

Neue Modder braucht das Land

Workshop – Einführung in LUA

Manuel Leithner aka Face

SFM-Modding

Kontakt: Sfm_f@web.de

Homepage: <http://forum.landwirtschafts-simulator.de>

Literatur

- Grundlagen des Workshops:
 - GIANTS Engine 4.1.3
 - <http://gdn.giants.ch>
 - Programming in LUA
 - <http://www.lua.org/pil/index.html>

Zur Person

Name: Manuel Leithner

Alter: 23

Beruf: Student der Wirtschafts-Informatik (6. Semester – Bachelor)
Werkstudent der Siemens AG

LS: Mitglied von SFM-Modding

Mods:

- Grasmod LS08
- KuhnVenta 2008
- Fliegl Hakenlift 2008
- Fendt GTA 380
- Pöttinger Jumbo

Soon:

- Case CVX 195
- Kirovets 700 K
- Güllemod
- JohnDeere STS

Gliederung

1. Grundlagen des Game-Development
2. Bäume, Bäume, Bäume aber kein Wald in Sicht
3. Grundlagen XML
4. LUA-Scripting – Einführung
5. LUA-Scripting – Fortgeschrittenes
6. Grundlagen der GIANTS – Engine
7. LUA-Scripting – Landwirtschafts-Simulator 2009
8. Zusammenfassung / Feedback / Fragen

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Game-Engine

- Was sagt Wikipedia:

*„Eine **Spiel-Engine** (engl. **Game Engine**) ist ein Computerprogramm, das den Spielverlauf steuert und für die visuelle Darstellung des Spieleablaufs verantwortlich ist. “*

- Bestandteile einer Game-Engine:

- Grafikengine
 - Physiksystem
 - Soundsystem
 - Steuerung
 - Datenmanagement
 - Netzwerkanbindung
 - Skript-Unterstützung
- } Optional

- Unterschied 3D-Engine – Game-Engine:

- 3D-Engine Teilbereich der Game-Engine

Warum Script-Unterstützung?

- Vorteile der Script-Unterstützung:
 - Script-Sprache leicht erlernbar (Für Artists)
 - Schnelles Prototyping
 - Keine Neucompilierung des Engine-Code
 - Erleichterte Modding-Möglichkeiten

- Nachteile:
 - Script-Sprache langsamer als Nativer Code (C++)
 - Einbindung einer Script-Unterstützung mit Aufwand verbunden
 - Für jede neue Funktion müssen Script-Schnittstellen bereit gestellt werden

- Fazit:
 - Vorteile überwiegen die Nachteilen
 - Script-Unterstützung hat sich in den letzten Jahren etabliert
 - Beispiele: UnrealScript, C-Script, Lua (GIANTS, Blizzard, CryTek)

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Multithreading

- Was ist Multithreading:

„Multithreading (Mehrfädigkeit) bezeichnet die Möglichkeit mehrere Tätigkeiten/Prozesse (nahezu) gleichzeitig auszuführen.“

- Warum ist Multithreading im Engine-Bereich notwendig:

- Aufteilung der Engine in deren Bestandteile
- Ohne Thread keine Möglichkeit Dinge gleichzeitig zu tun (z.B. Sound + Rendering)
- Multithreading ist ein MUSS bei Game-Engines

- GIANTS-Engine:

- Entgegen allen Gerüchten ist die Engine Multithreading-fähig
- Threads:
 - 1 x Sound
 - 1 x Logik + Rendering (D3D/OpenGL mit eigenen Threads)
 - 2-n x Physik
- Aufteilung der Threads auf Kerne wird von Windows gemanagt

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Der Gameloop

- Was ist der Gameloop:

„Der Gameloop ist die wichtigste Komponente eines Spiels. Er bezeichnet den reibungslosen, an den jeweiligen Rechner angepassten (FPS) Spielablauf“

- Funktionsweise (Dummy-Code):

1. Solange das Spiel nicht geschlossen wird:

1.1 Checke Input (Tastatur, Maus, Joystick, etc.)

1.2 Berechne Physik

1.3 Berechne Spiellogik

1.4 Gebe Sound aus

1.5 Male Spiel (Rendering)

2. Starte wieder bei 1.

- 1.1 – 1.5 werden nahezu parallel ausgeführt (Multithreading)

Funktion des reibungslosen Spielablaufs

- FPS (Frames-Per-Second):

„Frames-Per-Second definiert die Anzahl der gerenderten Bilder pro Sekunde und ist damit ein Indiz für das Empfinden des Spielablaufs“

- Faktoren der Beeinflussung:

- Polygonanzahl und andere Einflüsse
- Qualität der Programmierung
- Anderweitige Tätigkeiten am PC
- Potential des Rechners

- Benutzung der FPS um Spiel flüssig darzustellen:

