

Capítulo

2

Jogos no Design de Experiências de Aprendizagem de Programação Engajadoras

Tancicleide C. S. Gomes, Patricia C. de A. R. Tedesco e Jeane C. B. de Melo

Abstract

The challenges encountered in the process of teaching and learning programming have been discussed on the global stage, resulting in a number of hypotheses that seek to clarify what are the barriers faced and the reasons why beginners feel unhappy with their experiences introductory programming. Thus, a variety of initiatives involving the creation, application and combination of different pedagogical approaches and tools have been proposed, among which stand out the games. The games favor the creation of more engaging learning experiences, as well as promote the practice and the effort and the error as part of learning, important aspects for beginners in programming. In this perspective, this chapter seeks to show the direction and discuss relevant challenges the integration of games and approaches based games in the design of introductory experiences more engaging programming.

Resumo

Os desafios encontrados no processo de ensino e aprendizagem de programação têm sido discutidos no cenário global, resultando em diversas hipóteses que visam esclarecer quais são as barreiras enfrentadas e por quais razões os iniciantes sentem-se descontentes com suas experiências introdutórias de programação. Assim, uma variedade de iniciativas envolvendo a criação, a aplicação e a combinação de diferentes abordagens pedagógicas e ferramentas foram propostas, dentre as quais destacam-se os jogos. Os jogos favorecem a criação de experiências de aprendizagem mais engajadoras, assim como fomentam a prática e valorizam o esforço e o erro como parte do aprendizado, aspectos importantes para os iniciantes em programação. Nesta perspectiva, este capítulo busca indicar direcionamentos e discutir sobre os desafios pertinentes a integração dos jogos e de abordagens baseadas em jogos no design de experiências introdutórias de programação mais engajadoras.

2.1. Introdução

A integração de conteúdos de Ciência da Computação na educação básica, especialmente programação, tem se destacado por englobar o desenvolvimento de habilidades e competências consideradas necessárias aos profissionais do século XXI [CAS 2012, IFTF 2009, McGettrick 2005, Stephenson *et al.* 2012, Wing 2006, Wing 2008]. Os adeptos do ensino das denominadas linguagens digitais argumentam que a prática aprimora o raciocínio lógico e a criatividade, no entanto esta não é uma iniciativa recente [Goulart 2016].

Papert defendia o ensino de conceitos de programação para crianças e adolescentes desde o fim da década de 60 [Logo Foundation 2015, Papert 1979]. A partir dos esforços de pesquisa de sua equipe surgiu a primeira versão da linguagem Logo e, desde então, têm surgido propostas para experiências introdutórias de ensino de programação que englobam linguagens visuais de programação, robótica pedagógica, jogos digitais e interfaces tangíveis. Além disso, uma série de ações têm sido idealizadas no intuito de democratizar o acesso e desmitificar o aprendizado de programação.

No entanto, o ensino de programação na educação superior assim como na educação básica compartilham o mesmo desafio: poucos iniciantes parecem considerar o aprendizado de programação uma atividade fácil e prazerosa e projetar um ambiente de ensino e aprendizagem que seja envolvente não é uma tarefa trivial [Huggard 2004, McGettrick 2005, Pereira Júnior e Rapkiewicz 2004, Scott e Ghinea 2013]. As dificuldades relacionadas ao processo de ensino e aprendizagem de programação vêm sendo amplamente discutidas na literatura, de tal modo que a *British Computer Society* declarou programação como um grande desafio para a pesquisa em educação [McGettrick 2005].

Neste sentido, muitos estudos têm tentado compreender quais as barreiras enfrentadas pelos iniciantes e buscam estabelecer as razões pelas quais os alunos sentem-se descontentes com suas experiências de introdutórias de programação. Há propostas que abrangem diferentes possibilidades para auxiliar nas experiências introdutórias de ensino de programação, perpassando por novos métodos, técnicas, ferramentas, ou mesmo a combinação destes.

Segundo Pereira Júnior e Rapkiewicz (2004), a origem dos problemas associados ao processo de ensino e aprendizagem de programação é considerada complexa visto os diversos elementos envolvidos: os perfis distintos de sujeitos (professores e alunos), as práticas didático-pedagógicas e mesmo o grau de dificuldade considerado intrínseco à área. Outros pesquisadores, por sua vez, argumentam que uma das maiores dificuldades está relacionada à combinação e à utilização apropriada dos conceitos básicos de programação para a construção de um programa, pois ao que parece os alunos entendem os conceitos e estruturas que compõem uma linguagem de programação, mas não sabem como utilizá-las corretamente durante a construção de seus próprios programas [Caspersen e Kölling 2009, Lahtinen, Ala-Mutka e Järvinen 2005 *apud* Aureliano e Tedesco 2012].

Huggard (2004) afirma que os estudos psicológicos envolvendo o aprendizado de programação são realizados há mais de 40 anos, mas aparentemente não têm sido incorporados às práticas de ensino, de modo que os alunos continuam a ter pouca ou

nenhuma confiança em suas próprias habilidades. As experiências negativas são frequentemente associadas com dificuldades de aprendizado e poucos alunos parecem ser capazes de superá-las, porém não há ainda um consenso sobre as dificuldades intrínsecas ao aprendizado de programação, de modo que alguns estudiosos argumentam que há apenas a falta de aptidão [Jenkins 2002]. No entanto, os resultados de trabalhos visando mensurar a aptidão dos estudantes têm se mostrado inconclusivos: enquanto Byrnes e Lyons (2001) *apud* Huggard (2004) argumentam que existe uma relação entre habilidades de programação e de matemática, um estudo conduzido pela Universidade de Leeds concluiu que não havia qualquer relação entre o desempenho obtido em programação e a aptidão mensurada [Davy e Jenkins 1999 *apud* Huggard 2004].

Dweck (2002), entretanto, sugere que as barreiras enfrentadas pelos iniciantes podem estar relacionadas às suas próprias crenças implícitas e propõe classificá-los em duas categorias para ilustrar as influências em potencial dessas mentalidades: *entity-theorist* e *incremental-theorist*. Os *entity-theorists* acreditam que suas aptidões são características naturais fixas, ao passo que os *incremental-theorists* acreditam que suas aptidões são uma habilidade flexível e que pode ser aprimorada através do esforço.

Estes dois grupos, segundo Dweck (2002), demonstram comportamentos distintos ao se depararem com dificuldades no processo de aprendizado. Por exemplo, ao passo que os *entity-theorists* consideram o erro como um indicador de uma baixa aptidão em programação e a sua performance diminui após encontrarem dificuldades, os *incremental-theorists* consideram o erro como um indicativo de falta de esforço, má estratégia ou falta de conhecimentos prévios necessários e a sua performance se mantém ou melhora mesmo após se depararem com alguma dificuldade.

A classificação proposta por Dweck oferece indícios sobre alguns dos obstáculos relacionados ao aprendizado de programação por iniciantes que vão além das discussões sobre quais as linguagens ou paradigmas de programação mais adequados e enaltecem a importância do esforço, da prática e do próprio erro para uma melhor consolidação do conhecimento endossando o discurso de Winslow (1996), de que a chave para alcançar a proficiência em programação é a prática intensa.

Neste contexto, surgem os desafios enfrentados pelos educadores ao projetarem experiências de ensino de programação: “*como encorajar os estudantes a praticar programação e como estabelecer métodos que avaliem de maneira efetiva as habilidades individuais de programação?*” [Gove 2012, Huggard 2004]. Outro desafio é manter os estudantes motivados, sentindo prazer intrínseco ao longo de sua aprendizagem e superando suas próprias dificuldades [Dweck 2002, Huggard 2004, Pereira Júnior e Rapkiewicz 2004, Scott e Ghinea 2013].

Aparentemente, a questão chave do processo é como motivar o aluno e mantê-lo engajado buscando superar suas dificuldades de aprendizagem [Baeza-Yates 1995, Henderson 1987, Koliver, Dorneles e Casa 2004, Winslow 1996 *apud* Pereira Júnior e Rapkiewicz 2004]. Considerando, então, a prática como elemento chave para a aquisição de habilidades e competências em programação e as categorias de mentalidades propostas por Dweck, podemos formular o seguinte questionamento: *Como os educadores podem criar ambientes de aprendizagem mais engajadores cuja pedagogia fomente uma mentalidade incremental, encorajando a prática, valorizando o esforço e incorporando o erro como parte do processo de aprendizado?*

Há vários estudos que contemplam a apropriação ou o desenvolvimento de artefatos educacionais, e a proposição de métodos e práticas pedagógicas visando prover um ambiente de aprendizado mais efetivo [Scott e Ghinea 2013]. Dentre as possibilidades propostas, a incorporação de jogos digitais desponta como uma das abordagens mais utilizadas e apresenta resultados significativos: “97% dos estudos relatam que a utilização de jogos é uma ferramenta eficaz de ensino e aprendizagem de programação” [Silva et al. 2014].

Os jogos fazem parte do fenômeno denominado *game thinking*, o qual consiste em usar a diversão e princípios de jogos a fim de projetar soluções para problemas do mundo real [GauravMishra 2010]. De modo geral, os jogos têm características que podem auxiliar os educadores a criar ambientes de ensino mais engajadores. O encorajamento à prática, a valorização do esforço e a incorporação do erro ao aprendizado se combinam em uma experiência de aprendizagem imersiva. A experimentação é um dos pontos altos do ato de jogar em que, notoriamente, o erro exerce um papel fundamental ao longo deste processo e as consequências são destituídas de seu caráter punitivo e a prática reflexiva passa a compor um esforço despendido para atingir o objetivo final.

