Fast, Cheap and In Control

Tuomas J. Lukka, Janne V. Kujala, Asko Soukka and Matti Katila
Hyperstructure Group
Agora Center, P.O. Box 35
FIN-40014 University of Jyväskylä
Finland
lukka@iki.fi, jvk@iki.fi, humppake@iki.fi, mudyc@iki.fi

Abstract

We show how custom controllers can be built cheaply, with minimal requirements for mechanical and electronic skills, by using a standard computer mouse and LEGO bricks. The utilization of commodity components allows anyone to reproduce a particular design without any tools and to easily prototype new designs.

Our main contribution is a simple, practical and reliable way of interfacing LEGO bricks to an optomechanical computer mouse. The internal mechanism of the mouse is utilized as far as possible, and a small rubber tyre is used to drive the mechanism from the outside, pushed towards the workings with a small pressure.

We present two controllers as examples of what is possible within this framework: a controller with two levers and a horizontal wheel, and a joystick-like 2D controller with enough friction to retain the position it is set to.

Constructing rugged prototypes while avoiding mechanical looseness with LEGO bricks requires some care but is possible. We list a number of design principles towards these goals, derived from practical experience.

Finally, we evaluate our framework for prototyping user interface devices.

1. Introduction

Producing customized controllers for new user interfaces is generally expensive and time-consuming and additionally requires electronics and machine shop skills not usually found in computer science or psychology departments. Because of this, most user interfaces are still operated with the mouse and keyboard, even if the interfaces could benefit from customized controllers.

Experiments [8] (and intuition) indicate that adjusting a physical slider is much easier than adjusting a virtual slider on the screen with a mouse. Therefore, it would be desirable to have controllers suited for the specific features of the user interface.

For example, navigating fisheye views [4] requires altering the magnification of the view and the amount of the fisheye lens' distortion. As these two quantities are independent from each other, they may be manipulated naturally by a controller with two distinct knobs or levers.

Despite this, virtual sliders or scrollbars are far more common with computers than physical sliders, due to economics. Adding a new virtual slider to a program costs next to nothing, while adding a new physical slider for each user amounts to a considerable sum of money.

Still, complicated devices such as computer mice and joysticks can be cheap due to commodity economics; the cost of user interface devices has little to do with manufacturing and much more to do with the number produced, as the development costs are amortized over the whole production. Thus, computer mice cost approximately $10-$30, joysticks some more, steering wheels even more, and specialized flexible input devices can easily cost tens of thousands.

In this article, we attempt to alter the economics of custom controllers by providing a simple design for building controllers out of commodity parts, without requiring electrical or mechanical engineering experience. The components that suit our purposes best are LEGO [5] Technic bricks and optomechanical computer mice. LEGO Technic bricks are popular toys because they can be used to build working miniature models of just about any real devices. However, the existing ways to get information from a LEGO construction into a computer are expensive, clumsy or inaccurate. Our solution to this lies in interfacing the LEGO bricks with an optomechanical mouse — a cheap but accurate commodity controller.

In the following Sections, we first discuss related work, then present our interface between LEGO bricks and computer mice, show two example controllers. Following that, we discuss practical experiences from our work so far and conclude.

2. Related work

Both LEGO bricks and computer mice appear in the scientific literature as components, but not together.

The LEGO Group has published a robotics construction series called Mindstorms, which contains bricks with touch and light sensors and motors, and a central microcontroller unit to control them. This series is gaining popularity in universities for teaching about robotics, and even among researchers. However, even the central unit alone is quite expensive and the existing sensors are not accurate enough for use in user interface devices.

In [1], devices built from LEGO bricks and electric toolkit parts are used to navigate 3D virtual world. In [3], physical LEGO-sized bricks are used as controllers for applications on a large horizontal display surface, and a sorting experiment shows how two hands can be used flexibly in parallel.

The mouse itself was originally invented as a custom controller for the Augmentation Research Center project [2]. It is ironic that it has since become a commodity so that custom controllers are cheapest to build by purchasing mice for parts.

