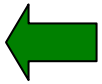
Projektaufgabe Multimedia-Programmierung Sommersemester 2007

LMU München

LFE Medieninformatik

Outline

Ludwig Maximilians Universität München

- Aufgabenstellung 
- Werkzeugunterstützung und Hilfsmittel
- Projektablauf

Thema: Minigolf-Spiel

Gestaltung

- Festlegung eines Themas für das Spiel und die Spielumgebung
- Wird durchgehend verwendet (Menüs, Hintergründe, Hindernisse, usw.)
- Passende festgelegte Farbkonzepte, Schriftzüge, Sounds, Benutzungsoberfläche, Logos, usw.
- Umfangreiche, verschiedenartige Bahnen; keine direkte Anlehnung an reale (Standard-)Minigolfbahnen

Funktionale Anforderungen: Spiel (1/2)

Ludwig Maximilians Universität München

1. Schlagen des Balles: Konzeption eines geeigneten Steuerungs-Mechanismus für angenehme Benutzung und sinnvollen Schwierigkeitsgrad (Spielspaß)
2. Realistische Ballphysik (z.B. Verhalten bei Kollisionen, Gravitation usw.)
    1. Verschiedene Bälle (z.B. hart, weich)
3. Kamera-Perspektive:
    1. Blickwinkel von schräg oben („Pseudo 3D“) oder ersatzweise von oben
4. Loch (oder sonstiges Ziel des Balles)
    1. Bei zu hoher Geschwindigkeit kann Ball nicht in das Loch fallen

Funktionale Anforderungen: Spiel (2/2)

5. Verschiedene Typen von Bahn-Elementen
    1. Einfache ebenerdige Hindernisse, die Ball abprallen lassen
    2. Hindernisse, die Ball verlangsamen (z.B. Sand, Wasser)
    3. Hindernisse, mit Steigung (z.B. Hügel, Schanzen, Rampen)
    4. Elemente, die Ball an andere Stelle bringen (z.B. Lifte, Rohre, Teleporter)
    5. Elemente, zur Beschleunigung des Balls (z.B. Rollbänder, Federn)
    6. Bewegliche Elemente (z.B. Schranken, Krabbeltiere)
    7. Hindernisse, die Ball von der Bahn werfen bzw. Strafpunkte kosten (z.B. Wassergraben, Bomben, ballfressende Wesen)

Funktionale Anforderungen: Gesamtanwendung

Ludwig Maximilians Universität München

1. Einspieler-Modus:
    1. 12 Bahnen; jedes Teammitglied soll (mindestens) 2 Bahnen entwerfen
2. Die Anwendung besteht neben dem Spiel selbst aus
    1. Startmenü
    2. Impressum: Angabe der Autoren und der Lehrveranstaltung
    3. Highscore: für Einzelspielermodus; wird lokal auf Festplatte gespeichert
    4. Anleitung/Hilfe
    5. Optionen

Funktionale Anforderungen (4/4): Mehrspielermodus

Ludwig Maximilians Universität München LMU

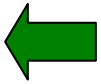
1. Verwendung eines einfachen Standard-Socket-Servers (wie in der Übung); Anwendung darf keine spezifischen server-seitigen Funktionen erfordern
2. (Mindestens) 2-4 Spieler können gegeneinander Spielen
3. Chatfunktion
4. Vor jeder Bahn: Spieler dürfen Hindernisse auf die Bahn setzen
5. Festlegung des konkreten Spielprinzips. Vorschläge:
   - Sichtbare oder unsichtbare Hindernisse
   - Hindernisse können für alle, nur für die anderen, oder für einzelne Spieler gesetzt werden
   - Begrenzter Vorrat an Hindernissen für das gesamte Spiel (jeder Spieler entscheidet, an welchen Bahnen er Hindernisse für die anderen platziert)
   - Kaufen oder Ersteigern von Hindernissen, die platziert werden dürfen
   - Hinweis: zur einfacheren Implementierung darf das Spielprinzip rundenbasiert bleiben, d.h. es ist nicht notwendig, dass zeitkritische gleichzeitige Aktion beinhaltet sind

Allgemeine/nicht-funktionale Anforderungen

1. Schwerpunkt sind Animation und Interaktion
2. Festlegung eines festen, durchgehend verwendeten gestalterischen Themas (siehe Folie 4)
3. Einbindung von Sound
4. Gute Steuerbarkeit der Spielobjekte; Spiel soll auch tatsächlich spielbar sein (und idealerweise auch Spaß machen… ☺)
5. Änderungsfreundlichkeit der Anwendung (Struktur)
6. Benutzbarkeit, Fehlerfreiheit, Robustheit, ...
7. Das Ergebnis soll ins Web gestellt werden!
    1. Keine Verwendung Copyright-geschützter Objekte (z.B. Logos, Sound)
    2. Lauffähigkeit auch ohne Multiplayer-Server (oder sonstige Zusatzsoftware)
    3. Benutzung möglichst selbsterklärend