Os jogos digitais, em especial, oferecem ainda *feedback* customizado, indicando as consequências das ações/decisões e auxiliando na compreensão das regras que regem o jogo, permitindo o reajuste das estratégias utilizadas. Outro aspecto relevante é que os jogos oferecem curvas de progresso que são planejadas visando desafios graduais às habilidades intelectuais e motoras dos jogadores.

Os jogos e artefatos lúdicos, disponíveis para diferentes plataformas com temáticas e abordagens pedagógicas variadas, possibilitam uma experiência de aprendizagem mais engajadora. No entanto, o simples acréscimo de jogos ao ambiente de aprendizado, não pressupõe melhorias no processo educacional e faz-se necessário compreender os benefícios e limitações oferecidas por esses recursos a fim de um melhor aproveitamento nas práticas pedagógicas. Assim, o presente capítulo discorre sobre o pensamento de jogo (*‘game thinking’*) enquanto um fenômeno mais amplo, englobando diversas manifestações dos jogos: jogos sérios, brinquedos, jogos diversos, estratégias de gamificação e *design* orientado a jogo, bem como a sua incorporação (com ênfase para a utilização de jogos digitais) em contextos educacionais introdutórios de programação.

Este trabalho encontra-se organizado da seguinte forma: na Seção 2 é introduzida uma breve discussão sobre o pensamento de jogo, suas definições e abordagens relacionadas. As implicações e os benefícios relacionados à incorporação dos jogos digitais nos cenários educacionais são discutidos na Seção 3. Um conjunto de recursos lúdicos úteis para experiências de aprendizagem de programação categorizados conforme a linha estabelecida pelo pensamento de jogo e sugestões para a integração em sala de aula são apresentados na Seção 4. As considerações finais são apresentadas na Seção 5.

2.2. O Pensamento de Jogo: Definições e Discussões

Os jogos, em sua essência, são elementos inerentes à cultura humana, de modo que o lúdico e o brincar estão presentes desde a mais tenra idade. Huizinga (2004) descreve os jogos como uma atividade naturalmente livre e ficcional, um momento de evasão do

mundo real e que carrega consigo um quê de inventivo. Algo que se distingue da vida real por meio dos limites espaciais e temporários estabelecidos e que proporcionam ao jogador um estado de imersão, preparando-o para a realização de tarefas ao longo da vida.

Uma outra definição apresenta os jogos como um sistema formal baseado em regras, que possui um resultado variável e quantificável, em que o jogador, respeitando as regras estabelecidas, se empenha em influenciar o resultado final [Juul 2003]. Salen e Zimmerman (2012), por sua vez, definem os jogos como um sistema no qual os jogadores se envolvem em um conflito artificial, definido por regras, que implica em um resultado passível de ser quantificado.

Dentre as conceituações apresentadas é possível identificar alguns elementos comuns que constituem o que considera-se como o cerne da definição do que são jogos [Alves 2014, Ranhel 2009, Sales e Zimmerman 2012, Wangenheim e Wangenheim 2012], tais como:

- Regras, que exercem um papel de fundamental importância, delimitando a sua estrutura e funcionamento e estabelecendo os meios de como alcançar os resultados esperados;
- Interação Social, que desperta à medida em que os jogadores se relacionam, disputam e/ou cooperam entre si em busca da resolução do problema proposto pelo jogo;
- Desafio, um problema a ser solucionado ou mesmo um conflito a ser vencido;
- Recompensas, os indicadores de que os objetivos propostos foram alcançados com sucesso, certificando aos jogadores de que a vitória foi atingida.

Segundo Mishra (2010), os jogos são parte de um fenômeno mais amplo, denominado pensamento de jogo, que consiste em usar a diversão e princípios de jogos para projetar soluções para problemas do mundo real. Marczewski (2016) descreve pensamento de jogo como um conceito mais amplo, delineado a partir de quatro categorias: (i) gamificação, (ii) jogos sérios, (iii) design orientado a jogos e (iv) jogos/brinquedos, que compõem a matriz apresentada na Figura 2.1.

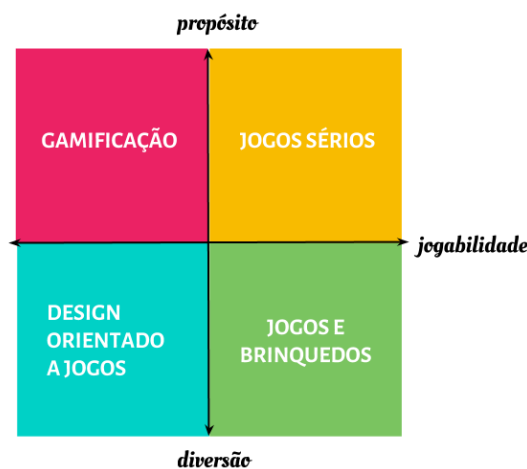


Figura 2.1. Matriz das abordagens do pensamento de jogo traduzida e livremente adaptada de AndrzejMarczewski (2016)

Os eixos delineiam a matriz considerando o objetivo primário de *design* (vide a Tabela 2.1). O eixo horizontal é orientado a jogabilidade, que designa o quão um jogo proporciona uma experiência fluida e divertida aos jogadores, uma experiência de interação lúdica e significativa, sem desestimulá-los com um alto grau de dificuldade, tampouco aborrecê-los com uma experiência tediosa. O eixo vertical indica o objetivo, variando entre pura diversão e um propósito específico. Estes eixos descrevem como estes três aspectos (propósito, diversão e jogabilidade) se relacionam e como estão presentes dentro destas abordagens de pensamento de jogo.

Tabela 2.1. Jogos e experiências baseadas em jogos segmentados por objetivos de *design*

	Pensamento de jogo	Elementos de jogos	Jogabilidade	Propósito
<i>Design</i> orientado a jogos	X			
Gamificação	X	X		
Jogos	X	X	X	
Jogos sérios	X	X	X	X

Nesta matriz (Figura 2.1), os quadrantes inferiores, *design* orientado a jogos e jogos/brinquedos, abrangem manifestações do pensamento de jogo que têm como objetivo primário de *design* a ênfase na diversão sem objetivos específicos secundários. Os quadrantes superiores, gamificação e jogos sérios, englobam abordagens que buscam um alto nível de engajamento e motivação, tendo como principal característica a existência de um propósito em seu cerne. O diagrama a seguir apresenta os principais desdobramentos das abordagens do pensamento de jogo (Figura 2.2).



Figura 2.2. Diagrama dos possíveis desdobramentos das categorias do pensamento de jogo traduzido e livremente adaptado de AndrzejMarczewski (2016)

2.2.1. Design Orientado a Jogos

O *design* orientado a jogos (do termo original em inglês, *game design inspired*, ou ainda *gameful* ou *playful design*) consiste em incorporar a diversão inerente aos jogos, além dos elementos de jogos propriamente ditos, para tornar a experiência do usuário mais agradável [Marczewski 2013].

Neste quadrante estão englobadas abordagens inspiradas em jogos e que utilizam ideias destes, tais como narrativas e elementos visuais (estéticos), para aperfeiçoar a experiência vivenciada pelos indivíduos, priorizando a diversão sem enfatizar mecânicas ou jogabilidade. Esta abordagem é comumente integrada ao desenvolvimento de produtos e serviços (Figura 2.3).



Figura 2.3. Exemplos de produtos desenvolvidos sob a perspectiva do *design* orientado a jogos

No cenário educacional, por sua vez, um bom exemplo são as adaptações de narrativas interativas (ou livros-jogo) que permitem ao leitor participar da história, proporcionando um maior envolvimento. As narrativas interativas convencionais comumente incluem regras para a construção de um personagem fictício, incluindo magias, técnicas de combate e a escolha de itens que o personagem possuirá no

inventário apresentados no início, de modo que o leitor é colocado como o personagem principal da história¹.

A narrativa se desdobra em caminhos distintos, de modo que o leitor pode experimentar a mesma história sob diferentes perspectivas, seja considerando escolhas de personagens ou mesmo, de acordo com as respostas às perguntas realizadas ao longo da narrativa, normalmente indicadas por páginas ou parágrafos numerados, como nos exemplos a seguir:

- Se quiser seguir o velho sábio, vá para a página 12;
- Se preferir retornar para a entrada da caverna, vá para a página 08;
- Se desejar entrar no labirinto sagrado, vá para a página 13.

De um modo geral, as adaptações para cenários educacionais, incorporam os conteúdos ao longo da narrativa de modo que as escolhas realizadas pelo leitor levem-no a refletir sobre a sua compreensão de um determinado conceito e respostas erradas levem o personagem perder itens, sofrer danos ou mesmo tornem a sua trajetória mais longa ou mais penosa, ou finais alternativos inesperados.

2.2.2. Jogos/ Brinquedos

Segundo Huizinga (2004), as características primordiais dos jogos residem em seu poder de fascinação e em sua capacidade de excitar, e obviamente na diversão que estes promovem. Naturalmente, os jogos são elementos constituintes do universo do entretenimento assim como a televisão, o cinema e a literatura, por exemplo [Petry 2016]. Os jogos essencialmente têm como principal objetivo promover a diversão, de modo que a interação é livre e não há a imposição de objetivos extrínsecos. Assim, o entretenimento puro e simples é um dos principais intuitos dos artefatos deste quadrante, que abrange jogos de cartas, jogos de tabuleiro, jogos eletrônicos e jogos digitais diversos, dentre outros.