In [9], one of the prototypes for using PDA as a peephole to a larger canvas uses mice and fishing lines for triangulating the position of the PDA (which the user can move around). The distance to a reference point was measured by having the fishing line go through a grid wheel in the mouse with small weight at the end maintaining tension.

3. Interfacing LEGO bricks to a computer through an optomechanical mouse

By far the most difficult part of our work with LEGO-based controllers has been the interfacing of the LEGO pieces to the optomechanical mice.

Figures 1 - 4 explain the internal mechanism of optomechanical mice,

Our first attempts involved replacing the slitted disks moving through the light gate with hand-made cardboard or plastic cutouts skewered on LEGO axles. This turned out not to be a good construction, requiring far too much handwork in the construction as well as not being very accurate.

The next approach was to try using small LEGO gears as the slitted disks. This made the construction far simpler, requiring no additional materials or tools, but the accuracy was severely affected: the gate missed a large fraction of the events.

The third and so far final solution is to use the mouse's own mechanism, simply driving it using a

Figure 1. The non-LEGO parts and tools required for a controller. A USB mouse would be preferred to the PS/2 mouse shown in the picture, since several can be connected to a computer simultaneously as separate input sources.

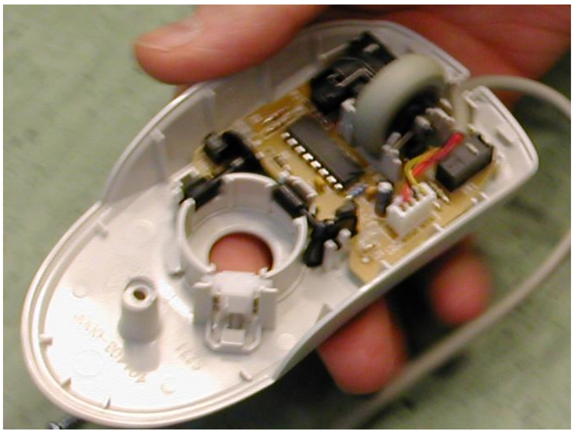

Figure 2. The mouse, opened with the screwdriver. The screwdriver will not be needed after this step.

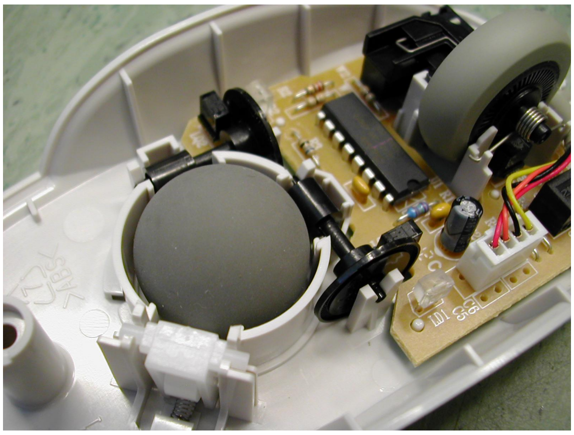

Figure 3. The mechanism of the mouse: the mouse ball, which rotates from contact with the underlying surface, rotates the two axles whose motion is then detected.

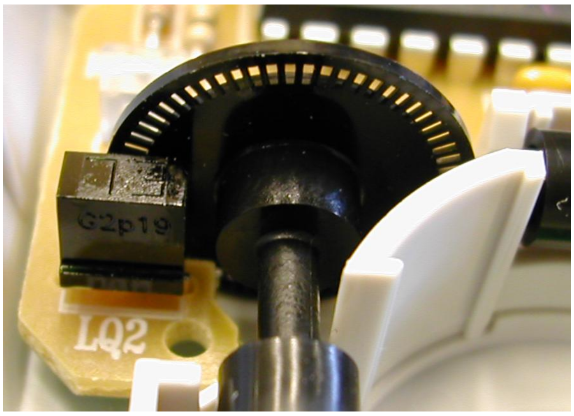

Figure 4. In optomechanical mice, the motion of the axles is detected by the IR light gate whose beam is cut by the slitted disk.

