



1999-2011

# challenges

2011

VII Conferência Internacional de Tecnologias de Informação e Comunicação na Educação

## [ PERSPECTIVAS DE INOVAÇÃO ]



12/13.MAIO.2011

UNIVERSIDADE DO MINHO BRAGA | PORTUGAL

+info>> [www.nonio.uminho.pt/challenges2011/](http://www.nonio.uminho.pt/challenges2011/)

### DATAS IMPORTANTES

- » até 28.02
- » 29.03 a 01.04
- » 01.04 a 08.04
- » até 27.04
- » após 28.04
- » até 08.04

### CONTACTOS

UNIVERSIDADE DO MINHO  
CENTRO DE COMPETÊNCIA DA  
UNIVERSIDADE DO MINHO  
INSTITUTO DE EDUCAÇÃO  
CAMPUS DE GUALTAR  
4710-037 BRAGA  
TEL: +351 253 601 870  
FAX: +351 253 601 871  
E-MAIL: [ic@uminho.pt](mailto:ic@uminho.pt)

### ORGANIZAÇÃO

**CCUM**  
CENTRO DE COMPETÊNCIA DA  
UNIVERSIDADE DO MINHO

**IETI**  
INOVAÇÃO EDUCACIONAL COM AS  
TECNOLOGIAS DE INFORMAÇÃO

### APOIOS



Título

VII Conferência Internacional de TIC na Educação

Challenges 2011

Perspectivas de Inovação

Organizadores

Paulo Maria Bastos da Silva Dias

António José Osório

Capa

Candeias Artes Gráficas

Paginação e Orientação Gráfica

Centro de Competência da Universidade do Minho

ISBN 978-972-98456-9-7

Edição do Centro de Competência da Universidade do Minho

© Centro de Competência da Universidade do Minho

Braga, Portugal

Maior, 2011

500 Exemplares

Apoio à publicação

**FCT** Fundação para a Ciência e a Tecnologia

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

## **BITS TANGÍVEIS: A CONSTRUÇÃO DE NARRATIVAS TANGÍVEIS**

Pedro Rito, ESE/Instituto Politécnico de Viseu, Portugal, rito@esev.ipv.pt

Ana Loureiro, ESE/Instituto Politécnico de Santarém, CIDTFF/Universidade de Aveiro, Portugal  
accloureiro@gmail.com

**Resumo:** A utilização de objectos tangíveis possibilita que se criem interacções, ou dinâmicas, que são alternativas ao rato e ao teclado, no processo de comunicação com o computador. A construção destes objectos incorporando componentes electrónicos permite trazer essa dinâmica para um outro nível. Este encontro com a tecnologia permite que as crianças tenham um papel activo, ao mesmo tempo que surge um efeito do controlo sobre os objectos, o que se torna importante para elas. Como reforço desse efeito de controlo, a introdução de componentes electrónicos e digitais programáveis permite também que a criança desenvolva, fortaleça e sinta o impacto do seu papel enquanto designer competente e criador de tecnologia. A tecnologia existente permite que a construção desses objectos e a comunicação com os computadores seja feita a baixo custo, através de micro-controladores, recorrendo por um lado a software *open source* e por outro à utilização de *open hardware*.

Iremos neste artigo apontar algumas das possíveis utilizações de objectos tangíveis para a construção de narrativas tangíveis, interactivas e dinâmicas.

**Palavras-chave:** computação física, interfaces tangíveis, narrativas tangíveis, *open software*, *open hardware*

**Abstract:** The use of tangible objects makes it possible to create interactions, or dynamics, which are alternatives to the mouse and keyboard in the process of communicating with the computer. The construction of these objects incorporating electronic components lets us bring that momentum to another level. This meeting with the technology allows children to take an active role, while there is a purpose of control over the objects, which becomes important to them. With the reinforcement of that control, the introduction of programmable digital electronic components also allows the child to develop, strengthen and feel the impact of their role as competent designer and creator of technology.

Current technology allows the construction of these objects and the communication with computers at a low cost through micro-controllers, using, on one hand, the open source software and on the other the open hardware.

In the article we intend to describe some of the possible uses of tangible objects to create tangible, interactive and dynamic storytelling.