Os jogos podem ainda agregar conteúdos e elementos linguísticos de outras formas de arte e comunicação e transformá-los de acordo com as necessidades de seu próprio meio [Petry 2016], alcançando novos espaços e recebendo novas ressignificações. Em 2012, a exposição *The Art of Videogames*², sob a curadoria de Chris Melissinos, explora os videogames sob uma nova perspectiva: enquanto formas de arte. No ano de 2014, em meio as galerias de Arquitetura e Design do Museu de Arte Moderna (MoMa) de Nova Iorque, jogos como *Pacman*, *Tetris*, *SimCity*, *Misty* e mais outros dez jogos tomaram espaço entre a exposição *Applied Design*,³ organizada por Paola Antonelli e Kate Carmody, inaugurando uma nova categoria de trabalhos de arte do MoMa⁴.

Existe uma profícua discussão sobre o fato dos videogames serem ou não considerados obras de arte. Melissinos [2015] argumenta que os *games* são uma das

¹<https://pt.wikipedia.org/wiki/Livro-jogo>

²<http://americanart.si.edu/exhibitions/archive/2012/games/>

³<http://www.moma.org/calendar/exhibitions/1328?locale=pt>

⁴https://www.moma.org/explore/inside_out/2012/11/29/video-games-14-in-the-collection-for-starters

mais importantes formas de arte da história por serem a única forma de mídia que proporciona a personalização da experiência artística, mantendo a autoridade do artista:

[...] Em jogos de videogames encontramos três vozes distintas: o criador, o jogo e o jogador. Aqueles que jogam um jogo estão seguindo a narrativa do autor e estão vinculados pelas construções das regras, mas com base nas escolhas que fazem, a experiência pode ser completamente pessoal. Se você pode observar o trabalho de outro e encontrar nele conexão pessoal, então a arte foi alcançada.

Neste sentido, surgem jogos como *Braid*⁵, *Fez*⁶, *Journey*⁷, *No Man's Sky*⁸, *Rainy Day*⁹, dentre diversos outros. *Braid* é um jogo eletrônico multiplataforma (Xbox 360, Windows, OS X, PlayStation 3, Linux) lançado em 2008. Neste jogo são exploradas algumas das questões mais inquietantes para os seres humanos: as capacidades de mudar o passado e de conhecer o futuro. As relações com o tempo são exploradas a partir de jogabilidades diferenciadas como a oportunidade de explorar o mundo à sua volta apenas com a sombra do personagem ou a possibilidade de voltar no tempo para impedir de ser morto por um inimigo, reagindo de maneira diferente.

Journey é um jogo digital desenvolvido exclusivamente para os consoles Sony® (PlayStation3 e PlayStation4) lançado em 2012, que alcançou vários prêmios'. *Journey* é descrito por seus desenvolvedores como “uma parábola interativa, uma aventura online anônima para experimentar a passagem da vida de uma pessoa e as suas intersecções com as vidas de outras pessoas[...]”.



Figura 2.4. Screenshot *Journey*

Em uma apologia à jornada da vida, com seus altos e baixos, felicidades e tristezas, nascimentos e mortes, amizades e solidão, um protagonista sem nome e sem expressão facial desperta em uma região desértica e desolada, vislumbrando uma montanha distante realçada por um ponto de luz brilhante no cume (Figura 2.4). A

⁵ <http://braid-game.com/>

⁶ <http://fezgame.com/>

⁷ <http://thatgamecompany.com/games/journey/>

⁸ <http://www.no-mans-sky.com/>

⁹ <https://thaisa.itch.io/rainy-day>

jornada se desenrola em meio a ruínas, tempestades de areia, vendavais de neve e cavernas sem vida, com quebra-cabeças simples e segredos que são lentamente revelados ao jogador [Teixeira 2015]. Além disso, a inovação no sistema cooperativo de jogo, que não fornece qualquer informação a respeito do parceiro (apenas no fim do jogo), é uma das características mais marcantes de *Journey*.

2.2.3. Gamificação

Deterding *et al.* (2011) definem gamificação como “*o uso de elementos (em contraste ao uso de jogos propriamente ditos) de design (ao invés de tecnologias baseadas em jogos) de jogos (ao invés de apenas atividades lúdicas ou brincadeiras) em contextos de não-jogo*”. Existem vários elementos típicos de *design* de jogos que podem ser incorporados em abordagens de gamificação tais como: *avatares*, recursos colecionáveis, recompensas, níveis (progressão), desbloqueio de conteúdos, combates, entre outros.

No entanto, comumente, a clássica tríade de elementos de jogos utilizada em abordagens de gamificação são pontos, emblemas e quadros de pontuação, do acrônimo PBL (oriundo dos termos em inglês: *points, badges e leaderboards*). Os pontos são os elementos mais simples e conhecidos, constituindo uma oportunidade de oferecer *feedback* ao jogador sobre o seu desempenho e apresentando-se, na maioria das vezes, conectados às recompensas (por exemplo, é necessária uma quantidade específica de pontos para alcançar um nível bônus).

Os emblemas, por sua vez, são elementos colecionáveis e são intrinsecamente ligadas ao *status* do jogador, atuando como credenciais indicando que o jogador desenvolveu determinadas habilidades e/ou alcançou determinadas conquistas, atribuindo distinção aos jogadores que os detém. Os quadros de pontuação comumente são projetados a partir do *ranking* das pontuações dos jogadores e oferecem um *feedback* geral da competição. Algumas variações, em especial, oferecem a possibilidade de visualizar uma comparação entre a pontuação dos jogadores e seus amigos.

No entanto, embora estes elementos sejam amplamente utilizados, a gamificação não se reduz à essa perspectiva. Segundo Schlemmer e Lopes (2016), estes componentes tornam mais fácil, escalável e de baixo custo estruturar abordagens de gamificação. Contudo, o uso dos elementos PBL não é capaz de tornar algo considerado entediante em algo motivador, pois, falham, sobretudo, no que se refere ao engajamento dos sujeitos envolvidos. Chou (2015) *apud* Schlemmer e Lopes (2016) argumentam que a perspectiva PBL, embora amplamente difundida, é uma visão reducionista da gamificação que pode ocasionar em uma utilização superficial do conceito e em abordagens gamificadas com baixo poder de inovação, e sobretudo, de baixo poder de engajamento e diversão.

O uso de PBLs de maneira descontextualizada pode ser problemática por concentrar-se em uma camada motivação extrínseca que pode funcionar de maneira adequada para os alunos mais jovens, porém não observa-se o mesmo para os alunos mais velhos, sobretudo quando consideram-se experiências de longo prazo. Veen e Vracking (2009) indicam que quando os jogadores são questionados sobre o porquê de gostarem de jogar, estes não se referem aos elementos PBL, mas, sim, aos desafios, às missões, às estratégias. Deste modo, em uma estratégia de gamificação é necessário compreender o problema a ser resolvido, o contexto, o ambiente e os sujeitos

envolvidos a fim de estruturar desafios e missões condizentes com os objetivos pessoais dos ‘jogadores’ [Schelmmmer e Lopes 2016].

A gamificação na educação se propõe, então, a utilizar estilos, dinâmicas e pensamento de jogo em contextos educacionais como meio para a resolução de problemas e engajamento dos sujeitos da aprendizagem.

2.2.4. Jogos Sérios

Segundo Alvarez e Michaud (2008) há uma variedade de formas de classificar os jogos sérios, mas estes podem ser definidos da seguinte forma: o objetivo de um jogo sério é proporcionar que usuários interajam com aplicações de tecnologia da informação que combinem aspectos de tutoria, ensino, formação, comunicação e informação, com alguns elementos de jogos. Esta combinação tem como propósito tornar o conteúdo prático, útil (sério) e jogável, o que é alcançado por meio do desenvolvimento de cenários que são ao mesmo tempo práticos e agradáveis.

Michael e Chen (2005) definem os jogos sérios como jogos em que o entretenimento, a diversão não são os objetivos primários de *design*, ao invés disto, promovem a aprendizagem e mudanças de comportamento em várias áreas, tais como negócios, indústrias, *marketing*, educação, assistência médica e educação/ treinamento.

“*Where in the World Is Carmen Sandiego?*” é um exemplo clássico de jogo sério voltado para educação lançado na década de 80. O objetivo do jogo consistia em capturar Carmen Sandiego, uma das maiores ladras do mundo e seus aliados. O jogo se inicia indo para o primeiro país onde o crime ocorreu e em seguida são fornecidas diversas dicas de várias fontes próximas de onde o ladrão esteve, levando à uma perseguição em todo mundo para encontrar para onde os ladrões foram antes que o tempo se esgote. Em meio à narrativa, o jogador é um detetive que junta diversas pistas ao redor do mundo para solucionar os casos, para tanto é necessário aprender sobre a história, geografia e cultura de diversos países¹⁰. No intuito de tornar o aprendizado de conceitos de algoritmos e estruturas de dados mais lúdico,

Na área de saúde, os jogos também podem ser utilizados no tratamento de disfunções¹¹. O projeto NeuroGames busca utilizar os jogos para diminuir os efeitos do Transtorno de Déficit de Atenção e Hiperatividade (TDAH). No jogo Caçador de Dragões, o jogador é perseguido por um dragão que possui uma aura protetora ao seu redor que se enfraquece em um intervalo de tempo que pode variar entre 2 e 20 segundos. O objetivo é o jogador (paciente) seja capaz de resistir à ansiedade causada pelo TDAH e possa controlar os seus próprios impulsos e realizar o ataque no momento adequado.

