

Criando aplicativos sobre o descarte adequado de lixo: experiências utilizando uma abordagem temática em um clube de programação

Andrea S. Charão¹, Ana Luísa V. Solórzano², Rafael G. Trindade¹

¹ Departamento de Linguagens e Sistemas de Computação

²Curso de Ciência da Computação
Universidade Federal de Santa Maria
Santa Maria, RS, Brasil

{andrea, alsolorzano, rtrindade}@inf.ufsm.br

Abstract. *Programming clubs are an effective approach to bring computing fundamentals to school-age students. As non-formal, after-school activities, clubs face the challenge of creating stimulating learning environments with meaning to the participants. This paper presents an experience that embeds a social responsibility theme across multiple projects in a programming club, as a meaningful way to engage students while strengthening their environmental awareness. During 9 workshops mentored by undergraduate students, the club participants created mobile applications using MIT App Inventor, around the theme of proper waste disposal. The article discusses some key decisions and lessons learned that may be helpful in replicating the experience.*

Resumo. *Clubes de programação são abordagens efetivas para disseminar fundamentos de computação ao público da educação básica. Por se tratarem de atividades extra-classe, de ensino não-formal, clubes enfrentam o desafio de criar ambientes de aprendizado estimulantes e com significado aos participantes. Neste trabalho, apresenta-se uma experiência que inseriu um tema de responsabilidade social em atividades de um clube de programação, envolvendo os participantes em uma temática de conscientização ambiental. Ao longo de 9 oficinas tutoradas por universitários, os participantes do clube criaram aplicativos usando MIT App Inventor, em torno da temática do descarte adequado de lixo. O artigo discute decisões-chave e lições aprendidas que favorecem a replicação da experiência.*

1. Introdução

Clubes de programação são iniciativas notáveis, que fazem parte de uma tendência mundial de disseminação de noções de Computação a estudantes em idade escolar. Em contraste com ambientes de ensino tradicionais, os clubes desenvolvem o ensino não-formal, que é não-obrigatório, mais adaptável e centrado nos alunos [Hamadache 1991, Eshach 2007]. Clubes têm sido empregados não apenas para envolver os estudantes em uma intersecção entre educação e recreação, mas também para complementar a educação formal [Sahin 2013, Gottfried and Williams 2013].

Relatos de experiência de clubes de programação e iniciativas similares são bastante diversificados, variando conforme a idade do público-alvo, objetivos de aprendizado, atividades e ferramentas. Existem clubes voltados ao ensino fundamental [Smith et al. 2014] e ao ensino médio [Weintrop and Wilensky 2015]. As atividades variam desde exercícios de computação desplugada e quebra-cabeças lógicos [Bellettini et al. 2014] até o desenvolvimento de jogos e animações [Lakanen et al. 2014]. Em alguns casos, as atividades são feitas para serem reutilizadas [Dee et al. 2017].

Mesmo com recursos reutilizáveis, o planejamento de atividades extra-classe pode ser desafiador [Giaya et al. 2015], visto que as experiências são únicas, devido às expectativas individuais e heterogêneas dos estudantes e tutores. Mesmo quando os tutores propõem atividades que estimulam a criatividade e levam em conta as preferências pessoais, permanece o desafio de agregar significado às criações feitas utilizando programação [Lye and Koh 2014]. Os estudantes tendem a se engajar e aprender mais construindo programas que fazem sentido em suas vidas diárias [Kafai and Resnick 1996].

Em muitos casos, os autores exploram abordagens temáticas para atingir o objetivo de dar significância ao processo de aprendizagem [Tessier and Tessier 2015]. Nessa abordagem, um tema específico conecta várias atividades, unidades de ensino ou disciplinas em um currículo completo. Em sua maioria, as revisões de literatura sobre abordagens temáticas se concentram na educação formal, e discutem as vantagens e dificuldades de implementar a aprendizagem temática em sala de aula [Tessier and Tessier 2015, Retnawati et al. 2017].

