



FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics

*Fortgeschrittene objektorientierte Programmierung
LVA 185.211, VU, 3 ECTS, 2013 S*



2. Übungsaufgabe

Complang
Puntigam
Über mich
Research
Lehre

Lernen Sie Smalltalk und entwickeln Sie ein Programm in Smalltalk (siehe unten). Ziel der Aufgabe ist das Sammeln von Erfahrungen zur dynamischen objektorientierten Programmierung. Es gibt mehrere frei verfügbare Smalltalk-Systeme. Prinzipiell sind alle davon zur Lösung der Aufgabe geeignet. Empfohlen wird das Smalltalk-System Squeak, das (im Vergleich zu einigen anderen, eher historisch interessanten Systemen) relativ neu ist und sich gut zum Programmieren von Spielen ohne besondere Vorkenntnisse eignet.

Squeak:

Squeak ist eine Open-Source-Implementierung von Smalltalk-80, die auf vielen Plattformen verfügbar ist. Details der Installation hängen von der Plattform ab. Beachten Sie, dass Sie neben dem eigentlichen Interpreter noch einige Dateien brauchen, in denen der aktuelle Systemzustand enthalten ist. Wenn Sie in Smalltalk programmieren, verändern Sie diese vorgegebenen Dateien, statt für Ihren Quellcode eigene Dateien anzulegen. Wenn Sie mehrere Versionen haben wollen, brauchen Sie auch mehrere solche vorgegebenen Dateien. Am wichtigsten ist die Datei "squeak.image", die das gesamte System in einem ausführbaren Zwischencode enthält. Die Datei "squeak.changes" enthält alle Änderungen des Smalltalk-Codes und "SqueakV41.sources" (vielleicht mit einer anderen einkodierten Versionsnummer) den Source-Code des Grundsystems. Wenn Sie "squeak" nicht in dem Verzeichnis aufrufen, in dem diese Dateien zu finden sind, müssen Sie die Dateien beim Aufruf explizit angeben.

Squeak unterstützt zwei grundsätzlich verschiedene Subsysteme zur Programmierung grafischer Oberflächen – das klassische "MVC" (Model-View-Controller, wie in Smalltalk-80) und das neuere "Morphic". Sie können eines der beiden Subsysteme wählen. Vermeiden Sie die gemischte Verwendung, bei der es zu unerwarteten Wechselwirkungen kommen kann.

Aufgabe:

Schreiben Sie das Spiel aus der 1. Aufgabe in einer Smalltalk-Umgebung neu, wobei das ganze Spiel aber nur auf einem Rechner in einem Fenster laufen soll. Es ist also nicht nötig, dass Spieler sich remote verbinden können. Um trotzdem mehreren Spielern das gleichzeitige Spielen zu ermöglichen, sollen über dieselbe Tastatur (aber über unterschiedliche Tasten) mindestens zwei Schlangen gleichzeitig steuerbar sein.

In Smalltalk schreibt man (anders als in eher statischen Sprachen) in der Regel nicht einfach ein Programm für einen bestimmten Zweck, sondern man adaptiert und erweitert die vorhandene Umgebung. Ein Ziel der Aufgabe ist es, diesen Programmierstil kennenzulernen. Vor allem zur Programmierung der grafischen Oberfläche empfiehlt es sich, ein anderes Programm als Vorlage zu verwenden und dieses abzuändern. Dabei werden Sie lernen, sich im Smalltalksystem zurechtzufinden. Eine Schwierigkeit (vor allem in der GUI Programmierung) besteht darin, dass das API (Application Programming Interface) keine gute statische Dokumentation hat. Smalltalk-Programmierer lernen eher durch das Lesen und die Verwendung bestehenden Smalltalk-Codes als aus der Dokumentation. Aus diesem Grund sind Kommentare besonders wichtig, und jede Klasse und Methode, die Sie selber schreiben, soll durch Kommentare klar spezifiziert sein.

Für Squeak gibt es unter anderem die Zusatzpakete “Etoys” und “Scratch”, die die Erstellung einiger Spiele stark vereinfachen. Zur erfolgreichen Lösung der Aufgabe ist es notwendig, dass Sie im eigentlichen Smalltalk-System Code schreiben bzw. adaptieren, also nicht einfach nur solche Zusatzpakete verwenden, um sich ein Spiel generieren zu lassen.

Die einfache Änderbarkeit des Systems kann sich auch negativ auswirken. Insbesondere wird es dadurch schwieriger, Programme von einem System auf ein anderes zu übertragen. Zur Verringerung dieses Problems wurden Versionsverwaltungssysteme wie z.B. Monticello eingeführt. Verwenden Sie ein solches System, um das von Ihnen entwickelte Spiel portabel zu gestalten.

Ein Grundkonzept hinter Smalltalk nennt sich “Personal-Computing” (PC), das heißt, jeder Benutzer hat sein eigenes System – ein Begriff, der schon lange losgelöst von Smalltalk verwendet wird. Obwohl dieses Konzept sehr erfolgreich war und ist, stößt es bei der Zusammenarbeit mehrerer Personen gelegentlich an seine Grenzen. Um diese Problematik klarer zu machen (und aus offensichtlichen didaktischen Gründen) sollen alle Mitglieder jeder Übungsgruppe gemeinsam an der Lösung dieser Aufgabe arbeiten. Es reicht nicht, wenn nur ein Teil der Gruppe die Arbeit macht und dem Rest der Gruppe die Lösung erklärt. Die Mitarbeit an dieser Aufgabe kann auch bei Abgabegespräch und Prüfung überprüft werden.

Anfang | HTML 4.01 | letzte Änderung: 2013-03-21 (Puntigam)