TILES: an inventor toolkit for interactive objects

Simone Mora, Monica Divitini and Francesco Gianni
Department of Computer and Information Science, NTNU
Sem Sælandsvei 7-9
7491-Trondheim (Norway)
{simone.mora, monica.divitini, francesco.gianni}@idi.ntnu.no

ABSTRACT
We present the groundwork for Tiles: an inventor toolbox to support the development of interactive objects by non-experts. Tiles is composed by (i) a set of physical input/output primitives to describe interaction styles with technology-augmented objects, (ii) extensible hardware modules easily embeddable in everyday things that implement the primitives, (iii) APIs to code application logics using popular programming languages. We are currently exploring the opportunities of using Tiles to develop applications for learning, games and advanced visual interfaces.

CCS Concepts
• Human-centered computing→Ubiquitous and mobile computing systems and tools *• Human-centered computing→Gestural input • Human-centered computing→User interface toolkits.*

Keywords
Tangible User Interfaces; Interactive Objects; Internet of Things.

1. INTRODUCTION
Making interactive objects, everyday things augmented with technology to provide some interactive behaviour on top of a traditional physical appearance, is a challenging process that requires a wide range of tools and skills from the fields of design and engineering. While the design phase is challenged by the lack of guidelines to understand how affordances and design of a tangible interface are related [1], the technical complexity of working with electronics and embedded programming languages obstruct the ability to explore design choices and to iterate by implementing functioning prototypes [2].

The goal of our research is to lower the threshold of technical skills required to design and build prototypes of interactive objects, allowing non experts to create simple applications. At the same time, we aim at higher the ceiling providing experts with extended tools and hacking opportunities to build complex interfaces.

In this poster we describe Tiles, a toolbox to support the iterative process of building prototypes of interactive objects providing design and prototyping tools. Our target users are designers, researchers and makers with no competence in electronics and basic skills in programming. We are currently completing the first design iteration for Tiles which points at exploring application domains by providing the toolbox prototype to our users.

In the following we present the toolkit architecture starting from design principles, further describing possible application domains.

AVI '16, June 07-10, 2016, Bari, Italy
ACM 978-1-4503-4131-8/16/06.
http://dx.doi.org/10.1145/2909132.2926079

2. DESIGN PRINCIPLES
The first prototype of Tiles has been developed based on the authors' experience with prototyping augmented artefacts for training and play [3, 4]. Our toolkit design is motivated by the following guidelines.

Feedbacks loops based on simple interactions. Support the development of interactive objects interfaces based on simple loops of physical inputs (e.g. gestural) and unobtrusive feedbacks.

Hardware tools easily embeddable into everyday objects. Enable the design of affordances for interactive objects without being limited by form factor of the technology used for prototyping.

Centralised, language-agnostic and ecologies-oriented software tools. Enable writing software to control coordinated ecologies of interactive objects scattered in space from a centralised, language-independent environment.

Support for hacking and modifiability and different levels of expertise. Enable experts to extend the platform and build complex interfaces without compromising ease of use for novices.

Compared to other prototyping platforms, Tiles aims at supporting the process of making interactive objects holistically, providing an integrated set of design, software and hardware tools to guide the user from idea generation to prototype implementation.

3. TILES ARCHITECTURE
Tiles is composed by:

Interaction Primitives: a set of abstracted common primitives to describe interaction styles with augmented objects

Squares: hardware modules with sensors and actuators to implement capture and generation of primitives

Cloud API: a software framework to allow manipulation and use of primitives within specific application logics

3.1 Interaction primitives
We developed a set of input/output interaction primitives representing some of the most common manipulations performed on everyday objects that can be sensed with technology. A subset of the primitives developed is reported in Table 1. These primitives may seem to carry little semantic value if considered alone, yet they assume meaning when associated to a physical-digital context. For example, primitives become meaningful when they relate to the affordances of a specific object, when they are mapped to the behavior of a digital system, when they are performed in sequence with other primitives, or in a context of ecology of objects.

Interaction primitives support the transition between design and implementation phases by providing a unified language that can be easily understood by non-experts. Primitives are also provided in the form of a *Card Game*[1] to be used with end-users in design workshops. We believe they can establish the foundation for an interaction language for interactive objects.

