

Diplomarbeit  
LEGO Mindstorms Simulator - JORGE  
Installationshandbuch

christof.seiler@datacomm.ch  
nik@netstyle.ch  
stefan.feissli@hispeed.ch

16. Dezember 2005



# Inhaltsverzeichnis

<b>1 Einfache Installation</b>	<b>5</b>
1.1 Einleitung . . . . .	5
1.2 Installer . . . . .	5
1.3 Öffnen von JORGE Projektdateien . . . . .	7
<b>2 Entwickler Installation</b>	<b>9</b>
2.1 Einleitung . . . . .	9
2.2 Referenzierte Pfade . . . . .	9
2.3 DirectX . . . . .	9
2.4 OGRE . . . . .	9
2.5 ODE . . . . .	9
2.6 wxWidgets . . . . .	10
2.7 Visual Studio . . . . .	10
2.7.1 JorgeMain.exe . . . . .	10
2.7.1.1 Output . . . . .	11
2.7.2 JorgeLib . . . . .	11
2.7.2.1 Output . . . . .	11
2.7.3 CompassSensor . . . . .	11
2.7.3.1 Output . . . . .	11
2.7.4 LeftTouchSensor . . . . .	11
2.7.4.1 Output . . . . .	11
2.7.5 RightTouchSensor . . . . .	12
2.7.5.1 Output . . . . .	12
2.8 Bemerkungen zu Linux . . . . .	12
2.8.1 OGRE . . . . .	12
2.8.2 ODE . . . . .	13
2.8.3 JORGE . . . . .	13

## Vorwort

Dieses Dokument dient als Installationsanleitung und beschreibt alle erforderlichen Schritte um JORGE zu installieren.

Die eine Möglichkeit besteht darin, nur die ausführbaren Dateien von JORGE zu installieren, dazu stellen wir einen Windows Installer bereit. Eine nähere Beschreibung wird im Kapitel 1 gegeben.

Entwickler, die JORGE kompilieren wollen, müssen die komplette Entwicklungsumgebung einrichten. Dies wird ausführlich im Kapitel 2 beschrieben.

# Kapitel 1

## Einfache Installation

### 1.1 Einleitung

Um JORGE möglichst einfach installieren zu können, haben wir einen Installer erstellt. Sie finden die Installationsdatei, JORGEsetup.exe, im Verzeichnis [Pfad zum CVS Repository]/jorge/installer/. Dieser Installer funktioniert so, wie die meisten seiner Art.

### 1.2 Installer

Der Ablauf ist dank dem Installer selbsterklärend und beinhaltet folgende Schritte:

