

Dokumentation PhidgetLab 2014

Abhängigkeiten zu externen Klassen

Alle Abhängigkeiten unserer Klassen zu Klassen anderer Packages, sind über ‘* Package-Name Kategorie’ gekennzeichnet. Dadurch werden diese beim Erstellen, der SAR-Files mitexportiert.

Portieren auf Etoys

Beim Portieren auf Etoys haben folgende Klassen, bzw. Methoden gefehlt, die wir dann über oben beschriebene Abhängigkeiten zu unserem SAR-File hinzugefügt haben.

Fehlende Packages:

- MultiColumn Lists-Core

Fehlende Klassen:

- MixedCurveMorph (nur Etoys5)
- CurvierMorph (nur Etoys5)

Fehlende Methoden:

- PolygonMorph>>rectOval (nur Etoys5)
- PolygonMorph>>diamondOval (nur Etoys5)
- PolygonMorph>>createLineFrom:to:color:width: (nur Etoys5)
- IconicButton>>initializeWithThumbnail: aThumbnail withLabel: aLabel andColor: aColor andSend: aSelector to: aReceiver with: aPhidget

Bilder laden, speichern

In Squeak gibt es zwei Möglichkeiten auf Bilder zugreifen. Entweder werden sie beim Zugriff ins Image herein geladen oder sie liegen bereits im Image als Strings vor und müssen nur noch konvertiert werden.

Die Klassenmethode PhdImageImporter>>getString: aName im Phidget-Tools Package erleichtert das Laden von Bildern ins Image und das Speichern der Bilder als String. Um die Methode zu nutzen, müssen die zuladenden PNG-Bilder im Verzeichnis Resources liegen. Die Methode wird vom Workspace aus aufgerufen (z.B. PhdImageImporter getString: #vibration), wobei die Ausgabe der Methode im Workspace ausgegeben werden kann (Strg+P). Zusätzlich wird die Ausgabe auch in ein eventuell geöffnetes Transcript geschrieben.

Der ausgegebene String muss anschließend in die Rückgabe der entsprechenden Methode (Kategorie: importedImages) im PhdETImageLoader kopiert werden.

Erstellen des SAR-Files (inklusive Bibliothek)

Die Klassenmethode `PhdSarCreator>>createSar` erstellt ein SAR-File mit allen benötigten Abhängigkeiten. Die Methode wird vom Workspace aus aufgerufen (z.B. `PhdSarCreator createSar`). Beim Ausführen wird nach den Commit Messages befragt; diese müssen nicht spezifiziert werden und können einfach bestätigt werden.

Damit das Packet `Morphic-Basic-NewCurve` exportiert wird, muss es vorher zum Monticello Browser hinzugefügt werden.

Unter Resources findet sich dann die `PhidgetLab.sar` (enthält bereits `PhidgetPlugin.dll`, Projektklassen und alle Abhängigkeiten).

Umrechnung der Sensorwerte

Für das Umrechnen der Sensorwerte in verständliche Einheiten, haben wir in der Beschreibung der Hardwarekomponenten Formeln gefunden, die wir zur Berechnung verwenden.

Die Formel für den Drucksensor liefert (leider) keine passablen Werte. Deshalb haben wir Werte ermitteln und aus diesen eine Funktion berechnet.

Formeln zur Umrechnung lassen sich finden unter <http://www.phidgets.com/products.php?category=1> mit Klick auf den entsprechenden Sensor und anschließend bei "Resources" auf "User Guide" klicken.

Warum gibt es eine eigene Klasse, die von Iconic Button erbt?

Die Bilder für die dargestellten Phidgetsensoren im InterfaceKit sind als `IconicButtons` realisiert. Damit hier auch das richtige Bild angezeigt werden kann, muss in der `initialize` Methode das zu erzeugende Phidget mit übergeben werden. Deshalb wurde die Methode

```
IconicButton>>initializeWithThumbnail: aThumbnail withLabel: aLabel andColor: aColor
andSend: aSelector to: aReceiver with: aPhidget
```

zusätzlich eingefügt.

Weiterhin sieht die Klasse `Iconic Button` unter `Etoys5` im Gegensatz zu einem `Squeak Image` geringfügig anders aus. Dies hat den Effekt, dass unter `Etoys5` beim Klick auf einen solchen Button die Methode `IconicButton>>launchPartFromClick` ausgeführt wird. Diese existiert allerdings in einem `Squeak Image` nicht. Deshalb wurde eine leere Methode diesen Namens hinzugefügt, um dieses Event auszuführen.