- Gameloop (update-Funktion) wird alle paar ms (Millisekunden) aufgerufen
- Größe der ms abhängig von den „Faktoren der Beeinflussung“
- Möglichkeit ms mit in die Berechnungen einzubeziehen, um so die Bewegungen abhängig von der aktuellen Framezahl zu steuern (Smooth)

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Wichtige Prinzipien

- Gameloop abhängig von den „Faktoren der Beeinflussung“:
 - Performance-orientiertes Modding wichtig
 - Saubere Scripte (siehe Kapitel 4,5,7)
 - Abfangen unnötiger Berechnungen
 - Weglassen unnötiger Schleifen-Durchläufe
 - Sauberes Polygonmodell
 - Weglassen unnötiger Polygone
 - Weglassen unnötiger Details
 - Darstellung von Details durch Texture-, Normal-, Environment-Maps
 - Richtige Einstellung der 3D-Files (Bsp: I3D - siehe Kapitel 6)
- Auswirkungen des Moddings beachten:
 - Modding ist der Eingriff in die Standard-Spielroutinen (getestet)
 - Veränderung der Standard-Routinen -> erneutes Testen erforderlich
 - Vor dem Benutzen von Standard-Routinen erst deren Abhängigkeiten checken
 - Dazu später mehr

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Recherchieren

- Was bedeutet Recherche:
„Bezeichnet die nicht-beiläufige, „professionelle“ Suche nach Informationen.“
- Warum ist Recherche wichtig?
 - Man kann nicht alles wissen
 - Oft sind es nur Kleinigkeiten, die einem das Scripten erleichtern
 - Es ist wichtig zu wissen **WIE** man an die richtigen Informationen kommt
 - Eine der wichtigsten Aufgaben beim Scripten/Programmieren

■ **Grundsatz:** Erst selber informieren, dann fragen!

- Erste Anlaufquellen
 - Google
 - Google weiß so gut wie alles
 - Wichtig ist das richtige Googlen
 - Entwickler-Homepage (Bsp: <http://gdn.giants.ch/>)
 - Informationen über die Engine
 - Scriptdokumentation (siehe Kapitel 5)
 - Forum

Gliederung

1. Grundlagen des Game-Development

1.1. Game-Engine

1.2. Multithreading

1.3. Gameloop

1.4. Wichtige Prinzipien

1.5. Recherchieren

1.6. Grundlagen des Modelling

Grundlagen des Modelling

- Bestandteile eines Modells:
 - Vertex: Ein Eckpunkt eines Modells
 - Polygon: Fläche zwischen 3 Eckpunkten
 - Face: Fläche aus mehreren Polygonen oder Eckpunkten
 - Mesh: Ansammlung von zusammengehörenden Faces/Polygonen
 - TG: Transformgroup zum Verschieben/Drehen/Skalieren von Meshes
 - Modell: Ansammlung von Meshes und Transformgroups
- Sichtbarkeit:
 - Per Definition sind Faces/Polygone nur von einer Seite sichtbar!
 - Sichtbarkeit wird über sogenannte Normals gesteuert
- Normals:
 - Benötigt zur Berechnung von Lichteinfall und Reflexionen

Grundlagen des Modelling

- Grundbegriffe :
 - Shading: Definition der Oberfläche eines 3D-Objektes
 - Textur: Muster/Struktur die auf ein 3D-Objekt aufgebracht werden kann
 - Mapping: Prozess des Auflegens einer Textur auf ein 3D-Modell
 - Lightning: Prozess des Positionierens von Lichtquellen im 3D-Raum
 - Rendering: Erzeugen einer 2D-Ansicht eines 3D-Modells

- Lichtarten:
 - Pointlight: haben einen Ursprung und strahlen rundum Licht ab
 - Directionallight: strahlt auf der ganzen Breite aus einer definierten Richtung
 - Spotlight: wie Pointlight, ist aber auf einen Abstrahlkegel begrenzt

Grundlagen des Modelling

- Orientierung und Ausrichtung:
 - Jedes Objekt (TG/Mesh) hat einen Nullpunkt (Pivot)
 - Alle Aktionen werden von diesem Nullpunkt aus berechnet
 - Translation (Verschiebung)
 - Rotation (Drehung)
 - Scale (Skalierung)

- Local vs. Worldspace: (jeweils vom Pivot des Objektes aus)
 - Worldspace: Koordinaten **ABSOLUT** zum Root-Nullpunkt
 - Localspace: Koordinaten **RELATIV** zum Parent-Nullpunkt
 - Genauere Beispiele später

Fragen / Feedback

- Gibt es noch Fragen?
- Was sollte besser gemacht/geändert werden?
- Welche Inhalte würden euch interessieren oder fehlen noch?