No intuito de tornar o aprendizado de conceitos de algoritmos e estruturas de dados mais lúdico, Wassila e Tahar (2012) desenvolveram um jogo sério inspirado no clássico jogo Super Mario (Figura 2.5). O jogo mantém alguns elementos estéticos (cenários, personagem) e mecânicas do jogo original e abrange, basicamente conceitos relacionados às árvores AVL, distribuídos da seguinte forma: a primeira fase é uma revisão dos conceitos de árvores de busca binária. A segunda fase é um exercício sobre os diferentes tipos de rotação presentes em árvores do tipo AVL. O terceiro nível busca

¹⁰https://pt.wikipedia.org/wiki/Where_in_the_World_Is_Carmen_Sandiego%3F

¹¹<http://link.estadao.com.br/noticias/geral,psicologos-usam-game-contra-deficit-de-atencao,10000029466>

ensinar ao jogador métodos de buscar um elemento, o quarto estágio demonstra vários casos de inserção de elementos e finalmente, o último nível trata sobre a exclusão de um elemento.



Figura 2.5. Jogo inspirado em Super Mario Bros para ensinar AVL [Wassila e Tahar 2012] - Nível 03

Os jogos também são desenvolvidos visando a promoção de marcas, produtos e serviços. Estes jogos, em especial, são denominados advergames e também estão incluídos neste quadrante. Convém mencionar embora estes jogos tenham como propósito explícito os fins publicitários, suas estratégias de jogabilidade não estabelecem uma relação direta com os conteúdos promovidos.

Segundo Obringer¹², existem três categorias destes jogos. A primeira categoria agrupa os jogos que são inseridos nos sites de uma companhia (ou ainda em um site que seja de propriedade ou patrocinado pela empresa, especialmente desenvolvido para o jogo) visando atrair os visitantes web ou mantê-los no site por mais, ou ainda permitir que os clientes experimentem algumas das funcionalidades do produto, embora os jogos não necessariamente devam estar ligados a algum produto em específico.

A segunda categoria se aproxima mais dos jogos tradicionais, podendo ser jogos de videogames comerciais desenvolvidos e vendidos para serem jogados em computadores ou consoles de games, mas essencialmente a diferença é que os games são criados com um objetivo específico de promover um produto, serviço ou instituição. O Exército dos Estados Unidos patrocinou um jogo de sucesso chamado 'America'sArmy' em um esforço visando fomentar o recrutamento.

A terceira categoria envolvem os anúncios inseridos dentro dos próprios jogos, em que um produto ou um anúncio para um produto fazem parte do jogo, como por exemplo um 'Ford Mustang' cruzando ruas virtuais de um jogo ou outdoors em meio ao jogo fazendo propaganda de produtos do mundo real.

Os jogos sérios são amplamente utilizados para resolver problemas em diversas áreas e envolvem desafios e recompensas usando o entretenimento e os componentes de engajamento comumente obtidos quando o usuário está jogando jogos, tendo como objetivo principal o desenvolvimento de habilidades e competências específicas. Os

¹² <http://money.howstuffworks.com/advergaming1.htm>

jogos também podem ser utilizados como um meio de transmitir ou reforçar mensagens ou hábitos, promover a mudança de comportamento.

Um exemplo de jogo utilizado para resolver problemas na área de saúde é o jogo Foldit que ganhou notoriedade em 2011, desenvolvido por pesquisadores da Universidade de Washington. O jogo tinha como objetivo montar estruturas de proteínas em modelos tridimensionais, seguindo as regras que existem na natureza, tais como componentes que são sensíveis a água não ficarem expostos ao meio, enquanto outros componentes obrigatoriamente têm que estar. O esforço coletivo de aproximadamente 46 mil participantes ao longo de 10 dias resultou em uma descoberta que persistia em aberto ao longo de 15 anos¹³.

A incorporação do pensamento de jogo na criação de experiências de aprendizagem pode ser realizada de diversas formas: na integração de estratégias de gamificação, no uso de jogos comerciais ou mesmo jogos educacionais, na utilização de ambientes virtuais e de simulação, ou, mesmo, no desenvolvimento de jogos. A escolha da abordagem de pensamento de jogo mais adequada ao contexto educativo depende do balanceamento entre o propósito, a jogabilidade e a diversão.

2.3. Jogos Digitais nos Contextos Educacionais

A afirmação atribuída ao filósofo grego Platão, de que todo aprendizado tem uma base emocional, encontrou recentemente ancoragem em evidências científicas de que as emoções influenciam o cérebro e por conseguinte, a aprendizagem [Chiesa 2014]. Segundo Guerra (2010), “[...]aprendemos tudo aquilo que nos emociona”, pois as emoções são intimamente ligadas ao entendimento e à motivação e neste sentido os jogos digitais despontam como recursos relevantes para os processos educacionais justamente por sua habilidade em provocar emoções [Frome 2007, Sales e Zimmerman 2012].

As emoções que os jogos provocam são estudadas nas mais diversas áreas do conhecimento: psicologia, neuropsicologia, neurociência cognitiva, educação, *design* de jogos, interação homem-computador, dentre outras [Frome 2007]. Na área de educação, uma das questões mais relevantes relacionadas ao uso dos jogos em contextos escolares é como estabelecer maneiras de manter os estudantes mais engajados considerando as emoções geradas pelos jogos e por conseguinte promover melhorias na aprendizagem. Atualmente, os sistemas educacionais tradicionais são amplamente focados na motivação extrínseca, envolvendo punições e recompensas, sejam esses reais ou simbólicas e desconsideram que o indivíduo aprende com maior facilidade se estiver engajado, fazendo-o deliberadamente, com o real desejo de entender [Chiesa 2014, McGonigal 2011, Prensky 2010]. No entanto, quando as estratégias pedagógicas se apropriam de recursos multissensoriais, promovem a reexposição aos conteúdos e a experimentação sob formas diferentes e níveis de complexidade crescentes o aprendizado é promovido de maneira mais consolidada [Guerra 2010].

Na universidade de Chicago, nos Estados Unidos, em um estudo realizado com 300 crianças em idade escolar utilizando jogos inspirados no *Gameboy*¹⁴, os resultados mostraram alta motivação, tanto das crianças familiarizadas com a tecnologia quanto

¹³ <http://blogs.estadao.com.br/radar-cientifico/2011/11/10/a-estrategia-dos-jogadores-de-fold-it/>

¹⁴ Console de jogos da empresa Nintendo®

das que não tinham acesso a ela fora da escola. Em um outro estudo realizado, que visava compreender a opinião dos professores sobre os limites e potencial dos videogames, um dos resultados obtidos é que tais profissionais, reconhecem, em sua maioria, que os jogos são relevantes no desenvolvimento de estratégias importantes para o aprendizado [Chiesa 2014].

Kishimoto (2008) argumenta que o jogo é um instrumento pedagógico relevante, uma atividade livre e alegre, de grande valor social, oferecendo inúmeras possibilidades educacionais por favorecerem o desenvolvimento corporal, o estímulo da vida psíquica, a inteligência, a adaptabilidade ao grupo, sobretudo na educação infantil e o ensino fundamental. Os jogos digitais, por sua vez, possuem características inerentes que promovem uma experiência de aprendizagem mais lúdica, imersiva e engajadora, e as evidências apontam que o aprendizado baseado em jogos digitais pode melhorar o engajamento e a motivação.

A tecnologia e especialmente o jogo digital, pode ser usado como uma maneira de simular experiências reais e permitir a prática privada para evitar possíveis constrangimentos ao cometer erros na frente dos colegas [Chiesa 2014]. Essa experimentação é um dos principais benefícios da utilização dos jogos digitais, pois além do jogador poder vivenciar variadas alternativas, desfechos e consequências, os erros cometidos não têm implicações no mundo real, e incorporam elementos de ludicidade e motivação [Alves 2014, McGonigal 2011, Prensky 2010, Wangenheim e Wangenheim 2012].

Existem alguns fatores pelos quais os jogos podem ser considerados excelentes instrumentos pedagógicos: (i) oferecem desafios possíveis de serem resolvidos ecoerentes com as habilidades desenvolvidas [Alves 2014, Prensky 2010, Wangenheim e Wangenheim 2012], (ii) fomentam a motivação intrínseca e o engajamento [Dickey 2007, Krendl e Lieberman 1988, Salen e Zimmerman 2012], (iii) capturam a atenção das pessoas, (iv) permitem que as pessoas participem ativamente ao invés de assistirem passivamente [Krendl e Lieberman 1988] e (v) permitem a prática repetitiva [Kozma 1991].

Prensky (2010) argumenta que os jogos de computador e videogames são envolventes para um número significativo de pessoas por diversas razões, dentre as quais, algumas são porque eles proporcionam: (i) satisfação e prazer, (ii) envolvimento intenso e apaixonado (iii) estrutura, (iv) motivação, (v) algo a ser feito, (vi) fluxo e, além de tudo, propiciam aprendizagem. Essas características corroboram para proporcionar uma interação lúdica significativa, e, uma vez inseridos em espaços educativos permitem que os estudantes participem ativamente de sua aprendizagem enquanto se divertem.

Notadamente, os jogos e seus elementos podem ser relevantes, enquanto artefatos educacionais, para promover experiências de aprendizagem mais lúdicas e atraentes. A incorporação de jogos digitais que detém valor educacional ou ainda variados tipos de aplicações de *software* que utilizam jogos computacionais para o ensino é denominada aprendizagem baseada em jogos digitais [Prensky 2010, Tang, Hanneghan e El-Rhalibi 2009 *apud* Monsalve 2014].