**Keywords:** Physical computation, tangible interfaces, tangible storytelling, open software, open hardware

### **1. Introdução**

As crianças expressam as histórias que ouvem de diferentes formas. Com a ajuda de um qualquer objecto ou conjunto de objectos elas conseguem criar e contar as suas histórias, desde que sintam que têm controlo sobre esses mesmos objectos (Montemayor, Druin, Chipman, Farber, & Guha, 2004). Com a utilização de objectos físicos que lhes são familiares elas reforçam o conhecimento, através de actividades exploratórias e de entendimento (Scarlatos, Mbogho, & Jaworska, 2006).

A tecnologia que se encontra disponível permite que operações, consideradas de exigência técnica média/alta, como manipular texto, vídeo ou som, possam ser feitas por crianças. A

tecnologia existente permite ainda que as interfaces de comunicação com computadores possam ir além das interfaces tradicionais como o rato e o teclado, ao mesmo tempo que os computadores podem ser substituídos por outros dispositivos com características similares (Montemayor et al., 2002, p. 299).

Um dos exemplos dessa tecnologia são os micro-controladores baseados no *Arduino*<sup>1</sup>, que são ferramentas *open source*, programáveis e que permitem criar interactividade com os computadores. A utilização destes tipos de micro-controladores permite potenciar ambientes e processos de criatividade, pois ao invés do utilizador perder tempo com os detalhes relacionados com o seu funcionamento técnico, esse mesmo tempo pode ser utilizado na experimentação e descoberta de outros e diferentes ambientes e cenários onde ele possa ser utilizado. A sua programação é feita através de um software, similar ao *IDE Arduino (IDE - Integrated Development Environment)* onde, apesar de existir a necessidade de escrever linhas de código para ensinar o micro-controlador, este permite a comunicação com outros programas, como é o caso do *Scratch*<sup>2</sup>. Sendo um hardware de fácil aprendizagem, surge a oportunidade de se construírem interfaces tangíveis, possibilitando que exista uma comunicação entre o software e o hardware através do código construído num ambiente gráfico, intuitivo, de fácil e rápida aprendizagem, com que uma criança pode facilmente trabalhar (Peta & Purchase, 2002, p. 774).

## 2. O open hardware

O *open hardware* é definido como todo o hardware cujo desenho é disponibilizado publicamente para que qualquer pessoa o possa estudar, modificar, distribuir, construir ou comercializar como outro desenho ou como hardware baseado no desenho original (“OSHW - Definition of Free Cultural Works,” sem data).

Um objecto definido como *open hardware* permite que as pessoas tenham a liberdade para controlar a tecnologia que desenvolvem, ao mesmo tempo que o conhecimento acerca da mesma é partilhado de forma aberta. Esta definição tem como fonte a definição de *open software* que também fomenta a partilha livre do conhecimento acerca do desenvolvimento de software e do seu uso (“The Open Source Definition (Annotated) | Open Source Initiative,” sem data).

O hardware *Arduino* é um caso de *open hardware* que foi desenvolvido por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. Apresentado ao público em 2005, este micro-controlador permite que se estabeleçam comunicações entre objectos tangíveis e objectos digitais. Esta comunicação é estabelecida entre dois ambientes e é designada por computação física (*physical computing*) (Noble, 2009, p. 245). Para Noble os ambientes que desencadeiam este tipo de comunicação estão relacionados com dois mundos, o mundo digital

---

<sup>1</sup> <http://www.arduino.cc/>

<sup>2</sup> <http://scratch.mit.edu/>

e o mundo físico (real). A computação física traduz-se assim no envio de informação do mundo digital para o mundo real e do mundo real para o mundo digital. A transmissão de informação pode ser observada a um nível mais baixo, traduzindo-se numa conversão da energia física para energia eléctrica, sendo esta percebida pelos computadores. Esta conversão de energias é designada por transdução (Igoe & O'Sullivan, 2004, p. 3).