No presente trabalho, apresenta-se uma experiência na qual um clube de programação incorporou um tema de responsabilidade social em vários projetos extra-classe, de modo a envolver os participantes em uma temática de conscientização ambiental, incentivando o descarte adequado de lixo. Esta iniciativa integra um programa de extensão universitária iniciado em 2014, envolvendo professores, funcionários e alunos de graduação em Ciência da Computação que participam como tutores voluntários. A experiência de um semestre envolveu alunos de 11 a 17 anos de 6º ao 9º ano de uma escola localizada em um bairro socialmente desfavorecido, na cidade de Santa Maria, RS.

2. Trabalhos relacionados

Dentre as iniciativas atuais para ensinar programação a iniciantes, em larga escala, destacam-se Code Club¹ e Code.org². O Code Club propõe atividades e guias para que jovens de 9 a 13 anos aprendam programação enquanto constroem jogos, animações e sites, utilizando Scratch, HTML, CSS e Python, mas sem opções para criar aplicativos móveis. O Code.org oferece diversos recursos, com diferentes níveis de dificuldade e para diferentes grupos de idade. Essa iniciativa disponibiliza atividades e tutoriais, popularizados no evento Hora do Código³, e planos de aula que podem ser usados em intervenções mais longas. Alguns tutoriais de Code.org focam na criação de aplicativos móveis com MIT App Inventor, Thinkable, App Lab ou CodeHS.

¹<http://codeclub.org>

²<http://code.org>

³<http://hourofcode.com>

Lakanen et al. [Lakanen et al. 2014] descrevem experiências de cinco anos oferecendo um curso de verão (25 horas, 5 dias), inserido em um programa de extensão universitária na Finlândia. Ele foi direcionado a estudantes do ensino fundamental e médio (12-18 anos) e focou no desenvolvimento de jogos usando C# e Microsoft XNA. Bellettini et al. [Bellettini et al. 2014] apresentam 4 oficinas interativas (2 horas cada) para estudantes (10-17 anos) na Itália. As oficinas incluíram atividades desplugadas, seguidas de um desafio de programação usando o Scratch.

Sullivan et al. [Sullivan et al. 2015] relatam sobre um clube de programação voltado a meninas do ensino médio (12-17 anos) na Irlanda. Em intervenções de 20 horas, as alunas fizeram atividades de programação e de desenvolvimento do pensamento computacional usando os jogos *on-line* do Blockly, CS-Unplugged, Scratch, Kinect2Scratch e MIT App Inventor. Meyer e Batzner [Meyer and Batzner 2016] descrevem um programa para jovens (9-10 anos) na Alemanha criarem jogos com Scratch, durante uma intervenção de 18 horas (2 horas/semana). Todas essas iniciativas enfatizam a programação como uma atividade divertida e criativa, mas nenhuma integra alguma temática às criações.

O relato mais próximo ao presente trabalho foi o de Ni et al. [Ni et al. 2016], que descreve um acampamento de verão em que alunos do ensino médio nos EUA criaram aplicativos móveis para fins socialmente benéficos. Seus resultados reforçam a ideia de que o foco em temas socialmente relevantes é uma alternativa para motivar os alunos ao mesmo tempo em que aprendem programação. Eles mencionam alguns aplicativos informativos desenvolvidos para abordar questões de comunidades (por exemplo, mercados de agricultores locais ou informações sobre reciclagem), mas não parecem explorar um mesmo tema em vários projetos, como proposto no presente trabalho.

3. Clube de programação

O clube de programação integra um programa de extensão universitária iniciado em 2014, envolvendo professores, funcionários e alunos de graduação em Ciência da Computação, que participam como tutores voluntários. Nesse programa, explora-se a educação não-formal, visando proporcionar oportunidades para estudantes em idade escolar se engajarem em atividades de Computação, com ênfase em utilizar a programação como uma ferramenta criativa. Nos últimos anos, foram beneficiados estudantes de instituições públicas e privadas, de ensino fundamental e médio, localizadas em Santa Maria, RS.

Em experiências anteriores, foram utilizadas diferentes atividades e ferramentas com estudantes de variadas idades, de escolas distintas ou de uma mesma escola. Geralmente, as atividades são realizadas no formato de oficinas que duram 20 horas, distribuídas em encontros semanais de 2 a 4 horas, em laboratórios de informática da universidade ou nas escolas.