As abordagens DGBL (do termo em inglês *Digital Game-Based Learning*) têm como principal finalidade prover o suporte à aprendizagem, à avaliação do desempenho dos alunos e melhorias no ensino, se apropriando das características inerentes aos jogos,

tal qual sua natureza experiencial, para o *design* de propostas pedagógicas de ensino mais imersivas e significativas para os estudantes.

Deste modo, considerando o questionamento trazido no início deste capítulo, sobre como os educadores poderiam criar ambientes de aprendizagem mais engajadores cuja pedagogia fomente uma mentalidade incremental, encoraje a prática, valorize o esforço e incorpore o erro como parte do processo de aprendizado, compreende-se que os jogos, sobretudo os digitais, possuem características que possibilitam a criação de ambientes com esta configuração.

Todavia, de um modo geral, ainda são encontrados obstáculos na integração de recursos tecnológicos na educação, sobretudo quando se trata de jogos. Estas barreiras vão desde a competência do profissional, o quão este está disposto a usar as tecnologias em seu espaço de ensino-aprendizagem, e mesmo a cultura escolar, *hardware* e *software* disponíveis, o suporte técnico institucional, dentre outras questões como os estigmas associados ao uso de jogos em contextos educacionais [Bingimlas 2009]. Neste sentido, a seção a seguir apresenta diversos artefatos lúdicos, assim como sugestões para a integração em contextos educacionais de programação.

2.4. Jogos em Experiências Introdutórias de Aprendizagem de Programação

No ensino de programação, comumente, as iniciativas para incorporar os jogos e recursos lúdicos se concentram em estratégias de gamificação, aprendizagem baseada em jogos e no desenvolvimento de jogos. Há uma ampla variedade de jogos digitais e artefatos lúdicos que englobam conceitos de programação, disponíveis na web e/ou para dispositivos móveis e que atendem faixas etárias variadas, alguns destes são apresentados a seguir considerando a abordagem de ‘game thinking’ incorporada.

2.4.1. Brinquedos e Jogos

Os brinquedos programáveis são brinquedos que podem executar sequências de instruções definidas por crianças. Comumente estes brinquedos apresentam-se na forma de um veículo com rodas e assumem aparências diversas como carro, tanque, abelha, e outras figuras representativas do imaginário infantil [Raabeet *al.* 2015]. São em sua maioria desenvolvidos visando apresentar conceitos de programação e resolução de problemas através da programação e/ou interação com artefatos robóticos.

As instruções executadas relacionam-se com o deslocamento do veículo e com o movimento de rotação do veículo, tais como: andar para frente ou para trás alguns centímetros e girar 90 graus para direita ou para esquerda. Em parte a concepção destes brinquedos foi influenciada pela pesquisa realizada por Seymour Papert que propôs a criação da linguagem Logo para movimentar “tartarugas” robóticas [Raabeet *al.* 2015].

Um exemplo deste tipo de brinquedo é o *BeeBot*¹⁵, projetado para ensinar sequências de instruções, estimativas e resolução de problemas para crianças. O brinquedo dispõe de um conjunto de cenários e uma vasta gama de materiais de apoio ao professor, além de acessórios extras (Figura 2.6). As crianças podem controlar o *BeeBot* para que ele faça movimentos de rotação e ande para frente ou para trás.

¹⁵<https://www.bee-bot.us/>



Figura 2.6. Brinquedo de programar BeeBot

O *Ozobot Bit*¹⁶ é um outro exemplo de brinquedo programável destinado a ensinar conceitos de programação para crianças e adolescentes (Figura 2.7). O *Ozobot Bit* é controlado através de códigos desenvolvidos por meio de uma linguagem de programação visual baseada em blocos, o *OzoBlocklyProgramming Editor*¹⁷.

A utilização do brinquedo é uma combinação de abordagens híbridas envolvendo tabuleiros físicos, jogos digitais e desafios *online*. No site oficial estão disponíveis vários recursos adicionais para a utilização do brinquedo. Outros brinquedos similares são *Cubelets*¹⁸, *Dash andDotRobot*¹⁹ e *Sphero BB-8*²⁰.



Figura 2.7. Brinquedo de programar Ozobot

Existem ainda os jogos de tabuleiro e de cartas que objetivam apresentar conceitos introdutórios de programação como condicionais, sequências de instruções e laços. O *CodeMonkeyIsland*²¹ é um exemplo de jogo de tabuleiro voltado a apresentar conceitos de programação para crianças a partir dos oito anos. O jogo abrange conceitos como estruturas de controle, estruturas de dados, lógica booleana, operadores e operações matemáticas (Figura 2.8). Outros exemplos de jogos similares são o *C-Jump*²², o *CodeMaster*²³ e o *RobotTurtles*²⁴, todos jogos de tabuleiro.

¹⁶ <http://ozobot.com/products/ozobot-bit>

¹⁷ <http://ozoblockly.com/>

¹⁸ <http://www.modrobotics.com/cubelets/>

¹⁹ <https://www.makewonder.com/>

²⁰ <http://www.sphero.com/>

²¹ <http://codemonkeyplanet.com/>

²² <http://www.c-jump.com/>

²³ <http://www.thinkfun.com/products/code-master/>



Figura 2.8. CodeMonkeyIsland: Exemplos de cartas de regras do jogo e o tabuleiro

2.4.2. Gamificação

Ao longo dos últimos observou-se uma crescente quantidade de artefatos destinados à auxiliar no ensino de programação que incorporam estratégias de gamificação. Uma das plataformas mais conhecidas foi desenvolvida mediante a iniciativa *Code Hour*²⁵ (Hora do Código), que é um movimento de amplitude global iniciado nos Estados Unidos. Esta iniciativa é realizada anualmente durante um evento homônimo promovido durante a *Computer Science Education Week* (Semana de Educação em Ciência da Computação).

Este movimento tem por objetivo apresentar e desmitificar a temática de programação para o público em geral, sobretudo crianças e adolescentes, provendo atividades lúdicas, desafios e jogos inspirados em diversos elementos do cenário infanto-juvenil, os quais abrangem tópicos de programação, todos estes recursos estão compilados no *site* oficial *Code Studio*. No *Code Studio*, há cursos sequenciais sobre fundamentos de Ciência da Computação abrangendo tanto atividades desplugadas quanto atividades digitais.

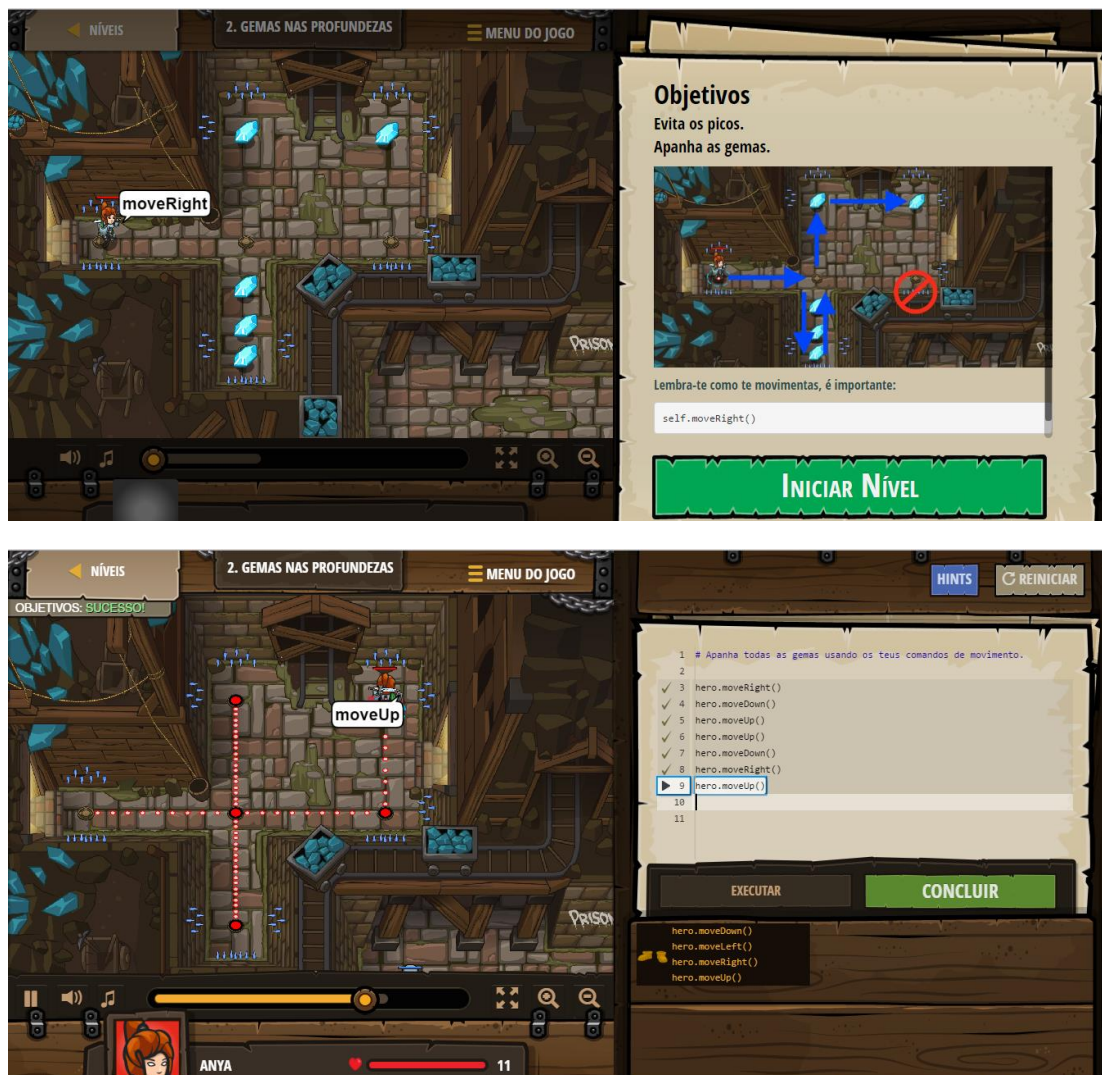
O *Bitsbox*²⁶ é originalmente um clube de assinaturas de artefatos lúdicos para a aprendizagem de programação, cujos produtos são destinados às crianças. Além disso, o *Bitsbox* disponibiliza ainda uma versão online e gratuita com uma série de atividades de programação em que os usuários têm a oportunidade de praticar programação usando a linguagem *JavaScript* para criar jogos simples e animações. Outro aspecto interessante no *Bitsbox* é o espaço destinado ao professor, que conta com uma série de planos de aula abrangendo diversas disciplinas e oferece um recurso para cadastro de vários alunos de maneira simultânea na plataforma. No mais, também é possível adquirir os kits do clube de assinaturas em grandes quantidades por um valor mais baixo, além de materiais extras envolvendo temáticas diversas.