## 2.1 O Arduino

O *Arduino* é um micro-controlador que permite que, por um lado, sejam criados e desenvolvidos outros tipos de interfaces para comunicar com o computador (para além do tradicional rato e teclado) e, por outro lado, que se criem objectos, físicos ou não, que reajam à informação que é emitida pelo computador. O *Arduino*, versão *Duemilanove*<sup>3</sup>, tem como principais características ter catorze portas de entradas/saídas digitais e 6 portas de entrada analógicas, a comunicação ser feita através do protocolo serial e apenas necessitar de uma fonte de alimentação, que pode ser uma entrada *USB (Universal Serial Bus)* ou uma pilha de nove volts. Das catorze portas digitais, seis são portas moduladas (*PWM, Pulse With Modulation*), isto é, portas que permitem que o sinal varie dentro de uma determinada escala, sendo que as restantes portas apenas recebem como valores o zero e o um. Foram desenvolvidas pela mesma equipa várias versões do *Arduino* que têm como principais diferenças e características o tamanho físico (espaço que ocupam), o número de portas digitais e analógicas que estão disponíveis e a forma de comunicar.

Um desses exemplos é o *Arduino Lilypad* que tem por base o *Arduino* e foi desenvolvido por Leach Buechley e pela empresa canadiana *SparkFun Electronics*<sup>4</sup>. Esta versão foi apresentada à comunidade em 2007 e permite que um micro-controlador possa ser integrado e interagir com e-texteis ou que funcione como objecto de decoração (*wearables*). O *Lilypad* tem cerca de dezoito centímetros, pode ser 'cosido' a uma peça de e-textil e ligado a outros sensores ou peças similares (Buechley & Eisenberg, 2008, p. 13). Para os criadores deste micro-controlador, o *Lilypad* tem por objectivo a criação, por qualquer pessoa, das suas próprias peças de e-textil, expandindo assim a sua criatividade.

## 2.2 Funcionamento do Arduino

O hardware *Arduino* tem associado um software *open source*, com o mesmo nome de *IDE Arduino*<sup>5</sup>, software que é compatível com todas as versões do hardware, e que foi desenvolvido a partir do *IDE* do *Processing*. O *IDE Arduino* tem por base a linguagem de programação *Java* e o *Wiring*. Neste *IDE* são escritas linhas de código que estão encapsuladas, para que seja perceptível a qualquer pessoa e se consiga trabalhar, mesmo não tendo conhecimentos

<sup>3</sup> <http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>

<sup>4</sup> <http://www.sparkfun.com/>

<sup>5</sup> <http://www.arduino.cc/en/Main/Software>

prévios de uma linguagem de programação. O *Processing* é uma linguagem de programação, desenvolvida por Ben Fry e Casey Reas, no seguimento do trabalho feito por John Maeda com o trabalho *Design by Numbers*. Trata-se de uma linguagem de programação de fácil compreensão que permite que artistas visuais desenvolvam o seu trabalho sem as limitações das ferramentas como as *tool boxes* ou comandos previamente programados e que para os autores do *Processing* podem limitar a criatividade.

Para além de receber informações de um computador, o micro-controlador *Arduino* pode receber informações a partir de sensores (sinal de entrada) e enviar informações através de actuadores (sinal de saída) para um computador (Banzi, 2008, pp. 29-30). Tanto os sensores como os actuadores são dispositivos electrónicos que permitem que o *Arduino* comunique com o mundo físico. São exemplos desses sensores objectos físicos os botões, sensores de movimento, infravermelhos, sensores de temperatura, sensor *CMOS (Complementary Metal-Oxide-Semiconductor)* e como actuadores os Leds (luz) ou motores. O *Arduino* não só recebe informação proveniente de sensores mas pode também receber informação a partir de outros softwares. Exemplos desses softwares são o *Processing*, *Adobe Flash*<sup>6</sup>, *Max/MSP*<sup>7</sup> ou o *Scratch*. Estes programas podem não só enviar informação para o *Arduino* mas também têm a capacidade de receber. O *Arduino*, depois de ensinado, não necessita de estar ligado a um computador, ele pode actuar sozinho, isto é pode ser criada uma interface que comunica com o computador através do *Arduino* e este pode ser alimentado a partir de uma bateria.

### 2.3 Outras versões do Arduino

Existem também outras versões do *Arduino*, algumas são compatíveis com o *IDE Arduino* ou com outros softwares. São exemplos desses micro-controladores: o *Paperduino*<sup>8</sup> desenvolvido por Guilherme Martins, o *Freeduino*, o *Boarduino*<sup>9</sup>, o *Sanguino*<sup>10</sup> ou o *Funnel I/O*<sup>11</sup>. Podem ligar-se a este tipo de micro-controladores os *Shields*<sup>12</sup>, que são placas que permitem expandir o *Arduino*, expansão essa relacionada com o espaço de trabalho (para estabelecer mais ligações) ou com a inclusão de hardware que não foi projectado originalmente nos micro-controladores.

