

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>

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

6. Grundlagen der GIANTS – Engine

6.1. I3D-Einführung und wichtige Einstellungen

6.2. Physik-Elemente

6.3. RootNode

6.4. Start-Script

6.5. Script-API

Gliederung

6. Grundlagen der GIANTS – Engine

6.1. I3D-Einführung und wichtige Einstellungen

6.2. Physik-Elemente

6.3. RootNode

6.4. Start-Script

6.5. Script-API

Gliederung

6. Grundlagen der GIANTS – Engine

6.1. I3D-Einführung und wichtige Einstellungen

6.2. Physik-Elemente

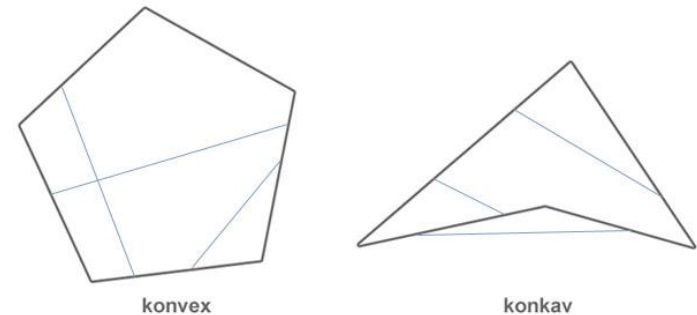
6.3. RootNode

6.4. Start-Script

6.5. Script-API

Physik-Elemente

- No Rigid Body:
 - Werden nur gerendert
 - Keine physikalischen Kräfte, keine Kollision
- Static:
 - Statische Kollision
 - Können nicht bewegt werden!
 - Reagieren nicht auf Gravitation
- Kinematic:
 - Haben Kollision gegen Dynamic-Objekte
 - Können bewegt werden
- Dynamic:
 - Dynamische Reaktion auf alle anderen Kollisions-Objekte
 - Bewegung ist durch physikalische Kräfte vorgegeben
 - Können über Compound & Compound-Child fest verbunden werden
 - Können auch über Joints verbunden werden



Physikalische Objekte dürfen nicht gescaled werden!!!

Dynamic & Kinematic Objekte müssen konvexe Objekte sein!

Gliederung

6. Grundlagen der GIANTS – Engine

6.1. I3D-Einführung und wichtige Einstellungen

6.2. Physik-Elemente

6.3. RootNode

6.4. Start-Script

6.5. Script-API

RootNode

- Merkmale:
 - Hauptknoten des I3D-Baumes des Spiels
 - Absoluter Null-Punkt des Spiel-Koordinatensystems
 - An diesen Knoten werden z.B. die Map und Fahrzeuge gelinkt
 - Gravitation wirkt von diesem Punkt aus

Gliederung

6. Grundlagen der GIANTS – Engine

6.1. I3D-Einführung und wichtige Einstellungen

6.2. Physik-Elemente

6.3. RootNode

6.4. Start-Script

6.5. Script-API

Start-Script

-- Start-Initialisierung. Wird beim Spielstart einmal aufgerufen

```
function init()
end;
```

-- Fängt Maus-Events ab

-- posX, posY: aktuelle X- bzw. Y-Pos der Maus (0-1)
-- isDown, isUp: Boolean ob Taste gedrückt ist
-- button: Mausknopf

```
function mouseEvent(posX, posY, isDown, isUp, button)
end;
```

-- Fängt Tastatur-Events ab

-- unicode, sym: Representation der Taste
-- modifier: Strg, Alt, etz
-- isDown: Boolean ob Taste gedrückt ist

```
function keyEvent(unicode, sym, modifier, isDown)
end;
```