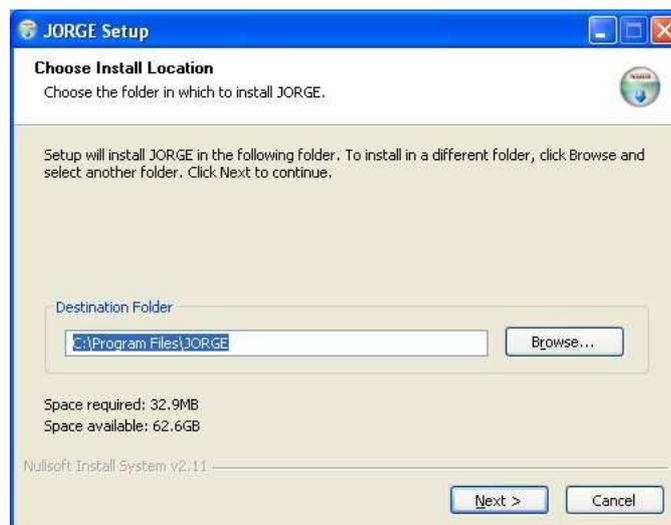


Figure 1.1: Installationsverzeichnis

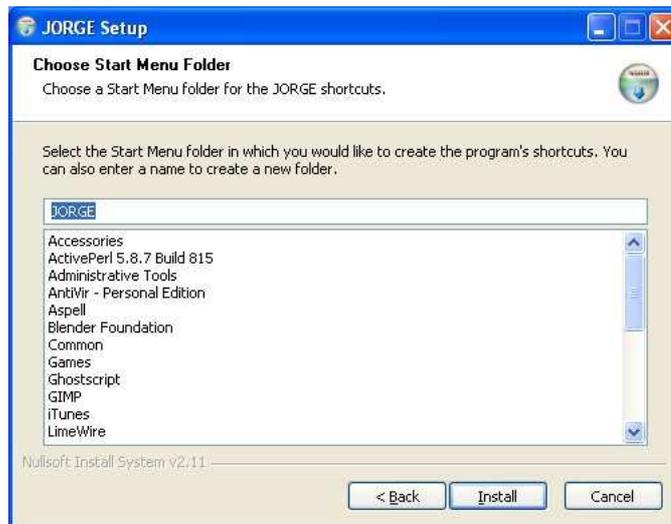


Abbildung 1.2: Start Menü Verzeichnis

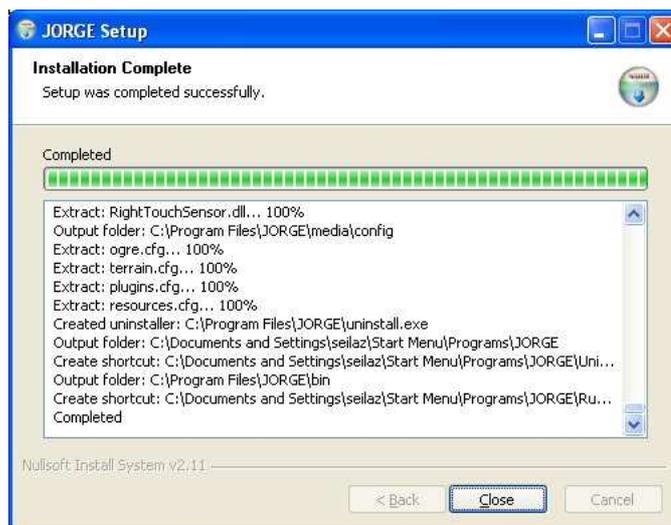


Figure 1.3: Installation komplett



Abbildung 1.4: Start Menü

Bei erfolgreicher Ausführung wird eine Verknüpfung auf das JORGE Hauptprogramm und das Deinstallationsprogramm im “Start Programme”-Ordner eingefügt, wie auf Abbildung 1.4 zu erkennen ist.

### 1.3 Öffnen von JORGE Projektdateien

Während der Installation wird die Dateiendung *.jorge* mit JORGE verknüpft (es wird ein Eintrag in der Windows Registry vorgenommen). Dies ermöglicht dem Benutzer, durch einen Doppelklick auf eine Projektdatei eine Simulation zu starten.



## Kapitel 2

# Entwickler Installation

### 2.1 Einleitung

JORGE verwendet die drei folgende Frameworks:

**OGRE** ist für die Realisierung des graphischen Teils (3D) von JORGE zuständig

**ODE** ist ein Physik Framework zur Berechnung der physikalischen Eigenschaften in der virtuellen 3D Welt.

**wxWidgets** ist ein GUI Toolkit für die Benutzeroberfläche

Diese Frameworks müssen vor JORGE kompiliert werden.

### 2.2 Referenzierte Pfade

In den nachfolgenden Kapitel beziehen wir uns auf die CVS Dateistruktur. Alle Pfadangaben sind relativ zu [Pfad zum CVS Repository]/jorge/.