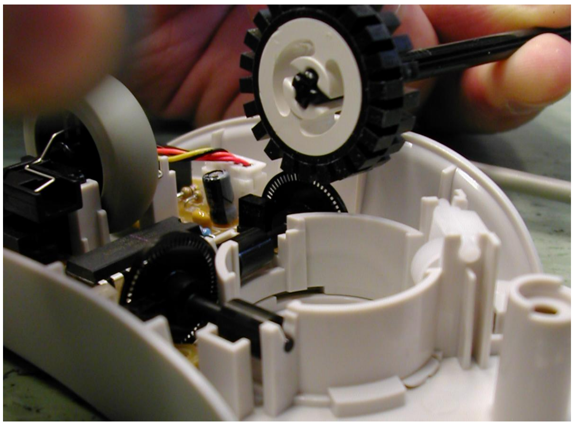

Figure 5. The principle behind the current version of our interface between the optomechanical mouse and LEGO pieces: driving the mouse axles by a rubber LEGO tyre. A smooth tyre (Bricklink part no 132-old) works even better, but they are currently in short supply.

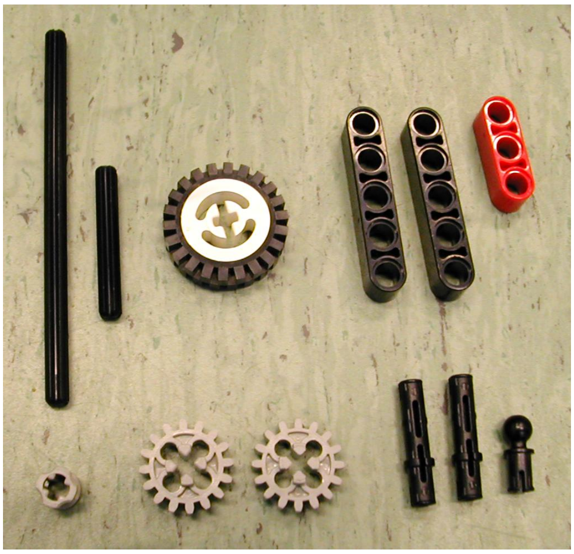

Figure 6. The LEGO parts for the driving mechanism

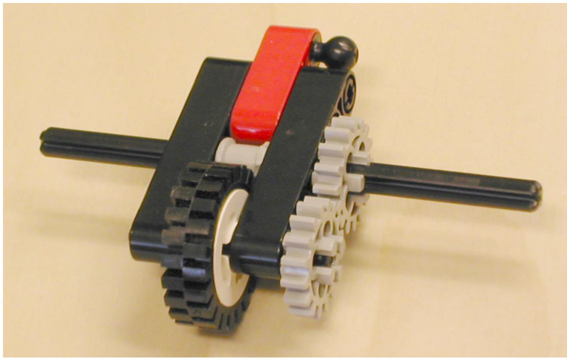

Figure 7. The driving mechanism assembled. The mechanism is attached to the LEGO framework by the long axle; the driving tyre is on one end of the 1x5 liftarms and a towball for attaching a rubber band at the other. The rubber band is used to pull the towball upwards, which in turn presses the tyre downwards. The force is easily adjustable by adjusting the other end of the rubber band.

LEGO rubber tyre (Fig. 5). In order to drive the mouse axle properly, a small amount of pressure needs to be applied to the wheel; this is provided by a rubber band through a small LEGO mechanism shown in Figs. 6 - 7.

With the pressure mechanism, it is simple to build a housing for the lower part of the mouse case and place the tyre holders there, as in Figs. 8 and [ref-figframedrivers] (the part of the housing where the rubber bands are attached is not shown in the these images).

