



# 'Air Espresso'

## Abgabe 3

Ein Fun-Flugsimulator  
von

Gerold Plotzer	0025686	926	gerold_plotzer@hotmail.com
Ulrich Kreuzer	0025784	935	e0025784@student.tuwien.ac.at

## Reminder

Bei dem Computerspiel 'Air Espresso' handelt es sich um einen Fun-Flugsimulator. Diese Bezeichnung, Fun-Flugsimulator, soll schon andeuten, dass weniger Wert auf eine korrekte Replikation der physikalischen Wirklichkeit und Gesetze gelegt wird, sondern dass im Vordergrund der Spielspaß steht.

Der furchtlose Spieler steuert sein Flugzeug durch eine enge gefährliche Schlucht. Seine Gegner sind seine eigene Geschicklichkeit, die Zeit die unaufhaltsam dahinfließt, und die Punkte die der Spieler sammeln muss. Der Pilot wird mitten in das Spielgeschehen hinein gestoßen – seine Ziele sind klar definiert: den Canyon so schnell wie möglich zu durchfliegen und Punkte zu sammeln. Nur harte Kerle kommen durch.

## Installation

Das Spiel „Air Espresso“ wird als Zip-File angeboten. Nach dem Extrahieren von *espresso.zip* in einen beliebigen Ordner kann das Spiel durch das Ausführen der Dateien *AirEspresso-DEBUG.exe* beziehungsweise *AirEspresso-RELEASE.exe*, welche sich im Ordner *bin* befinden, gestartet werden.

## Steuerung

Das Spiel wird durch die Tasten „W“, „A“, „S“ und „D“ gesteuert.

- „W“ führt zu einem Steigen des Flugzeugs.
- „S“ läßt das Flugzeug sinken.
- „A“ leitet eine Linkskurve ein.
- „D“ leitet eine Rechtskurve ein.

Die einzelnen verschiedenen Modi werden mit folgenden Tasten gewechselt:

- F3: Wireframe Modus
- F4: Lineares / Nearest – Filtering
- F5: Mip Mapping On/Off
- F6: Immediate-Mode / Vertex-Array / Vertex-Buffer-Objects
- F7: Display-Lists

Die Geschwindigkeit des Flugzeugs paßt sich dem jeweiligen Flugzustand an; steigt das Flugzeug so reduziert sich die Geschwindigkeit. Sinkt es, so beschleunigt der Dreiecker.

## Ziel des Spiels

Sie müssen einen alten Dreiecker (Muster Fokker Dr.1, „Richtofen“) durch eine gefährliche Schlucht fliegen und währenddessen so viele Punkte wie möglich sammeln. Die Zeit ist Ihr Feind! Vergessen sie nicht auch die Zeit-Items einzusammeln um einen Timeout hinauszuzögern, beachten Sie deshalb stets die Zeit-Anzeige.

Ein allzu waghalsiges Fliegen, verbunden mit Kollisionen mit dem Canyon, führt zu einer Beschädigung Ihres Dreieckers! Fliegt man direkt auf den Canyon zu, so wird am Meisten des Status abgezogen, streift man hingegen nur den Canyon, so wird etwas weniger des Status weggenommen. Der Status kann über die Statusanzeige abgelesen werden. Sinkt der Status des Flugzeugs auf Grund zu vieler Kollisionen auf Null, so wird

Ihr Flugzeug explodieren, danach können Sie das Spiel durch Drücken auf „R“ neustarten beziehungsweise mit „Esc“ beenden.

### **Step-by-Step-Walkthrough**

Sie starten in der Mitte des Canyons. Sammeln Sie möglichst alle Münzen ein, jede von Ihnen bringt Punkte. Große Münzen bringen mehr Punkte als die kleinen Münzen, wählen Sie also Ihren Flugweg sorgfältig aus. Achten Sie auf die Zeitanzeige! Läuft die Zeit ab, so bricht das Spiel ab und es wird „GAME OVER“ ausgegeben und das Spiel kann mittels Tastendruck auf „R“ neu gestartet werden. Um die verfügbare Zeit zu verlängern müssen Sie die blauen Items einsammeln. Neben den Münzen und den „Checkpoints“ sind auch noch Items in Form eines Fragezeichens vorhanden. Sie dienen der Überraschung des Spielers, denn Sie können entweder dem Spieler Punkte und Status abziehen, oder Punkte und Status erhöhen.

Sind Sie am Ende des Canyons angelangt, so müssen Sie durch zwei Pylonen durchfliegen, danach wird „ZIEL ERREICHT“ ausgegeben und sie können das Spiel mittels Tastendruck auf „R“ neustarten.

Kollidieren Sie nicht mit dem Canyon! Jede Kollision führt zu einer Beschädigung an Ihrem Flugzeug. Sinkt der Status auf 0, so wird „GAME OVER“ ausgegeben und das Flugzeug wird unweigerlich explodieren! Kollidieren Sie auch nicht mit den im Spiel enthaltenen Hängebrücken, denn Sie führen zu einer sofortigen Zerstörung des Flugzeugs!

