

## Übungen zum Aufbau interaktiver 3D-Engines

Sommersemester 2013

### Blatt 1

#### Übungsbetrieb

In der Übung am Dienstag wird ein Aufgabenblatt verteilt, das bis zum Freitag um 23:59 Uhr der darauffolgenden Woche zu bearbeiten ist. Die Aufgabenblätter befinden sich auch auf der Veranstaltungswebseite (<http://www-lehre.inf.uos.de/~ai3de>) sowie im Stud.IP.

Die Übungen finden dienstags von 12:15 – 13:45 in Raum 31/304 statt. In der Übung wird alle zwei Wochen (beginnend am 23.04.2013) eine Kurzpräsentation einer Lösung des letzten Blattes und die Vorstellung des nächsten Übungsblattes durchgeführt. An den restlichen Terminen wird alle zwei Wochen (beginnend am 16.04.2013) ein Austausch der Teilnehmer und Hilfestellung zum laufenden Übungsblatt stattfinden.

#### Testbetrieb

Jede Gruppe wird mindestens einmal ihre Lösung wie oben beschrieben während einer Übung vorstellen. Neben der Besprechung des aktuellen Lösungsverlaufes in der Übung wird es für jede Gruppe ein Feedback per Mail geben.

#### Mailingliste

Jeder Teilnehmer, der sich bis zum 08.04.2013 in der Veranstaltung im Stud.IP eingetragen hat, wurde automatisch bei der Mailingliste angemeldet. Die restlichen sollten dies unter <http://list.serv.uni-osnabrueck.de/mailman/listinfo/ai3de> tun. Über die Mailingliste werden Änderungen im Vorlesungs- oder Übungsbetrieb, sowie eventuelle Fehlerkorrekturen der Aufgabenblätter mitgeteilt. Zudem soll sie als Hilfestellung dienen, damit Probleme untereinander diskutiert werden. Dazu einfach eine Mail an [ai3de@list.serv.uni-osnabrueck.de](mailto:ai3de@list.serv.uni-osnabrueck.de).

#### Benotung

Die ordentliche Bearbeitung der Arbeitsblätter und Mitarbeit und Anwesenheit bei den Übungsterminen geht zu 1/5 in die Endnote ein.

## Aufgabe 1: git Zugang einrichten (20%)

Folgen Sie diesen Punkten, um das Übungs-Repository und das der eigenen Gruppe zu beschaffen:

1. Installieren Sie git von <http://git-scm.com/>.
2. Starten Sie "git bash" und konfigurieren Sie git mit folgenden Befehlen
  - a. 'git config --global user.name "Your name here" ' <-- Stellen Sie Ihren Benutzernamen ein
  - b. 'git config --global user.email "xxx@uos.de" ' <-- Stellen Sie ihre E-Mail-Adresse ein
3. Prüfen Sie mit 'pwd', ob Sie sich in ihrem Nutzerverzeichnis befinden und wechseln sie gegebenenfalls mit 'cd \$HOME' dorthin.
4. Prüfen Sie mit 'ls .ssh', ob schon ein .ssh-Verzeichnis mit den Dateien id\_rsa und id\_rsa.pub vorliegt. Wenn nicht, bei Schritt 5. fortfahren, wenn schon, bei Schritt 7.
5. Erstelle einen SSH-Key mit folgenden Befehlen
  - a. 'ssh-keygen -t rsa -C "xxx@uos.de" ' <-- Die interaktiven Abfragen mit Enter bestätigen
6. Nach erfolgreicher Erstellung sollten nach Eingabe von 'ls .ssh' die gewünschten Dateien vorhanden sein.
7. Senden Sie eine Mail mit der Datei 'id\_rsa.pub' und dem Namen Ihrer/s Gruppenpartnerin/s an [ewittkorn@uos.de](mailto:ewittkorn@uos.de) - Sie erhalten Schreibrechte und den Namen Ihres Gruppen-repositories.
8. Begeben Sie sich mit der "git bash" zu einem Ordner für die Daten der Übung.
9. Führen Sie 'git clone git@vm205.rz.uos.de:ai3de-uebung.git' aus, um die Daten der Übung zu erhalten.
10. Führen Sie 'git clone git@vm205.rz.uos.de:ai3de-gruppeX.git' aus, um das repository Ihrer Gruppe zu erhalten.
11. Kopieren Sie den Code des ersten Übungsblattes in ihr Gruppenverzeichnis und bearbeiten Sie ihn wie in der Übung besprochen.

Weitere Hinweise:

- Ein gutes git-GUI für Windows kann man unter <http://code.google.com/p/gitextensions/> finden.
- Die Einstellung von git wurde von <https://help.github.com/articles/set-up-git> und die Erstellung des ssh-keys von <https://help.github.com/articles/generating-ssh-keys>
- Das Tutorial unter <http://try.github.io/levels/1/challenges/1> lohnt sich für das bessere Verständnis von git
- Sämtliche Funktionalitäten von git sind im folgenden Buch nachzulesen <http://git-scm.com/book>

## Aufgabe 2: eine OpenGL- Test-Applikation starten (10%)

Folgen Sie diesen Punkten, um ein kleines OpenGL-Programm in Ihrer IDE einzurichten und zu starten:

