

Interface de programação tangível para produção de algoritmos sonoros

Cassiano Pereira Viana, André Luis Alice Raabe

Curso de Bacharelado em Ciência da Computação – Universidade do Vale do Itajaí

cassiano.viana@edu.univali.br, raabe@univali.br

Abstract. *With the popularization of Computational Thinking, fundamental concepts of Computing have reached an increasing audience, however, the analysis of scientific production in the area indicates that there are still few solutions that aim to teach visually impaired users. This work presents a tangible low cost programming interface that uses a cell phone and blocks with fiducial markers in order to allow users to create sound algorithms working concepts such as conditional and repetition. The interface was developed with the perspective of universal and inclusive design in order to try to reach the widest possible audience. It was adopted an incremental development that allowed to evolve the interface from tests with users. The tests performed with 9 users, being a blind one, revealed that: (1) even without knowing Braille the blind user was able to use the interface satisfactorily; (2) the interface serves a wide age range from 8 years and; (3) the users performed better when the interface symbols were expanded.*

Resumo. *Com a popularização do Pensamento Computacional, conceitos fundamentais de Computação tem alcançado um público cada vez maior, no entanto, a análise da produção científica na área indica que ainda existem poucas soluções voltadas a ensinar usuários deficientes visuais. Este trabalho apresenta uma interface de programação tangível de baixo custo que utiliza um telefone celular e blocos com marcadores fiduciais a fim de possibilitar que usuários criem algoritmos sonoros trabalhando conceitos como condicional e repetição. A interface foi desenvolvida com a perspectiva de design universal e inclusivo de forma a tentar atender ao maior público possível. Foi adotado um desenvolvimento incremental que permitiu evoluir a interface a partir de testes com usuários. Os testes realizados com 9 usuários, sendo um cego, revelaram que: (1) mesmo sem saber braille o usuário cego conseguiu utilizar satisfatoriamente a interface; (2) a interface atende a uma ampla faixa etária a partir de 8 anos e; (3) os usuários se saíram melhor quando foi ampliada a condução por meio dos símbolos da interface.*

1. Introdução

A computação e o ensino de Pensamento Computacional estão sendo levados a um público cada vez maior. O Pensamento Computacional é uma habilidade cada vez mais demandada em todas as áreas do conhecimento ao mesmo tempo que pode ser utilizado para resolver problemas do dia-a-dia (Wing, 2010). Isso evidencia a necessidade de se levar o Pensamento Computacional a todas as pessoas, independentemente de idade, cultura, nível de escolaridade, ou necessidade especial. No entanto, são poucas as

alternativas que contemplam os deficientes visuais por exemplo. Trabalhos como os de Dantas, Pinto e Sena (2013), e Santana et al (2017) evidenciam a preocupação em incluir os deficientes visuais, porém sem abordar a perspectiva de Pensamento Computacional. Segundo Kakehashi (2013), no âmbito escolar, o ensino de programação tradicional com uso de monitores afasta os deficientes visuais do processo de interação que acontece quando dois indivíduos discutem em uma ambiente acessível a ambos. Em contrapartida, Kuss (2016) indica a ampliação gradativa de estudantes com deficiência visual nas redes de ensino regulares.

Nesse contexto, este trabalho criou uma maneira de viabilizar o ensino de noções de algoritmos (repetição, desvio, sequenciamento de instruções) de forma inclusiva e de baixo custo. Foi criada uma interface de programação tangível, utilizando apenas um telefone celular e uma base de madeira, que possibilita aos usuário criarem algoritmos sonoros inserindo peças que representam ações sonoras ou de controle de fluxo. A figura 1 ilustra a interface construída em uso. No topo da haste de madeira encontra-se o telefone celular que através da câmera reconhece e executa os algoritmos representados nos blocos inseridos pelos usuários no painel.