As atividades escolhidas envolvem programação baseada em blocos (usando Scratch, Blockly, tutoriais da Hora do Código ou MIT App Inventor) ou programação com linguagens textuais (Python ou Javascript), dependendo do público participante. Em menor escala, já foram utilizados jogos como Lightbot, Code Combat e Code Hunt, e atividades de computação desplugada envolvendo problemas lógicos para treinamento de participantes na Olimpíada Brasileira de Informática (OBI). Nos parágrafos seguintes, apresenta-se nossa experiência mais notável, pois envolveu uma abordagem diferente, temática, que contrastou com as anteriores.

3.1. Escola e estudantes

Em 2016, buscando expandir as atividades do programa para mais escolas, alcançou-se uma escola particular localizada em um bairro socialmente desfavorecido, focada em programas filantrópicos e comunitários no ensino primário e secundário. A maioria de seus estudantes recebe bolsas que cobrem taxas escolares, materiais didáticos e alimentação/uniforme. A escola preocupa-se com a inclusão digital e com questões ambientais, tendo um histórico de atividades extra-curriculares orientadas a *hardware*, por exemplo, desmontando e remontando computadores descartados, e introduzindo robótica com kits de baixo custo. Para orientar essas atividades, a escola emprega alguns professores com formação em informática e eletrônica.

Após apresentar o programa à escola no final de 2016, decidiu-se propor a eles duas atividades complementares durante 2017: treinamentos para a OBI (no primeiro semestre) e oficinas de criação de aplicativos móveis (segundo semestre). Esses assuntos atraíram a atenção dos professores da escola, que ajudaram a divulgar informações sobre o programa e a reunir estudantes interessados em atividades extra-classe.

Assim, formou-se um time de 21 alunos com idades entre 11 e 17 anos, de 6º à 9º ano, sendo que alguns já haviam participado de oficinas de robótica. A maioria das atividades ocorreram em um laboratório da universidade, com custos de transporte compartilhados entre a escola e a universidade, sempre com pelo menos um professor da escola junto aos estudantes.

3.2. Abordagem temática

Antes das oficinas, foi proposto aos professores da escola que trocassem ideias com seus alunos, em busca de um tema para os projetos de aplicativos. Obteve-se retorno de professores envolvidos em um curso chamado “Protagonismo Social”, oferecido a todas as séries, que engloba assuntos relacionados a transformações sociais e comunitárias. No bairro da escola, há problemas ambientais relacionados ao descarte inadequado de lixo, assim, esse tema de grande relevância foi o selecionado para os projetos.

Em seguida, os tutores de graduação foram desafiados a reunir ideias viáveis e envolventes de aplicativos móveis sobre o tema. Essa abordagem contrasta com outras experiências [Ni et al. 2016] em que os estudantes decidiram o que criar. Essa escolha foi motivada por duas razões observadas em experiências anteriores: (i) Os estudantes geralmente não se sentem à vontade para sugerir suas ideias verbalmente, mesmo quando incentivados nessa direção. Acredita-se que é melhor guiá-los no início das atividades, permitindo que se expressem através de suas customizações. (ii) Ter tutores envolvidos em atividades criativas pode impulsionar sua motivação, gerando impacto positivo em suas atitudes de tutoria.

3.3. Atividades desenvolvidas

As oficinas ocorreram em 9 encontros, de duas horas cada, durante quatro meses no segundo semestre. As atividades foram divididas em duas fases: tutoriais introdutórios e projetos temáticos. Em todos os encontros, contou-se com pelo menos 4 tutores de um total de 8 graduandos voluntários em 2017, que auxiliaram os estudantes nas atividades. Os estudantes trabalharam individualmente ou em duplas, sendo que o laboratório da universidade tinha cerca de 30 computadores disponíveis.