On the software side, the Linux kernel is able to pass events from each USB mouse to a different virtual

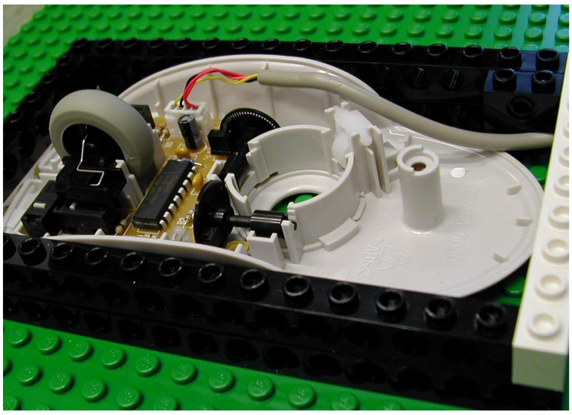

Figure 8. The beginnings of the housing for the mouse case.

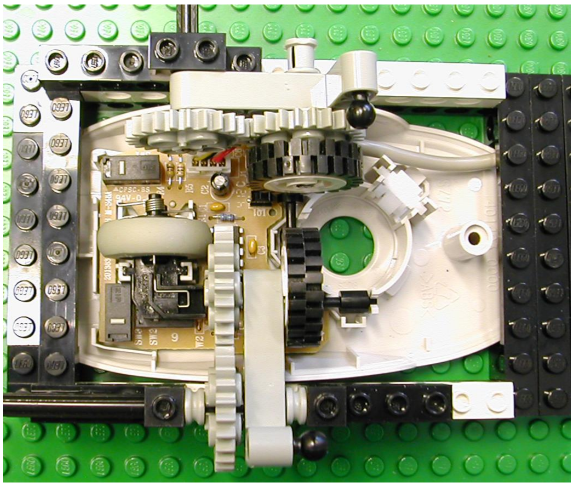

Figure 9. The driving mechanisms (using a slightly different version than the one depicted in the previous figures) attached. After the towballs are attached to rubber bands for pressing the tyres downwards slightly, motions from the LEGO axles are accurately translated to mouse X and Y motions.

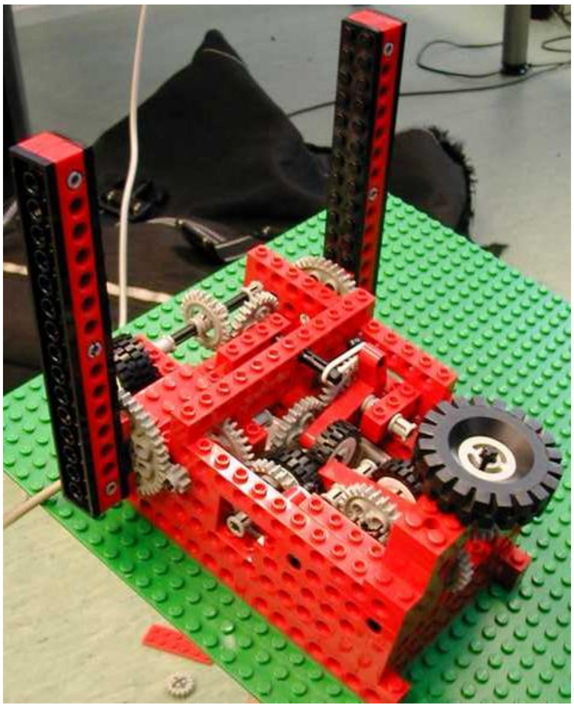

Figure 10. A fully functional controller prototype, with two absolute arm controls and a wheel control. All three controls have enough friction to retain their position.

device using an enhanced PS/2 protocol. This makes reading the numerical values from all mice connected to the computer trivial.

4. Examples

In this Section, we discuss two example controllers which use the above interface to the host computer. These controllers are not particularly novel but they give an idea of the magnitude of possibilities within our framework.

A. A custom controller for a specific user interface

As an example of a novel controller built using these methods, we present the controller in Fig. 10. This controller is intended for an interface where rotating the multiple left- and right- heading connections around the currently focused node is the main action. For this, the horizontal wheel corresponds geometrically closely to what the user sees on the screen. The handles can be used to control zoom factors for the focused node and the peripherally shown nodes.

