

# Aplicação de Métodos Lúdicos para o Desenvolvimento e Avaliação da Capacidade de Pensamento Algorítmico em Crianças

Vanessa M. de Oliveira<sup>1</sup>, José R. A. Aranha Junior<sup>1</sup>, Alex O. Barradas Filho<sup>1</sup>

<sup>1</sup>Coordenação do Curso de Engenharia da Computação – Universidade Federal do Maranhão (UFMA)

Av. dos Portugueses, 1966 – 65.080-805 – São Luís – MA – Brasil

{vanessaoliveira2706, robertoaranhajr}@gmail.com,  
alex.barradas@ecp.ufma.br

**Abstract.** *In recent years, technological evolution has led to major changes in contemporary society and demanded more qualified professionals, especially regarding the logic of programming. Thus, some strategies have been directed to premature contact with concepts related to computational thinking. In this context, the present research combines the application of playful tools with the problem-solving teaching and learning methodology to develop and evaluate the ability of algorithmic thinking in children.*

**Resumo.** *Nos últimos anos, a evolução tecnológica tem proporcionado grandes mudanças na sociedade contemporânea e exigido profissionais mais qualificados, principalmente no que se refere à lógica de programação. Dessa forma, algumas estratégias têm sido direcionadas ao contato mais prematuro com os conceitos relacionados ao pensamento computacional. Nesse contexto, a presente pesquisa combina a aplicação de ferramentas lúdicas com a metodologia de ensino e aprendizagem baseada em resolução de problemas para desenvolver e avaliar a capacidade do pensamento algorítmico em crianças.*

## 1. Introdução

A evolução tecnológica a partir dos conceitos computacionais tem se tornado uma realidade na sociedade contemporânea e exigido profissionais mais qualificados, principalmente no que se refere à lógica de programação. Essa demanda tem proporcionado um espaço de debate, entre as instituições de pesquisa e ensino, sobre o desenvolvimento e a difusão da capacidade do pensamento computacional para a população, de tal forma a promover a resolução automatizada de problemas em diferentes áreas do conhecimento [Heintz *et al.* 2016].

Nesse contexto, as estratégias têm sido direcionadas ao contato mais prematuro com os conceitos e a lógica de programação, o que tem proporcionado um ambiente de proliferação para o desenvolvimento de ferramentas e metodologias de ensino de programação a crianças [Williams *et al.* 2015]. Entre o desenvolvimento de tais instrumentos de ensino se destacam as abordagens lúdicas, tais como os jogos sérios e a construção de ambientes de gamificação [Monasor *et al.* 2010].

Os jogos sérios permitem a transmissão do conhecimento de forma lúdica, prática e imersiva aos alunos [Michael e Chen 2005]. A ludicidade é um fator importante quando se trata da necessidade de criar engajamento com o público, ou seja, o divertimento pode ter um papel influente na aquisição de conhecimento pelo estudante e pode ser utilizado para capturar a atenção de crianças para o estudo do pensamento computacional [Giannakos 2013].

Entende-se por pensamento computacional a habilidade para a resolução de problemas, criação de sistemas, além da compreensão do comportamento humano pelo desenvolvimento de ideias fundamentais para a ciência da computação [Wing 2006]. Assim sendo, o pensamento representa a capacidade de analisar e resolver uma variedade de problemas presentes na sociedade contemporânea [Shute *et al.* 2017].

Como o pensamento computacional está centrado na capacidade de abstração para a resolução de problemas de forma que possa ser implementada por computadores [Barr e Stephenson 2011], uma estratégia de aprendizagem ativa que pode ser utilizada, no contexto dos jogos sérios e pensamento computacional, é a aprendizagem baseada em resolução de problemas (PBL, do inglês *Problem Based Learning*).

O método PBL consiste em uma abordagem pedagógica que habilita o aprendizado de estudantes enquanto estão engajados ativamente na resolução de problemas significativos, o que possibilita uma filosofia construtiva, autodirecionada, colaborativa e contextual [Yew e Goh 2016].

No trabalho de Nascimento *et al.* (2018), os autores executam um mapeamento sistemático a respeito do pensamento computacional como uma ferramenta auxiliadora no processo de ensino e aprendizagem de disciplinas da Educação Básica. A reflexão dos autores aponta que há pouca pesquisa envolvendo pensamento computacional de forma interdisciplinar, mas indicam a viabilidade da aplicação dos conceitos do pensamento computacional.