Apesar de ter sido projectado inicialmente para Artistas, Arquitectos e Designers, já se encontram diferentes situações onde estão a ser criadas e utilizadas novas funcionalidades para este tipo de micro-controlador, como é o caso da sua aplicação na robótica (Balogh, sem data), na música (Vallis, Hochenbaum, & Kapur, sem data), na prototipagem de interfaces para

---

<sup>6</sup> <http://www.adobe.com/products/flash/>

<sup>7</sup> <http://cycling74.com/products/>

<sup>8</sup> <http://lab.guilhermemartins.net/2009/05/06/paperduino-prints/>

<sup>9</sup> <http://www.ladyada.net/make/boarduino/>

<sup>10</sup> <http://sanguino.cc/>

<sup>11</sup> <http://funnel.cc/Hardware/FIO>

<sup>12</sup> <http://shieldlist.org/>

jogos (Bekker, Hummels, Nemeth, & Mendels, 2010) ou em material programável para crianças (Eisenberg, Elumeze, MacFerrin, & Buechley, 2009).

### 3. Software e a programação

De acordo com Eisenber *et al* tem existido alguma ênfase na discussão acerca de crianças a programar, nomeadamente na construção de linguagens de programação adequadas para crianças, na necessidade das crianças aprenderem alguns tópicos particulares relacionados com a programação ou até mesmo no tema principal, relacionado com a necessidade ou não das crianças aprenderem a programar (Eisenberg et al., 2009, pp. 3-5). Contudo, estes autores reforçam que a aprendizagem de programação por parte das crianças pode ser um reforço para a sua criatividade, nas áreas da matemática, das ciências e artísticas.

#### 3.1 O software do ToonTalk

Para Kahn (Kahn, 1996, pp. 198-199) aprender a usar computadores sem aprender a programar é o mesmo que aprender a ler sem aprender a escrever. Este autor e a sua equipa desenvolveram o *ToonTalk*<sup>13</sup>, um programa que permite que as crianças programem usando objectos animados e o ambiente de trabalho é o de um jogo de computador. Com a ênfase de criar um ambiente de programação agradável e divertido para as crianças, o *ToonTalk* permite que sejam criados jogos, ao mesmo tempo que permite controlar os sensores e motores do Lego, sem existir a necessidade de instrução de outras pessoas acerca da programação.

#### 3.2 O software Scratch

Existem algumas experiencias na utilização de *open software* e de *open hardware* com crianças, como no caso do *Dialando* (Smith, 2010), que foi desenvolvido através do uso do *Processing*, do *Arduino* e do *Scratch*, uma interface que permite às crianças aprender a programar, usando deste forma tecnologia livre e objectos físicos que funcionam como interfaces físicas de baixo custo.

O *Scratch* foi desenvolvido no *MIT Media Lab*, e trata-se de um software para aprendizagem de programação através de uma interface de blocos individuais. A utilização de uma sintaxe de programação que faça uso de blocos torna a aprendizagem mais explícita do que implícita (Richards & Smith, 2010, p. 16). Para estes autores a utilização do *Scratch* permite que os alunos consigam obter noções básicas para iniciarem o trabalho em qualquer linguagem de programação.

---

<sup>13</sup> <http://www.toontalk.com/>

Associado ao software *Scratch* existe a *Picoboard*<sup>14</sup>. Trata-se de um micro-controlador que possui vários sensores embutidos na mesma placa e que permite a comunicação com o software *Scratch*. Tendo por base a *Picoboard* foi desenvolvida a *SenseBoard* (Richards & Smith, 2010), com a preocupação de acrescentar novas funcionalidades, nomeadamente a inclusão de Leds, a possibilidade de trabalhar com programas mais complexos, a possibilidade de escrita através da internet e de ligar motores. Já o projecto *GogoBoard*<sup>15</sup> apresenta um micro-controlador com características similares, mas inclui um kit com recurso a componentes electrónicos que podem ser adquiridos em qualquer loja. Tanto a *Picoboard* como a *SenseBoard* ou a *GogoBoard* têm capacidades similares ao *Arduino*, mas o *Arduino* apresenta como principal potencial o facto de não ser exclusivo de um único software e não de não estar direccionado apenas para um determinado tipo de utilizadores.

