

Doku zu

doppeldecker

Christoph Graf	0300309	033 532	christoph.graf@gmx.at
Birgit Wlaschits	0502944	e532	bw.223@gmx.at

Allgemein

Im Spiel gibt es 2 PlayerCharacter (oranger und roter Flieger, **PlayerCharacter.cs**), ein paar Gegner (grüne Flieger, **Enemy.cs**) und 4 Levels

Tastenbelegung:

Cursor rechts / D	Rotation um Z-Achse nach rechts
Cursor links / A	Rotation um Z-Achse nach links
Cursor rauf / W	Rotation um X-Achse nach oben
Cursor runter / S	Rotation um X-Achse nach unten
F1	Toggle Debug-Text-Anzeige
F2	Toggle Wireframe-Modus
F3	Toggle Fullscreen Modus
F4	Spieler 2 aktivieren / deaktivieren
Leertaste / E	schießen

Gamepadbelegung:

Thumbstick links	Steuerung
B-Button	Feuern

Licht & Texturen

Es gibt ein Licht in der Szene und alle Objekte werfen Schatten.

Für die 3D-Modelle werden Normalmaps und Specularmaps unterstützt.

3D-Modelle

Für die 3D-Modelle gibt es einen eigenen Content-Processor (FbxProcessor) der den 3D-Modellen den Shader (incl. Parametern) und die Maps zuweist.

Player und Gegner

PlayerCharacter und Gegner haben **Character.cs** als Überklasse mit gemeinsamen Funktionen.

Die gegnerischen Flieger bewegen sich gerade auf den Spieler zu. Bewegung erfolgt in der Funktion move die in der update-Funktion aufgerufen wird

Rendering

Alle 3D-Objekte sind Instanzen der Klasse **FbxNode**. Diese werden über einen Scenegraph verwaltet. Alle 3D-Objekte werden in der Funktion Load in **Game1.cs** instanziiert.

Game logic

Tastatur-, Maus und Gamepad-Input wird von den Klassen **KeyboardHandler.cs**, **MouseHandler.cs** und **GamepadHandler.cs** verwaltet. Die UpdateGame-Funktion in Game1.cs ist quasi der Gameloop des Spiels.

Jeder Spieler hat 3 Leben, Gegner haben 3 Hitpoints. Sinken die HP auf 0 wird ein Gegner zersört, bzw. dem Spieler ein Leben abgezogen.

Partikel

Für Patronen beim Schießen verwenden wir Partikel (Character/Bullet.cs) Partikel werden mit Hilfe von QuadRenderer.cs realisiert und verfügen auch über Billboarding.

Skybox

Rund um die Szene gibt es eine Skybox, bestehend auf 6 Quads mit jeweils passender Textur.

Menü

Über den Menüpunkt **tutorial** gelangt man zu den beiden Tutorial-Missionen, die nacheinander gespielt werden können. In der ersten Mission muss man durch eine bestimmte Anzahl Ringe (siehe Debug Output) fliegen, in der zweiten eine bestimmte Anzahl Gegner abschießen.

Mit **new game** startet man ein neues Spiel. Es gibt zwei Levels, die nacheinander durchfliegen werden können.

Das Menü haben wir uns von einem XNA-Sample ausgeliehen, da mein eigenes Menüsystem nicht so richtig funktionieren wollte.

Levels

Levelbeschreibung, Level-Objective und Levelinhalt werden von einem Text-File gelesen und dann in der Level-Klasse (LevelStuff-Ordner) weiterverarbeitet.

Font

Zur Sicherheit hab ich noch den verwendeten Font beigelegt.