-- Globaler Game-Loop

-- dt: Zeit seit letztem Update-Aufruf

```
function update(dt)
end;
```

-- Globale Draw-Funktion

-- Zeichnet und Schreibt auf den Bildschirm

```
function draw()
end;
```

Gliederung

6. Grundlagen der GIANTS – Engine

6.1. I3D-Einführung und wichtige Einstellungen

6.2. Physik-Elemente

6.3. RootNode

6.4. Start-Script

6.5. Script-API

Script-API

- Grundfunktionen – XML-Lesen und verarbeiten – Version 1:

D:/myXML.xml

```
<myXMLFile>
  <data>
    <singleData index="1" />
    <singleData index="2" />
    <singleData index="3" />
  </data>
</myXMLFile>
```

```
local data = {};
local xmlFile = loadXMLFile("My XML", "D:/myXML.xml");

local i=0;
while true do
  local xmlPath = string.format("myXMLFile.data.singleData(%d)", i);
  local singleData = getXMLInt(xmlFile, xmlPath .. "#index");
  if singleData == nil then
    break;
  end;

  table.insert(data, singleData);
  i = i + 1;
end;
```

Script-API

- Grundfunktionen – XML-Lesen und verarbeiten – Version 2:

D:/myXML.xml

```
<myXMLFile>
  <data itemCount="3">
    <singleData1 index="1" />
    <singleData2 index="2" />
    <singleData3 index="3" />
  </data>
</myXMLFile>
```

```
local data = {};
local xmlFile = loadXMLFile("My XML", "D:/myXML.xml");

local itemCount = getXMLInt(xmlFile, "myXMLFile.data#itemCount");

for i=1, itemCount do
  local xmlPath = string.format("myXMLFile.data.singleData%d", i);

  local singleData = getXMLInt(xmlFile, xmlPath .. "#index");
  table.insert(data, singleData);
end;
```

Script-API

■ Grundfunktionen – Key-Events:

--Abfangen eines Tastendrucks

```
function keyEvent(unicode, sym, modifier, isDown)
  if sym == Input.KEY_n and isDown then
    -- do something
    print("Taste N gedrueckt ");
  end;
end;
```

■ Grundfunktionen – Text-Rendern:

--Abfangen eines Tastendrucks

```
function draw()

  setTextColor(1,1,1,1);
  -- RGB in Prozent (0-1)
  -- Alpha (0-1)
  setTextBold(true);

  renderText(0.5,0.5,0.022, "Ich bin ein Text");
  -- 0.5,0.5: Position des Textes (Anfangspunkt)
  -- 0.022: Größe des Textes
end;
```

Fragen / Feedback

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

Übungen bis zur nächsten Veranstaltung

- Aufgabe (XML-Lesen und Text schreiben):
 - Erweitere die letzte Aufgabe
 - Schreibe eine XML, in der man beliebig viele Autos definieren kann
 - Jedes Auto kann beliebig viele Gänge haben, die je Auto verschieden sein können
 - Auch die anderen WICHTIGEN Eigenschaften sollen definiert werden können
- Erweitere Dein Script um einen XML-Reader, der die XML beim Spielstart liest und automatisch daraus Autos generiert
- Diese Autos sollen in einem GLOBALEN Table gespeichert werden
- Per Taste G (`Input.KEY_g`) soll es möglich sein zwischen den verfügbaren Autos zu wechseln
- Per Taste H (`Input.KEY_h`) soll es möglich sein die Infos des aktuellen Autos zu erhalten
- Per Taste M (`Input.KEY_m`) und Taste N (`Input.KEY_n`) soll es möglich sein die Geschwindigkeit um jeweils 10kmh zu erhöhen bzw zu senken.
- Per Taste B (`Input.KEY_b`) kann man das aktuelle Auto ein- und ausschalten
- Sofern nicht die Infos angezeigt werden, sollen die letzten 10 Aktionen angezeigt werden
Tip: Statt `print()` eine eigene Funktion schreiben die Aktionen zu einem Table hinzufügt

Übungen bis zur nächsten Veranstaltung

- Infos:
 - Abgabe: Sonntag 18.07.2010
 - Nächster Termin: Sonntag 18.07.2010 – 20 Uhr