3.3.1. Tutoriais introdutórios

No primeiro encontro foi proposta uma introdução aos fundamentos de Computação usando o tutorial da Hora do Código chamado Labirinto Clássico ⁴. Experiências anteriores mostraram que esse primeiro contato com um ambiente de programação baseado em blocos é uma ponte eficaz para o MIT App Inventor. Os estudantes que completaram rapidamente o primeiro tutorial foram incentivados a escolher outros da Hora do Código que utilizassem blocos.

No segundo encontro, apresentou-se o MIT App Inventor explicando suas áreas de edição e propondo a criação de um aplicativo simples, a partir do zero. Esse aplicativo consistiu em uma imagem que, quando clicada, mudava para outra e tocava um áudio.

No terceiro encontro, foi proposta a criação de um aplicativo “Quiz” (jogo de perguntas e respostas), usando variáveis, estruturas de dados (listas), declarações condicionais e funções. Todos esses assuntos foram revisados e detalhados posteriormente nos outros projetos. Como parte dos tutoriais introdutórios, esse aplicativo foi atrativo para os estudantes, que puderam escolher o tema e as questões do seu jogo.

A partir desse encontro, foram propostos projetos base, como mostrado na Figura 1). Com a ajuda dos tutores, esperou-se que os alunos entendessem os blocos já existentes e inferissem quais estavam faltando para terminar o aplicativo.

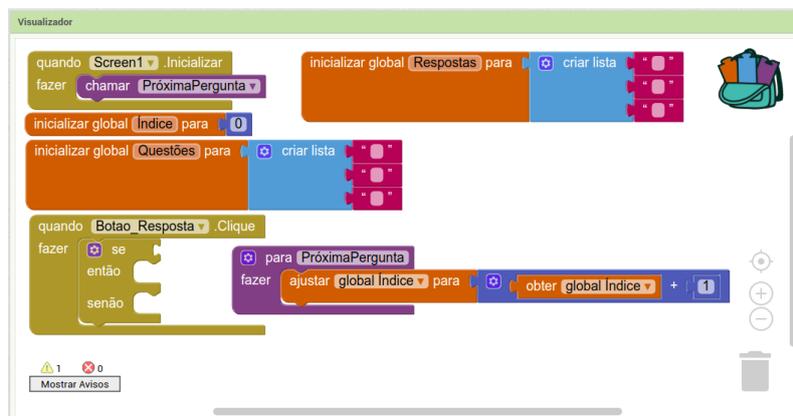


Figura 1. Exemplo da área de desenvolvimento de um projeto base usado

3.3.2. Projetos temáticos

As ideias dos projetos foram desenvolvidas gradualmente. Os tutores utilizaram o tempo entre encontros subsequentes para desenvolver suas ideias no MIT App Inventor antes de as propor nas oficinas, e para discutir sobre observações feitas quanto ao progresso dos estudantes.

Nos encontros temáticos, os tutores primeiramente apresentavam o cronograma do dia. Após, explicavam os conceitos de Ciência da Computação que seriam utilizados e mostravam uma prévia do app proposto, sendo trabalhado um app temático por encontro.

⁴<https://studio.code.org/hoc/1>



(a) Derrote os Monstros (b) Locais de Reciclagem (c) Acerte a Lixeira! (d) Guia de Descarte de Lixo

Figura 2. Tela inicial dos quatro projetos sobre descarte de lixo, executando no emulador do AI2

Em seguida, era fornecido um *link* para *download* do projeto base e de seus arquivos de mídia (imagens e sons).

3.3.3. Recursos de apoio

Além do apoio dos tutores e dos projetos base, foram oferecidos recursos adicionais aos estudantes, para incentivá-los a completarem seus projetos: (i) *Guia do AI2*: mesmo havendo livros abrangentes sobre o MIT App Inventor⁵, não encontrou-se um que fosse adequado à nossa experiência, principalmente por serem em outras línguas. Assim, decidiu-se criar um breve Guia do AI2 em português brasileiro, abordando uma introdução ao ambiente e aos nossos aplicativos introdutórios. As versões impressa e digital do guia estavam disponíveis durante as atividades. (ii) *Cheat Sheets*: foram criados *cheat sheets*, pequenas folhas com dicas para os ajudar com as perguntas mais frequentes: como fazer *login* na plataforma, conceitos básicos de computação e instruções para a conexão com o emulador. (iii) *Posts Motivacionais*: entre encontros, usou-se a página do programa no Facebook para postar fotos de atividades e desafios, premiados com camisetas. Ela também serviu como meio de comunicação com os professores, estudantes e seus pais.

