TouchControl

An interactive multi-touch 3D design tool

David Lemberski¹, Marco Hemmerling²
¹,²Hochschule Ostwestfalen-Lippe, University of Applied Sciences; Detmolder Schule für Architektur und Innenarchitektur, Germany
¹,²http://www.hs-owl.de/fb1
¹david.lemberski@hs-owl.de, ²marco.hemmerling@hs-owl.de

Abstract. Today’s architectural design is changing rapidly through the scope of new digital design tools (software like parametrical modelers) on the one and the use of digitally controlled fabrication techniques (hardware like 3D CNC-milling) on the other side. However the full intuitive and experimental potential of the software has to be accessed through the sometimes limiting bottleneck of the traditional mouse-screen user interface although alternative interaction methods like multi-touch became available. This paper focuses on extending the usability and spontaneity in a common architectural design process using an iPad as an external hardware controller.

Keywords. intuitive design tool; iPad; OSC; multi-touch; human-computer interaction.

INTRODUCTION

This paper shows the setup of an iPad with a customizable software multi-touch control-surface and the resulting possibilities to influence the design process in a parametric modeling environment. Parametric modelers – in particular graphical algorithm editors like Generative Components [1] and Grasshopper [2] – offer the ability to make fast design changes but are limited in interaction due to the restrictions of the traditional mouse-screen user-interface. With a mouse as classical input-device the user can only change one value at a time so it can take long to explore the whole spectrum of results and interdependency.

Multi-touch capable hardware controller – like fader-boxes or rotary controller – help to overcome the mouse-control paradigm and extend the range of human-computer interaction in multiple degrees of freedom (DOF). The mouse offers only two DOF (movement along the x- and y-axis) in comparison to a fader-box where every fader represents one DOF. E.g. using a hardware-controller which offers eight faders enables the user to access eight DOF at once. Some like the Behringer BCF2000 [5] offer even visual and haptical feedback where servo-motors move the faders and LED-rings around the knobs change their state to match a value send from the target device e.g. after a preset switch. These devices can
be easily customized on software level, e.g. a fader can be assigned to control different parameters, but they have a fixed layout and are not designed to be customized on hardware level like swapping the positions of a fader and a knob. The labeling is fixed and neutral (e.g. ‘fader1’, ‘fader2’) which makes it difficult to remember the right fader when using varying assignments. One possibility to overcome the labeling limitation is using sticky notes.

However, a device that integrates a flexible, customizable layout/labeling and additionally touchscreen xy-pads would fit the needs of an interactive design process much better.

**CUSTOMIZABLE VIRTUAL MULTI-TOUCH INTERFACES**

In 2004 a pressure sensitive and customizable music control surface named JazzMutant Lemur [6] had been released. The price for unlimited touchpoints functionality was still quite high - around $ 2,500 in 2006. Unfortunately in 2010 the production stopped. With the launch of Apple's iPhone in summer 2007 an affordable multi-touch device with visual feedback came to the market. In contrast to other multi-touch devices - like LCD-screens - it allows to track up to 12 touches simultaneously. The appearance of the iPad in April 2010 widened the usability of this application due to its larger screen-size. Other manufacturers released multi-touch enabled tablet computers with screen-sizes comparable to the iPad in the meantime. Since the iPad is longer on the market a larger software basis is available for it.

Since the release in 2010 over 25 million iPads has been sold [7], the average user’s age is between 35 and 44 years [8]. The multi-touch technology enables a direct, playful and intuitive access to the computer and is maybe a huge step after the appearance of the mouse in the ‘80s. Currently most of the applications used on the iPad are related to reading books, papers, watching pictures and movies or playing interactive games. A much more interesting, unconventional, experimental and creative use of the multi-touch functionality happens in the field of music creation and editing. Since music and architecture share a lot of similarities (Lemberski and Hemmerling, 2010; submission number 102 by Anke Tiggegamm: Camera Musica - Compositions in music and space), apps designed for music control are being used in this study to enable the same level of spontaneity and interactivity.

Apps like TouchOSC [10] and Control [11] feature a customizable and thus intuitive control-surface with (virtual) interface elements (widgets) like push/toggle buttons, rotary knobs, faders, xy-pads and labels. Through the touchscreen they give an immediate visual feedback to the user about value changes on the control device (e.g. movement of a virtual slider on the iPad) which are concurrently sent over a wireless connection to other (target) devices where they can be used to control parameters. Another aspect is the bidirectional functionality available in TouchOSC and Control allowing dataflow between the devices. Parameter changes made on the target device (like preset switching through the software running there) can be send back to the iPad to visually represent the

![Figure 2](TouchOSC on iPad [9])
actual state of parameters. Individually customizable interface layouts with labels and multiple pages can be created to deal with different tasks.