Outline

Ludwig Maximilians Universität München

- Aufgabenstellung
- Werkzeugunterstützung und Hilfsmittel 
- Projektablauf

ActionScript-Editierung

Ludwig Maximilians Universität München LMU

- Eclipse mit Plugins unter Linux installiert:
  Aufruf mit *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-Klasses 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://theresidentalien.typepad.com/ginormous/2005/10/eclipse_mtasc_f.html

Testwerkzeuge

Ludwig Maximilians Universität München

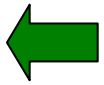
- *Autotestflash* (http://www.osflash.org/autotestflash): Automatisches simulieren von Benutzer-Aktionen in Flash-Anwendungen („Klick-Roboter“)
- *ASUnit* (http://www.asunit.org/): Unit-Test Framework für ActionScript

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
- Zugangänge zum Repository für Max Maurer und Andreas Pleuß mit einrichten

Outline

Ludwig Maximilians Universität München

- Aufgabenstellung
- Werkzeugunterstützung und Hilfsmittel
- Projektablauf 

Betreuung

Ludwig Maximilians Universität München

- Betreuer:
  - Rolle des “Kunden”,
  - Aber auch für Hilfestellung bei Problemen und gemeinsame Absprachen
- Regelmäßige Treffen: Team vereinbart individuell Termin mit dem Betreuer (unter Berücksichtigung der Meilensteine)
  - Vorstellung des Zwischenergebnisses des Teams
  - Kurzbericht jedes Teammitglieds über eigene Teilaufgaben
  - Besprechung von Problemen und des weiteren Vorgehens

Vorgehen: Phasen und Ergebnisse

Ludwig Maximilians Universität München

1. Team-Organisation.
   Ergebnis: Dokument mit wichtigsten Stichpunkten
2. Gestaltungskonzepte für die Benutzungsschnittstelle.
   Ergebnis: Horizontale Prototypen bzw. Mock-Ups (Bilder)
3. Spielephysik. Ergebnis:
   1. Funktionale (vertikaler) Prototypen
   2. Präsentationen der Prototypen aller Teams in der Übung
4. Entwurf der Gesamtanwendung (Architektur, Strukturierung).
   Ergebnis: Modell der Gesamtanwendung
5. Implementierung und Test in mehreren Iterationen.
   Ergebnis: verschiedene Versionen der Anwendung
6. Bereitstellung des Endergebnisses:
   1. Abgabe der zu Beginn der letzten Vorlesungswoche
   2. Präsentation aller Ergebnisse in der letzten Vorlesungsstunde (Kurzvortrag mit Folien)
   3. Bereitstellung der Web-fertigen Anwendung und des Quellcodes

Zu Phase 1: Team-Organisation

Ludwig Maximilians Universität München LMU

- Team-Organisation so bald wie möglich, spätestens Mittwoch, 16.5.
- 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 für Betreuer (+ für Andreas Pleuß)

Zu Phase 2: Gestaltungskonzept für die Benutzungsschnittstelle

Ludwig Maximilians Universität München

- Sammlung kreativer Ideen (Einsatz kreativer Arbeitsweisen sinnvoll, wie z.B. Brainstorming)

Dem Betreuer zu präsentieren:

- Thema des Spiels
- Titel des Spiels
- Mock-Ups:
  - Bilder von der Benutzeroberfläche
  - nachgebildete „Screenshots“ (Bildbearbeitungssoftware) unter Verwendung von existierenden Bildern und Screenshots
  - Sollen gestalterische Grundideen übermitteln
  - Möglichst mit wenig Aufwand erstellen
  - Punktuell bereits eigene gezeichnete Objekte
- Geplante konkrete Spielregeln des Spiels (in Stichpunkten fest spezifiziert)

Zu Phase 3: Spielephysik

Ludwig Maximilians Universität München

- Experimentieren mit Algorithmen zur sinnvollen Realisierung der Spielephysik
- Implementierung zum Ausprobieren als einfache Prototypen
- Graphische Gestaltung nur soweit zum Ausprobieren notwendig
- Präsentation in Form eines informalen Vortrags in der Übung (zusammen mit den anderen Teams)

Phase 4: Entwurf der Gesamtanwendung

Ludwig Maximilians Universität München

- Architektur und Gesamtstruktur der Anwendung festlegen
- Aufteilung wichtig, um parallele Bearbeitung durch verschiedene Team-Mitglieder zu ermöglichen (z.B. eine FLA-Datei kann nicht von mehreren Personen gleichzeitig bearbeitet werden!)
- Einführung in der Vorlesung und der Übung!
- Optional: Verwendung des Codegenerators
  - Vorteile:
    - Zeitersparnis durch automatische Erzeugung eines Codegerüsts aus dem Modell
    - Sinnvolle Strukturierung damit vorgegeben und muss nicht mühsam selbst erdacht und getestet werden
    - Erzeugte Codegerüste können trotzdem beliebig verändert und angepasst werden
  - Nachteile:
    - keine automatische Unterstützung zur Überprüfung der Modelle => Modelle müssen sorgfältig erstellt werden, um sinnvolle Codegenerierung zu erhalten
    - Kein “Round-Trip-Engineering”

Weiteres Material

Ludwig Maximilians Universität München

- Spiele-Programmierung:
  - Nächste Vorlesung
  - 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/MMPProjektaufgabeSS07