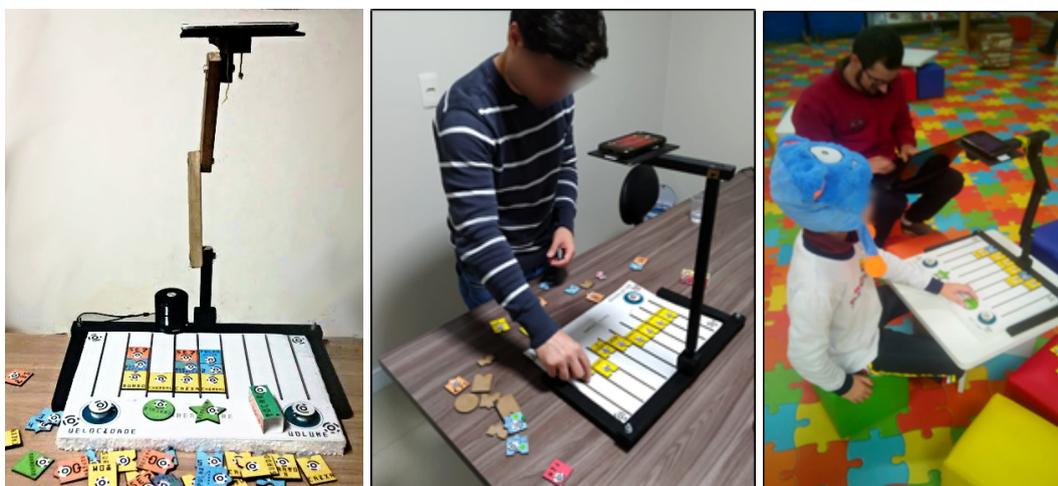


Figura 1: Interface tangível desenvolvida

Todo o projeto foi desenvolvido com base no design inclusivo universal. O design inclusivo foi utilizado ao acrescentar-se características (o Braille), como uma forma de incorporar um grupo específico (os cegos) a utilizar a interface de forma que não era atendido antes das alterações (DUBUC e EDGE, 2006). Já o Design Universal foi utilizado na visão de que embora tenham sido adicionadas características inclusivas específicas, o projeto foi pensado para ser utilizado por um público-alvo amplo. Segundo Newell e Gregor (2000), alcançar a totalidade dos usuários é uma tarefa praticamente impossível.

2. Interfaces Tangíveis

A interface desenvolvida se encaixa na categoria das interfaces tangíveis, as TUI (*Tangible User interfaces*) e mais especificamente na categoria da *Tangible Programming Interfaces*. Segundo Ishii (2008), as TUI proveem forma física para a informação digital. São construídas sobre as habilidades sensoriais humanas de forma a ligar o mundo digital com o físico, possibilitando libertação do confinamento gerado

pelos interfaces gráficas. Segundo Horn e Jacob (2007), dar à programação uma forma física facilita a compreensão de sintaxes complicadas, promove a colaboração e torna o ambiente da sala de aula positivo. Argumentam que as tecnologias usadas nessas interfaces devem ser baratas e duráveis, viabilizando sua implementação nas escolas públicas. Propõe então o termo “interface tangível passível” para designar uma coleção de componentes físicos que não necessitam de bateria ou de ligação contínua a um sistema digital.

As linguagens Tern e Quetzal (HORN e JACOB, 2007) utilizam marcas fiduciais sobre as peças encaixadas para controlar os robôs iRobot Create™ e LEGO Mindstorms™, respectivamente. Essas marcas são capturadas por uma câmera posicionada sobre as peças, interpretadas por um computador e enviadas para o robô via tecnologia sem fio. O trabalho usa a biblioteca TopCodes que é capaz de reconhecer cerca de 99 marcas menores de 25x25px (HORN, 2018).

Já projetos como a TurTan, de Gallardo et al. (2008), e a mesa musical reacTable (KALTENBRUNNER e BENCINA, 2007), posicionam a câmera sob as peças dispostas livremente sobre uma superfície transparente. Ao lado da câmera, para o retorno visual das peças dispostas, é utilizado um projetor. Em outra linha, a interface AlgoBlock, que é baseada na linguagem Logo, utiliza componentes eletrônicos nos blocos, os quais são ligados a um computador para executar o programa (SUZUKI e KATO, 2016). Essa interface diferencia-se das outras pelo fato de utilizar componentes eletrônicos individualmente nos blocos.

O projeto P-CUBE, é o único adaptado a deficientes visuais. Não utiliza câmera nem componentes eletrônicos nas peças, mas tags RFID em blocos que são encaixados em uma matriz. O resultado também é a movimentação de um robô (KAKEHASHI et al., 2013). Dessa interface, a parte da grade inspirou a divisão dos tempos musicais do tabuleiro. A figura 2 ilustra os trabalhos mencionados.

