LV MP1: Projektbewertung Semester: **Sommer 2011**

Team:

1. Name:	**Rabe**	Vorname:	**Jan**	Matr. Nr.: **766212**
2. Name:	**Wallroth**	Vorname:	**Tom**	Matr. Nr.: **768068**
3. Name:	**Börner**	Vorname:	**Tilman**	Matr. Nr.: **768148**

Projekt: Android-Spiel „Slack'n'Hay“ **Zusatzleistung(en):** keine

this space intentionally left blank

Selbsteinschätzung des Teams:

Wie auch im Multimedia-Engineering Projekt hatten wir uns die Programmierung eines Spiels vorgenommen, und auch hier war klar, dass wir unsere Ziele einschränken mussten: Für ein vollständiges Spiel war der uns zur Verfügung stehende Zeitrahmen zu knapp, deshalb hatten wir uns lediglich die Grundelemente des uns vorschwebenden Spiels ins Pflichtenheft geschrieben.

Als Spielprinzip haben wir uns für ein sogenanntes „**Hack-and-Slay**“ entschieden, dessen Reiz darin besteht, durch Gebrauch einer Spielfigur und ihres Schwerts unter einer Vielzahl von Gegnern möglichst keinen übrig zu lassen. Das Besondere daran sollte sein, einserseits eine Art grafisches Mash-up aus verschiedenen uns am Herzen liegenden klassischen Spielen herzustellen; und andererseits, das Ganze für **Android**-Plattformen zu entwickeln.

Unser **anvisiertes Endprodukt** sollte aus einer Spielwelt bestehen, in der eine mittels Touchscreen kontrollierte Spielfigur gegen andere Spielfiguren kämpfen kann, die mittels einer simplen künstlichen Intelligenz offensiv vorgehen. Alle Spielfiguren sollen sterben können und animiert sein; für elementare Ereignisse soll eine Sound-Ausgabe stattfinden; die Spielwelt beinhaltet außerdem dekorative Elemente, die zusätzlich Bewegungshindernisse darstellen.

All das haben wir erreicht.

Dazu mussten einige Herausforderungen überwunden werden, die hier nur kurz angerissen werden sollen. Für weitere Details und einen Überblick über unsere Lösungen verweisen wir auf die **Dokumentation**, die diesmal nicht in HTML-, sondern in PDF-Form auf der abgegebenen CD enthalten ist.

Die erste Schwierigkeit, mit der wir uns konfrontiert sahen, resultiert aus der starken **Ressourcenbeschränkung**, die den heutigen mobilen Endgeräten noch zu eigen ist und die sich in besonderen Eigenschaften der auf Android laufenden **Dalvik Virtual Machine** äußert. Entsprechend schlägt sie sich auf alle Bereiche des Spiels nieder, sodass wir allerorten oft nicht die uns geläufigen Programmierpraktiken anwenden konnten, sondern andere, der Situation angepasste gefragt waren, die wir uns erst aneignen mussten. Natürlich war es ebenso erforderlich, dass wir uns mit den Konventionen für Android-Applikationen vertraut machten.

Ein weiteres neues Feld war für uns der Umgang mit **Touchscreen-Eingaben**. In diesem Zusammenhang haben wir ein sinnvolles Interface zur Steuerung der Spielfigur entwickelt und als Backend dafür ein System, mittels dessen wir entsprechenden Input **kontextabhängig parsen** und in Folge mit Semantik versehen können.

Wichtig war ebenfalls die Entwicklung einer **Grafikengine mit OpenGL ES**, mit der wir das Spiel auf den mobilen Geräten performant visualisieren können, insbesondere mittels **Animationssequenzen**, die aus Einzelbildern generiert wurden. Das Ergebnis ist eine grafische Spielewelt aus isometrischer Perspektive, in der sich Figuren flüssig bewegen können und die Kamera frei positioniert werden kann, ohne dass die **Illusion von Räumlichkeit** einer eigentlich 2-dimensionalen Welt zerstört würde.