A pesquisa de Gomes *et al.* (2018) relata que os índices de reprovação e evasão nos cursos de computação estão relacionados principalmente às dificuldades de aprendizagem de algoritmos e programação. Como recomendação ao problema de evasão, os autores propõem a utilização de ferramentas visuais que permitem estruturar a solução de maneira macro com o objetivo de diminuir o índice de reprovação, cita-se como exemplo de ferramenta visual o *HelpBlock*.

Na proposta de Lima *et al.* (2018), uma ferramenta denominada de *Edubot* é desenvolvida para trabalhar de forma prática os conceitos de agentes inteligentes. Para tanto, os autores utilizam a abordagem PBL durante a execução dos experimentos nas salas de aula.

Dessa forma, na presente pesquisa, combina-se a aplicação de ferramentas lúdicas com a metodologia de ensino e aprendizagem ativa PBL para o desenvolvimento da capacidade de pensamento algorítmico em crianças, assim como avaliar os resultados obtidos pela resolução de problemas práticos e contextuais.

## 2. Metodologia

As principais etapas da proposta de relacionar jogos sérios (ferramentas lúdicas), pensamento computacional e PBL para avaliar a capacidade do pensamento algorítmico em crianças são ilustradas na Figura 1.

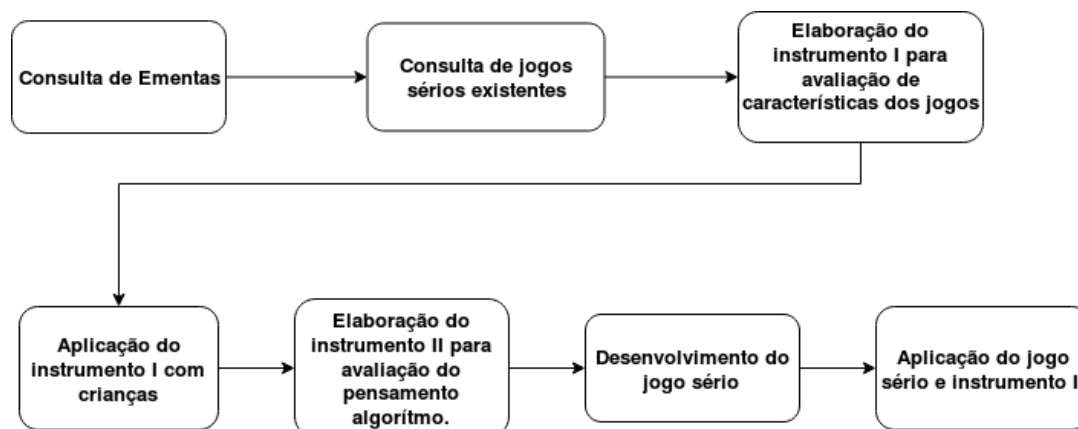


Figura 1. As etapas gerais da proposta

## 2.1. Seleção dos Conteúdos Abordados no Jogo a ser Desenvolvido

Nesta etapa, o primeiro passo foi realizar uma busca acerca do conteúdo abordado nas disciplinas introdutórias aos cursos de computação em universidades e institutos federais pelo Brasil. A busca foi realizada através das ementas onde se pode ter acesso ao conteúdo das disciplinas. No total foi verificado as ementas de 75 cursos, sendo 28 de Ciência da Computação, 24 de Sistema de informação, 18 de Engenharia da Computação e 5 de Engenharia de Software.

Após a busca, um ranking foi elaborado para verificar os tópicos mais frequentes nas disciplinas. Este passo serviu para que fosse encontrado um norte a respeito do conteúdo pertinente para o desenvolvimento do jogo, a Tabela 1 mostra os 5 tópicos mais frequentes.

Tabela 1. Tópicos mais frequentes

Tópicos	Contagens
Conceito de Algoritmos	66
Estrutura de Dados Homogêneas	58
Estruturas de Controle	58
Tipos de Dados	58
Linguagem de Programação	51

## 2.2. Seleção de Jogos Utilizados como Referência

Antes de iniciar o processo de desenvolvimento do jogo, a pesquisa realizou uma extração de características de jogos sérios. Para tanto, uma lista com 19 jogos foi elaborada e classificada conforme a abordagem de programação (bloco ou textual). Posteriormente, selecionou-se apenas um jogo para cada classe. Os critérios para a seleção dos jogos foram três:

- Jogos que continham a maior quantidade dos 5 tópicos encontrados com maior frequência nas ementas;

- Jogos em que o foco fosse estimular o pensamento algorítmico;
- Jogos que possuíssem uma versão demonstrativa que permitisse uma experiência de jogabilidade mais completa.

