

aMAZEment

Im Weltraum kann man sich nur einmal verirren

Queteschiner Daniel
Matrikel-Nr.: 9856024
SK: K810

WS 2005/2006



Einleitung

Wir schreiben das Jahr 2206. An Bord des Raumfrachters Amilon 5 befinden Sie sich gerade auf dem Weg zur Erde, als plötzlich ruchlose Weltraum-Piraten aus dem Hyperraum auftauchen und das Feuer eröffnen. Mit knapper Not entkommen Sie dem Inferno in einer Rettungskapsel und stranden auf dem nächstgelegenen Planeten. Dieser scheint zwar besiedelt zu sein, doch von den Bewohnern fehlt jede Spur. In den labyrinthartigen Anlagen machen Sie sich auf die Suche nach einem Transportmittel, das Sie wieder nach Hause bringen kann.

Spielbeschreibung

Das Spiel besteht aus 2 labyrinthartigen Umgebungen, in denen Sie den Ausgang finden müssen. Dabei gibt es unterschiedliche Probleme zu bewältigen. Im 1. Abschnitt müssen zum Beispiel bewegliche Plattformen aktiviert werden, damit der Ausgang zugänglich wird. Im 2. Abschnitt ist Ihre einzige Orientierungshilfe ein kleiner fliegender Roboter, dem Sie folgen müssen - allerdings ist die Sicht durch Nebel stark eingeschränkt, sodass Sie nur dem Geräusch des Roboters folgen können.

Spielsteuerung

Desktop

Start-Menü

Um sich nach links oder rechts zu drehen, bewegen Sie den Mauszeiger zum linken bzw. rechten Fensterrand.

Um das Spiel zu starten oder fortzusetzen drücken Sie die linke Maustaste.

Zum Verlassen des Spiels drücken Sie die Esc-Taste.

Spiel

Um sich vorwärts oder rückwärts zu bewegen, bewegen Sie den Mauszeiger zum oberen bzw. unteren Fensterrand. Um sich nach links oder rechts zu drehen, bewegen Sie den Mauszeiger zum linken bzw. rechten Fensterrand.

Bewegliche Plattformen und Transporter aktivieren Sie, indem Sie sich auf sie stellen und die linke Maustaste drücken. Schalter werden ebenfalls durch das Drücken der linken Maustaste betätigt, sofern Sie sich in unmittelbarer Nähe befinden.

Aktivierbare Plattformen, Transporter und Schalter erkennen Sie an der blauen Farbe, rote Farbe hingegen signalisiert, dass das jeweilige Objekt im Moment nicht benutzbar ist.

Durch Drücken der rechten Maustaste kehren Sie zum Start-Menü zurück.

Credits

Um sich nach links oder rechts zu drehen, bewegen Sie den Mauszeiger zum linken bzw. rechten Fensterrand. Um das Spiel zu verlassen drücken Sie die Esc-Taste.

CAVE

Start-Menü

Um sich nach links oder rechts zu drehen, drücken Sie den Joystick nach links bzw. rechts.
Um das Spiel zu starten oder fortzusetzen drücken Sie Button 1.
Zum Verlassen des Spiels drücken Sie Button 3.

Spiel

Um sich vorwärts oder rückwärts zu bewegen, drücken Sie den Joystick nach vorne bzw. hinten.
Um sich nach links oder rechts zu drehen, drücken Sie den Joystick nach links bzw. rechts.
Bewegliche Plattformen und Transporter aktivieren Sie, indem Sie sich auf sie stellen und Button 1 drücken.

Schalter werden ebenfalls durch das Drücken von Button 1 betätigt, sofern Sie sich in unmittelbarer Nähe befinden.

Aktivierbare Plattformen, Transporter und Schalter erkennen Sie an der blauen Farbe, rote Farbe hingegen signalisiert, dass das jeweilige Objekt im Moment nicht benutzbar ist.

Durch Drücken von Button 3 kehren Sie zum Start-Menü zurück.

Credits

Um sich nach links oder rechts zu drehen, drücken Sie den Joystick nach links bzw. rechts.
Um das Spiel zu verlassen drücken Sie Button 3.

