

Übung 7 – Multimedia-Programmierung

Inhalt

Einführung in Squeak: EToys und Smalltalk

Aufgaben

Kopieren Sie sich den Ordner `/home/proj/mi_mmp_ss06/squeak_workspace/` in ihr Homeverzeichnis. Wechseln Sie in das neu erstellte Verzeichnis und starten Sie dort Squeak mit dem Befehl `squeak`.

Hinweis:

Im CIP-Pool ist die Version 3.6 installiert (ist auf squeak.org im FTP-Archiv verfügbar).

1.) EToys

Erste Schritte

- Öffnen Sie ein neues *Morphic*-Projekt (*Open...* -> *Morphic Project*). Mit *Save project on file* können Sie das einzelne Projekt speichern.
- *Save...* bzw. *Save as...* speichert den gesamten Workspace, d.h. nach dem nächsten Öffnen befindet sich Squeak genau im gleichen Zustand wie im Moment des Speicherns. Mit *Jump to project...* oder mit den Knöpfen im *Navigator-Tab* können sie zwischen Projekten wechseln oder auf die oberste Welt-Ansicht gelangen.
- Öffnen Sie das Zeichnen-Werkzeug (z.B. im *Navigator-Tab*) und zeichnen Sie einen Käfer. Übernehmen Sie die Zeichnung mit *Keep*
- Öffnen Sie mittels des *Halo*-Menüs einen *Viewer*. Betrachten Sie die Werte und Anweisungen der verschiedenen Kategorien. Wichtig: Die Kategorie können Sie wechseln, indem Sie auf den Kategorie-Namen (z.B. *basic* oder *tests*) klicken.
- Beachten Sie, dass alle Werte im *Viewer* stets aktuell sind (z. B. beim Ziehen des Käfers mit der Maus). Tragen Sie in der Titelleiste des *Viewers* einen Namen für Ihr Käfer-Objekt ein (z.B. *Bug* statt *Sketch*).
- Lassen Sie die Anweisung *forward by* ausführen. Richten Sie den Käfer so aus, dass er durch *forward by* tatsächlich auch in Kopfrichtung bewegt wird. (Dazu rufen Sie im *Halo*-Menü *rotate* auf, dann erscheint nach Beendigung der Rotation ein grüner Pfeil auf dem Käfer, der dessen Ausrichtung definiert.)

Skripte visuell erstellen

- Erstellen Sie ein neues leeres Skript für den Käfer (Menü in der Titelleiste des *Viewers* oder Kategorie *scripts* und dort *empty script* auf die Welt ziehen). Benennen Sie das Skript als *move*.
- Fügen Sie dem Skript mittels Drag'n Drop die eine Anweisung hinzu, die den Käfer um 10 Einheiten vorwärts bewegt. Fügen Sie eine zweite Anweisung hinzu, die den Käfer um 10 Einheiten dreht.