We must stress that this controller is still very much work in progress; we are constantly experimenting with more advanced controllers for this application. The results will be reported in more detail later.

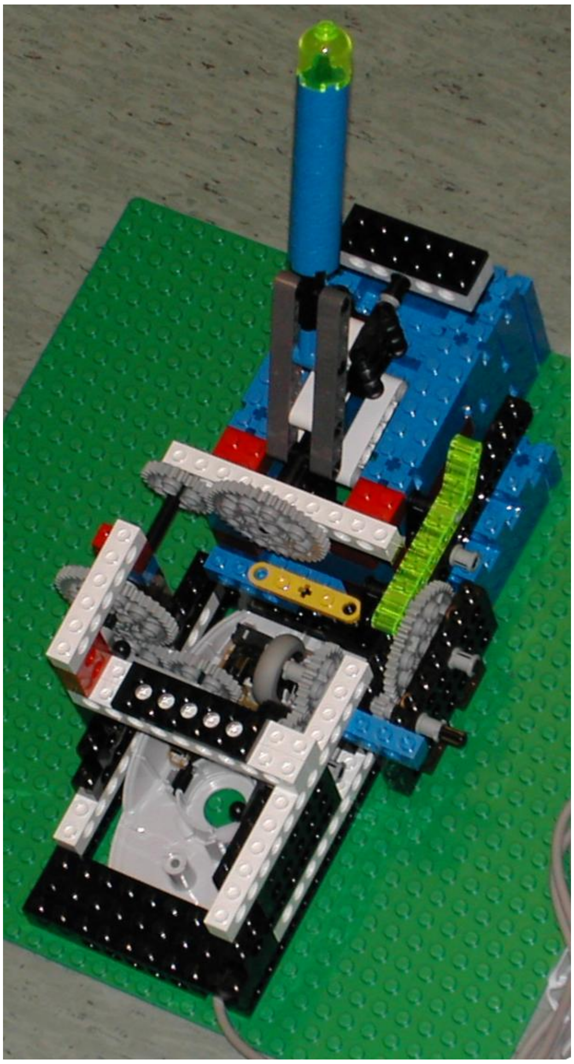

Figure 11. A 2D joystick-like controller built from LEGO bricks as a proof-of-principle. There is no force returning the joystick to its center position in this design, but if desired, that could easily be added using rubber bands or shock absorbers (although in that case simply purchasing a commodity game controller would be easier).

B. Combining two degrees of freedom

As a proof-of-principle example of connecting more than one degree of freedom to the same control, we present the 2D joystick-like controller in Fig. 11. The exact LEGO mechanism of the controller is beyond the scope of this paper..

5. Practical tips

In this Section, we summarize some of our practical experiences on building controllers.

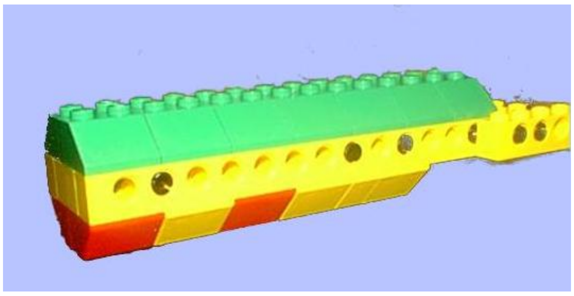

Figure 12. A rugged and relatively comfortable octagonal handle built from slopes and inverted slopes.