²⁴<http://www.thinkfun.com/robot-turtles/>

²⁵ code.org

²⁶<https://bitsbox.com/>

O *CodeCombat*²⁷ poderia ser facilmente classificado apenas como um jogo de RPG *online*, mas também é uma plataforma de aprendizagem de programação para iniciantes em que o jogador pode optar por aprender utilizando *Javascript* ou *Python* (Figura 2.9). Com um visual retrô e uma centena de níveis de jogos de complexidade crescente que abrangem sintaxe básica, argumentos, *strings*, laços, variáveis, algoritmos, funções, *arrays*, dentre muitos outros.



²⁷ <http://br.codecombat.com/>



Figura 2.9. ScreenshotCodeCombat em três momentos: no início da fase, na execução do código e no fim da fase.

Além disso, são oferecidos manuais para os professores, que abrangem uma visão geral dos níveis, os conceitos abordados, assim como as soluções das fases nas duas linguagens. Atualmente, um usuário com perfil de professor pode inscrever uma ou mais turmas e acompanhar o desempenho dos alunos. Uma das facilidades de incorporar o *CodeCombat* em sala de aula, é que ao criar a turma, o professor define a linguagem a ser utilizada, as idades dos alunos e o nível de proficiência em programação.

Uma outra plataforma é a *CodinGame*²⁸, que abrange diversos níveis de dificuldade, a partir de tutoriais, desafios e batalhas, envolvendo desde conceitos mais básicos até mesmo problemas de otimização, programação de agentes, dentre outros (Figura 2.10). Todos os problemas resolvidos estão inseridos em contextos de jogos, desde o início, por exemplo: o usuário aprende a declarar e manipular variáveis controlando mísseis de uma nave sob ataque. É possível praticar programação em uma vasta gama de linguagens, incluindo *Java*, *C++*, *C#*, *Python3*, *Javascript*, *PHP*, dentre diversas outras.

O *CodinGame* é completamente gamificado, todas as atividades contribuem para adquirir 'XP' e prêmios virtuais, além de desbloquearem conteúdos exclusivos. Um outro ponto forte desta plataforma é a possibilidade de interagir com os outros jogadores.

²⁸<https://www.codingame.com/>

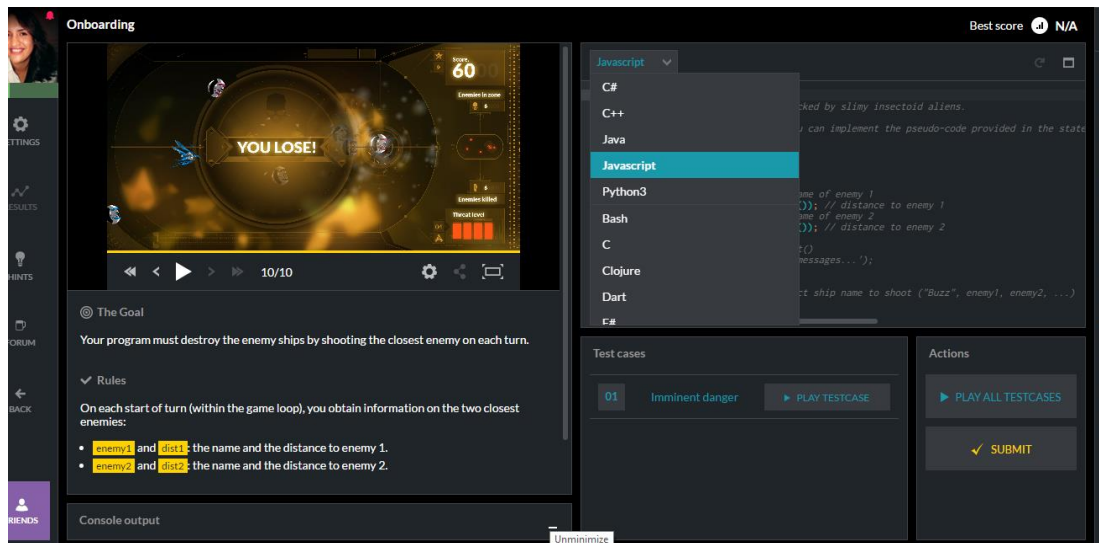


Figura 2.10. Screenshot da plataforma CodinGame

*Code Hunt*²⁹ é uma plataforma *online* desenvolvida e mantida pelo grupo de pesquisa RISE (*Research in Software Engineering*) da Microsoft®. No *Code Hunt*, os jogadores são imersos em uma narrativa em que códigos mal escritos estão atrapalhando o bom funcionamento de um sistema, então será necessário corrigir estes fragmentos de código, utilizando a linguagem *Java* ou *C#* (Figura 2.11).

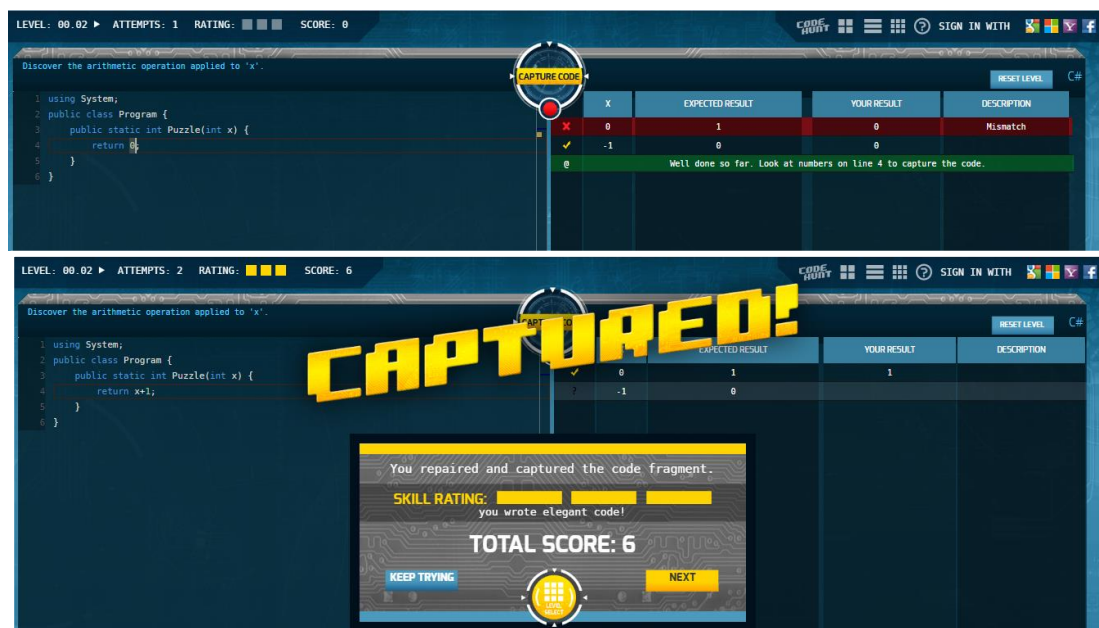


Figura 2.11. Screenshot da plataforma Code Hunt

Os conceitos são divididos em ‘setores’, que contém vários níveis, envolvendo conceitos como: aritmética, *loops*, condicionais, strings, *loops* aninhados, *arrays*, cifras, *puzzles*, dentre outros. Similarmente às outras plataformas apresentadas, no *Code Hunt*, os desafios resolvidos acumulam experiência e desbloqueiam novos níveis, no entanto,

²⁹<https://www.codehunt.com>

soluções mais elegantes ganham pontos extras. Além disso, também é possível criar novos níveis e por conseguinte outros setores (conjuntos de níveis) e zonas (conjuntos de setores).

Nacionalmente, por sua vez, destaca-se o Programaê³⁰, um portal que contém uma coletânea de projetos e desafios, baseado nas principais plataformas de ensino de programação para crianças e adolescentes. O Programaê possui uma estrutura voltada a possibilitar tanto o aprendizado autônomo e individual, quanto para apoiar o professor a gerenciar e acompanhar o desenvolvimento de suas turmas, de modo que é possível acompanhar o progresso individual dos alunos.