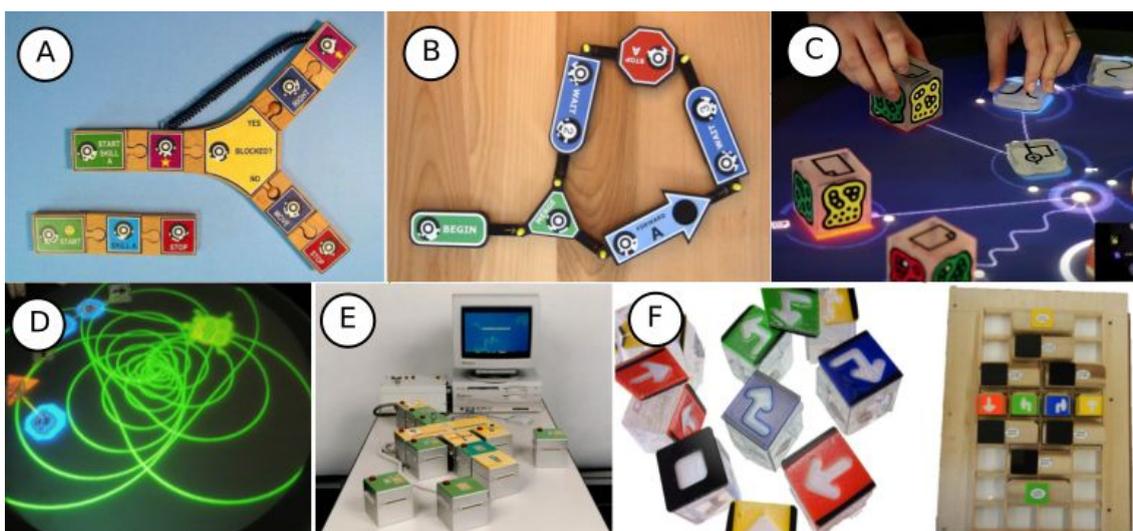


Figura 2: Tern (A), Quetzal (B), reacTable (C), TurTan (D), AlgoBlock (E), P-CUBE (F)

Esses trabalhos influenciaram as escolhas do projeto de forma que decidiu-se utilizar a biblioteca TopCodes como em (HORN, 2018), utilizando uma instalação com uma câmera focada para uma superfície como em (KALTENBRUNNER e BENCINA,

2007) e evitar o uso de recursos que aumentariam o custo do projeto tais como os componentes eletrônicos usados em (SUZUKI e KATO, 2016) ou o RFID usado em (KAKEHASHI et al., 2013).

3. Projeto da Interface Tangível

O projeto é uma interface tangível que permite ao usuário montar a sequência de sons de uma bateria ou sons gravados pelo usuário. A dinâmica de uso é colocar blocos com os nomes das peças (caixa, bumbo, chimbau etc) em uma grade de divisão sobre a qual é fixado um celular que captura as peças e emite a “música”. A ideia de programação está inserida em blocos que permitem repetir (conceito de loop), desativar ou pular peças (conceito de desvio condicional).

A figura 3 mostra como a interface foi organizada. A parte superior do tabuleiro é dividida em 8 partes onde as peças são encaixadas. Na parte inferior esquerda está o controle de velocidade e na direita o controle de volume. Na parte inferior central fica a área onde devem ser colocados os blocos de presença e o bloco de estados.

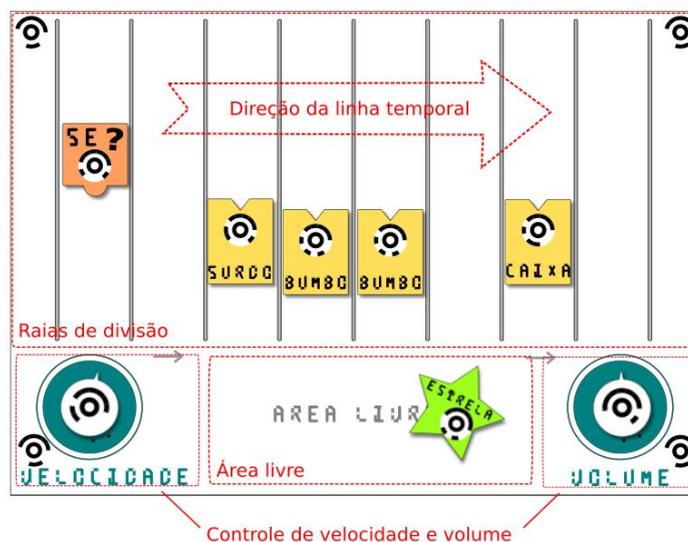


Figura 3: Organização do tabuleiro

O funcionamento acontece através de uma linha temporal que se move da esquerda para a direita, com velocidade constante. Quando a posição da linha temporal coincide com as peças de som, o celular executa o áudio associado à peça. Controles de início, interrupção e gravação são verificados a cada modificação na disposição das peças, independentemente da posição da linha. As peças são dispostas sobre um tabuleiro com divisões. Cada um dos quatro cantos do tabuleiro tem uma marca fiducial.