4. Projetos temáticos

Foram desenvolvidos quatro aplicativos (Figura 2) que abordam a consciência ambiental, incentivando o descarte adequado do lixo.

4.1. Derrote os Monstros do Lixo

Esse app é um jogo (Fig. 2a) cujo objetivo é derrotar três monstros do lixo, clicando neles até que seus pontos de vida acabem e eles desapareçam. Ele usa uma tela, que mostra a imagem de um monstro e sua respectiva pontuação, e no final mostra uma imagem de parabéns.

⁵<http://appinventor.mit.edu/explore/books.html>

Neste app são usadas declarações condicionais e funções, porém esses conceitos foram abstraídos ao serem fornecidos prontos no projeto base, para nesta oficina se focar apenas em listas e variáveis. As listas são usadas para armazenar as imagens dos monstros e seus pontos de vida iniciais, e as variáveis para indexar as listas.

4.2. Locais de Reciclagem

Este é um app informativo, que exibe locais de reciclagem em um mapa (Fig. 2b). Para isso, os tutores estudaram alternativas para criar e mostrar o mapa dentro do app. A solução encontrada foi utilizar o My Maps da Google para marcar os locais de reciclagem, e então criar o aplicativo no AI2 para iniciar a aplicação Google Maps, que precisava estar instalada no dispositivo.

Este projeto aborda a composição como um recurso poderoso em programação e desenvolvimento de *software*, conforme proposto por dois tutoriais inspiradores disponíveis no site do MIT App Inventor⁶⁷. Os dois utilizam GPS e o componente "Iniciador de Atividades", que pode iniciar uma atividade do Android, como um navegador para uma página da *web* ou o aplicativo da câmera. Usou-se a possibilidade de abrir o Google Maps para o mapa desenvolvido no My Maps, definido por um URL, ao clique de um botão na tela inicial.

4.3. Acerte a Lixeira!

Esse app foi inspirado em um tutorial do AI2 do jogo Space Invaders⁸. Nesta versão (Fig. 2c), o jogador lança uma bola de papel em direção a uma lixeira, que se movimenta aleatoriamente no topo da tela. A bola de papel sai de um saco de lixo, arrastado pelo jogador na extremidade inferior da tela. Para cada acerto é marcado um ponto e o jogo pode ser reiniciado a qualquer momento.

Este jogo explora o conceito de procedimentos/funções, chamados para mover a bola na tela. Ele também contém condicionais associados a itens na tela, como marcar um ponto quando a bola acerta a lixeira ou definir a nova posição do saco de lixo quando arrastado, servindo como uma prévia para o próximo projeto.

4.4. Guia de Descarte de Lixo

O quarto aplicativo (Fig. 2d) também é um jogo, em que o jogador deve determinar o descarte adequado de lixo para onze imagens de lixo. A tela mostra a imagem atual, a pontuação, um botão de reiniciar e quatro botões representando as opções de lixeiras com suas respectivas cores, como branco para lixo hospitalar e marrom para lixo orgânico.

Este projeto enfatiza o conceito de condicionais, além de rever outros anteriormente abordados. A implementação no AI2 consiste em inserir as imagens de lixo, conhecendo suas respectivas lixeiras, e conferir com uma declaração "quando verdadeiro" se a imagem selecionada para aquela lixeira pertence ou não a sua coleção de imagens.

⁶<http://explore.appinventor.mit.edu/ai2/android-wheres-my-car>

⁷<http://explore.appinventor.mit.edu/displaying-maps>

⁸<http://explore.appinventor.mit.edu/ai2/space-invaders>

5. Discussão

A avaliação da experiência ocorreu principalmente por meio de observações e *feedback* informal dos estudantes e tutores. Nos parágrafos seguintes, resume-se as lições aprendidas, enfatizando aspectos positivos e desafiadores enfrentados.