### **Beschreibung der Implementierung**

Bei der Implementierung wurde „Microsoft Visual C++ 2005 Express Edition“ und „Microsoft Visual C++ 2005 Professional“ verwendet.

In der main() Methode von „AirEspresso“ werden die MS3D-Modelle geladen, die Kollisionsmodelle des Flugzeugs, des Canyons und sonstiger Spielelemente erstellt und die eigentlichen Spielelemente (große Coins, kleine Coins, Checkpoints und Fragezeichen) initialisiert. Anschließend wird mit Hilfe von GLUT das OpenGL-Fenster initialisiert. Höhe und Breite des OpenGL-Fenster ergeben sich aus der jeweils aktuellen verwendeten Bildschirmauflösung.

In der init() Methode werden sämtliche Displaylists durch Aufruf der Funktion `erstelleListeADV(Model)` erstellt, welche sich in der Datei `scenery.cpp` befindet. Auch das Laden von manchen Texturen erfolgt in der `init()`-Methode.

In der `display` Methode wird zunächst die Kamera mittels `gluLookAt()` positioniert. Danach wird das Kollisionsmodell des Flugzeugs an die richtige Stelle transformiert.

Nun werden die zur Steuerung benutzen Keys abgefragt und je nachdem die Winkel `yaw`, `pitch`, `roll` und `pitchZusatz` des Flugzeugs manipuliert.

Wird das Flugzeug nicht bewegt, so hat es ein eigenstabiles Verhalten: es rotiert sich in Neutralstellung zurück.

Treten während dem Fliegen Kollisionen mit dem Canyon auf, so dementsprechend darauf reagiert. Das `Collisionhandling` befindet sich in der Datei `collisionHandler.cpp`.

Air Espresso arbeitet mittels Timing, sodass eine Ausführung auf unterschiedlich starken Rechnern mit der gleichen Spielgeschwindigkeit möglich ist.

## **Realisierung der einzelnen Punkte**

### **Spieldesign**

Ganz im Vordergrund der dritten Abgabe stand bei uns das Spieldesign. So wurden noch mehr Spielelemente in das Spielgeschehen eingebunden und diese vor allem sinnvoll angeordnet.

Im Spiel „Air Espresso“ werden folgende Spielelemente verwendet:

- Flugzeug: Der Spieler steuert sein Flugzeug, welches ein Dreidecker „Fokker Dr.1“ ist. Das Flugzeug wurde so gestaltet dass es stark an das Flugzeug von Freiherr Baron von Richthofen erinnert.
- Große Münzen: Der Spieler muss während dem Flug durch den Canyon Münzen einsammeln um seine Punkteanzahl zu erhöhen. Die Münzen sind so angeordnet, dass der Pilot am Steuerknüppel gefordert wird, wenn er alle Münzen vor Ablauf der Zeit einsammeln möchte.
- Kleine Münzen: Kleine Münzen haben einen geringeren Wert als die großen Münzen, obwohl sie durch ihre Größe etwas schwieriger einzusammeln sind. Auf Grund des Schwierigkeitsgrades befinden sich immer mehrere kleine Münzen in Flugrichtung, sodass es dem Piloten ermöglicht wird mehre Punkte einzusammeln. Da sich nun der Pilot zwischen zwei verschiedenen Münzarten entscheiden muss, sollte er sich den Flugweg genau überlegen, er steht vor folgenden Entscheidungen: „Wähle ich diesen Weg, so kann ich einige große Münzen einsammeln; wähle ich allerdings jenen Weg, so kann ich mehrere kleine Münzen einsammeln“. Diese Frage nach dem richtigen Flugweg kann entfallen, wenn der Pilot zuvor recht zügig geflogen ist und somit noch genügend Zeit hat um verbliebene Coins einzusammeln.
- Checkpoints: Da die Zeit unaufhörlich dahinfließt muss der Pilot rechtzeitig zu den blauen Checkpoints gelangen. Die Checkpoints verlängern die zur Verfügung stehende Zeit um einen sinnvollen Wert.
- Fragezeichen: Um noch mehr Spannung in das Spiel zu bringen wurden Items in das Spielgeschehen gesetzt, deren Verhalten bei Kontakt mit dem Flugzeug vorher nicht offensichtlich ist. So können die Fragezeichen sich positiv auf Pilot und Flugzeug auswirken in dem die Punkteanzahl und der Status erhöht wird, aber auch gegenteiliges kann der Fall sein.

### **Nichttriviale Objekte**

Sämtliche im Spiel vorkommende Objekte sind nichttrivial, sie wurden alle händisch von uns im Programm „Milkshape“ gestaltet. Für das Laden der Milkshapemodelle verwenden wir unseren Modelloader er uns geeignete Datenstrukturen zur Verfügung stellt.