O tabuleiro é capturado pela câmera do celular no qual está instalado um aplicativo que é responsável por acessar a câmera, obter os dados de posicionamento, angulação, diâmetro e tipo de cada peça, calcular os tempos de execução, repetições e intervalos, emitir instruções de ajuda ao usuário e emitir o som programado. O aplicativo foi desenvolvido usando as linguagens Kotlin e Java atendendo a plataforma Android a partir da versão 5.0.

Para montar a sequência sonora utilizou-se como base a gramática demonstrada

no Quadro 1: a música é formada por uma sequência de sons. Cada som é formado por uma instrução e um instrumento ou apenas o instrumento. As instruções são as estruturas de controle SE ou REPITA. As instruções de controle não permitem instruções aninhadas devido às restrições do tabuleiro.

```

1 <musica> ::= <musica><som> | <som>;
2 <som> ::= <instrucao><instrumento> | <instrumento>;
3 <instrucao> ::= <se> | <repita>;
4 <se> ::= se <parametro_se><som>;
5 <repita> ::= repita numero <som>;
6 <parametro_se> ::= <tem_estrela> | <tem_circulo>;
7 <instrumento> ::= som;

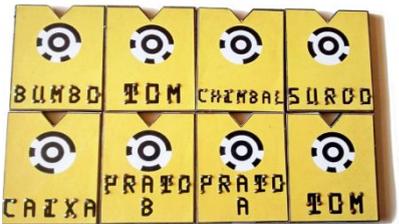
```

Quadro 1 - Gramática

A instrução SE aceita apenas um dos dois parâmetros “se tem estrela” ou “se tem círculo”, que se refere a presença ou ausência destes objetos no tabuleiro. A instrução REPITA aceita números até 5. O *token* instrumento é um som da bateria ou um som gravado pelo usuário.

4. Projeto dos Blocos

O blocos são feitos de madeira revestida por papel. Sobre cada peça há pelo menos uma marca fiducial. Os blocos possuem marcações em braille feitas com tinta relevo. São 7 tipos de blocos: som, repetição, condição, parâmetros, blocos de presença (estrela e círculo), bloco de estados e blocos giratórios de controle de volume e velocidade. Entre os blocos de parâmetro há números e testes condicionais relacionados com os blocos de presença. O quadro 2 detalha os tipos de blocos.

<p>Bloco de som</p> 	<p>Os blocos de som tem dimensões de 3,9cm x 5,4 cm, possuem a descrição do instrumento na parte inferior e a marca fiducial de 2cm no centro. Foram produzidos 7 pares de sons de partes da bateria. Na parte superior da peça podem ser encaixadas as peças de parâmetro de repetição ou teste condicional.</p>
<p>Bloco SE</p> 	<p>O bloco SE, para ser usado necessita de um dos dois blocos de parâmetro: “se tem estrela” ou “se tem círculo”. Se estas peças estiverem presentes no tabuleiro (círculo ou estrela) o bloco de som afetado é tocado, caso contrário não. Estar presente significa estar visível no ângulo de visão da câmera e dentro das coordenadas indicadas pelas quatro marcas das extremidades.</p>