Existem ainda diversos artefatos similares, em sua maioria gratuitos ou com recursos que podem ser utilizados gratuitamente, tais como: *Codecademy*³¹, *CodeAvengers*³², *Code Wars*³³, *Codesters*³⁴, *Eu Posso Programar*³⁵, *Robomind*³⁶, *You++*³⁷, dentre vários outros.

2.4.3. Jogos Sérios

Ao longo dos últimos anos, os jogos sérios voltados a auxiliar no ensino de programação têm apresentado um crescimento notável. Além disso, com a popularização dos dispositivos móveis e a crescente valorização de modelos híbridos de ensino e aprendizagem, vários jogos educacionais que visam auxiliar no aprendizado de programação figuram nas lojas de aplicativos das principais plataformas móveis (*Android* e *iOS*).

No intuito de apresentar e desmitificar os conceitos de programação, estes jogos concentram-se no público infanto-juvenil, sendo aqueles voltados para crianças os que apresentam uma maior variedade. Em sua maioria, versam sobre programação em uma abordagem introdutória e menos complexa, abrangendo conceitos como: (i) sequências de instruções, (ii) condicionais, (iii) laços, (iv) procedimentos/ funções, (v) parâmetros. Comumente, estes jogos foram desenvolvidos embasando-se em currículos de ensino de Ciência da Computação, oferecendo materiais e recursos diversos que dão suporte ao professor em suas práticas educativas.

Narrativas simples, personagens coloridos e cativantes, instruções de comando construídas a partir de blocos de código *drag-and-drop* são as principais características em comum entre estes jogos. Os exemplos para este tipo de abordagem são numerosos, dentre os quais podemos citar: *Bit by Bit*³⁸, *Beebot*³⁹, *Cargo-Bot*⁴⁰, *Cato'sHike*⁴¹, *Daisy*

³⁰ www.fundacaolemann.org.br/programae/

³¹ <https://www.codecademy.com/pt>

³² <https://www.codeavengers.com/>

³³ <http://www.codewars.com/>

³⁴ <https://www.codesters.com/HoC/>

³⁵ <http://www.eupossoprogramar.com/>

³⁶ <https://www.robomindacademy.com/go/robomind/home>

³⁷ <http://youplusplus.eu/>

³⁸ <http://rikaigames.com/bitbybit/>

³⁹ <https://www.bee-bot.us/>

⁴⁰ <http://twolivesleft.com/CargoBot/>

⁴¹ <http://hwahba.com/catoshike/>

*thedinosaur*⁴², *Hopscotch*⁴³, *Kodable*⁴⁴, *Learnwithel Chavo*⁴⁵, *Lightbot*⁴⁶, *MyRobotFriend*⁴⁷, *RobotSchool*⁴⁸, *Run Marco*⁴⁹, *The FoosCoding*⁵⁰, *Tynker*⁵¹, entre outros.

Bit by Bit possui uma linguagem totalmente visual e é destinado a crianças entre 6 e 12 anos de idade, este jogo explora conceitos como sequências de instruções, condicionais e laços, no controle de um ou mais personagens que precisam realizar tarefas simples como mover-se de um ponto ao outro ou coletar objetos ao longo de mais de 40 níveis (Figura 2.12).

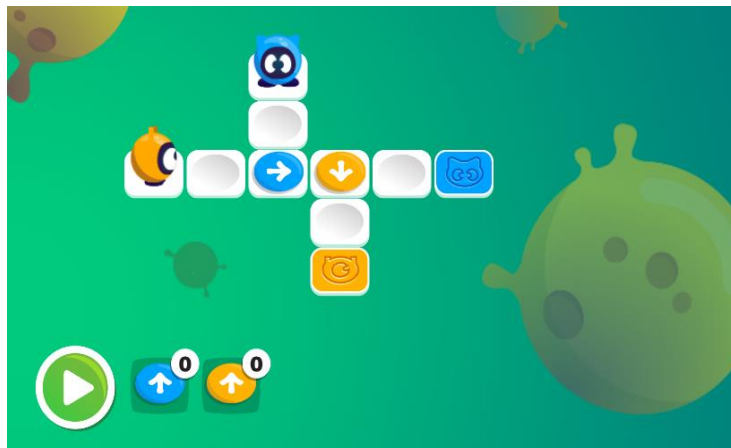


Figura 2.12. Screenshot do jogo Bit by Bit

Outro jogo similar é *The Foos*, voltado às crianças com idades entre 5 e 8 anos. Em mais de 40 níveis são apresentados conceitos introdutórios de programação, como: sequências de instruções, condicionais, *loops*, funções, parâmetros, dentre outros (Figura 2.13). O jogo dispõe também de espaços de exploração criativa e puro entretenimento, como o *Foo Studio*, o qual permite a resolução de níveis bônus, além de possibilitar a criação de novos níveis de jogo que podem ser compartilhados publicamente com todos os usuários do *The Foos*.

Existe ainda uma versão mais enxuta voltada à Hora do Código, que abrange 24 níveis e que pode ser encontrada gratuitamente na web. Além disso, estão disponíveis alguns recursos complementares que auxiliam na sua integração em sala, tais como:

- O currículo completo, abrangendo todas as fases do jogo *mobile*;
- O currículo associado à versão Hora do Código, disponível apenas na *web*;

⁴²<https://itunes.apple.com/br/app/daisy-the-dinosaur/id490514278?mt=8>

⁴³<https://www.gethopscotch.com/>

⁴⁴<https://www.kodable.com/>

⁴⁵http://www.aprendeconelchavo.com/us/site/main_us.php?lang=en

⁴⁶<https://lightbot.com/hocflash.html>

⁴⁷ <http://www.leapfrog.com/en-us/home>

⁴⁸ <http://www.robotschoolapp.com/>

⁴⁹<https://www.allcancode.com/web>

⁵⁰ <http://thefoos.com/>

⁵¹<https://www.tynker.com/>

- Um conjunto de atividades relacionadas aos conteúdos do jogo que podem ser impressas e realizadas com os alunos sem a necessidade de utilizar um computador;
- Atividades de colorir para imprimir.



Figura 2.13. Screenshot do jogo The Foos

O *Tynker*, disponível tanto para dispositivos móveis quanto para *web*, é uma plataforma completa repleta de módulos que variam desde o iniciante ao avançado e abrangem conceitos diversos de programação, focando no público infanto-juvenil (Figura 2.14).

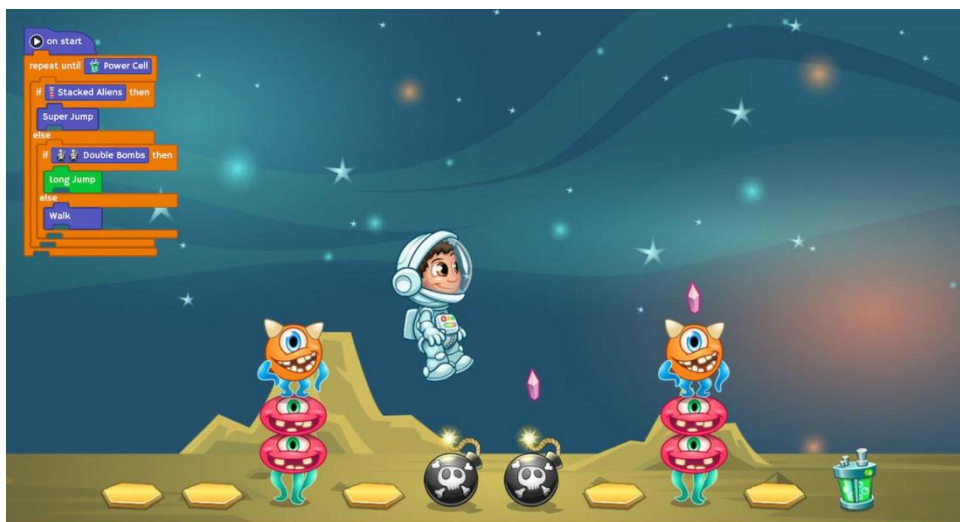


Figura 2.14. Screenshot do jogo Tynker

Alguns destes módulos estão disponíveis gratuitamente para experimentação. Um dos grandes diferenciais do *Tynker* é uma ampla oferta de material de apoio ao professor, tais como planos de aula e outros recursos adicionais.

O quadro a seguir (Quadro 2.1) apresenta os conteúdos abrangidos pelos aplicativos listados acima, assim como o custo e se estes aplicativos oferecem materiais

de apoio pedagógico. Para este quadro aplica-se a seguinte legenda: S - Sequências de Instruções, C - Condicionais, L - Laços, F - Procedimentos ou Funções, P - Parâmetros, R\$ - Custo, M - Materiais de apoio ao professor, SO - Sistema operacional.

Quadro 2.1. Aplicativos Móveis para Aprendizagem de Programação

	S	C	L	F	P	R\$	M	SO
Bit by Bit	X	X	X	X	X	Não	Sim	Android, iOS
Beebot	X	X	X			Não	Sim	Android, iOS
Cargo-Bot	X	X	X	X	X	Sim	Não	iOS
Cato'sHike	X	X	X	X	X	Não	Não	iOS
Daisy thedinosaur	X	X	X	X		Não	Não	iOS
Hopscotch	X	X	X	X	X	Não	Sim	Android, iOS
Kodable	X	X	X	X		Não	Sim	Android, iOS
Learnwithel Chavo	X	X	X	X		Não	Não	Android, iOS
Lightbot	X	X	X	X	X	Não	Sim	Android, iOS
MyRobotFriend	X	X	X	X	X	Não	Não	Android, iOS
RobotSchool	X	X	X	X	X	Sim	Não	Android, iOS
Run Marco!	X	X	X	X		Não	Sim	Android, iOS
The FoosCoding	X	X	X		X	Não	Sim	Android, iOS
Tynker	X	X	X	X	X	Não	Sim	Android, iOS

Conforme é possível observar, existem diversos aplicativos gratuitos disponíveis para os principais sistemas operacionais móveis. De um modo geral estes aplicativos oferecem a oportunidade de um contato inicial com conceitos de programação a partir de abordagens baseadas em linguagens visuais de programação.