### **Transparenz-Effekte**

An geeigneter Stelle wurden im Spiel „Air Espresso“ Transparenz-Effekte eingesetzt um das grafische Design etwas aufzubessern. So werden bei sämtlichen Explosionen mittels

Transparenz gute Ergebnisse erzielt.

## **Experimentieren mit OpenGL**

### *Geometrie-Modi*

Die Geometrie-Modi können mittels „F6“ umgeschaltet werden. Standardmäßig werden Vertex-Arrays zum Rendern benutzt.

Als Geometrie-Modi wurden sämtliche geforderte Modi implementiert:

- Immediate-Mode: Im Immediate-Mode werden sämtliche Modelle Dreieck für Dreieck gerendert, er ist somit der langsamste aller Geometrie-Modi.
- Vertex-Array-Modus: Im Vertex-Array-Modus werden die Modell mittels der erstellten Vertex-Arrays gerendert. In unserem Spiel „Air Espresso“ ist dieser Modus um Einiges schneller als das Rendern im Immediate-Mode.
- Vertex-Buffer-Objects: wie gefordert wurden auch von uns die Vertex-Buffer-Objects implementiert. Der VBO-Modus bringt nochmals einen Boost im Vergleich zu den Vertex-Arrays. Die schnellste aller Render-Arten sind aber bei unserem Spiel die Display Lists, da wir nur eine Szenerie von ca. 7000 Triangles rendern. Nach Rücksprache mit den Tutoren ist dies auf die relativ kleine Anzahl der Triangles zurückzuführen, da das „Binden“ und „Unbinden“ bei einer solch kleinen Anzahl recht teuer wird.

### *Mip Mapping*

Mip Mapping kann über die Taste „F5“ ein, bzw. ausgeschaltet sein. Ist „Mip Mapping“ eingeschaltet und möchte man die Filterung mittels „F4“ ändern, so muss „Mip Mapping“ zuerst wieder deaktiviert werden. Standardmäßig ist „Mip Mapping“ ausgeschaltet.

### *Texturqualitätseinstellungen*

Die Texturqualität kann über die Taste „F4“ geändert werden. Beachten Sie bitte, dass die Texturqualität nur dann geändert werden kann, wenn zuvor das „Mip Mapping“ per „F5“-Taste deaktiviert wurde.

### *Display Lists*

Display Lists können über die Taste „F7“ zum Rendern aktiviert werden. Display Lists sind bei „Air Espresso“ die mit Abstand schnellste Render-Möglichkeit.

### *Anzeige der Framerate*

Die Anzeige der Framerate erfolgt ständig in der linken oberen Ecke des OpenGL-Fensters.

### *Wireframe Modus*

Der Wireframe Modus kann über die Taste „F3“ aktiviert bzw. deaktiviert werden.

## **Spezial Effekte**

Laut der Tabelle wurden folgende Spezialeffekte in das Spiel eingebaut:

*Partikel-Effekt* mittels GL\_Point-Sprites-Extension für die Explosion des Flugzeugs falls der Status auf Null zurückgeht oder die Zeit abläuft. Hierbei werden mittels der Extension

Texturen auf Punkte gelegt, welche danach mittels eines transparenten Alpha-Kanals dargestellt werden. Die einzelnen Partikel bewegen sich mit zufälliger Geschwindigkeit und zufälliger Richtung im Raum aus. Da Anfangs mehrere Partikel übereinander liegen wird so ein Effekt einer Explosion erzielt.

Grundlagen dafür waren ein Artikel auf [nehe.gamedev.net](http://nehe.gamedev.net):  
<http://nehe.gamedev.net/data/articles/article.asp?article=19>

*Explosion* der Spielelemente falls eine Kollision mit diesen stattfindet. Die Explosion wird mittels Alphablending eines Bitmaps simuliert.

*Projected Plane Shadow*: Das Flugzeug wirft einen Schatten auf den Boden. Als Unterlagen für den Schatteneffekt haben wir den Artikel auf [www.devmaster.net](http://www.devmaster.net) verwendet: <http://www.devmaster.net/articles/shadows/>

*Sound Effekt*: Das Motorengeräusch des Flugzeugs ist abhängig von der Fluggeschwindigkeit. Bei einer Kollision mit den Spielelementen erfolgt je nach Art des Elementes eine Wiedergabe unterschiedlicher Sounds.

### **Verwendete Zusatztools**

Folgende Zusatztools bzw. externen Libraries wurden verwendet:

- GLUT: OpenGL-Fensterinitialisierung und Steuerungs-Callbacks.
- ColDet: Collision Detection Library
- GLF: Textausgabe im OpenGL-Fenster: [www.forexseek.com/glf/](http://www.forexseek.com/glf/)
- FMOD: Wiedergabe der Soundeffekte.

### **Tools zum Erstellen der Model**

Zum Erstellen der Models wurde das Programm „Milkshape 3D“ verwendet. Um die Texturen den komplexen Modellen anzupassen bzw. zu entzerren wurde „LithUnwrap“ verwendet.