<p>Bloco Repita</p> <p>1 </p> <p>2 </p>	<p>Funciona de forma semelhante ao bloco SE, sendo encaixado com parâmetros que modificam o comportamento dos blocos de som. Os parâmetros são os blocos de número, que indicam o número de vezes que o som da peça deve ser tocado. O efeito é de que a linha temporal pára sobre a peça para executá-la mais que uma vez, mas na verdade são criados mais blocos e os seguintes são movidos para a direita e a linha continua seu movimento normalmente.</p>
<p>Bloco de estados</p> 	<p>A interface pode ter quatro estados: iniciado, parado, gravando e em modo ajuda. O aplicativo inicia no primeiro modo, tocando as peças disponíveis. O bloco de estados é responsável por ativar cada modo. A gravação possibilita atribuir um som gravado a uma peça. E a Ajuda fornece auxílio para uso da interface com mensagens gravadas.</p>
<p>Blocos de volume e velocidade</p> <p>1 </p> <p>2 </p>	<p>Os blocos VOLUME e VELOCIDADE utilizam o ângulo para definir a intensidade. Ao girar para a esquerda o volume ou a velocidade diminuem e aumentam no sentido contrário. No caso da velocidade, o maior tempo que a linha leva para atravessar a interface é 4 segundos. Conforme a peça de velocidade é girada no sentido horário a linha temporal acelera até no máximo duas vezes.</p>

Quadro 2. Blocos e descrições

5. Testes Realizados

Para realização dos testes buscou-se envolver o usuário na condição real de uso e observar as reações e dificuldades em cada tarefa. Segundo Cybis, Betiol e Faust (2015), nessa técnica, o usuário é convidado a utilizar a interface, diante do olhar crítico do avaliador atento às perdas de tempo, retrabalho e mesmo aos bloqueios nas interações. Segundo os autores é a técnica mais confiável e mais dispendiosa.

Primeiramente definiu-se uma lista de cinco tarefas a ser realizada pelo usuários e em um segundo momento foi realizado um teste mais livre com o objetivo de avaliar o quanto a interface conduzia o usuário. A interface foi testada por 9 usuários. No primeiro teste foram observados 5 usuários, incluindo um deficiente visual. Foram realizadas modificações a partir dos resultados e foi conduzido um segundo teste com 4 pessoas incluindo uma criança de 8 anos e uma de 5 anos. No primeiro teste foram observados a consecução das tarefas (conseguiu, não conseguiu, conseguiu com ajuda), e no segundo teste os tempos dos eventos de uso.

5.1. Teste com o usuário cego

O voluntário cego que participou do teste disse não conhecer Braille de forma que as

marcações nas peças foram inúteis. Tinha algum conhecimento de programação, e isso o ajudou a entender os conceitos de repetição e desvio. Cada tarefa foi requisitada após a apresentação de exemplos. As peças foram sendo entregues ao usuário progressivamente conforme a tarefa que ele realizava.

Primeiramente o usuário foi guiado pelas divisões do tabuleiro, controles de volume e velocidade e posição da câmera. Para a identificação da posição da câmera foi adicionado um barbante de forma que o usuário poderia senti-lo quando as peças estivessem na altura requerida para os modos de pronúncia da ajuda e seleção da gravação. Em seguida foi iniciada a produção de sons. Primeiro, o usuário recebeu quatro peças e às dispôs sobre o tabuleiro. Ele conseguiu entender a lógica da sequência da música. Nesse momento, foi requisitado que utilizasse as peças de controle de volume e velocidade, ao que foi capaz de utilizá-las, embora ele tenha observado uma demora no tempo de resposta.

A segunda tarefa solicitada ao usuário foi que, depois de exemplificado, repetisse alguma peça utilizando o bloco REPITA. O usuário encontrou dificuldade, mas conseguiu entender e repetir algumas peças com ajuda. A terceira tarefa solicitada ao usuário foi que controlasse se uma peça tocava ou não tocava de acordo com a adição ou remoção da peça em formato de círculo em conjunto com o bloco SE. O usuário conseguiu realizar a tarefa facilmente.

Por último, foi testado o processo de gravação de um som personalizado em uma peça. Nesse teste, a operação de gravação só foi conseguida depois de algumas tentativas e com ajuda. O usuário não entendeu que devia produzir o som no momento em que a voz orientava quando iniciar com uma contagem regressiva dizendo “Produza o som em 3, 2, 1...”.

5.2. Resultados do primeiro teste

O Quadro 3 considera os 5 usuários que testaram a primeira versão da interface. Entre esses está o usuário cego (usuário 5).

Tarefa/usuário	1	2	3	4	5 (cego)
Montar sequência de sons	CA	CA	F	F	F
Controlar velocidade	F	F	F	F	F
Controlar volume	F	F	F	F	F
Repetir peças usando REPITA	F	CA	F	CA	CA
Controlar se toca ou não usando SE	F	CA	CA	CA	F
Gravar uma peça usando GRAVAR	NC	NC	CA	NC	NC
Ouvir explicação usando AJUDA	F	F	CA	CA	CA

Quadro 3. Tarefas e desempenho (F = facilmente, CA = com ajuda, NC = não conseguiu)

Os dados mostram que as maiores dificuldades concentraram-se nas tarefas de utilizar o bloco de gravação. Isso ocorreu porque os usuários não conheciam a possibilidade de gravar uma peça com um som próprio, pois a ajuda não ficou suficientemente clara. É interessante notar que não houve diferença significativa de uso entre o usuário cego em relação aos demais usuários.

