



LFE Medieninformatik

_____ **LMU**
Ludwig _____
Maximilians _____
Universität _____
München _____

Projektaufgabe
Multimedia-Programmierung
Sommersemester 2006

LMU München
LFE Medieninformatik



Outline

- Aufgabenstellung ←
- Werkzeugunterstützung und Hilfsmittel
- Projektablauf



Thema: Jump&Run-Spiel



<http://www.harveycartel.org/metanet/n.html>



Funktionale Anforderungen: Spiel

- Level:
 - Die Dimensionen des Levels sollen ungefähr einem Quadrat entsprechen
 - Zu jedem Zeitpunkt ist nur ein Teil-Ausschnitt des Levels sichtbar. Der sichtbare Ausschnitt bewegt sich entsprechend der Bewegung der Spielfigur ("Kamera").
 - Es befinden sich feste Objekte (z.B. Plattformen, Wände) im Level, auf die die Spielfigur springen kann.
- Spielfigur:
 - Animiert; kann stehen, laufen und springen. Jeder Zustand wird durch eine entsprechende Animationen dargestellt.
 - Befindet sich die Spielfigur in der Luft, fällt sie nach unten. Die Spielfigur kann nicht durch Plattformen oder die Wände des Levels hindurch fallen.
- Interaktive Objekte im Level:
 - Punkte-Objekte, die von der Spielfigur eingesammelt werden können und den Punktestand der Spielfigur erhöhen
 - Verschiedene Bonus-Objekte, die von der Spielfigur eingesammelt werden können und die Fähigkeiten der Spielfigur erhöhen (z.B. höheres Springen)
 - Verschiedene Hindernisse (z.B. feste und bewegliche), die die Spielfigur aufhalten oder ihr Schaden zufügen (z.B. Wegnahme von Fähigkeiten oder Rücksetzen an die Startposition)



Funktionale Anforderungen: Gesamtanwendung

- Einspieler-Modus:
 - Festlegen eines sinnvollen Spielziels
 - z.B. Erreichen des Ziels nach Einsammeln aller Bonus-Objekte
 - Z.B. Erreichen möglichst vieler Punkte in einer bestimmten Zeit
- Mehrspieler-Modus:
 - Für 2-4 Spieler, die sich gleichzeitig im gleichen Level befinden
 - Chat-Funktion in den Pausen zwischen den einzelnen Levels
 - Festlegen eines sinnvollen Spielmodus. Der Spielmodus darf so festgelegt werden, dass keine ganz exakte Synchronisation der Clients notwendig ist
 - z.B. jeder Spieler erhält abwechselnd einen Zeitraum, in dem er Punkte-/Bonus-Objekte einsammeln darf; die anderen Spieler können währenddessen nur ihre Position im Level verbessern, oder z.B. Hindernisse für die anderen Spieler setzen
- Die Anwendung besteht neben dem Spiel selbst aus
 - Startmenü
 - Impressum: Angabe der Autoren und der Lehrveranstaltung
 - Highscore: wird lokal auf Festplatte gespeichert
 - Anleitung/Hilfe
 - Optionen



Allgemeine/nicht-funktionale Anforderungen

- Schwerpunkt sind Animation und Interaktion
- Einbindung von Sound
- Graphische Gestaltung anhand durchgängig umgesetzter Idee/Thema
- Gute Steuerbarkeit der Spielobjekte; Spiel soll auch tatsächlich spielbar sein.
- Änderungsfreundlichkeit der Anwendung
- Das Ergebnis soll ins Web gestellt werden können
 - Keine Verwendung Copyright-geschützter Objekte (z.B. Logos, Sound)
 - Lauffähigkeit auch ohne Multiplayer-Server (oder sonstige Zusatzsoftware)
 - Einfache Bedienbarkeit ohne Einarbeitungsaufwand
 - Fehlerfreiheit, Robustheit, ...



Outline

- Aufgabenstellung
- Werkzeugunterstützung und Hilfsmittel ←
- Projektablauf



ActionScript-Editierung

- Eclipse mit Plugins unter Linux installiert:
`/soft/bin/eclipse-ide-3.1.2-mmp`
- Plugin *mtasc* (<http://mtasc.org/>): externer Compiler für ActionScript
(aufzurufen über `/soft/bin/mtasc`)
- Plugin *ASDT* (<http://www.asdt.org/>): Eclipse Editor für ActionScript.
 - Einstellungen in Eclipse unter *Window->Preferences*:
 - *mtasc* als Compiler angeben `/soft/bin/mtasc`
 - ActionScript Core-Klassen angeben: mit *mtasc* mitgeliefert (enthält nicht Flash-Komponenten) `/soft/IFI/lang/mtasc-1.12/iX86-unknown-linux/std`
 - “Bugs” unter <http://sourceforge.net/projects/aseclipseplugin/>
- Plugin *Flashout* (<http://www.potapenko.com/flashout/>): zum direkten Betrachten der kompilierten SWF-Dateien in Eclipse
 - Hier nochmal gleiche Einstellungen wie bei *ASDT* notwendig
- Tutorial zur Verwendung der Kombination *Eclipse + mtasc + ASDT + flashout* unter: http://theresidentialien.typepad.com/ginormous/2005/10/eclipse_mtasc_f.html