Umsetzung

Das Projekt umfasst :

- aMAZEment (Spiel-Applikation)
- ms3dasciid2vis (Modell-Konverter)
- 3D Modelle, Texturen

Entwicklungsumgebung : Windows, Irix

Verwendete APIs : CAVELib/KNAVELib(modifiziert), OpenGL, OpenAL

Programmiersprache : C++

3D Modelle wurden mit Milkshape3D erstellt, Texturen mit Corel Photo-Paint

Beschreibung der wichtigsten Komponenten

CAVEmain: enthält die grundlegenden CAVELib Funktionen.

CAVEApp: zentrale Applikationssteuerung

CAVEAudio: OpenAL-Implementation

CAVEEntityManager: Verwaltung der verschiedenen Entities (Plattformen, Schalter, ...), die die Spiellogik steuern. Die unterschiedlichen Objekte werden in einer eigenen Datei (.ent) im Textformat beschrieben und können somit einfach verändert werden.

CAVEMaTManager: Verwaltung von Materialien und Texturen. MaTs werden ebenfalls aus speziellen Dateien (.mat) im Textformat geladen.

CAVERenderer: OpenGL-Implementation

CAVEModelManager: Verwaltung von beweglichen 3D Modellen (.mdc)

CAVEWorld: Verwaltung der Leveldaten für Rendern und Kollision. Es werden dabei 2 verschiedene Datensätze (.vis, .col) verwendet, um bei der Kollisionsabfrage eine möglichst geringe Anzahl von Dreiecken testen zu müssen. Die beiden Datensätze sind jeweils in einem sloppy binary AABB-Tree organisiert, um schnelles Frustum-Culling bzw. schnelle AABB-Kollisionsabfragen zu ermöglichen.

CAVEMath: umfasst eine obligatorische Vektor- und Matrix-Klasse, sowie zahlreiche Funktionen zur Kollisionsabfrage (AABB, OBB, Ray, Plane, Sphere, Triangle)

Credits

Konzept, Programmierung, Level-Design: Queteschiner Daniel

3D Modelle: Queteschiner Daniel, Psionic (Fighter, <http://www.psionic3d.co.uk>)

Texturen: Queteschiner Daniel

Skybox Texturen: Evile Dick (Menu), Justin Fisher (Level 1), The Mighty Pete (Credits)

Music: Justin R. Durban (Menu), Marcus Gutierrez (Level 1), Tarantula Studios (Level 2), Jay Berlinsky (Credits)(<http://www.flashkit.com>)



Anhang A

Beschreibung der Dateiformate (Schlüsselwort, [Wert/Beschreibung])

VIS (Visibility)

Verwendung

3D Modelldaten zum Rendern größerer Modelle, in einem AABB-Tree organisiert für schnelles Frustum-Culling

Bemerkungen

Material-Indizes beziehen sich auf MaTs in zugehöriger .MAT-Datei

Aufbau

CAVEVis [float Version (1.0)]

Vertices: [unsigned int NumVerts - Anzahl der Vertices]

[Liste der Vertices

Format: float x float y float z float normal_x float normal_y float normal_z float u float v]

[AABB-Tree Nodes

Format:

Min: [float x float y float z - Minimum der AABB dieses Nodes]

Max: [float x float y float z - Maximum der AABB dieses Nodes]

Faces: [unsigned int NumFaces - Anzahl der Dreiecke in diesem Node

größer 0: der Node ist ein Leaf, es folgt eine Liste der Faces (Vertex- und Material-Indizes)

Format: unsigned int VIndex1 unsigned int VIndex2 unsigned int VIndex3 unsigned int MaT-Index

gleich 0: der Node besitzt 2 Unterknoten, es folgt die Beschreibung dieser ChildNodes]]

COL (Collision)

Verwendung

3D Modelldaten zur Kollisionsabfrage, in einem AABB-Tree organisiert für schnelle AABB-Kollisionstests

Aufbau

CAVECol [float Version (1.0)]

Vertices: [unsigned int NumVerts - Anzahl der Vertices]

[Liste der Vertices

Format: float x, float y, float z]

[AABB-Tree Nodes

Format:

Min: [float x float y float z - Minimum der AABB dieses Nodes]