Table 1: Sample interaction primitives

Input/ Output	Primitive	Degrees of freedom	Sample mapping
IN	Touch	Single, double, touch-and-hold	Send a command, log a quantity
IN	Shake	Horizontally, vertically	Throw a random option, discard a command
IN	Rotate	Clockwise, counterclockwise	Modify a quantity, select an option
OUT	LED feedback	Color shift, blink, fade	Continuous status notification
OUT	Haptic feedback	Vibration pattern	Discrete status notifications

3.2 Square modules

Tiles Square are electronic modules, small enough to be easily attached or embedded in everyday objects (Figure 1). They are simple autonomous computational units, pre-programmed to capture input primitives or produce outputs.

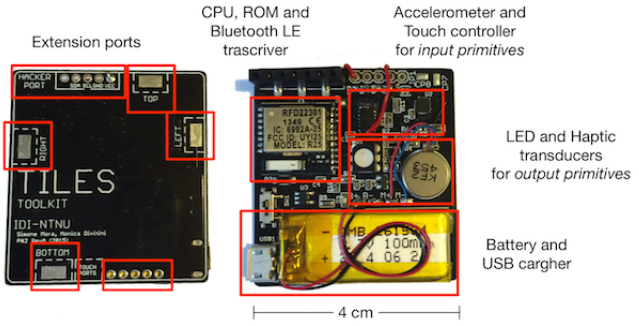

Figure 1: Tiles Squares modules

Squares don't implement any application logic, they just act as interface between the physical world and digital operations. Software applications that make use of Square modules run remotely, on top of a cloud API. Because of the small size and low cost, Squares can support the development of interfaces that make use of an ecology of distributed primitives/objects.

3.3 Cloud API

Tiles Cloud API provides a centralized, language-agnostic software interface to interact with objects embedding square modules. It allows developers that are not specialized coding for embedded devices to create applications using commonly known languages. This is a paradigm shift compared to traditional approaches of embedded programming: instead of implementing and deploying code for each specific square module, Cloud API enables developers to keep an "eye-bird view" on the design of an interface. Functionalities provided by multiple Square modules attached to several objects, and thus the behavior of the augmented objects, can be programmed from routines running in a centralized cloud environment; without requiring physical access to the hardware modules.

Tiles Cloud is provided as a server routine, an app for administration of Square modules and a set of libraries to be used to develop applications in Java, JavaScript, Python or C.

[1] http://tilestoolkit.io/cards

4. APPLICATION DOMAINS

We are exploring application domains to specialize our toolkit in the following directions:

Learning programming for the IoT. Tiles can be used to make non experts familiar with IoT concepts. Users can easily write simple applications to handle physical inputs captured by one square module (e.g. when an object is shaken) to produce a digital output on another square (e.g. change of LED color). In this way users are exposed to both concepts proper of the physical world like acceleration changes; and concepts from the digital world as conditional statements and event-driven programming. Tiles can be easily integrated with visual programming languages to further extend the learning to novices and kids.

Development of hybrid games. Tiles can be used to develop games that blend elements of physical and digital play such as pervasive and exer–games. For example, interaction primitives captured by square modules embedded into sport clothes or equipment can be used to feed a computer engine to trigger game dynamics.

Exploration of advanced visual interfaces. Square modules can be used as physical pixels; they can be scattered in space or linked side by side for increasing information granularity. They can be used as ambient interfaces for non-intrusive information awareness or to promote behavioral change. For example, a square module next to a water faucet can display over-average water consumption data in a household.

5. CONCLUSIONS

We presented the first prototype of a toolkit to support non-expert in making interactive objects. Tiles supports interactive objects development as a whole, with primitives useful for design as well as hardware and software tools for implementation.

We are currently in the process of validating our technology by letting students and designers build Tiles applications. We are also extending the set of primitives to better support the development of ecosystems of interactive objects.

6. ACKNOWLEDGMENTS

We thank students who took part in the implementation of the prototype and Nice Industridesign AS for providing feedbacks.

7. REFERENCES

[1] Kimura, H. et al. 2007. CookieFlavors: Rapid Composition Framework for Tangible Media. *Proceedings of Next Generation Mobile Applications, Services and Technologies*. (2007), 100–109.

[2] Lee, J.C. et al. 2004. The Calder Toolkit: Wired and Wireless Components for Rapidly Prototyping Interactive Devices. *Proceedings of DIS*. (2004), 167–175.

[3] Mora, S. and Divitini, M. 2014. WATCHiT: A Modular and Wearable Tool for Data Collection in Crisis Management and Training. *Proceedings of AMI*. 8850, Chapter 22 (2014), 274–289.

[4] Mora, S. et al. 2015. The Interactive-Token Approach to Board Games. *Proceedings of AMI*. 9425, Chapter 10 (2015), 138–154.