

Abschlussprojekt

Achtung: Das Projekt soll in Cocos2d-x umgesetzt werden.

In den Übungsblättern 1 bis 7 wurden Ihnen Grundlagen vermittelt um aufwendigere Multimedia-Anwendungen zu bauen. In diesem Projekt finden nun viele gelernte Aspekte Anwendung. Alle Techniken (und viel Code), den Sie für dieses Projekt benötigen, wurden während des Semesters in der Übung schon behandelt. Vor allem Sprite Animationen, Physiksimulationen und Kollisionen sind wichtig.

Es wird davon ausgegangen, dass Cocos2d-x bereits erfolgreich installiert wurde. Zudem empfehlen wir die Einrichtung einer geeigneten Versionsverwaltung (bspw. SVN oder Git) sowie eines Kommunikationskanals um die Gruppenarbeit möglichst effizient zu gestalten.

Bonusberechnung

Durch die erfolgreiche Abgabe können bis zu 10% Notenbonus erlangt werden. Der Bonus wird gestaffelt je nach Umsetzung zugeteilt:

- Alle funktionierenden Abgaben, die alle Basisanforderungen erfüllen erhalten 5% Bonus.
- Alle funktionierenden Abgaben, die alle Basisanforderungen und mindestens drei der gegebenen Zusatzanforderungen erfüllen erhalten weitere 5% Notenbonus (insgesamt 10%).

Description

Das Ziel des diesjährigen Bonusprojekts ist die Entwicklung eines ["Artillery" Spieles](#) (z.B: Pocket Tanks oder die Worms Spiele). Die Spielfigur, beginnt auf der linken Seite des Bildschirms. Der Spieler kann mit der Tastatur die Figur durch ein Terrain (z.B.: eine Ebene mit Bergen oder verschiedene in der Luft schwebende Inseln) bewegen und die Artillerie Kanone auf der Spielfigur in verschiedenen Winkeln ausrichten. Um zu schießen muss der Spieler zu erst die Stärke des Schußes einstellen, die dann die Entfernung des Schusses beeinflusst. Der Schußwinkel sollte über die Maus einstellbar sein. Die Schußstärke sollte entweder über eine Tastatur- oder Maustaste einstellbar sein. Je länger man diese Taste drückt, desto stärker der Schuß. Bei Loslassen der Taste wird geschossen. Das Ziel des Grundspieles ist es innerhalb mehrere über das Spiel Terrain verteilte Zielobjekte (mit HP, "Health Points") zu zerstören. Mit den Zusatzbedingungen kann das Ziel auch sein einen Gegner auszuschalten, der von einer KI oder einem anderen Spieler gesteuert wird. Ein taktisch veranlagter Spieler sollte dann das Terrain nutzen können um dem gegnerischen Feuer auszuweichen. Seid kreativ und entwickelt eure eigenen Spielkonzepte! Viel Spaß!

Basisbedingungen (5%)

Die folgenden Basisbedingungen müssen erfüllt werden:

- a) Das Spiel soll mit einem Start-Bildschirm anfangen auf dem ein Menü und mögliche Optionen angezeigt werden.
- b) Der Spieler sollte anhand des Menüs, die Möglichkeit haben, ein bestimmtes Level auszuwählen. Es sollte mindestens 3 verschiedene Levels geben. Diese unterscheiden sich was Thema, Hintergrund, Terrain, Objekte und Hindernisse betrifft. Es sollte verschiedene Hindernisse pro Level geben, welche verschiedene Größen, Formen und Eigenschaften haben.
- c) Der Spieler sollte den Winkel und die Stärke des Artillerieschusses verändern können. Die Physik sollte dem entsprechend realistisch und richtig sein. Außerdem muss der Spieler die Figur bewegen können.

- d) Es sollte mindestens 3 Ziele geben welche als Gewinnbedingung zerstört werden müssen.
- e) Jedes Spiel sollte eine Uhr bzw. Zeit haben, die sobald sie 0 erreicht, das Spiel beendet.
- f) Die Ziele sollten eine Anzahl Lebenspunkte haben.
- g) Jeder Schuss bzw. Angriff sollte eine bestimmte Menge an Schaden anrichten, welche abhängig von der Nähe zum jeweiligen Ziel ist.
- h) Die folgenden Animationen müssen enthalten sein:
 - a. Der Artillerieschuss, abhängig vom Winkel und der Stärke.
 - b. Die Bewegung der Spielerfigur.
 - c. Der Schusswinkel.
 - d. Verlieren des Spiels (bspw. Eine Animation mit „Verloren“)
 - e. Gewinnen des Spiels (bspw. Eine Animation mit „Gewonnen“)

Es steht euch frei, die Level so zu entwerfen wie ihr wollt. Die Grafiken für alle Objekte, den Hintergrund, Terrain, Figuren, etc. müssen alle eigenständig entworfen werden. Passt bitte auf, dass ihr kein urheberrechtlich geschütztes Material benutzt. Und gebt eurem Spiel einen Namen.