#### 4. Construção de interfaces/narrativas tangíveis

Nos anos oitenta Mitchel Resnick e Stephen Ocko<sup>16</sup> desenvolveram um projecto de nome *LEGO/Logo* que permitia a integração de peças de Legos com a programação, através da linguagem *Logo*. Através de motores e sensores podiam ser construídos robôs ou outros objectos, que eram ensinados e que podiam funcionar sem estarem ligados de forma permanente a um computador. A visão destes autores permitiu que a computação física fosse aprofundada ao mesmo tempo que a utilização de uma linguagem de programação com os Legos fomentava a criatividade das crianças (Millner, 2009, p. 12).

Actualmente o recurso a componentes electrónicos, actuadores e sensores não é dispendioso, ao mesmo tempo que o espaço físico que eles ocupam permite que esses componentes estejam escondidos, decorados de forma livre e dentro de objectos cujas dimensões físicas são adequadas às crianças (Eisenberg et al., 2009, p. 4). Não é nossa intenção aprofundar o trabalho para o nível da computação ubíqua (interagir com o computador apenas com acções invisíveis), pois pretendemos que as crianças construam as suas interfaces, os objectos das suas narrativas e também queremos evitar que as crianças sintam a falta de controlo.

Para Montemayor (Montemayor et al., 2004) a utilização de tecnologia frágil por parte das crianças, como é o caso de alguns sensores e actuadores, não lhes permite construir comportamentos cognitivos consistentes e comportamentos previsíveis. Deve ser também tida em conta a situação de que quando a tecnologia necessita de um determinado grau de precisão para funcionar ou é interrompida por mau funcionamento isso distrai a criança, retirando-lhe o controlo. Assim, os objectos utilizados devem ser de confiança, duradouros, sólidos e baseados em tecnologias flexíveis.

---

<sup>14</sup> <http://www.picocricket.com/picoboard.html>

<sup>15</sup> <http://gogoboard.stanford.edu/>

<sup>16</sup> <http://llk.media.mit.edu/papers/ll.html>

Actualmente encontramos alguns exemplos da utilização de electrónica com crianças: *Blackbox* (Andersen, 2008), experiencia com crianças da pré-escola, onde elas são convidadas a interagir com uma caixa com sensores e leds; *Storyroom* (Montemayor et al., 2004), onde crianças, entre os 4 e os 6 anos, interagem com diversos objectos, nomeadamente blocos do Lego Duplo, que incorporam sensores e são utilizados para que construam e contem as suas histórias; os *Electronic Blocks* (Wyeth & Wyeth, 2010) destinado a crianças entre os 3 e os 8 anos, permite que elas façam uso de blocos de Lego Duplo que contêm circuitos electrónicos e com a disposição desses blocos consigam programar e construir outros objectos como robôs, animais, entre outros.

## 5. Conclusão e Trabalho Futuro

A utilização do *Arduino* apresenta como principais vantagens ser *open hardware*, ser um micro-controlador compatível com vários sistemas operativos, entre os quais o *Linux*, ter um preço de mercado reduzido e ter uma interface que permite a comunicação com outros softwares. A curva de aprendizagem é rápida e prevemos que se obtenham resultados quase imediatos com a sua utilização. Sendo um software livre, as bibliotecas necessárias para o adoptar estão disponíveis e são também livres. A utilização de objectos programáveis, ou seja, ambientes de programação direccionados para crianças e a constante envolvência dos Professores na aprendizagem irá, como afirmam (Eisenberg et al., 2009, p. 5) (e com o qual concordamos) conduzir as crianças a uma construção enriquecida de narrativas, arte e simulações e possibilitar uma presença diária, dentro e fora da sala de aula, no intelectual delas.

Não pretendíamos debater a questão das vantagens e desvantagens de serem softwares e hardwares livres ou não, mas sim de fomentar na comunidade o conhecimento que existe e que pode ser desenvolvido e utilizado em diversos contextos. Tentámos ao longo deste documento destacar as principais características e valências, para que a comunidade tenha interesse em utilizar e expandir esta rede do conhecimento.

