
Programmieren in Java

<http://proglang.informatik.uni-freiburg.de/teaching/java/2010/>

Java-Projekt 1

2010-06-14

1 Subversion (jeder für sich)

Subversion. Wir verwenden die Versionsverwaltung *Subversion*. Ein Subversion-Server hält zeitliche Schnappschüsse eines Quellcodebaums als *Versionen* fest, wobei jede Version einen Änderungskommentar hat. Versionen können auch miteinander verglichen werden, und man kann sich Dateien aus einer früheren Version holen, wenn man etwas verschlechtert hat.

In der Eclipse im Pool ist als Subversion-Client das Plug-in Subversive installiert. Aus Lizenzgründen wird bei Subversive die Bibliothek zum Ansprechen des Servers separat installiert. Das machen wir also als erstes.

Aufgabe 1 (Subversive einrichten)

Folgen Sie den Instruktionen auf

<http://proglang.informatik.uni-freiburg.de/teaching/java/2010/project/svn.html>

(auch von der Vorlesungshomepage erreichbar: Projekt-„Auscheck-Anleitung“).

Aufgabe 2 (Commit+Update)

Das Java-Projekt „svnplayground“ ist für Experimente mit Subversion gedacht.

- Legen Sie dort im Package `animals` eine Klasse an, die nach Ihrem Lieblingstier benannt ist (z.B. `Horse`).
- Rechtsklicken Sie auf das `svnplayground`-Projekt und wählen Sie „Team/Synchronize with Repository“. Sie gelangen nun in die Synchronize-Perspektive.
- In der Synchronize-View sehen Sie sowohl hereinkommende (\Leftarrow) als auch ausgehende (\Rightarrow) Änderungen am Projekt. Rechtsklicken Sie auf Ihre Lieblingstier-Klasse und wählen Sie „Commit“ (also veröffentlichen, hochladen, eben committen). Tragen Sie im Änderungskommentar ein, dass Sie Ihr Lieblingstier neu angelegt haben.
- Wenn Ihre Teampartner auch alle ein Tier veröffentlicht haben, synchronisieren Sie erneut. Sie sollten jetzt fremde Lieblingstiere in „incoming changes“ sehen. Markieren Sie alles und wählen Sie „Update“ aus dem Kontextmenü. Jetzt sollten die anderen Lieblingstiere auch bei Ihnen verfügbar sein.
- Wenn mehrere Leute auf derselben Datei Änderungen anbringen, gibt es Konflikte. Auch damit kann Subversion umgehen, aber das lernen wir heute nicht. Konflikten kann man ausweichen, indem man Absprachen trifft, wer gerade in welchen Dateien und Packages unterwegs sein darf.

2 twodeedoo (teamweise)

Das Beispiel-Spiel zur twodeedoo-Engine ist schon installiert. Dort gibt es eine Klasse `Runner`. Führen Sie sie als Java-Application aus. Wenn alles gutgeht, sehen Sie ein Auto, das über ein Fenster fährt und mit rechts+links gesteuert werden kann.

Aufgabe 3 (Drivester umbauen)

Kern des Spiels "Drivester" ist die Klasse `DrivesterGameApp`.

- Die Datei `carsprites.png`, einem Hobby-Game-Designer-Forumsposting entnommen¹, enthält viele Ansichten verschiedener Autos. Momentan wird das schwarze Auto verwendet. Ändern Sie die Initialisierung der Puppets so ab, dass das rostrote Auto (256 Pixel weiter rechts in derselben Datei) verwendet wird. (Interessanter Weg: Navigieren Sie mit `Ctrl+Shift+T` zu `IPuppet`, lassen Sie mit `F4` die Typhierarchie anzeigen, laufen Sie so zu der `Image-Puppet`-Klasse, dort zum Konstruktor und suchen Sie mit `Ctrl+Shift+G` nach allen Stellen im Programm, wo die Klasse instanziiert wird.)
- Das Auto wird von der Klasse `CarActor` gesteuert, die auch Tastatureingaben für das Auto behandelt. Wenn der Spieler die Leertaste drückt, soll das Auto auf die Hälfte der Geschwindigkeit abgebremst werden. Basteln Sie das in `CarActor` hinein.

3 Bomberman (teamweise)

Bomberman ist ein Spiel, in dem mehrere Figuren in einem rechtwinkligen Labyrinth aus Mauern (auf einem Gitter) herumlaufen und versuchen, einander umzubringen. Dazu legt man Bomben, die nach einiger Zeit explodieren und die umgebenden Felder vorübergehend mit tödlichem Feuer ausfüllen.

Diese Woche fangen wir mit dem Gitter an. Die folgenden Fragen helfen vielleicht beim Nachdenken:

- Wie wollen Sie das Gitter repräsentieren²?
- An jeder Stelle des Gitters kann zunächst ein Mauerstein oder nur Luft stehen (später aber auch eine Bombe oder eine Spielfigur). Vielleicht ein gemeinsames Interface³?
- Vermutlich ist es geschickt, jeden Mauerstein durch eine eigene Puppe zeichnen zu lassen? Was bedeutet das für Ihre Klassen?
- Die Klasse `DrivesterGameApp` eignet sich vielleicht als Kopiervorlage für die Hauptklasse Ihres Bomberman-Spiels. Was müssen Sie entfernen, damit nichts von Autos mehr übrigbleibt?

¹<http://www.rpgmakervx.net/index.php?showtopic=2310>

²Tipp: Blatt 5.

³Tipp: Blatt 3.