Testwerkzeuge

- *Autotestflash* (<http://www.osflash.org/autotestflash>): Automatisches simulieren von Benutzer-Aktionen in Flash-Anwendungen
- *ASUnit* (<http://www.asunit.org/>): Unit-Test Framework für ActionScript



Versionsmanagement

- Versionsmanagement-System *Subversion* (SVN)
- Einrichtung eines Repositories durch RBG unterstützt:
<http://www.rz.ifi.lmu.de/Dienste/Subversion>
- Terminalserver: Subversion-Clients installiert => Subversion-Befehle im Kontextmenü verfügbar
- Generell:
 - Empfehlung: Alle Dateien in Subversion verwalten
 - „Checkout“ zur Bearbeitung einer Datei
 - Häufig „Update“ zum aktualisieren verwenden
 - Nur getestete Änderungen in das System einstellen mit „Checkin“
 - Kleine Änderungen vornehmen (auf aktuellster Version!), testen und wieder einchecken
- Zugang zum Repository auch für den Betreuer des Teams mit einrichten



Socket-Server für Multiplayer-Modus

- Electroserver (<http://Electrotank.com/electroserver>): Kommerzieller Server mit kostenloser Version (für bis zu 20 Verbindungen)
- Eigenen einfachen Socket-Server erstellen; in Java sehr einfach zu erstellen. Tutorials im Web, z.B. <http://java.sun.com/docs/books/tutorial/networking/sockets/index.html>



Strukturierung der Anwendung

- Gesamter Code in externe ActionScript-Klassen (nur Instanziierung der Hauptklasse in Zeitleiste)
- Aufteilen in mehrere FLA-Dateien, dynamisches Laden von SWFs (Klasse *MovieClipLoader* und Methoden von *MovieClip*)
- Model-View-Controller Paradigma aus Vorlesung (Kap. 1.3)
- Vordefinierte Klassen für Observer-Pattern:
<http://www.moock.org/eas2/>
(Download example code => Kap. 16)



Outline

- Aufgabenstellung
- Werkzeugunterstützung und Hilfsmittel
- Projektablauf ←



Betreuung

- Betreuer: für Fragen und für Rolle des Kunden
- Regelmäßige Treffen
 - Vorstellung des Zwischenergebnisses des Teams
 - Kurzbericht jedes Teammitglieds über eigene Teilaufgaben
 - Besprechung von Problemen und des weiteren Vorgehens