Os jogos selecionados foram o *CodeCombat* para a programação textual e o *Lightbot* para a programação visual.

### **2.3. Elaboração e Aplicação do Instrumento I de Avaliação de Características dos Jogos**

Para a avaliação das características dos jogos pelos usuários, elaborou-se um formulário com base em duas ferramentas, o MEEGA+KID [Petri *et al.* 2018] e o artigo de Giannakos (2013).

Após a elaboração do instrumento de avaliação, selecionou-se 25 estudantes de 7 a 14 anos de idade da rede municipal de ensino para que pudessem jogar os jogos selecionados e responder ao formulário. Cada estudante jogou por até duas horas em dias separados.

### **2.4. Instrumento II de Avaliação de Habilidade do Pensamento Algorítmico**

Um instrumento de avaliação para acessar a habilidade de pensamento algorítmico foi utilizado a fim de se realizar um teste A/B, onde o grupo de controle foi constituído de alunos do ensino tradicional que realizaram o teste sem o contato prévio do jogo desenvolvido e o grupo de teste foi constituído de alunos que utilizaram o jogo para posteriormente responder o teste.

O instrumento utilizado foi extraído do artigo de Tsukamoto *et al.* (2017), o qual é baseado em três categorias de operações para construção de algoritmos: operações sequenciais; condicionais; e iterativas. De forma mais específica, o teste apresenta três questões discursivas em que cada uma avalia se o aluno entendeu o conceito de cada categoria de operação.

### **2.5. Desenvolvimento do Jogo Sérió**

O jogo sérió desenvolvido foi baseado no clássico *Bombberman*, um jogo 2d que apresenta características similares aos jogos utilizados para a extração de características. O *Bombberman* é um jogo de estratégia em forma de labirinto, com o objetivo de destruir os obstáculos. Nesse contexto, a adaptação realizada no jogo foi permitir que o personagem interaja com o ambiente por comandos programáveis. Ressalta-se também que as fases do jogo foram elaboradas com base no método PBL, ou seja, pela proposta de apresentar desafios (problemas) contextuais ao ensino do pensamento computacional a serem realizados durante o jogo.

A *engine Godot* foi utilizada para o desenvolvimento do jogo, assim como um projeto *open source* que exemplifica a estrutura básica do jogo *Bombberman*. O projeto utilizado como base do desenvolvimento do jogo é licenciado pela licença da MIT, que permite a modificação do software livremente desde que seja feito as devidas citações aos desenvolvedores originais (a propriedade da licença está atribuída a comunidade *Godot*).

## 2.6. Aplicação do Jogo Sérió e do Instrumento II de Avaliação

Na última etapa da pesquisa, ambos os grupos (controle e teste) foram formados com cinco crianças. O primeiro grupo apenas respondeu o instrumento de avaliação, enquanto o segundo grupo jogou por 1 hora e 30 minutos e posteriormente respondeu o instrumento de avaliação. O tempo concedido para a resolução do instrumento de avaliação foi de 25 minutos para ambos os grupos.

## 3. Resultados

### 3.1. Aplicação do Jogo Sérió e do Instrumento I de Avaliação

Os jogos sérios selecionados para a execução da etapa de extração de características foram o *Codecombat* (programação textual) e o *Lightbot* (programação visual). Os resultados da etapa de extração podem ser visualizados na Tabela 2, onde 25 alunos jogaram ambos os jogos e responderam os questionários do Instrumento I de avaliação. O jogo em que houve melhor avaliação para um determinado item do instrumento I foi assinalado com X, enquanto para empates foi assinalado com ---.