- Starten und beenden Sie das Skript testweise mittels Klicken auf das Uhr-Symbol im Skript-Fenster. Beachten Sie, dass **nicht** zwischen Editier- und Ausführungs-Modus unterschieden wird, d.h. Sie können während der Skript-Ausführung weiter editieren oder den Käfer mittels Drag'n Drop an eine andere Position setzen. Die Werte im Viewer (z. B. die x-Koordinate) werden stets aktualisiert.
- Beachten Sie die Auswahl-Möglichkeiten für die Skript-Ausführung, die sich direkt rechts neben dem Uhr-Symbol befinden (Standardwert *paused* bzw. *ticking*). Hier können Sie z.B. definieren, wenn das Skript Ereignis-gesteuert ausgeführt werden soll.
- Ziehen Sie aus dem *Supplies-Tab* ein *Joystick*-Objekt auf Ihre Welt. Öffnen Sie dessen Viewer und darin die Kategorie *joystick*. Definieren Sie mittels Drag'n Drop (vom Viewer des Joysticks zum Skript des Käfers), dass der Käfer sich nun abhängig von der rechts-links-Ausrichtung des Joysticks drehen soll (multipliziert mit dem Faktor 10).
- Erstellen Sie eine Blume, die sich dreht. Erstellen Sie mittels des Halo-Menüs mehrere Zwillinge der Blume (im Halo-Menü den Punkt *Menu* aufrufen). Das Skript soll dabei mit dupliziert werden.
- Ziehen Sie aus dem *Widgets-Tab* eine Knopf-Leiste *All Scripts* auf Ihre Welt und verwenden Sie diese, um alle vorhandenen Skripte gleichzeitig zu steuern.
- Fügen Sie dem Käfer eine Instanzvariable *flowerCount* hinzu (Menü in der Kopfleiste des Viewers) und initialisieren diese mit 0.
- Erweitern Sie das Skript des Käfers (*move*) um eine If-Abfrage (If-Abfrage kann aus der Titelleiste des Skript-Fensters gezogen werden!). Wenn der Käfer über eine Blume fährt, soll seine Variable *flowerCount* um 1 erhöht werden. Verwenden Sie dazu den Test auf Farben (*color sees*). Damit der Test nicht zu oft *true* liefert, können Sie z.B. der Blume einen kleinen Mittelpunkt zeichnen. Änderungen an der Vorlage können für alle Zwillinge übernommen werden (im Halo-Menü den Punkt *Menu* -> *siblings...*-> *make all siblings look like me* aufrufen). Falls das Skript nicht funktioniert, überprüfen Sie, ob auch alle Teile beim Drag'n Drop tatsächlich eingerastet sind (wird durch grüne Farbe **vor** dem Loslassen angezeigt).
- Fügen Sie Ihrer Welt einen *Watcher* für die Variable *flowerCount* hinzu, damit der Variablenwert auch ohne Viewer sichtbar bleibt.

2.) Smalltalk

- Öffnen Sie ein Browser-Fenster und machen Sie sich damit vertraut (siehe Vorlesung).
- Erstellen Sie eine neue Kategorie *MyCategory* für Ihre eigene Klasse.
- Graphische Objekte sind Unterklassen der Klasse *Morph*. Es existieren bereits viele vordefinierte Unterklassen von *Morph*, z.B. in den Kategorien (*Morphic-Kernel*, *Morphic-Basic*, *Morphic-Demo*, *Morphic-...*). Erstellen Sie in Ihrer eigenen Kategorie (*MyStuff-Ue6*) eine Unterklasse von *EllipseMorph* mit dem Namen *TestMorph*. Beachten Sie, dass alle Änderungen am Code (z.B. im Browser) immer mit *accept* (Menü oder *Alt+s*) bestätigt werden müssen.
- Öffnen Sie ein Workspace-Fenster, um dort direkt Smalltalk-Anweisungen einzugeben. Legen Sie in einer Variablen eine Instanz Ihrer Klasse *TestMorph* an. Machen Sie die Instanz auf dem Bildschirm sichtbar, indem Sie die geerbte Methode *openInWorld* aufrufen.
- Ergänzen Sie Ihre Klasse *TestMorph* um eine Methode *initialize*. Diese soll zuerst die Methode *initialize* Ihrer Oberklasse aufrufen (*super initialize.*). Anschließend soll die Farbe der Ellipse auf grün gesetzt werden (*color := Color green.*). Vergessen Sie nicht, mit *accept* zu bestätigen.

Verbindung von EToys und Smalltalk-Code:

- Legen Sie für Ihren Käfer in dessen Viewer ein neues Skript an, beim Ereignis *mouseLeaveDragging* ausgelöst wird (in der Auswahl recht des Uhr-Symbols festlegen).
- Schalten Sie für Ihr Skript-Fenster die Ansicht auf *textual Code* um (in der Titelleiste). Erstellen Sie dort eine Instanz Ihrer Klasse *TestMorph* in der Welt. Die x- und y-Koordinaten sollen der aktuellen Position des Käfers entsprechen:

```
script1
| tm |
tm := TestMorph new openInWorld.
tm x: self getX.
tm y: self getY.
```

(script1 entspricht hier dem Skriptnamen in der Titelleiste des Fensters)

Abgabe der Aufgabe:

Speichern Sie Ihr Projekt als pr-Datei mittels *Save project on file...* (im World-Menü) unter dem Namen *VornameNachname*. Senden Sie die Datei bis spätestens 31. Juli 2006 per Email an Andreas Pleuß. Es werden nur individuelle Lösungen angenommen.