projektskizze erstellt, Entwicklungs Umgebung definiert, Projektskizze Präsentation erstellt
Elaboration	40	2	Abgabe Projektskizze, Präsentation Projektskizze , Spielprinzip definiert, Applikations Grundgerüst erstellt
	41		Anwendungsfälle detailliert ausformuliert, Rest identifiziert und priorisiert, 1. Entwurf Domänenmodell
	42	3	Use Cases fertig, Supplementary Specification fertig falls benötigt, System-Sequenzdiagramm erstellt, Glossar erstellt, Vision fertig, Domänenmodell erstellt, Erste Java Tests bezüglich Client / Server verhalten; welche Daten werden übertragen
	43		Abgabe Analyse, Präsentation Analyse , erste Version UML Klassendiagramm, erste Version Architekturdokumente
Construction	44	4	UML Klassendiagramm, UML Interaktionsdiagramm für ausgewählte Systemoperationen, Architekturdokumente fertig, Implementation des UML Klassendiagramms in Java, Test der Zusammenhänge: ev. Klassendiagramm überarbeiten
	45		UML Klassendiagramm, UML Interaktionsdiagramm für ausgewählte Systemoperationen, Architekturdokumente fertig, Design Präsentation fertig, erste Implementation der Klassen inklusive Methoden und Eigenschaften
	46	5	Abgabe Design, Präsentation Design , Implementation der einzelnen Klassen, Client / Server Testing, Implementation einer simplen Spieloberfläche mit ersten Elementen (ein Turm, ein Gegner, vier Spieler).
	47		Simple Spieloberfläche fertig, Testing mit Client / Server, Ein Spiel von A-Z Durchspielen sollte ab hier möglich sein, Spiel erweitern auf unterschiedliche Türme, Unterschiedliche Gegnertypen, Unterschiedliche Gegnerwellen, Testing der ganzen Klassen
Transition	48	6	System Test fertig, Schlusssdokumentation erstellen, Schlusspräsentation erstellt, Spiel fertig stellen, Alle Spielelemente fertig, Spiel Ballancing (Schwierigkeit: nicht zu einfach, nicht zu schwer)
	49		Projekt Ende, Schlusspräsentation, Alles abgeben

7 Kundennutzen

Der grösste Nutzen des Kunden ist der Spassfaktor, den dieser bei diesem Spiel haben wird. Der grosse Unterschied zu anderen bekannten Tower Defense Spielen ist die zwingende Zusammenarbeit der unterschiedlichen Spieler durch upgraden der Towers der Mitspieler, Geld spenden und gemeinsame Towers bauen. Ein weiterer wichtiger Vorteil von CCTD ist die Möglichkeit sich und seine Mitspieler retten zu können, falls viele Creeps bei einer Runde nicht besiegt wurden und die nächste Runde bereits beginnt.

Das grundsätzliche Spielprinzip, Towers bauen und Creeps mit diesen töten, ist das Gleiche wie bei anderen Tower Defense Spielen.

8 Wirtschaftlichkeit

Wir haben grundsätzlich drei Möglichkeiten durch unser Produkt einen Gewinn zu erzielen. Die erste Möglichkeit wäre es, unser Spiel an Unterhaltungsplattformen zu verkaufen. Die zweite Möglichkeit wäre es unser Produkt direkt zu vertreiben.

Die dritte Variation, die wir schlussendlich anstreben, ist es, unsere eigene Plattform zu betreiben und Einnahmen mittels Werbebanner und ähnlichem zu erzielen.

Unser Zielmarkt sind primär Jugendliche und Junge Erwachsene, da das Spielprinzip in dieser Kategorie am ehesten bekannt ist.

Wir spekulieren darauf, dass sich unser Spiel hauptsächlich durch Mund-zu-Mund Propaganda verbreitet.