Max: [float x float y float z - Maximum der AABB dieses Nodes]

Faces: [unsigned int NumFaces - Anzahl der Dreiecke in diesem Node

größer 0: der Node ist ein Leaf, es folgt eine Liste der Faces (Vertex-Indizes)

Format: unsigned int VIndex1, unsigned int VIndex2 unsigned int VIndex3

gleich 0: der Node besitzt 2 Unterknoten, es folgt die Beschreibung dieser ChildNodes]]

MDC (ModelCAVE)

Verwendung

3D Modelldaten zum Rendern kleinerer Modelle

Bemerkungen

Material-Indizes beziehen sich auf MaTs in zugehöriger .MAT-Datei

Aufbau

CAVEModel [float Version (1.0)]

Vertices: [unsigned int NumVerts - Anzahl der Vertices]

[Liste der Vertices

Format: float x float y float z float normal_x float normal_y float normal_z float u float v]

Min: [float x float y float z - Minimum der AABB dieses Modells]

Max: [float x float y float z - Maximum der AABB dieses Modells]

Faces: [unsigned int NumFaces - Anzahl der Dreiecke in diesem Modell]

[Liste der Faces (Vertex- und Material-Indizes)

Format: unsigned int VIndex1 unsigned int VIndex2 unsigned int VIndex3 unsigned int MaT-Index]

MAT (Material and Texture)

Verwendung

Beschreibung von Materialien und zugehöriger Textur

Aufbau

CAVEMat [float Version (1.0)]

Materials: [unsigned int NumMaTs - Anzahl der beschriebenen MaTs]

[Liste der MaTs

Format:

float DiffuseR float DiffuseG float DiffuseB float DiffuseA

float AmbientR float AmbientG float AmbientB float AmbientA

float SpecularR float SpecularG float SpecularB float SpecularA

float EmissiveR float EmissiveG float EmissiveB float EmissiveA

float SpecPower

float Alpha

string TexturName]

ENT (Entity)

Verwendung

Beschreibung von Objekten, die den Spielablauf steuern

Aufbau

CAVEEnt [float Version (1.0)]

Entities: [unsigned int NumEntities - Anzahl der beschriebenen Entities]

[Liste der Entities

Format:

Type: [string EntityType - Entity-Typ, bestimmt die nachfolgenden Einträge]

[Entity spezifische Einträge]]

Entity-Typen:

- CHANGELEVEL
- LEVELSETUP
- PATHPOINT
- PLATFORM
- ROBOTGUIDE
- SKYBOX
- TRIGGER

CHANGELEVEL [zum Wechseln in einen anderen Level]

Position: [float x float y float z - Position des Entities]

NewLevel: [string LevelName - Name des neuen Levels]

NewLevelDir: [string LevelDirName - Name des Verzeichnisses, in dem sich der neue Level befindet]

NewTextureDir: [string TextureDirName - Name des Verzeichnisses, in dem sich die Texturen des neuen Levels befinden]

LEVELSETUP

PlayerStart: [float x float y float z - Position, an der der neue Level gestartet wird]

PlayerRotY: [float y - bestimmt die Richtung, in die man am Start blickt]

Music: [string MusicName - Name (inklusive Verzeichnis) der WAV-Datei, die als Hintergrundmusik dient]

FarClipPlane: [float FarClipPlane - Distanz, nach der nichts mehr gerendert wird]

ClearColor: [float R float G float B - Farbe, mit der die Anzeige gelöscht wird]

UseFog: [string Boolean - **true** oder **false**, soll (exponentieller) Nebel verwendet werden?]