**CASE STUDY – PARAMETRIC PAVILION**

A parametrical pavilion model has been chosen to research and exemplify the concept of an intuitive design tool. For the practical setup of the case study an iPad as a controller on the one side and a standard PC running Grasshopper (GH) as host and target device on the other side has been defined. Data is being send via a standard wireless connection (WiFi) between the devices using UDP (User Datagram Protocol) and OSC (Open Source Control) messages. The apps on the iPad - either TouchOSC or Control - use their own OSC sender/receiver, inside GH a UDP sender/receiver written by Luis Fraguada [12] (part of the gHowl tools) links the parametric part to the control device. A clear customized control layout has been created which contains only the needed parameters (Fig. 3).

Both devices are identified by their unique IP-address in the wireless network (e.g. 169.254.69.32), the respective UDP receiver/sender by a unique port-number (e.g. 8.000). Every widget like a fader has also its unique ID so it can be distinguished from the others and its value is directed to the corresponding parameter input inside GH. Faders and knobs send single continuous values between 0 and 1, xy-pads a pair of continuous values between 0 and 1 for off/on.
The whole parametric model of the pavilion is built in GH which combines core functions like associativity and parametrization with a clear user friendly GUI (Graphical User Interface) and the concept of a graphical algorithm editor (Fig. 4). Components have input and output pins, are placed on the canvas and interconnected by wires like a growing horizontal tree-structure (Fig. 5). There is a library of components to choose from, while every component executes a different task referred to an algorithm like multiplication of values or creation of geometry. Custom components and thus functions can be either scripted in common languages like Python [13], C#.Net and VB.Net [14] or created by wiring existing modules. The specific order of connections and the type of used components define the resulting model next to the values of input parameters. Input values are fed via numerical
fields or through virtual sliders inside GH, in the study setup however they can be routed to come from the UDP-receiver component (Fig. 6).

It takes effort to build up the GH definition especially that every new project needs its own individually tailored logic. It is however clever to reuse groups or customized components from other similar projects or from a user library. Once the definition is ready design variations can be easily made by changing the corresponding parameters and so forcing a recomputation of the whole model.

CONCLUSION
The concept of extending the intuitive part in a parametric design process by adding a customizable multi-touch remote control subjectively enables a higher interactivity level during the exploration phase in comparison to the standard mouse-screen interface. However a double-blind comparative survey is needed to objectify the added value.

Today a mouse is preferably required during the building phase of the GH definition, which may change in the future when multi-touch hardware with bigger screens will be available and multi-touch GUIs will be more favored. Both phases - the building and exploration phase - are interdependent and both underlay the same dynamic process of trial and error. Changes in the definition require changes in the parameter values and vice versa until the design evolves into the desired direction and approaches the objectives. The responsiveness of the setup depends on the complexity of the definition where sophisticated constructions need up to several seconds to update in GH due to computation impact.

The biggest shortcoming of the actual setup is the layout customization of the control-surface. Currently a custom layout for the mentioned apps cannot be created or edited on the iPad within the app itself. A free graphic layout editor for TouchOSC which runs on the most popular platforms (OSX, Windows, Linux) is available on the developers page. The layout must be (re)transferred to the iPad after creation or after minor changes. Such a graphic editor does not even exist for Control, one has to create the layout in a plain text editor. There is so far no possibility to automatically update the layout on the iPad after parameters have been added, deleted or swapped in GH so every change their needs a manual modification of the layout definition file in the layout editor and an upload to the iPad.

OUTLOOK
A further study will take into account how to solve the above described shortcoming and look at the bidirectional communication possibilities between the control and target software. This could allow a more complex visual feedback information on the control device.

The necessity for an external layout editor could be bypassed by incorporating editing possibilities into the app. This however would still require a manual reedit of the layout after definition changes in GH. A more practical solution would be to develop a GH component which automatically changes the control layout and sends it to the iPad when changes affecting it had been made inside of the GH definition.

An interesting application area would be extending the current single person system to a team or collaborative working environment. Clients, architects and civil engineers could sit all together around a multi-touch table and collaboratively change a design in a discussion through the help of a controlled parametric modeler.

REFERENCES
Hauschild, M and Karzel, R 2010, Digitale Prozesse, Institut für internationale Architektur-Dokumentation GmbH & Co. KG, München