**Tabela 2. Comparativo dos resultados do instrumento I**

Questões	Lightbot	CodeCombat
Como eu me senti durante o jogo?	X	
Eu me diverti ao jogar este jogo.		X
Eu prefiro aprender com este jogo do que de outra forma.	X	
Eu estava tão envolvido(a) com o jogo que perdi a noção do tempo.	X	
Houve algo interessante no início do jogo que capturou a minha atenção.	X	
As metas e tarefas do jogo me motivaram a continuar jogando.	X	
Gosto do visual deste jogo.		X
O conteúdo e a organização do jogo me deixaram confiante em aprender.	---	---
Gosto dos elementos do jogo. (personagens, história, figuras, cores)	X	
Consegui entender as regras e instruções do jogo.	X	
Foi fácil entender e utilizar os comandos para jogar.		X

Observa-se que o *Lightbot* teve uma predileção maior na preferência geral dos alunos, pois prevaleceu em 7 dos 12 itens perguntados. Dessa forma, o *Lightbot* foi escolhido como a principal referência para o desenvolvimento do jogo sérió.

No entanto, ressalta-se que o *Codecombat* obteve melhor aceitação para um item fundamental, ou seja, os alunos tiveram como preferência a programação textual. Portanto, o desenvolvimento do jogo terá referência maior o *Lightbot*, mas seguirá a mecânica por programação textual do *Codecombat*.

### 3.2. O Jogo Sérioso Desenvolvido

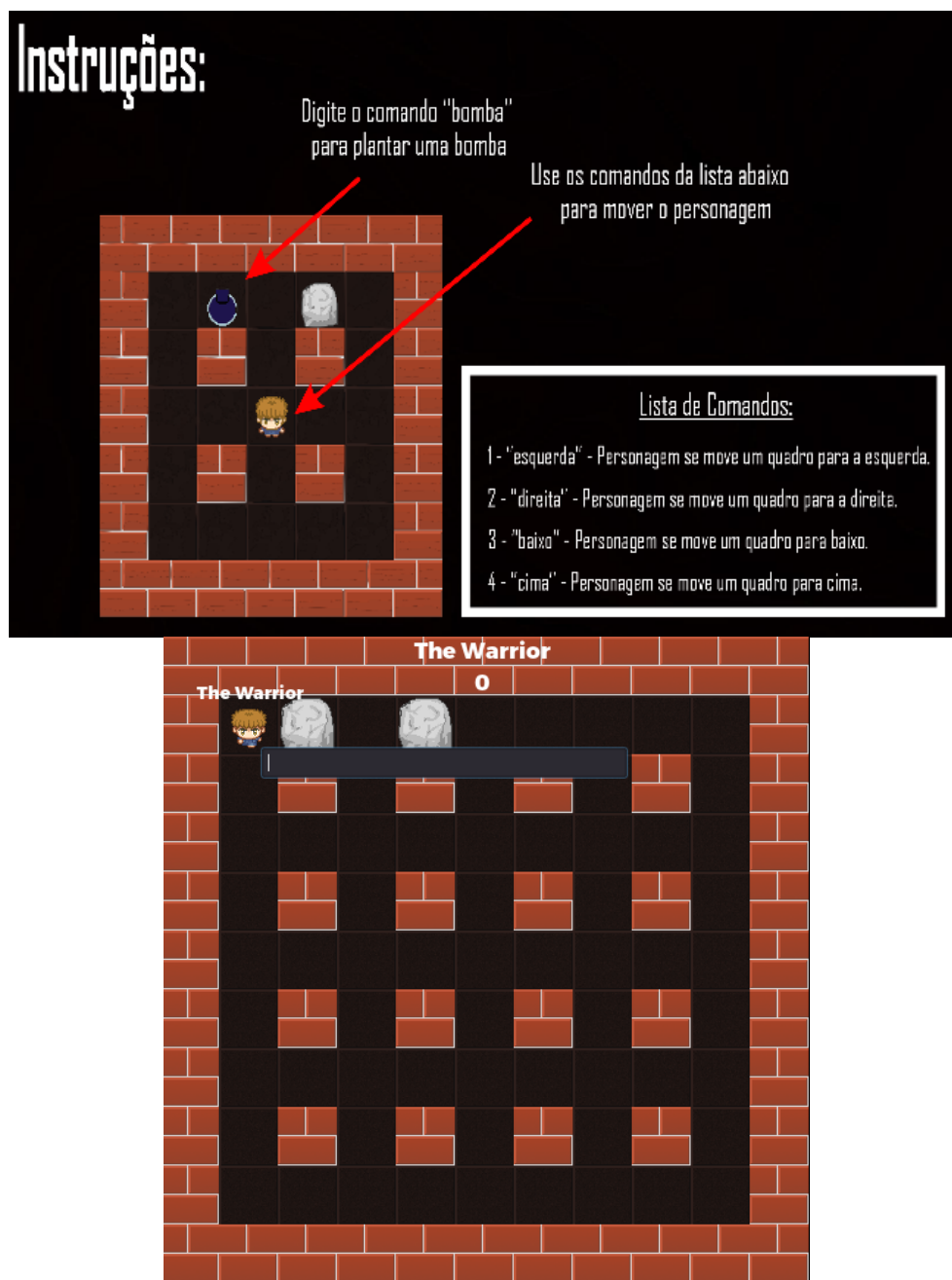
O desenvolvimento do jogo sério foi feito com base em dois direcionamentos: o conteúdo das ementas consultadas e o resultado da extração das características dos jogos sérios mais interessantes aos alunos. De forma geral, o jogo trabalha com a assimilação do conceito de algoritmos, onde assim como no *Lightbot* é necessário guiar o personagem por uma sequência de instruções ordenadas tendo como objetivo remover os obstáculos e passar de fase. Ressalta-se que cada fase se relaciona como umas das três operações essenciais (sequencial, iterativa e condicional).