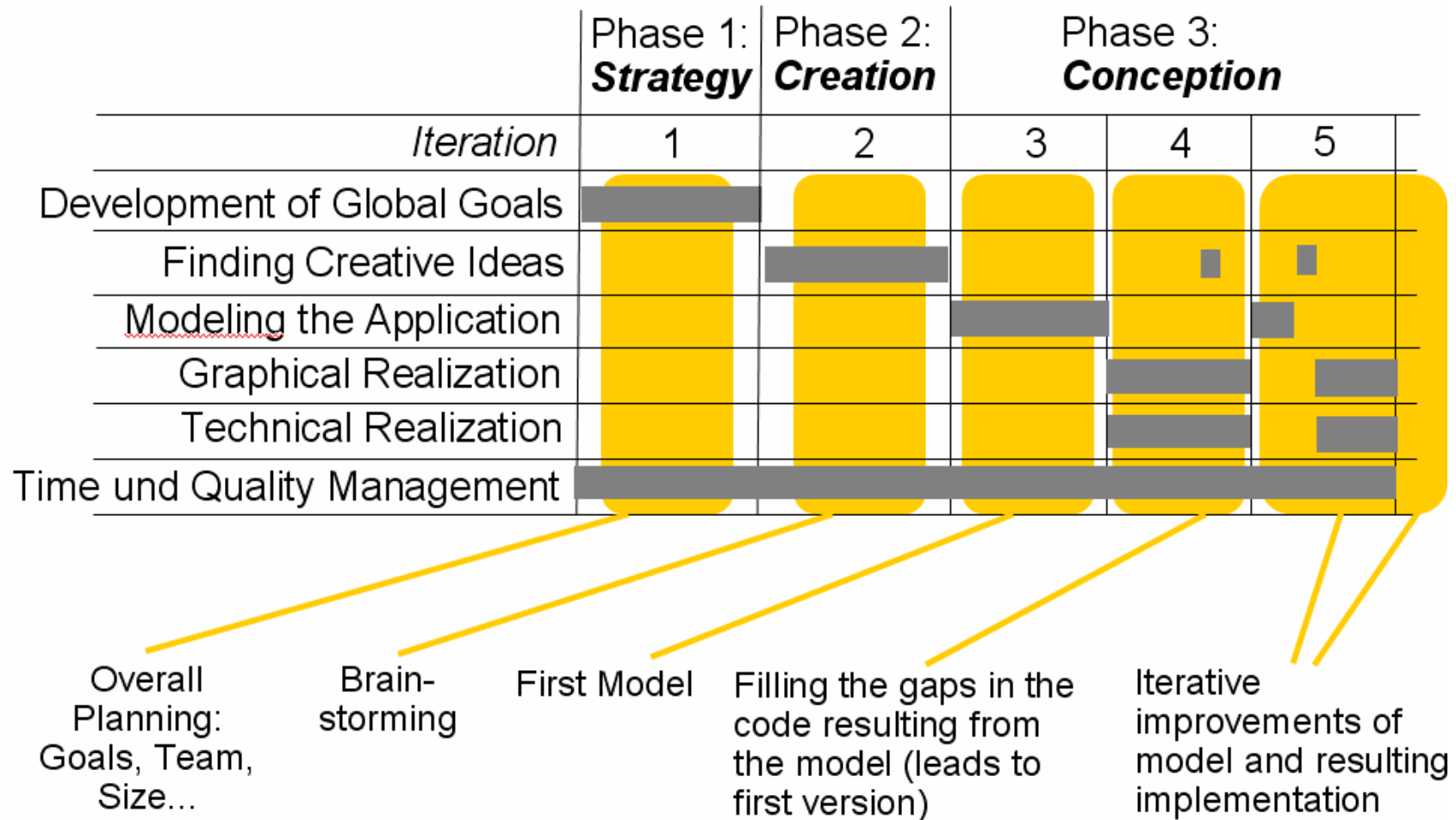


Vorgehen: Phasen und Ergebnisse

- Team: Phasen, Meilensteine
 1. Team-Organisation
 2. Einarbeitung; Erstellung von Prototypen
 - Graphisch: Gestalterisches Thema des Spiels
 - Funktional: Bewegung der Spielfigur, Kollisionserkennung
 3. Entwurf der Anwendung (Architektur, Strukturierung)
 - MML-Modell der Anwendung
 4. Implementierung und Test in mehreren Iterationen
 - Versionen der Anwendung
- Pro Teammitglied abzugeben (Material wird noch bereit gestellt):
 - Ausgefüllter Stundenbogen am Ende jeder Woche
 - Ausgefüllter Fragebogen am Ende der Projektaufgabe



Vorgehen: Übersicht





Nächste Schritte: Phase 1

- Team-Organisation bis nächsten Montag, 22.5.
- Dem Betreuer einzusenden: Textdatei mit Kurzinfo zu
 - Team,
 - Teamleiter (Ansprechpartner für Betreuer),
 - Team-Kommunikation (wie ist vorgesehen, zu kommunizieren),
 - Team-Treffen (wann und wo wird sich das Team treffen),
 - SVN-Zugang



Nächste Schritte: Phase 2

- Sammlung kreativer Ideen (z.B. mittels Brainstorming)
 - Thematische und gestalterische Umsetzung
 - Konkrete Spielregeln, Features, ...
- Einarbeitung in notwendige Spiele-Logik/-Physik
 - Kollisionserkennung
 - Steuerung der Kamera
 - Verhalten der Spielfigur, ...
- Erstellung von Prototypen zum Experimentieren
- Bis 2. Juni: Vorführung der Prototypen, d.h.
 1. Ausgewählter aussagekräftiger Grafiken und Animationen zur Illustration des gestalterischen/thematischen Konzepts
 2. Vorführung von ausgewählten Prototypen zur Demonstration der erarbeiteten Algorithmen und der Spiel-Physik
- Erstes Treffen mit Betreuer nach Bedarf/Absprache; spätestens bei Abschluß der Phase 2



Phase 3: Modellierung

- Einführung in der Vorlesung und der Übung!



Weiteres Material

- Spiele-Programmierung:
 - Buch *Flash MX 2004 Games Most Wanted* (in der Bibliothek)
 - Kollisionserkennung: <http://www.harveycartel.org/metanet/tutorials.html>
 - Weitere:
 - <http://www.gamedev.net>
 - <http://www.gamasutra.com>
 - <http://www.devmaster.net>
- Große Sammlung von freien Flash-Werkzeugen:
<http://www.osflash.org/>
- Flash-Forum: <http://www.flashforum.de/>
- Wiki des Lehrstuhls zum Austausch von Links, Tips, Hinweisen und Fragen (in Kürze):
<https://wiki.medien.ifi.lmu.de/view/Main/MMPPProjektaufgabeSS06>