5.3. Resultados do segundo teste

No primeiro teste foi observado que os usuário não entenderam o propósito da área livre em relação aos blocos de estrela e círculo. Também tiveram dificuldade para entender a relação entre as peças SE e REPITA e as peças de parâmetro. Além disso, foi sugerido que todas as divisões do tabuleiro fossem percorridas e não somente as que possuíam peças amarelas. Por isso foram tomadas algumas medidas: (1) Foram demarcadas as áreas do círculo e da estrela na área livre; (2) Foi eliminada a necessidade das peças REPITA e SE, sobre as peças de parâmetro; e (3) Foi alterado o controle do início e fim da linha para percorrer todas as casas. Além dessas alterações foi implementada uma modificação sugerida pelo usuário cego: fazer com que as peças mais próximas a ele tivessem um som mais alto que as peças de trás.

A análise do Quadro 4 foi baseada nos eventos observados durante o segundo teste. Quando os usuários conseguiam, produzir um som, utilizar as peças de controle ou descobrir a relação do volume com a posição da peça no eixo y.

Usuário	1	2	3	4
Idade	8	21	5	21
Sexo	M	F	F	F
Eventos de uso				
Produzir som	23s	1m 23s	2m 29s CA	2m 33s CA
Entender que seria uma bateria	35s	2m	-	3m CA
Começar encaixar/entender encaixes	58s	1m 30s	-	4m
Compreender funcionamento/direção da linha	-	2m 30s	-	-
Compreender peça SE	1m47s	8m CA	-	12m CA
Compreender peça REPITA	5m CA	5m 30s CA	-	8m CA
Entender efeito do volume por aproximação	6m CA	11m CA	-	14m CA
Entender VELOCIDADE	2m	-	-	5m
Entender VOLUME	8m	-	-	4m 20s

Quadro 4. Análise baseada no tempo dos eventos (CA = com ajuda)

Os resultados mostram que o menino de 8 anos conseguiu utilizar a interface facilmente, descobrindo seu propósito e funções rapidamente. Já a menina de 5 anos não obteve sucesso, apenas conseguiu produzir sons, mas não entendeu como funcionava. Concentrou seu interesse em encaixar as peças da mesma cor, sem se preocupar com os efeitos sonoros. Acredita-se que a interface não está atrativa o suficiente ou é muito complexa para essa idade.

Os outros usuários, de 21 anos apresentaram um comportamento diferente: demoraram em média 2 minutos para começar a encaixar as peças, tentando compreender todo o funcionamento antes de começar. Uma questão comentada foi que não conheciam os sons das peças da bateria pelo nome e que por isso teriam mais dificuldade em compreender a sequência.

6. Conclusões

Como conclusões deste trabalho foi possível entender que a mera inclusão de instruções em Braille não é suficiente para alcançar o público dos deficientes visuais. Independente disso, o usuário cego conseguiu realizar as tarefas de forma semelhante aos usuários com visão normal. Foi também possível concluir que as diferenças de uso se dão muito mais pela idade do que pelo fato de a pessoa ter ou não a visão.

Quanto ao desempenho da biblioteca TopCodes, foi satisfatória no sentido da precisão dos dados de posição das marcas fiduciais, embora tenha ocorrido casos de ativação de outros estados quando o bloco de estados não estava presente. A biblioteca comportou-se muito bem mesmo com a variação de luz, ocorrendo erros apenas nos casos onde havia reflexo sobre as peças, revelando que a impressão das marcas deve ser o mais fosca possível.

Comparando-se com a interface P-CUBE (KAKEHASHI et al., 2013), pode-se afirmar que a utilização de uma grade de divisão foi utilizada em ambas as interfaces, mas com propósitos diferentes. Enquanto na P-CUBE, a grade serviu apenas para auxiliar no sequenciamento das instruções, neste trabalho a divisão foi utilizada também para controlar a linha de tempo, uma vez que o feedback ao usuário é sonoro. Uma diferença notável é que em P-CUBE são utilizados vários lados dos blocos, enquanto neste trabalho foi utilizada apenas uma face dos blocos, o que facilitou a interação.