Aplicativos móveis. Acredita-se que focar em aplicativos para dispositivos móveis foi uma estratégia motivacional bem sucedida. O teste dos apps em dispositivos reais claramente motivou os estudantes, que ficaram animados ao ver suas criações no *smartphone* e ao mostrá-las aos colegas. Essas observações ratificam outras experiências de educação não-formal usando o App Inventor [Roy 2012, Wagner et al. 2013]. Como mencionado antes, os estudantes participaram de uma experiência no primeiro semestre do ano, sobre um assunto diferente mas relacionado (problemas lógicos da OBI). Nessa experiência, teve-se uma taxa de frequência menor. O foco em apps atraiu os estudantes que haviam abandonado o programa e, o mais importante, a maioria deles participou de todos os encontros.

Programação baseada em blocos. A abordagem baseada em blocos confirmou ser uma alternativa acessível para introduzir programação a iniciantes. Os estudantes rapidamente se familiarizaram com a movimentação e montagem dos blocos para alcançar um objetivo. Como era esperado, também tiveram algumas dificuldades, não relacionadas a problemas de lógica. Os estudantes relutaram principalmente com os blocos condicionais, pois eles exigem que o usuário selecione e monte mais "peças" em comparação a outros blocos. Os tutores foram de grande auxílio aos estudantes, evitando que eles passassem muito tempo montando os condicionais corretamente.

Atividades temáticas. A abordagem temática foi eficaz em trazer valor e significado aos apps criados. Após uma experiência, um aluno afirmou que "*Criar aplicativos é um jeito legal de nos lembrar que temos que cuidar do ambiente*". De fato, criar aplicativos em torno de um tema integrativo não requer um esforço adicional dos estudantes e ainda tem a vantagem de fortalecer assuntos que já são recorrentemente trabalhados em sala de aula. Mesmo para uma atividade extra-classe, é benéfico promover alguma colaboração com a escola, o que conseguiu-se quando os professores se envolveram sugerindo um tema. Tutores têm mais trabalho com essa abordagem, mas observou-se que isso foi motivador e os desafiou de maneira positiva, indo além de apenas replicar o que aprenderam em tutoriais do AI2.

Ambiente de programação. Mesmo o MIT App Inventor servindo bem para a proposta, observaram-se algumas desvantagens. (i) Ele requer que os estudantes façam *login* com um e-mail da Google, e a maioria não tinha o costume de usar e-mail. Assim, gastou-se um tempo criando contas no primeiro encontro, sendo que nos encontros seguintes alguns haviam esquecido sua senha. Descobriu-se que é possível hospedar AI2 em um servidor local, para que os usuários não precisem se registrar, porém, não houve tempo e recursos para isso ser efetivado nesta experiência. (ii) O AI2 é baseado em um navegador, mas o emulador de dispositivo é uma aplicação *desktop* e, portanto, requer um tempo para

ser instalado nos computadores. O emulador é um recurso valioso, pois alguns estudantes não têm dispositivos reais ou conexão para testar seus apps, mas pode ser uma fonte de problemas quando eles tentam instalá-lo e iniciá-lo sozinhos. Recomenda-se que esse processo seja orientado pelos tutores no início, assim os estudantes podem se concentrar apenas na criação dos apps.

6. Considerações finais

Neste trabalho, explorou-se uma abordagem temática inserida em um clube de programação extra-classe, oferecido a estudantes de uma escola localizada em um bairro socialmente desfavorecido. O tema foi escolhido colaborativamente, envolvendo estudantes, professores da escola e tutores de graduação.

Alcançou-se o objetivo de envolver os estudantes em atividades de programação ao mesmo tempo em que se enfatizava o assunto do descarte adequado de lixo. Acredita-se que envolver um único tema em vários projetos foi uma abordagem vantajosa, pois valorizou um assunto visto pelos estudantes em sala de aula, chamando sua atenção e dando sentido aos aplicativos criados com blocos de programação.