A. Mechanics

The two most difficult goals to achieve are avoiding mechanical looseness and obtaining a rugged structure. While interrelated, the second problem is more relevant to robotics and has therefore been dealt with in that context due to the popularity of the LEGO Mindstorms. See e.g. [6] [7]

Mechanical looseness is a different matter; it suggests the following design rules:

• No frictionless pins. For controllers, the frictionless pins are next to useless
except in special circumstances due to their looseness.
• A high gear ratio right after the gear attached to the control stick is vital to alleviate the looseness in gears after that.
• Axles used in gear trains must be supported at several points, far enough from each other.
• All moving joints should be designed symmetrically, using e.g. 3-long pins with friction. This is similar to taking the numerical derivative of a function $f(x)$ at x as $(f(x+\epsilon/2) - f(x-\epsilon/2))/\epsilon$ instead of $(f(x+\epsilon) - f(x))/\epsilon$
• Even if it is a horrifying thought to LEGO purists, bricks can be glued together to obtain an even sturdier structure. However, this is seldom necessary.

B. Handles

For handles, basic LEGO brick designs are rather angular and uncomfortable. In our work, we have found that there are several ways to build comfortable and rugged handles by using some more specialized bricks such as slopes and inverse slopes or rounded bricks. As an example, an octagonal handle is shown in Fig. 12.

With some more effort, ergonomic handles of any imaginable shape can also be created by covering LEGO bricks with air-hardening modelling paste, as in Fig. 13.

Figure 13. An ergonomic handle implemented using air-hardening modelling paste. The handle was molded directly to the left hand of one of our research group's members.

6. Conclusion

The most important contribution of this article is the simple interface between LEGO bricks and an optomechanical computer mouse, which alters the economics of creating custom controllers.

There are several positive attributes to our approach: prototyping new controllers is cheap, in the region of $30-$100 per controller. Controllers are easy to build and the parts are reusable. When using USB mice, it is possible to attach as many controllers as desired to a single computer and the controllers are easy to use programmatically e.g. in Linux. Additionally, this approach is able to tap a hidden resource: a considerable fraction of scientists and students actually *do* have previous experience with LEGO bricks. And of course, building controllers using LEGO bricks is both fun and motivating.

Of course, there are also problems with this framework: construction of controllers does require care (see Practical tips above). Even with careful construction, there will be some (not much) mechanical looseness in the controller; however, humans *are* used to compensating some mechanical looseness. Prototypes built from LEGO bricks are also often relatively large compared to what the size of the desired controller would ideally be. However, we have found that redesigning the controller a few times, with some internal peer review helps to reduce the controller size greatly. For knobs or levers with absolute position, our approach does unfortunately require calibration; there is no way for the computer to know the initial position of the lever. On the whimsical side, there is also the problem of having to explain the budget for purchasing LEGO sets with strange names.

The down-sides are still quite acceptable for prototypes, and of course using LEGO bricks for prototyping won't rule out the possibility to create a further prototype from different materials.

References

[1] M. Ayers. Interface toolkit. In *UIST Proceedings of the Symposium on User Interface and Software Technology.* ACM, 1996.

[2] D. C. Engelbart and W. K. English. A research center for augmenting human intellect. In *AFIPS Conference Proceedings of the 1968 Fall Joint Computer Conference*, volume 33, pages 395–410, 1968.

[3] G. W. Fitzmaurice, H. Ishii, and W. Buxton. Bricks: Laying the foundations for graspable user interfaces. In *CHI*, pages 442–449, 1995.

[4] G. W. Furnas. Generalized fisheye views. In *Proceedings of CHI '86*, pages 16–23. ACM Press, 1986.

[5] LEGO. Lego group. `http://www.lego.com/`.

[6] B. U. Marina, P. Iris, J. Catherine, V. Alison, and S. Jonathan. Teachers as designers: Integrating robotics in early childhood education. *Information Technology in Childhood Education Annual*, 14:123–145, 2002.

[7] M. Orazio, L. H. Hautop, and C. Maurizio. Robotics as an educational tool. *Journal of Interactive Learning Research*, 10:25–47, 1999.

[8] R. C. Stuart. *Evaluating alternative controllers using the midi protocol for human-computer interaction.* PhD thesis, University of Washington, US [Rosenberg], 1995.

[9] K.-P. Yee. Peephole displays: Pen interaction on spatially aware handheld computers. In *Proceedings of CHI '03.* ACM Press, 2003.