A Tabela 3 mostra os principais comandos que podem ser utilizados pelos usuários durante a execução de cada fase.

**Tabela 3. Comandos para cada fase**

Fases	Comandos
Fases: 1 a 3	<ol style="list-style-type: none"> <li>1. “esquerda” - move um quadro para esquerda</li> <li>2. “direita” - move um quadro para direita</li> <li>3. “baixo” - move um quadro para baixo</li> <li>4. “cima” - move um quadro para cima</li> </ol>
Fases: 4 a 6	<ol style="list-style-type: none"> <li>1. Comandos anteriores com parâmetros para repetição, “comando(n)” - Move o comando n quadros.</li> </ol>
Fases: 7 a 9	<ol style="list-style-type: none"> <li>1. Se duas pedras consecutivas encontradas: utilizar comando “bomba2”, caso contrário: utilizar comando “bomba”.</li> </ol>

As instruções podem ser visualizadas na Figura 2 que ilustra a tela da fase inicial, as fases posteriores desbloqueiam novos comandos e a dificuldade é aumentada pela inserção de mais obstáculos.



**Figura 2. Tela do jogo sério na fase 1**

*Código disponível em: <<https://gitlab.com/vaannessamota/bombercode>>*

### 3.3. Desempenho no Instrumento II de Avaliação

Os resultados mostrados nas Tabelas 4 e 5 referem ao desempenho dos alunos no instrumento II de avaliação, que possui como objetivo avaliar a capacidade do pensamento algorítmico dos alunos.

**Tabela 4. Resultados do grupo controle**

<b>Grupo B</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>
Questão 1	0	0	3	4	0
Questão 2	1	3	1	5	0
Questão 3	0	3	0	0	0
Total	1	6	4	9	0

Observa-se na Tabela 4 uma dificuldade para obter a nota máxima de cada questão (5 pontos), onde apenas o participante 4 (P4) alcançou uma nota 5 na questão 2. De forma geral, a média do score final (Total) dos participantes foi de 3,4, tendo 0 como a menor nota e 15 como maior nota.

**Tabela 5. Resultados do grupo teste**

<b>Grupo A</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>
Questão 1	5	3	5	5	3
Questão 2	3	3	5	3	0
Questão 3	5	3	3	3	3
Total	13	9	13	11	6

Para o grupo de teste que tiveram o contato com o jogo antes da aplicação do instrumento II de avaliação, a média do score final dos participantes foi de 10,4, uma nota bem superior quando comparado com o desempenho médio do grupo de controle. Ressalta-se que durante a aplicação do jogo os alunos demonstraram maior motivação para a resolução dos desafios. Além disso, acredita-se que a exposição de problemas referente aos tópicos abordados no instrumento II de avaliação permitiu aos alunos exercitar a capacidade de abstração para a resolução de problemas.

