

Maturarbeit von Alex Türler und Fabian Lyck

Alex Türler, Fabian Lyck

15. Januar 2014

1 Zusammenfassung

Thema dieser Maturarbeit ist die Entwicklung eines serverbasierten Multiplayer-Spiels. Schauplatz ist ein virtuelles, dreidimensionales Sonnensystem, in welchem physikalische Gesetze simuliert werden. Der Spieler steuert ein Raumschiff, welches er individualisieren kann, wie zum Beispiel Antriebe an verschiedenen Stellen befestigen. Wenn möglich soll dem Spieler eine Skriptsprache zur Verfügung stehen, mit der Prozesse automatisiert werden können.

Das Spiel soll einer Mischung aus Action und Strategie entsprechen. Die Idee ist Ressourcen zu sammeln und sein Schiff zu verbessern, um sich mit den anderen Spielern zu messen, wobei das Spiel jedoch auf keinen Fall in einem langweiligen Abbauwettbewerb enden soll, sondern dies auch Teil der Action ist.

2 Aufgaben

2.1 Klassendesign

Das Klassendesign umfasst die Definition von Variablen und Klassen sowie deren Verwendung. Vor der eigentlichen Implementation müssen hier die Details festgelegt werden. Bei Unity muss man darauf achten, dass alle Klassen, die Teil eines Unity GameObjects sind, von MonoBehaviour erben.

Zuständig: FABIAN

2.2 Implementierung

Bei der Implementierung handelt es sich um die gesamte Programmierarbeit, welche sich nach dem Klassendesign und Applikationsprotokoll richtet. Der Code wird vorraussichtlich in Unity-JavaScript geschrieben.

Zuständig: FABIAN

2.3 Applikationsprotokoll

Das Applikationsprotokoll beschreibt die Kommunikation zwischen Client und Server und beinhaltet bestimmte Kommunikationscodes. Diese Kommunikation besteht in diesem Fall grösstenteils aus Remote Procedure Calls. Unsere Arbeit besteht daraus, diese Liste zu erstellen und die Hintergründe zu erklären.

Zuständig: FABIAN

2.4 Dateisystem

Das Dateisystem welches vom Server benutzt wird, um die Informationen über vorhandene Spieler zu speichern, ebenfalls können damit Einstellungen des aktuellen Spielers gespeichert werden. Die Syntax ähnelt lua. Dafür wird im Projekt eine Klasse erstellt.

Zuständig: FABIAN

2.5 Dokumentation

Eine Erklärung der verwendeten Funktionen und Klassen inklusive eines Klassendiagramms. Sie wird in Form eines Javadocs erstellt.

Zuständig: ALEX

2.6 Erarbeitung von Theorien

Wir beschreiben die Theorien welche in der Maturarbeit zum Lösen von Problemen nötig sind. Zum Beispiel die Gravitationsberechnung bzw. -approximation, bestenfalls mit einem Octtree, oder auch die Projektion von dreidimensionalen Objekten auf eine zweidimensionale Textur.

Zuständig: ALEX, FABIAN

2.7 Design GUI

Die GUI ist das Interface des Spielers, bei dieser gilt es darauf zu achten, dass Sie übersichtlich und verständlich bleibt, ohne jedoch an Funktionen einzubüssen. Die Arbeit besteht hier aus zwei Teilen, zuerst muss eine Skizze erstellt werden, danach eine passende GUI implementiert werden.

Zuständig: ALEX, FABIAN

2.8 Design 3D

Hierbei handelt es sich um das Design von dreidimensionalen Objekten, die im Spiel vorkommen. Darunter fallen Raumschiffe, Planeten und Asteroiden. Dies besteht aus der Modellierung und Texturierung der Objekte.

Zuständig: ALEX

2.9 Testing

Ein ausführliches Testen der Software ist in jedem Stadium nötig. Dies verhindert spätere Probleme mit Programmierfehlern, die zu spät erkannt werden. Ebenso dient es der Qualitätssicherung. Es wird ein Plan mit nötigen Tests sowie deren Ausführungsweise erstellt und daraufhin durchgeführt.

Zuständig: ALEX

2.10 Terminplan

Der Terminplan muss von Beiden regelmässig nachgeführt werden, um Verzögerungen rechtzeitig zu erkennen. Der voraussichtliche Plan ist diesem Text angehängt.

Zuständig: ALEX, FABIAN

2.11 Vorläufige Themen des Maturberichts

1. Introduction
 - 1.1 Vorwort
 - 1.2 Ideas
 - 1.3 Der Vorgänger
 - 1.4 Versioning
 - 1.5
 - 1.6
2. Implementation
 - 2.1 Engine, Language, Environment
 - 2.2 Program Execution
 - 2.3 Gravitationsberechnung
 - 2.4
 - 2.5
3. Application Design
 - 3.1 Class Diagram

- 3.2 Network Communication
 - 3.3
 - 3.4
- 4. Graphic Design
 - 4.1 Modeling
 - 4.2 Texturing
 - 4.3 Rendering
 - 4.4 GUI
 - 4.5
 - 4.6
- 5. Testing
 - 5.1 Einleitung
 - 5.2 Tests
 - 5.3
 - 5.4
- 6. Attachments
 - 6.1 Documentation
 - 6.2 Schedule
 - 6.3 Sourcecode
 - 6.4 References
 - 6.5
 - 6.6

2.12 Versionsverwaltung

Zur Versionsverwaltung wird git benutzt. Dies verhindert Versionskonflikte mit verschiedenen Versionen in der Schule und daheim. Im Moment ist das Projektverzeichnis einzusehen unter <https://github.com/DeinFreund/Matur>. Dieses wird nur als Arbeitsverzeichnis genutzt.