PhidgetLab

Crossing the Border from Squeak to Real World Objects

Lysann Kessler, Stephanie Platz,
Thomas Klingbeil, Philipp Tessenow, Frank Schlegel
{firstname.lastname}@student.hpi.uni-potsdam.de
with Michael Haupt, Michael Perscheid, Robert Hirschfeld
{firstname.lastname}@hpi.uni-potsdam.de
Software Architecture Group, Hasso-Plattner-Institut
University of Potsdam, Germany

Keywords

Squeak, Phidgets, Etoys, sensors, actuators, sensing, controlling, USB.

Description

This project is about controlling Phidgets in Squeak. Phidgets are a set of plug-and-play sensors and actuators for USB sensing and controlling from a PC. Our goal is to access these Phidgets within the Squeak environment and integrate them in *Etoys*. In particular we want to give children the ability to playfully combine the Etoys environment with "real world" motors, sliders, joysticks and many other gadgets.

Therefore we developed a plugin for the Squeak VM and an API that manages the communication between the VM, the plugin and the Phidget API, and implemented an Etoys integration for the Phidgets.

Figure 1: Pupils controlling an LED using Squeak and the Phidget InterfaceKit

Motivation

At the Hasso-Plattner-Institut, we held several workshops for pupils. We taught them OOP and wanted to work with practical examples and projects. Using Phidgets the pupils created some successful and very creative projects. But during those short-term workshops we also experienced that teaching them a new programming language means that they cannot fully concentrate on the project itself. Therefore we wished for a way to make Phidgets accessible in Squeak and Etoys.

With that wish in our minds, we developed PhidgetLab as a coursework project in the "Software Engineering I" undergraduate course at HPI. We wrote an API that enables other developers to access Phidgets and use them in their applications. The foundation of the API is a plugin for the Squeak VM written in *Slang* and *C* that handles the communication between the VM and the Phidget API. We also extended Etoys so that it is possible to access and control Phidgets in an easy visual way. For example a child could draw a car in Etoys and then control it using a Phidget Mini-Joystick.

To provide some insight into the abilities of using Phidgets in Etoys we built a motor-powered marble maze. It can be controlled with any analog input like a joystick or an acceleromter. Details will be given below.

Realisation

The project mainly consists of two packages; one called "Phidget-Core", and another one called "Phidget-Etoys".

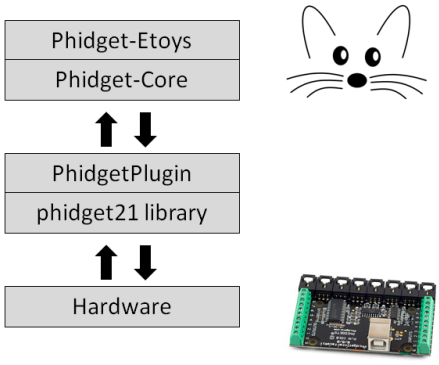

Figure 2: General Architecture of PhidgetLab

The Core package provides the Phidget API for Squeak. Its classes and methods can be used to communicate with Phidgets directly from the Squeak environment. For instance, the value of the first sensor attached to a connected Phidget InterfaceKit (`ifkit`) can be obtained by executing `(ifkit sensors at: 1) value`.

The Core API addresses people with a certain fluency in programming Squeak. They can make use of it when programming new applications using Phidgets. It organically adapts the official Phidget manufacturer's API to Squeak. We also expanded the API to provide some handy converted values in additon to the "raw" sensor values.

The "Phidget-Etoys" package integrates this functionality into Etoys. To that end, it depends on the "Phidget-Core" package. But in contrast to the Core API it also addresses people without any prior Phidgets or even Squeak programming knowledge.

We want every Phidget to have an equivalent in the Squeak Objects Menu and an Etoys visualization. One can simply view the Phidget's state there, set values, or make the device part of an Etoys script to let it interact with other Etoys entities.

To communicate with the Phidgets we make use of the Phidget manufacturer's programming API [3] written in C. A plugin for the Squeak VM, compiled as a shared library, interfaces Phidget-Core with that API. The general architecture of all components is shown in Figure 2.

Example: Marble Maze

To demonstrate the functionality of the plugin and the integration in Etoys we made an easy-to-implement but nevertheless illustrative example: a servo-driven marble maze. The goal of this game is to navigate a ball trough a maze with walls and holes. The ball can be controlled by tilting the maze using two axes. The unmodified game is shown in figure 3, next to our Phidget-controlled version.

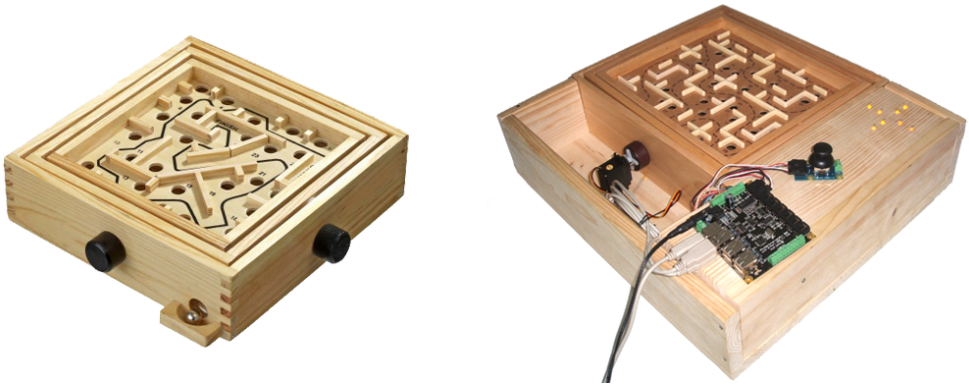

Figure 3: A simple marble maze game [1] without and with Phidgets

We use two servo motors to control the axes. To control the servos themselves we can use any analog sensor like a joystick, sliders or an accelerometer. A simple Etoys script assigns the values of the sensors to the servo-positions (see figure 4). That way the user can navigate the ball through the maze.

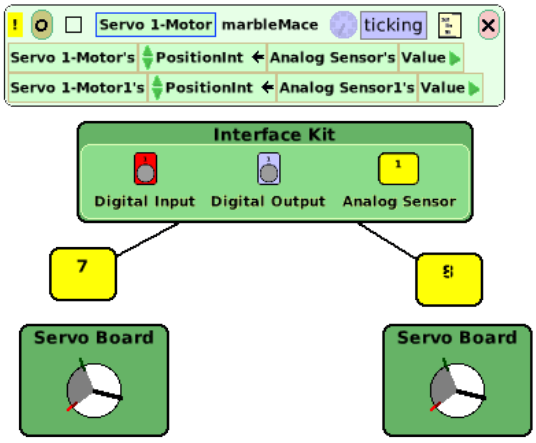

Figure 4: The script and the visualisation of the phidgets in Etoys

Licensing Information

PhidgetLab, i. e., our implementation in Squeak and the VM plugin, are licensed under the MIT Licence. The Phidgets library, which is not part of but relied-upon by our contribution, is licensed under the LGPL.

Affiliation Details

The surface mail address at which the developers can be reached is Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Str. 2-3, D-14482 Potsdam, Germany.

References

[1] http://images.mytoys.com/intershoproot/ecs/store/de/images/173/14/1731481-l.jpg (30.06.2009 15:18).

[2] http://www.hpi.uni-potsdam.de/swa/.

[3] http://www.phidgets.com/.