No entanto, os participantes de ambos os grupos tiveram dificuldades para responder adequadamente à questão 3. Tal resultado para questões de maior complexidade pode indicar a necessidade de mais contato com o jogo, ou seja, mais prática na resolução de problemas. Em compensação, a dinâmica mostrou-se eficiente para a questão 1, que aborda o conceito de operação sequencial.

#### **4. Conclusão**

A utilização de jogos sérios existentes e metodologias de avaliação de jogos educacionais mostrou-se adequada para obter indicativos do melhor direcionamento para o desenvolvimento de um jogo sério com o objetivo de ludificar o ensino do pensamento computacional.

Além disso, a aplicação do jogo sério desenvolvido durante a pesquisa e do instrumento II de avaliação permitiram observar indicativos da influência positiva da



utilização da ferramenta lúdica nos participantes para o aprimoramento da capacidade do pensamento algorítmico.

Para trabalhos futuros, pretende-se aumentar o tempo de contato dos participantes com o jogo sério, tendo como objetivo aumentar a prática de atividades relacionadas à habilidade do pensamento algorítmico. Para tanto, deve-se implementar no jogo sério um conjunto número maior de desafios e fases.

### **Agradecimentos**

Os autores agradecem à FAPEMA pelo apoio financeiro concedido, ao Laboratório de Análise de Dados e Inteligência Computacional (LADIC) pelos recursos disponibilizados e a todos que ajudaram diretamente ou indiretamente no presente trabalho.

### **References**

- Barr, V; Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?, *ACM Inroads*, 2(1), p. 48-54.
- Giannakos, M. N. (2013). Enjoy and Learn with Educational Games: Examining Factors Affecting Learning Performance, *Computers & Education*, 68, p. 429-439.
- Heintz, F., Mannila, L., Färnqvist, T. (2016). A Review of Models for Introducing Computational Thinking, *Computer Science and Computing in K-12 Education*, In: *IEEE Frontiers in Education Conference (FIE)*, p. 1-9.
- Lima, T. M. S., Menezes, R. F. S., Barradas Filho, A. O., Santos, D. V., Bottentuit Junior, J. B. (2018). Edubot: Um Estudo Prático de Aprendizagem Baseada em Problemas no Contexto de Agentes Inteligentes e Jogos Sérios, *Revista Tecnologias na Educação*, 27, p. 1-15.
- Michael, D. R., Chen, S. L. (2005). *Serious Games: Games That Educate, Train, and Inform*, Cengage Learning PTR.
- Monasor, M. J., Vizcaino, A., Piattini, M., Caballero, I. (2010). Preparing Students and Engineers for Global Software Development: A Systematic Review, In: *5th IEEE International Conference on Global Software Engineering*, p. 177-186.
- Nascimento, C., Santos, D. A., Tanzi, A. (2018). Pensamento Computacional e Interdisciplinaridade na Educação Básica: Um Mapeamento Sistemático, In: *WAlgProg. Anais dos Workshops do Congresso Brasileiro de Informática na Educação (CBIE)*, p. 709-718.
- Petri, G., Von Wangenheim, C. G. (2017). How games for computing education are evaluated? A systematic literature review, *Computers & Education*, 107, p. 68-90.
- Schorr, M., Gomes, E. R., Pretto, F. (2018). Aprendizagem de Algoritmos e Programação por meio da ferramenta visual HelpBlock, In: *WAlgProg. Anais dos Workshops do Congresso Brasileiro de Informática na Educação (CBIE)*, p. 560-568.
- Shute, V. J., Sun, C., Asbell-Clarke, J. (2017). Demystifying Computational Thinking, *Educational Research Review*, 49, p. 142-158.

- Tsukamoto, H., Oomori, Y., Nagumo, H., Takemura, Y., Monden, A., Matsumoto, K. (2017). Evaluating Algorithmic Thinking Ability of Primary Schoolchildren Who Learn Computer Programming, In: IEEE Frontiers in Education Conference (FIE), p. 1-8.
- Williams, C., Alafghani, E., Daley, A., Gregory, K., Rydzewski, M. (2015). Teaching Programming Concepts to Elementary Students, In: IEEE Frontiers in Education Conference (FIE), p. 1-9.
- Wing, J. M. (2006). Computational Thinking, Communications of the ACM, 49(3), p. 33-35.
- Yew, E. H. J., Goh, K. (2016). Problem-based Learning: An Overview of its Process and Impact on Learning, Health Professions Education, 2(2), p. 75-79.