Wie wird dafür gesorgt, dass beim Wechsel der Projekte immer das aktuelle Interface-Kit angeschlossen ist?

Beim Schließen eines Projektes ruft das geöffnete InterfaceKit immer die Funktion "closeConnection" auf und wird grau. Beim wieder öffnen dieses Projektes (oder auch ein anderes Projekt) wird die Methode PhdETPhidget>>refresh aufgerufen. Sie versucht über PhdETPhidget>>openLastConnection eine Verbindung mit dem zuletzt angeschlossenen InterfaceKit aufzunehmen. Anschließend wird noch über PhdCrPhidget>>checkValid die entsprechende ID erneuert, so dass die bereits herausgezogenen Ports registriert sind und Werte liefern.

Bekannte Bugs

- Der Anschluss von 2 Interface-Kits ist u.U. problematisch
- Teilweise wird beim Schließen eines Interface-Kits die zugehörige cr Klasse nicht gelöscht. Diese wird dann auch immernoch im PhidgetManager angezeigt. Wenn man dann die Hardware anschließt, verbindet sich automatisch die alte cr Klasse mit der Hardware und man kann kein Interface-Kit benutzen. Wir haben noch nicht herausgefunden, wann das passiert.
Dirty Fix: Beim Durchlaufen der Tests werden diese gelöscht.

Future Work

- Rotoscope für ServoMotor
Das Bild des ServoMotors den Bildern der Sensoren anpassen, also eine realistische Abbildung des SensorMotors.
- Animation der Sensoren
Die Sensoren haben zur Zeit noch statische Bilder. Beispielsweise beim Joystick könnte der aktuelle Ausschlag durch eine Linie dargestellt werden.
Als Referenz kann dazu die Vorgängerversion des PhidgetLabs (SWT-Projekt 2013) verwendet werden. Hier sind diese Animationen implementiert.
- Interface-Kit in Ausklappleiste (Flap)
Das Interface-Kit soll man unter Etoys nicht nur aus dem Objects-Katalog verwenden können, sondern auch direkt aus der Ausklappleiste/Flap/Lager rausziehen können.
- Sprache der Bedienelemente automatisch anpassen
Die Begriffe des Phidget-Lab sind alle englisch. Unter Etoys kann man seine Sprachpräferenzen angeben und alle Begriffe werden automatisch übersetzt. Dies sollte auch für die Begriffe des Phidget-Lab funktionieren. Die Übersetzung der Begriffe steht in .po Dateien. Neue PO-templates lassen sich über "GetTextExporter new exportTemplate". Das erstellt POTs. "Mit den gettext-Werkzeugen (<http://www.gnu.org/software/gettext/>) kann man die mit den existierenden POs mergen, dann die neuen Übersetzungen eintragen, und zum Schluss nach .MO compilieren. Es sollte eigentlich auch gehen, dass ihr eure Übersetzungen in ein weiteres .PO packt, das so heißt wie euer MC-Paket. Bzw. der GetTextExporter sollte das erzeugen, wenn ihr da #translated Sends drin habt. Nur die Etoyskachelübersetzungen landen in der globalen Etoys-

Tiles.po. Die sollten aber trotzdem gefunden werden, wenn ihr sie da rausnehmt und in eure eigene .PO schiebt. So ließe sich das leichter warten, als wenn ihr das mit dem Rest vermischt.”

- Werte der Sensoren (Kategorie phidgetport) beim Öffnen der Seitenleiste anzeigen, statt der Kategorie basic
Wenn man die Seitenleiste zu einem Sensor öffnet, dann öffnen sich standardmäßig die Kategorien “Einfach” und “Tests” geöffnet. Man könnte nun standardmäßig die Kategorie phidgetport öffnen.
- Bezeichnungen der Sensoren anpassen (z. B. “Bar” vs. “Pressure”) (zur Abwechslung mal etwas einfaches ;))
- Anregung der Schüler: Werte der Sensoren sofort zugreifbar, nicht erst über Seitenleiste
Wenn man auf die Werte eines Sensors für ein Skript verwenden möchte, muss man zunächst die Seitenleiste öffnen, die Kategorie wechseln und kann dann auf die Werte zugreifen. Das ist ziemlich umständlich, und es wäre schön, wenn man zum Beispiel direkt unter jedem Sensor seinen Wert anzeigen würde und diesen dann für Skripte verwenden kann.