1. Kopieren Sie den Aufgaben-Code aus dem "OpenGL"-Verzeichnis in ihr Gruppen-repository.
2. Erstellen Sie ein neues Java-Projekt von diesem existierendem Code und fügen Sie die Bibliotheken "lwjgl\_util.jar" und "lwjgl.jar" aus dem "libs/lwjgl-2.8.5/jar/"-Verzeichnis des Übungs-repositories dem Build Path hinzu.
3. Setzen Sie den Pfad zu den natives
  - a. Eclipse: Rechtsklick aufs Projekt -> Properties -> lwjgl.jar aufklappen -> "Native library location" auf "Übungs-repository / libs / lwjgl-2.8.5 / native / "Das vorliegende Betriebssystem" " stellen.
  - b. NetBeans: Rechtsklick aufs Projekt -> Properties -> Run -> Bei "VM Options" " -Djava.library.path="Übungs-repository"/libs/lwjgl-2.8.5/native/<linux|macosx|solaris|windows>" eintragen.
4. Starten Sie das Projekt und erfreuen Sie sich an dem Symbol von Kraft, Mut und Weisheit ☺

## Aufgabe 3: Der Logger (15%)

Folgen Sie diesen Punkten, um den prototypischen Logger zu vervollständigen:

1. Kopieren Sie den Aufgaben-Code aus dem "Logger"-Verzeichnis in ihr Gruppen-repository
2. Erstellen Sie ein neues Java-Projekt von diesem existierenden Code.
3. Machen Sie sich mit der Klasse "Logger.java" vertraut.
4. Implementieren Sie die Methode "getSrcString()" so, dass sie den String "Dateiname der aufrufenden Datei, Zeilennummer: " zurückgibt. Benutzen Sie dazu den Inhalt des Arrays stack[] und die Java-API Seite <http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/StackTraceElement.html>

Der Output:

```
[INFO] KeineAhnung.java, 42: Eine nette Info  
[WARNING] KeineAhnung.java, 42: Eine kleine Warnung  
[ERROR] KeineAhnung.java, 42: Ein gefaehrlicher Fehler
```

Soll nacher so aussehen:

```
[INFO] Main.java, 23: Eine nette Info  
[WARNING] Main.java, 25: Eine kleine Warnung  
[ERROR] Main.java, 27: Ein gefaehrlicher Fehler
```

## Nützliche Zusatzaufgaben (5%)

1. Richten Sie sich im Editor ihrer Wahl ein Code-Template für die verschiedenen Logger-Aufrufe ein.
2. Implementieren Sie eine Art, den Logger-Output zu filtern und nur Ausgaben von bestimmten Klassen anzuzeigen.

#### Aufgabe 4: XML-Parsing (50%)

Folgen Sie diesen Punkten um eine XML-Datei nach dem DOM-Model und mittels SAX-Parsing in eine Liste von Java-Objekte zu parsen:

1. Kopieren Sie den Aufgaben-Code aus dem "XMLParsing"-Verzeichnis in ihr Gruppen-repository.
2. Erstellen Sie ein neues Java-Projekt von diesem existierendem Code und fügen Sie die Bibliothek "lwjgl\_util.jar" aus dem "libs/lwjgl-2.8.5/jar/"-Verzeichnis des Übungs-repositories dem Build Path hinzu.
3. Machen Sie sich mit den Klassen im "model"-package und der XML-Datei xml/Actors.xml vertraut.
4. Implementieren Sie für jede der "model"-Klassen doe "toString()"-Methode, eine Beispielausgabe könnte so aussehen:

```
Actor with Name: CrazyCube and ComponentCount: 3
```

```
----- ActorComponents start -----
```

```
RenderComponent with objMesh: meshes/cube.obj and SphereCount: 1
```

```
----- BoundingSpheres start -----
```

```
BoundingSphere mit Radius: 1.75
```

```
----- BoundingSpheres end -----
```

```
PhysicsComponent with BoxCount: 2
```

```
----- Boxes start -----
```

```
Box with Size (2.0, 2.0, 2.0) and Density: 10.0
```

```
Box with Size (1.5, 7.3, 6.8) and Density: 5.0
```

```
----- Boxes end -----
```

```
TransformComponent with Position: (0.0, 10.0, 8.0) and Rotation: (40.0, 20.0, 10.0)
```

```
----- ActorComponents end -----
```

5. Implementieren Sie in der Klasse "DOMSceneParser.java" die Methode "parse(...)" so, dass die zurückgegebene Liste korrekt erstellte Instanzen der model-Klassen enthält.
6. Implementieren Sie in der Klasse "SAXSceneParserHandler.java" die Methoden "startElement()" und "endElement()" so, dass die gleiche Liste, wie beim DOM-Parsing zurückgegeben wird.
7. Überprüfen Sie den Inhalt der Liste anhand der Ausgabe (Sofern die toString() Methoden korrekt implementiert werden).

#### Zusatzaufgabe (10%)

Folgen Sie diesen Punkten, um eine XML-Datei zu erstellen:

1. Implementieren Sie in der Klasse SceneXMLWriter.java die Methode "createSceneDocument()" so, dass durch Lesen der geparsen ActorListe ein DOM Document-Objekt erstellt wird, das mit der Hilfsfunktion writeSceneXML in eine neue XML-Datei geschrieben werden kann (zur Hilfe in die weiteren Hinweise gucken!).
2. Vergleichen Sie die erstellte XML-Datei mit der Actors.xml.

Weitere Hinweise:

- API fürs DOM-Parsing <http://docs.oracle.com/javase/1.5.0/docs/api/org/w3c/dom/package-summary.html>
- API fürs SAX-Parsing <http://docs.oracle.com/javase/1.4.2/docs/api/org/xml/sax/package-summary.html>
- Tutorial zur XML-Erstellung: <http://www.mkyong.com/java/how-to-create-xml-file-in-java-dom/>

Viel Erfolg 😊