

# Softwareprojekt II

## **SCHLUSSBEWERTUNG**

### **Anaphylactic Engagement**

Ein 3D Spiel von

Kai Böttcher - 714846

Juljan May – 758282

Felix Moser - 756878

Nella Klopotek v. Glowczewski – 757988

Tim Strehlow - 749991

## **Planung allgemein**

Im zweiten Semester unseres Software-Projekts hat sich herausgestellt, dass die Anforderungen aus SWP I für die Spieleentwicklung relativ ungeeignet sind. Da speziell für ein 3D-Spiel sehr viel technischer Aufwand vor der Entwicklung des eigentlichen Spiels notwendig ist, hätte das erste Semester genutzt werden müssen, weniger Zeit auf den Entwurf der Spiellogik zu verwenden, als auf den Start mit der Implementierung des Engine-Pakets. So hätte das zweite Semester mehr dafür genutzt werden können, die geplante Spiellogik umzusetzen. Somit hätte das Spiel komplexer gestaltet werden können. Im aktuellen Zustand umfasst es nur die wesentlichen Interaktionsmöglichkeiten. Spielmodi und weitere Features, die das Spiel erst interessant machen, waren zeittechnisch nicht möglich zu implementieren.

## **XNA und das Arbeiten damit**

XNA bietet dem unerfahrenen Entwickler die Möglichkeit relativ schnell ein ordentliches Spiel zu entwickeln. Trotzdem bietet es auch genug Freiheit um mit damit auch professionelle Projekte zu realisieren.

Zusätzlich aber war die Umsetzung der Menüstruktur komplexer als erwartet, da XNA keine Control-Klassen zur Verfügung stellt, die für unsere Applikation geeignet gewesen wären. So mussten die ganzen Controlklassen von Grund auf selbst implementiert werden. So ist jedoch im Engine-Paket ein GUI-Framework entstanden, womit in Zukunft einem neuen XNA-Spiel wesentlich schneller eine Menüstruktur hinzugefügt werden kann. Zusätzlich haben wir so Einblick in einen Bereich erhalten, der normalerweise von der Entwicklungsumgebung bereitgestellt wird.

## **SVN als Versionsverwaltungssystem**

Das Arbeiten mit SVN als Versionsverwaltungssystem kam uns in dem Projekt sehr zur Hilfe. So konnte problemlos an vielen Stellen parallel gearbeitet werden und auch Interaktionen innerhalb Klassen waren möglich.

## **Visual Studio als Entwicklungsumgebung**

Ist man Eclipse oder Eclipse-ähnliche Entwicklungsumgebungen gewohnt, ist Visual Studio im Gegensatz dazu relativ unkomfortabel. Importe (bei C# *using*) und Autovervollständigung werden hier nicht so benutzerfreundlich unterstützt. Auch Refaktorisierungen sind um einiges umständlicher gestaltet.

### **Fazit**

Als Fazit können wir sagen, dass sich unser Projekt als relativ unkonventionell im Gegensatz zu üblichen Softwareprojekten herausgestellt hat. Somit haben viele Richtlinien nicht auf das Projekt gepasst und mussten angeglichen werden. Dennoch haben wir viel gelernt und einen interessanten Einblick in die Spieleentwicklung erhalten.