Pretendemos continuar a aprofundar os temas aqui apresentados e, num futuro próximo, disponibilizar formação junto de um grupo de alunos de mestrado do Ensino Básico (futuros Professores) para que possam desenvolver este projecto durante o período de estágio, criando actividades relacionadas com computação física e a construção de narrativas digitais tangíveis, junto de crianças entre os 4 e os 8 anos de idade.

## Agradecimentos

<sup>1</sup>Financiado pelo PROTEC e PTOFAD desde Abril de 2010. <sup>2</sup>Co-financiada pela FCT e pelo Fundo Social Europeu (ESF) como parte do Programa Operacional Potencial Humano Europeu

(POPH) inserido no QREN, de 1 Janeiro de 2009 a Agosto de 2010 e financiada pelo PROTEC desde Setembro de 2010.

## Referências

- Andersen, K. (2008). Black box: exploring simple electronic interaction. Em *Proceedings of the 2nd international conference on Tangible and embedded interaction* (pp. 207–208).
- Balogh, R. (sem data). Educational Robotic Platform based on Arduino.
- Banzi, M. (2008). *Getting Started with Arduino* (1st ed.). Make.
- Bekker, T., Hummels, C., Nemeth, S., & Mendels, P. (2010). Redefining toys, games and entertainment products by teaching about playful interactions. *International Journal of Arts and Technology*, 3(1), 17–35.
- Buechley, L., & Eisenberg, M. (2008). The LilyPad Arduino: Toward Wearable Engineering for Everyone. *IEEE Pervasive Computing*, 7(2), 12-15.
- Eisenberg, M., Elumeze, N., MacFerrin, M., & Buechley, L. (2009). Children's programming, reconsidered: settings, stuff, and surfaces. Em *Proceedings of the 8th International Conference on Interaction Design and Children* (pp. 1–8).
- Igoe, T., & O'Sullivan, D. (2004). *Physical Computing: Sensing and Controlling the Physical World with Computers* (1st ed.). Course Technology PTR.
- Kahn, K. (1996). ToonTalkTM--An Animated Programming Environment for Children. *Journal of Visual Languages & Computing*, 7(2), 197-217. doi:doi: DOI: 10.1006/jvlc.1996.0011
- Millner, A. (2009). *Computer as Chalk*. Massachusetts Institute of Technology.
- Montemayor, J., Druin, A., Chipman, G., Farber, A., & Guha, M. L. (2004). Tools for children to create physical interactive storyrooms. *Computers in Entertainment (CIE)*, 2(1), 12.
- Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W., & D'Amour, A. (2002). Physical programming: designing tools for children to create physical interactive environments. Em *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* (pp. 299–306).
- Noble, J. (2009). *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks* (1st ed.). O'Reilly Media.
- OSHW - Definition of Free Cultural Works. (sem data). . Obtido Fevereiro 22, 2011, de <http://freedomdefined.org/OSHW>
- Richards, M., & Smith, N. (2010). Teaching UbiComp with sense. Em *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10 (pp. 765–768). New York, NY, USA: ACM. doi:10.1145/1868914.1869024

- Scarlatos, L. L., Mbogho, A., & Jaworska, M. (2006). Teaching with Tangibles: A Tool for Defining Dichotomous Sorting Activities.
- Smith, A. C. (2010). Dialando: tangible programming for the novice with scratch, processing and arduino.
- The Open Source Definition (Annotated) | Open Source Initiative. (sem data). . Obtido Fevereiro 23, 2011, de <http://www.opensource.org/osd.html>
- Vallis, O., Hochenbaum, J., & Kapur, A. (sem data). A Shift Towards Iterative and Open-Source Design for Musical Interfaces.
- Wyeth, P., & Wyeth, G. F. (2010). Electronic blocks: Tangible programming elements for preschoolers. Em *IFIP TC. 13 International Conference on Human-Computer Interaction* (Vol. 1, pp. 496–503).
- Wyeth, P., & Purchase, H. C. (2002). Tangible programming elements for young children. Em *CHI '02 extended abstracts on Human factors in computing systems*, CHI EA '02 (pp. 774–775). New York, NY, USA: ACM. doi:10.1145/506443.506591