Zusatzanforderungen (weitere 5%)

Von den Zusatzanforderungen müssen drei erfüllt werden für einen weiteren Notenbonus:

- a) Es sollte einen Multiplayer Modus geben, in dem ein zweiter Spieler anstatt der Ziele in den Leveln vorhanden ist. Spieler 1 und 2 sollten Lebenspunkte haben und beide Spieler sollte auf einer Tastatur spielen können. Das Spiel sollte in dem Fall Rundenbasiert stattfinden. Gewinnbedingung sollte es sein die Lebenspunkte des Gegners auf 0 zu bringen.
- b) Der Spieler hat die Fähigkeit zwischen Schwierigkeitsgraden auszuwählen und die Spielerfigur entsprechend Lebenspunkte haben. Anstelle der Ziele, treten nun einer oder mehrere Gegner, die zurückschießen. Auf dem Schwierigkeitsgrad „einfach“ bewegen sich diese nicht und schießen in eine zufällige Richtung. Auf dem mittleren Schwierigkeitsgrad sind sowohl Schießen wie auch Bewegen komplett zufällig. Bei „Schwer“ sollten die gegnerischen Aktionen auf eine state-machine basieren (für mehr Informationen schaut in die Vorlesung).
- c) Der Spieler sollte in jedem Level und im Start Bildschirm andere Musik hören. Außerdem sollte es Audioeffekte für Angriffe, Treffer, besiegte Gegner und ein besiegen des Spielers geben.
- d) Parallax-Scrolling des Hintergrunds während des Spiels. Dazu müssen die Level natürlich signifikant größer als das Fenster in dem das Spiel läuft sein, damit der Hintergrund dann sinnvoll „gescrollt“ werden kann. Ob dies dann horizontal, vertikal oder in alle Richtungen ist euch überlassen.
- e) Vollständig zerstörbares Gelände, welches permanenten Schaden durch (mindestens) die Schussaktionen der Spieler und Gegner nimmt.
- f) Zufällig generierte Level, welche zufällig, Terrain generieren und zufällig Hintergrund, Objekt und Hindernisse zuweisen.

Organisation

1. Teams bilden

Versuchen Sie baldmöglichst Teams zu bilden. Jedes Team muss aus 3-4 Studierenden bestehen. Die Teams sollten bunt gemischt sein und mind. eine/n Kunst und Multimedia Studierende/n beinhalten. Es gibt zwar keine Frist für die

Teambildung aber die dafür verbrauchte Zeit wird Ihnen bei der Umsetzung fehlen.

2. Kommunikation & Koordination

Sprechen Sie sich mit Ihrem Team ab. Treffen Sie sich zu einem ersten Brainstorming. Wie soll das Spiel aussehen? Teilen Sie das Projekt in Einzelaufgaben und weisen Sie jedem Teammitglied geeignete Aufgaben zu. Erstellen Sie einen Zeitplan zur Realisierung Ihres Projekts.

3. Setup

Falls noch nicht geschehen, richten Sie Cocos2d-x ein. Suchen Sie aus den Übungen nach geeigneten Codestellen. Viele der benötigten Codestellen sind bereits in den Übungen behandelt worden. Richten Sie eine Versionskontrolle ein. Einigen Sie sich auf einen Kommunikationskanal. Helfen Sie Ihren Teamkollegen. Am besten richten Sie die Geräte während des ersten Treffens gemeinsam ein.

4. Fragen & Probleme

Es wird Ihnen während die Tutorien zwischen 02.07 und 13.07 ein Tutor für Rückfragen zur Verfügung stehen. Nutzen Sie diese Zeit um Ihre Fragen bezüglich Inhalt / Spielstrategie und Optik zu klären und Ihre Probleme bei der Implementierung zu lösen. Zudem bieten die Tutoren in der letzten Juliwoche zu den bisherigen Übungsstunden (Di, Mi, Fr) Hilfestellung bei der **Klausurvorbereitung** und beim letzten Bug-Fixing an.

5. Abgabe

Die Abgabe muss als ZIP Datei bis zum **30.07.2018 23:59 Uhr** im UniWorX Portal (<https://uniworx.ifi.lmu.de/>) erfolgen. Das ZIP muss folgendes enthalten:

- Den Code und alle nötigen Bibliotheken.
- Eine kurze Anleitung zur Ausführung/Kompilierung des Spiels.
- Eine ca. einseitige Spieleanleitung/-beschreibung. Diese sollte mit Screenshots unterstützt sein.
- Eine kurze Übersicht die darstellt welcher Studierende welche Tätigkeiten übernommen hat.
- Ein Video von 1-3min, dass den Spielablauf und -ziel, die Interaktionen, etc. zeigt.

Es werden keine Präsentationen stattfinden.