Essa experiência teve um escopo limitado, e pode não ser facilmente generalizada a outros temas, mas acredita-se que o tema de descarte de lixo poderia ser reutilizado em contextos similares, visto que problemas ambientais são uma preocupação global. Visando incentivar a replicação desta experiência, discutiram-se lições aprendidas que podem ser úteis a outros programas. Também, disponibilizam-se os projetos base e aplicativos completos para *download* em: <https://goo.gl/RG5XLz>.

Agradecimentos

Os autores agradecem à direção, professores e alunos da Escola Marista Santa Marta, pela receptividade e apoio incondicional ao projeto, e a todos os universitários tutores que colaboraram nesta experiência.

Referências

- Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, M., and Zecca, L. (2014). Extracurricular activities for improving the perception of informatics in secondary schools. In *Proceedings of the 7th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, (ISSEP)*, pages 161–172. Springer.
- Dee, H., Cufi, X., Milani, A., Marian, M., Poggioni, V., Aubreton, O., Rabionet, A. R., and Rowlands, T. (2017). Playfully coding: Embedding computer science outreach in schools. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '17*, pages 176–181. ACM.
- Eshach, H. (2007). Bridging in school and out-of-school learning: formal, non-formal, and informal education. *Journal of Science Education and Technology*, 16(2):171–190.
- Giaya, D., Macmillan, C., Price, A., and Roche, M. (2015). Creating engaging computer programs for children in an after-school computer club. Interactive Qualifying Project E-project-121414-190803, Worcester Polytechnic Institute, Melbourne, Australia.

- Gottfried, M. A. and Williams, D. (2013). STEM club participation and STEM schooling outcomes. *Education Policy Analysis Archives*, 21(79).
- Hamadache, A. (1991). Non-formal education: A definition of the concept and some examples. *Prospects*, 21(1).
- Kafai, Y. B. and Resnick, M., editors (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Lawrence Erlbaum Associates.
- Lakanen, A.-J., Isomöttönen, V., and Lappalainen, V. (2014). Five years of game programming outreach: Understanding student differences. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14*, pages 647–652. ACM.
- Lye, S. Y. and Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming. *Comput. Hum. Behav.*, 41(C):51–61.
- Meyer, D. and Batzner, A. (2016). Engaging computer science non-majors by teaching k-12 pupils programming: First experiences with a large-scale voluntary program. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research, Koli Calling '16*, pages 174–175. ACM.
- Ni, L., Schilder, D., Sherman, M., and Martin, F. (2016). Computing with a community focus: Outcomes from an app inventor summer camp for middle school students. *J. Comput. Sci. Coll.*, 31(6):82–89.
- Retnawati, H., Munadi, S., Arlinwibowo, J., Wulandari, N. F., and Sulistyaningsih, E. (2017). Teachers' difficulties in implementing thematic teaching and learning in elementary schools. *The New Educational Review*, 48:201–212.
- Roy, K. (2012). App inventor for android: Report from a summer camp. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, SIGCSE '12*, pages 283–288. ACM.
- Sahin, A. (2013). STEM clubs and science fair competitions: Effects on post-secondary matriculation. *Journal of STEM Education: Innovations and Research*, 14(1):5–11.
- Smith, N., Sutcliffe, C., and Sandvik, L. (2014). Code club: Bringing programming to uk primary schools through scratch. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14*, pages 517–522. ACM.
- Sullivan, K., Byrne, J. R., Bresnihan, N., O'Sullivan, K., and Tangney, B. (2015). Code-plus — designing an after school computing programme for girls. In *Proceedings of the 2015 IEEE Frontiers in Education Conference (FIE), FIE '15*, pages 1–5. IEEE Computer Society.
- Tessier, L. and Tessier, J. (2015). Theme-based courses foster student learning and promote comfort with learning new material. *Journal for Learning through the Arts*, 12(1).
- Wagner, A., Gray, J., Corley, J., and Wolber, D. (2013). Using app inventor in a k-12 summer camp. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13*, pages 621–626. ACM.
- Weintrop, D. and Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. In *Proceedings of the 14th Int. Conf. on Interaction Design and Children, IDC '15*, pages 199–208. ACM.