### 2.3 DirectX

Um JORGE kompilieren zu können, muss das DirectX9 SDK von Microsoft installiert werden:

1. DirectX9 von <http://msdn.microsoft.com/directx/> herunterladen
2. Ausführen der heruntergeladenen Datei

### 2.4 OGRE

Um das OGRE Framework zu kompilieren, müssen folgende Schritte durchgeführt werden:

1. Entpacken der Datei *tools/Custom-Ogre.zip* im Verzeichnis *tools/*
2. Die Projektdatei *tools/ogrenew/OgreMain.sln* mit Visual Studio öffnen und einen Batch Build vornehmen, siehe Abbildung 2.1 (Bei allfälligen Fragen verweisen wir auf das README von Ogre im *tools/ogrenew* Verzeichnis)

### 2.5 ODE

Die statische Bibliothek vom ODE Framework ist bereits kompiliert.

## 2.6 wxWidgets

Ebenfalls muss das wxWidgets Framework heruntergeladen und kompiliert werden:

1. Herunterladen der Version 2.6.2 (stable release) von der wxWidgets Homepage: <http://www.wxwidgets.org/>. Dateiname: *wxMSW-2.6.2.zip*
2. Entpacken in *tools/*
3. Den Entpackten Ordner umbenennen in *wxWidgets/*
4. Datei *tools/wxWidgets/build/msw/wx.sln* mit Visual Studio öffnen und Batch Build ausführen
5. Datei *tools/editor/build/src/stc.sln* mit Visual Studio öffnen und die Konfigurationen Debug und Release kompilieren.

## 2.7 Visual Studio



Abbildung 2.1: Solution Explorer

### 2.7.1 JorgeMain.exe

Für die Entwicklung auf Windows wurde das Visual Studio .NET 2003 verwendet. Im Ordner *jorgeMain/* findet man das Verzeichnis “*scripts*” in dem eine Solutiondatei abgelegt ist, genannt *JorgeMain.sln*. Um JORGE zu kompilieren muss diese Datei geöffnet werden und ein Batch Build auf dieser Solution ausgeführt werden. Bei erfolgreicher Kompilierung werden alle benötigten Dateien an den Richtigen Ort kopiert und man kann die Applikation entweder im Debug oder Release Modus ausführen.

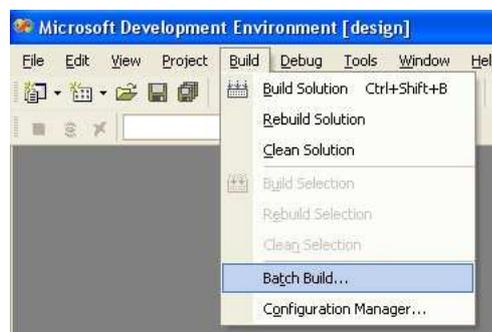


Abbildung 2.2: Batch Build

### 2.7.1.1 Output

Typ	Pfad
Debug	jorgeMain/bin/release/JorgeMain.exe
Release	jorgeMain/bin/release/JorgeMain.exe

Tabelle 2.1: JorgeMain Output

## 2.7.2 JorgeLib

Das JorgeLib Projekt stellt eine statische Bibliothek von JORGE dar. Sie wird benutzt von JorgeMain und den Sensor Plugins.

### 2.7.2.1 Output

Typ	Pfad
Debug	jorgeMain/lib/debug/JorgeLib_d.lib
Release	jorgeMain/lib/debug/JorgeLib.lib

Tabelle 2.2: JorgeLib Output

## 2.7.3 CompassSensor

Der Kompass Sensor ist eine Beispiel-Implementation eines Sensor Plugins.

### 2.7.3.1 Output

Typ	Pfad
Debug	jorgeMain/sensors/compassSensor/bin/debug/CompassSensor.dll
	media/plugins/debug/CompassSensor.dll
Release	jorgeMain/sensors/compassSensor/bin/release/CompassSensor.dll
	media/plugins/release/CompassSensor.dll

Tabelle 2.3: CompassSensor Output

## 2.7.4 LeftTouchSensor

Es bestehen spezielle Abhängigkeiten zwischen JORGE und diesem Sensor, daher empfehlen wir nicht, dieses Plugin als Referenzbeispiel zu übernehmen.