Für die **logische Repräsentation** dieser Welt haben wir ein **flexibles Raster**, das Grid, entwickelt, das seine interne Welt aus diskreten Zellen transparent übersetzt in die Koordinaten einer theoretisch unendlichen und kontinuierlichen Außenwelt. So kann es leicht an ein mit Fließkommawerten operierendes Grafiksystem angebunden, aber auch an anderer Stelle wiederverwendet werden.

Um Spielewelten leicht definieren zu können, haben wir einen grafischen WYSIWYG **Level-Editor** gebaut.

Unseren Spielobjekten haben wir ein **Framework aus modularen Komponenten** erschaffen, das wie die anderen erwähnten Elemente auch in weiteren Projekten Verwendung finden kann. Vor allem aber macht es möglich, dass wir das Spiel durch Schaffen neuer und Rekombination bestehender Komponenten **leicht erweitern** können.

All dies hat uns viel Mühe gekostet, mitunter Frustration beschert, aber auch viel Spass gemacht und vor allem, wie sicherlich erkennbar geworden ist, unser Wissen in vielen technischen, theoretischen und praktischen Aspekten von Multimedia-Anwendungen (speziell in Form von Android-Applikationen) bedeutend erweitert. Insgesamt denken wir, dass sich das von uns geschaffene Ergebnis im Rahmen dieser Lehrveranstaltung sehr gut sehen lassen kann, und daher schlagen wir eine eben solche, *sehr gute* Bewertung unserer Arbeit vor.

Technologien:

- Java (optimiert für Dalvik-VM)
- Android SDK
- OpenGL for Embedded Systems

Werkzeuge:

- Eclipse mit Android-SDK und
- DDMS (Dalvik Debug Monitor Server)
- Android-Emulation, div. Android-Geräte
- Android Developer Bridge
- Photoshop
- SVN

Architektur- und Designmuster:

- Factory-Klassen zur Erzeugung von Spielobjekten
- Singleton zur Erzeugung von UIDs
- Objekt Pool Pattern für die Wiederverwendung von Spielfiguren
- Composite Pattern für das modulare Zusammenfügen der Spielobjekte
- Monitor Object Pattern für die Synchronisation des Spielthreads und des Renderthreads

Wiederverwendbare Komponenten:

- OpenGL-ES Grafikengine, optimiert für Android Dalvik VM
- kontextbasiertes Touchscreen Input-System für Android-Geräte
- flexibles Komponenten-Framework für Spielobjekte (mit Grafik- und Soundkomponenten für Android)
- 2-dimensionales Zellenraster („Grid“) zur diskreten Strukturierung eines Spielfelds
- Statemachine zur repräsentation von Spielobjektzuständen

Welche externen Snippets, Klassen, Frameworks, Komponenten wurden verwendet?

- Android API

Was wurde (im Vergleich zum Pflichtenheft) nicht realisiert, was ist instabil oder unzureichend getestet?

Nicht realisiert:

- Punktevergabe an Spieler
- Speicherung des Spielzustands

Instabil:

- unsere Psyche

Welche Features des Projekts sind innovativ?

- Entwicklung auf mobiler Android Plattform, Steuerung per Touchscreen und Gestenerkennung

Hiermit gestatten wir Prof. Dr. Targo Pavlista, dem Fachbereich VI und der BHT Berlin die Video-Dokumentation unseres Projekts zu veröffentlichen und zum Download bereit zu stellen.

Berlin, den 14.7.2011

Empfohlene Projektnote: **1,0** Empfohlene Note für Zusatzleistung: n/a

Folgende kurze Textbeschreibung sollte neben dem Icon unseres Projektes sichtbar sein (was gemacht wurde, welche Technologien):

Ein Hack'n'Slay im Retro-Look für Android-Geräte (mit OpenGL ES 2.0, ab Android 2.2).

Wir haben ein Icon (Format max. 400 * 400 Pixel) selbst erstellt und mit auf die DVD/CD gepackt (**ja**, ~~nein~~):