FogDensity: [float Density - Dichte des exponentiellen Nebels]

FogColor: [float R float G float B - Farbe des Nebels]

PATHPOINT [zum Festlegen von Pfaden]

Position: [float x float y float z - Position des Wegpunktes]

PointID: [int ID - ID dieses Wegpunktes (≥ 0 , darf nur 1x in einem Level vorkommen)]

NextPointID: [int ID - ID des nächsten Wegpunktes in diesem Pfad, -1 bedeutet das Ende des Pfades]

PLATFORM

Position: [float x float y float z - Start-Position der Plattform]

Rotation: [float x float y float z - Start-Rotation der Plattform]

Translate: [float x float y float z - Vektor, um das Ende der Translation zu erreichen]

Rotate: [float x float y float z - Vektor, um das Ende der Rotation zu erreichen]

Model: [string ModelName - Name des zu verwendenden Modells im Verzeichnis *models*]

Min: [float x float y float z - Minimum der Kollisions-AABB (nicht implementiert - AABB aus MDC wird verwendet)]

Max: [float x float y float z - Maximum der Kollisions-AABB (nicht implementiert - AABB aus MDC wird verwendet)]
NoCollision: [string Boolean - **true** oder **false**, soll ein Kollisionstest mit diesem Modell durchgeführt werden?]
Rideable: [string Boolean - **true** oder **false**, wenn **true** dann wird der Spieler mit der Plattform mitbewegt]
Sound: [string SoundName - Name der WAV-Datei im Verzeichnis *audio*, die abgespielt wird, wenn sich das Modell bewegt, NULL wenn kein Sound verwendet werden soll]
LoopSound: [string Boolean - **true** oder **false**, soll der Sound wiederholt abgespielt werden?]
Trigger: [string TriggerName - Name des Trigger Entities, das diese Plattform aktiviert, spezielle Namen sind **Button1**, **Button2** und **Button3**, die als aktive Trigger gelten, wenn die entsprechende Wand- bzw. Maus-Taste losgelassen wird]

ROBOTGUIDE [leitet den Spieler entlang eines durch Wegpunkte definierten Pfades]
Position: [float x float y float z - Start-Position des Entities]
StartPoint: [int ID - ID des 1. Wegpunktes, den das Entity ansteuern soll]
Model: [string ModelName - Name des zu verwendenden Modells im Verzeichnis *models*]
SubModel0: [string SubModelName - Name des zu verwendenden SubModells im Verzeichnis *models*, NULL wenn keines verwendet werden soll; dieses Modell wird automatisch mit dem Haupt-Modell mitbewegt und zusätzlich um die y-Achse rotiert (+)]
SubModel1: [string SubModelName - Name des zu verwendenden SubModells im Verzeichnis *models*, NULL wenn keines verwendet werden soll; dieses Modell wird automatisch mit dem Haupt-Modell mitbewegt und zusätzlich um die y-Achse rotiert (-)]
Sound: [string SoundName - Name der WAV-Datei im Verzeichnis *audio*, die (looped) abgespielt wird, wenn sich das Modell bewegt, NULL wenn kein Sound verwendet werden soll]

SKYBOX

SkyTextureBase: [string TextureBaseName - Ausgangsname (inklusive Textur-Verzeichnis) für die 6 Skybox Texturen, an den die Endungen *_bk.bmp*, *_dn.bmp*, *_ft.bmp*, *_lf.bmp*, *_rt.bmp* und *_up.bmp* angehängt werden]

TRIGGER

Name: [string TriggerName - Name dieses Entities, dient zur Identifizierung (darf nur 1x im Level vorkommen)]
Position: [float x float y float z - Position des Entities]
ActiveTime: [float Seconds - Dauer (in Sekunden), für die der Trigger aktiv ist]
ActiveOnce: [string Boolean - **true** oder **false**, kann dieser Trigger öfter oder nur 1x aktiviert werden?]
Range: [float Range - der Spieler muss sich innerhalb dieser Distanz aufhalten, um dieses Entity mit einem speziellen Trigger (d.h. durch Drücken einer Taste) aktivieren zu können]
Sound: [string SoundName - Name der WAV-Datei im Verzeichnis *audio*, die abgespielt wird, wenn sich das Modell bewegt, NULL wenn kein Sound verwendet werden soll]
LoopSound: [string Boolean - **true** oder **false**, soll der Sound wiederholt abgespielt werden?]
Trigger: [string TriggerName - Name des Trigger Entities, das diesen Trigger aktiviert, spezielle Namen sind **Button1**, **Button2** und **Button3**, die als aktive Trigger gelten, wenn die entsprechende Wand- bzw. Maus-Taste losgelassen wird]

Anhang B

Screenshots