### 2.7.4.1 Output

Typ	Pfad
Debug	jorgeMain/sensors/leftTouchSensor/bin/debug/LeftTouchSensor.dll
	media/plugins/debug/LeftTouchSensor.dll
Release	jorgeMain/sensors/leftTouchSensor/bin/release/LeftTouchSensor.dll
	media/plugins/release/LeftTouchSensor.dll

Tabelle 2.4: LeftTouchSensor Output

## 2.7.5 RightTouchSensor

Gleich wie 2.7.4.

### 2.7.5.1 Output

Typ	Pfad
Debug	jorgeMain/sensors/rightTouchSensor/bin/debug/RightTouchSensor.dll
	media/plugins/debug/RightTouchSensor.dll
Release	jorgeMain/sensors/rightTouchSensor/bin/release/RightTouchSensor.dll
	media/plugins/release/RightTouchSensor.dll

Tabelle 2.5: RightTouchSensor Output

## 2.8 Bemerkungen zu Linux

JORGE funktioniert nicht unter Linux (OpenGL Problem). Wir erklären hier nur wie JORGE kompiliert werden kann.

Die Kompilation wurde auf einem Debian System (Kernel 2.6) mit gcc 3.2 durchgeführt.

### 2.8.1 OGRE

Da kleine Änderungen am OGRE Code vorgenommen werden mussten, haben wir uns entschlossen, OGRE mitzuliefern.

1. Autotools (Autoconf, Automake and friends) müssen zur Verfügung stehen. Unter Debian reicht ein `simple apt-get install automake1.7` oder höher
2. Wenn nicht bereits vorhanden, installieren Sie bitte folgende Developer Libraries: SDL 1.2.4, FreeType2, DevIL, zzip, [CEGUI 0.2.0], pkg-config sowie CG Toolkit ([http://developer.nvidia.com/object/cg\\_toolkit.html](http://developer.nvidia.com/object/cg_toolkit.html)). Diese Libraries bzw. Tools können ganz einfach per `apt-get` installiert werden. Speziell für Debian heisst das:

libfreetype6-dev	libmng-dev
libzip-dev	libglpng-dev
libtool	libglpng-dev
zlib1g-dev	libglpng
libdevil-dev	libpng12-dev
libsdl-dev	freeglut3-dev
libtiff4-dev	libglut3-dev
libtiffxx0	libglut3
libjpeg62-dev	freeglut3
libcms1-de	libxerces26-dev
libgtk2.0-dev	

3. Laden Sie `wxGTK-2.6.2.tar.gz` von <http://wxwindows.org> herunter.
4. Wechseln Sie ins Verzeichnis `tools/`
5. Entpacken Sie `wxGTK-2.6.2.tar.gz` und ändern den Namen des entpackten Ordners in `wxWidgets/`
6. Führen Sie `make, make install` im ordner `wxWidgets/` aus.
7. Wechseln Sie zurück ins Verzeichnis `tools/`
8. Entzippen Sie bitte `Custom-Ogre.zip` und `ode-0.5.zip`

9. Führen Sie nun folgende Befehle aus: `./configure --with-cfgtk=gtk`, `make` und `make install` (das letzere als root)
10. Mehr Informationen können dem README im Verzeichnis `jorge/tools/ogrenew` entnommen werden

### 2.8.2 ODE

1. Um Ode zu kompilieren wechseln Sie bitte in das `tools/ode-0.5` Verzeichnis
2. Führen Sie `make configure` und `make ode-lib` auf der Konsole aus
3. Anschliessend muss das lib sowie das include Verzeichnis an die entsprechenden Orte kopiert werden, auf Unix: `include/ode -> /usr/local/include/ode` und `lib/libode.a -> /usr/local/lib/libode.a`

### 2.8.3 JORGE

Wir haben ein Projekt für Codeblocks (<http://www.codeblocks.org>) erstellt:

1. Um JORGE zu kompilieren öffnen Sie im Verzeichnis `jorgeMain/scripts/codeForge` die Datei `jorgeMain.cbp`
2. Kompilieren Sie JORGE.
3. Und lösen Sie das OpenGL Problem!