A produção dos blocos em madeira foi simples e em conjunto com a impressão do papel couche resultou em blocos de boa durabilidade. Dessa forma pode-se afirmar que é uma maneira simples e barata de produzir os elementos da interface. Já na parte do desenvolvimento do aplicativo, a escolha da linguagem Kotlin foi acertada pela facilidade que ofereceu na parte de filtros dos objetos dos tipos de blocos.

Embora a interface tenha apresentado problemas de usabilidade a experiência lúdica de programar sons de uma bateria mantiveram o interesse dos usuários por um tempo razoável. O principal trabalho futuro é a ampliação dos testes com mais usuários deficientes visuais.

Referências

- BARROS, Renata Pitta et al. CardBot - Assistive Technology for Visually Impaired in Educational Robotics: Experiments and Results. *Ieee America Transactions*, [s.l.], v. 15, n. 3, p.517-527, mar. 2017. Institute of Electrical and Electro Engineers (IEEE). <http://dx.doi.org/10.1109/tla.2017.7867603>.
- BORDINI, Adriana et al. Computação na Educação Básica no Brasil: o Estado da Arte. *Rev. Informática Teoria Aplicada* (online), Porto Alegre, p.210-238, nov. 2016.
- CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. *Ergonomia e Usabilidade: Conhecimentos, Métodos e Aplicações*. 3. ed. São Paulo: Novatec, 2015. 496 p.
- DANTAS, André Luiz P.; PINTO, Gabriela R. P. Rezende; SENA, Claudia Pinto P.. Apresentando o BEM: Um Objeto de Aprendizagem para mediar o processo educacional de crianças com deficiência visual e videntes nas operações básicas de

Matemática. XXIV Simpósio Brasileiro de Informática na Educação SBIE, 2013.

DUBUC, L., Edge D. TUIs to ease: tangible user interfaces in assistive technology. Proceedings of the 3rd Cambridge Workshop on Universal Access and Assistive Technology (CWUAAT 2006). p.1–10, 2006. Cambridge University Engineering Department.

HORN, Michael S.; JACOB, Robert J. K.. Designing tangible programming languages for classroom use. Proceedings Of The 1st International Conference On Tangible And Embedded Interaction - Tei '07, [s.l.], p.159-162, 2007. ACM Press. <http://dx.doi.org/10.1145/1226969.1227003>.

ISHII, Hiroshi. Tangible bits. Proceedings Of The 2nd International Conference On Tangible And Embedded Interaction - Tei '08, [s.l.], p.15-25, 2008. ACM Press. <http://dx.doi.org/10.1145/1347390.1347392>.

KATO, Hiroko; TAN, Keng; CHAI, Douglas. Barcodes form Mobile Devices. New York: Cambridge University Press, 2010.

KUSS, Paulo Fernando. Análise da inclusão das crianças cegas na educação regular: um olhar para a tecnologia assistiva. 2016. Dissertação (mestrado em mestrado acadêmico em educação) - Universidade do Vale do Itajaí, .

NEWELL, Alan F.; GREGOR, Peter. “User sensitive inclusive design”--- in search of a new paradigm. Proceedings On The 2000 Conference On Universal Usability - Cuu '00, [s.l.], p.39-44, 2000. ACM Press. <http://dx.doi.org/10.1145/355460.355470>.

SANTANA, Kayo et al. Blinds, Basic Education: jogo digital inclusivo para auxiliar o processo de ensino-aprendizagem das pessoas com deficiência visual. Anais do Xxviii Simpósio Brasileiro de Informática na Educação (sbie 2017), [s.l.], p.877-886, 27 out. 2017. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC). <http://dx.doi.org/10.5753/cbie.sbie.2017.877>.

THIEME, Anja et al. Enabling Collaboration in Learning Computer Programing Inclusive of Children with Vision Impairments. Proceedings Of The 2017 Conference On Designing Interactive Systems - Dis '17, [s.l.], p.1-14, 2017. ACM Press. <http://dx.doi.org/10.1145/3064663.3064689>.

WING, Jeannette M.. Computational Thinking What and Why? Thelink, [s.l.], 17 de novembro de 2010. Disponível em <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>>. Acesso em 8 de julho de 2018.