No entanto, não foram encontrados programas similares que sejam destinados ao uso em computadores pessoais. Convém mencionar que buscou-se aplicativos que pudessem auxiliar em experiências introdutórias de programação, existe ainda uma gama de aplicativos voltados às práticas de programação para alunos que detêm mais experiência em programação que não foram mencionados neste trabalho.

2.4.4. Criação de Jogos Digitais

Uma outra possibilidade para incorporação de jogos em experiências introdutórias de programação é o desenvolvimento de jogos digitais por meio de linguagens visuais de programação. Estas linguagens têm como diferencial permitir que mesmo alunos que ainda não tenham conhecimento em programação possam criar jogos.

Uma das linguagens de programação visual mais utilizadas é o *Scratch*⁵², desenvolvida e mantida pelo grupo *LifelongKindergarten* no *Media Lab* do Instituto de Tecnologia de Massachussets (MIT). No *Scratch* é possível manipular mídias, tais como imagens e músicas, para a criação de histórias interativas, jogos ou animações. As instruções de código no *Scratch* são blocos encaixáveis que necessitam ser arrastados e soltos no espaço de comandos.

Os comandos no *Scratch* são similares às peças de quebra-cabeça e as suas formas indicam como eles podem ser combinados corretamente. Desta maneira, os comandos quando combinados formam programas que estão sempre sintaticamente corretos. Sem a preocupação com erros sintáticos, os usuários focam apenas na criação da lógica de funcionamento de seus projetos [Malan e Leitner 2007, Maloney et al. 2010 apud Aureliano e Tedesco 2012]. O *Scratch* é gratuito, tem versões *online* e *desktop* e está disponível em português brasileiro.

Similarmente ao *Scratch*, o *App Inventor*⁵³ também é uma plataforma cuja linguagem de programação visual é baseada em blocos (Figura 2.15). O diferencial do *App Inventor* é possibilitar a criação de aplicativos para dispositivos móveis *Android* sem exigir conhecimento prévio em programação [Gomes e Melo 2013].



Figura 2.15. Janela *App Inventor Designer*

O desenvolvimento de uma aplicação nesta ferramenta é realizado através de duas janelas: *Designer* e *Blocks*. A janela *Designer* é executada a partir do navegador e permite criar visualmente a interface do usuário, ao clicar e arrastar os componentes da *Palette*, para o *Viewer*. A janela *Blocks Editor* (Figura 2.16), por sua vez, permite controlar o comportamento dos componentes definidos na janela *Designer*.

⁵² <https://scratch.mit.edu/>

⁵³ appinventor.mit.edu

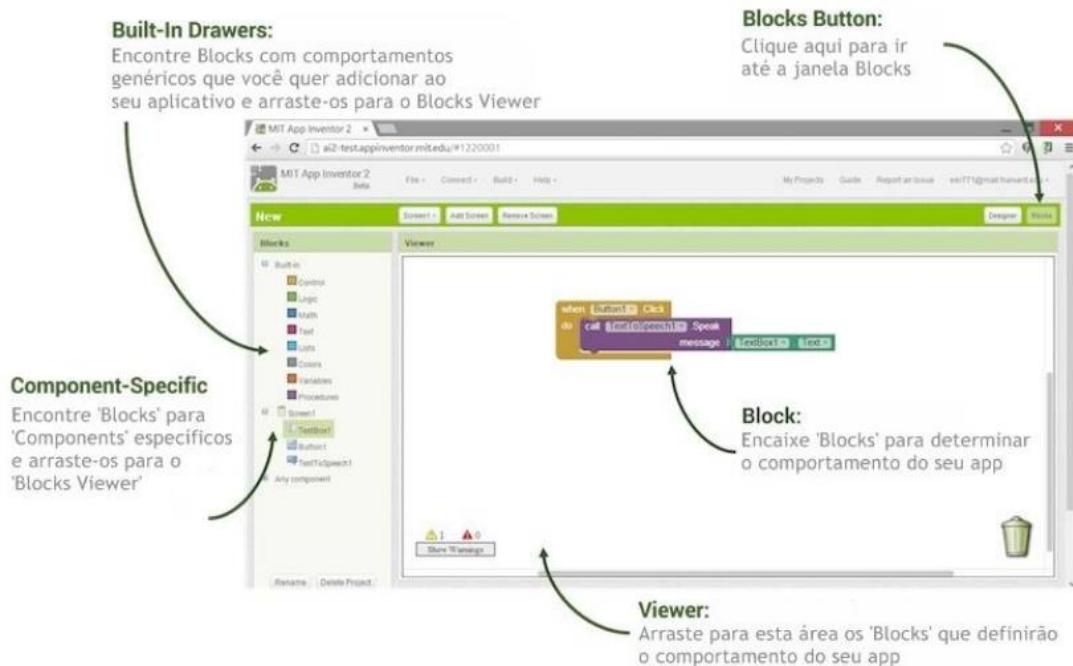


Figura 2.16. Janela Blocks Editor

Neste ambiente, o usuário encontra blocos conectáveis, que podem ser eventos ou métodos, em uma interface do tipo arrastar e soltar. Estes blocos operam strings e listas, realizam ações de controle (e.g. *if*, *else*, *foreach*, etc.) e operações matemáticas, entre outras funcionalidades. É possível executar o teste do aplicativo diretamente em um dispositivo *Android* (*smartphone* ou *tablet*) que esteja conectado ao computador [Gomes e Melo 2013]. O *App Inventor* é gratuito e está disponível em português brasileiro.

O *Kodu*⁵⁴, por sua vez, é um *software* para computadores pessoais que permite a criação de jogos digitais em 3D para o sistema operacional *Windows* e para o console *Xbox*⁵⁵ através de uma linguagem de programação visual (Figura 2.17). O usuário escolhe cenários de jogos pré-definidos e insere personagens e programa os personagens através da combinação de movimentos e ações possíveis listados no cenário escolhido, apenas arrastando e soltando estes componentes pré-definidos na tela. Os jogos criados podem ser compartilhados como os outros usuários. No entanto, apesar de ser disponibilizada gratuitamente, esta ferramenta está disponível apenas em inglês.

⁵⁴ <http://www.kodugamelab.com/>

⁵⁵ Atualmente, o *Kodu* está disponível para *Xbox* da Microsoft © apenas nos Estados Unidos.



Figura 2.17. Exemplo de comandos de instruções no *Kodu*

Em contraste à todas as outras ferramentas apresentadas, o aplicativo *PocketCode*⁵⁶ é um ambiente de desenvolvimento integrado (IDE) completamente desenvolvido e projetado para ser executado diretamente a partir de um *smartphone* (Figura 2.18). Atualmente, o *PocketCode* está disponível apenas para dispositivos móveis *Android*. O ambiente se assemelha muito às outras plataformas baseadas em linguagens de programação visual, sobretudo pela linguagem adotada que se parece com o *Scratch*.

Os usuários podem criar aplicativos diversos (jogos, animações e histórias interativas) para dispositivos móveis a partir deste aplicativo e compartilhá-los no site oficial do *PocketCode* para que outros usuários possam realizar o *download* e usar ou ainda podem compartilhar o projeto para que outros usuários possam modificar. No mais, é possível incorporar aos aplicativos criados alguns recursos específicos oferecidos pelos dispositivos móveis como NFC, acelerômetro, utilizar o som e a câmera, além de conectar-se via *Bluetooth* com placas Arduíno, robôs, como os *Lego Mindstorms*, ou até mesmo drones.

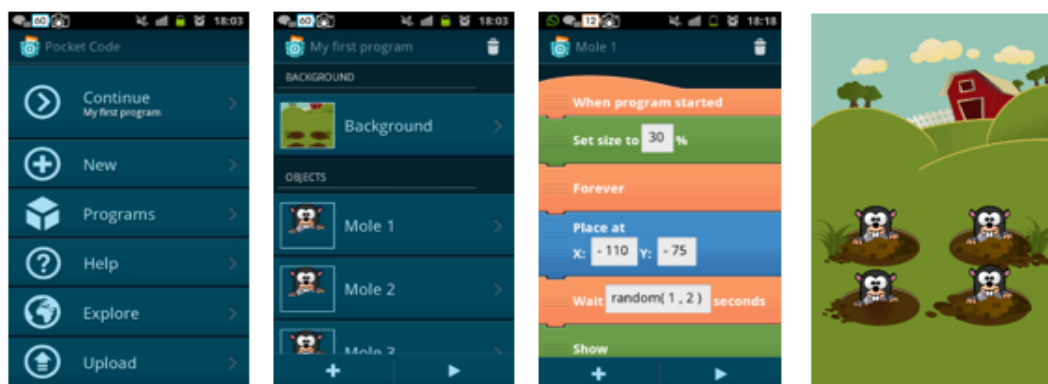


Figura 2.18. Screenshots: processo de criação de um aplicativo no *PocketCode*

⁵⁶ <http://www.catrobat.org/>

Existem ainda outras plataformas, também baseadas em linguagens visuais de programação, tais como o *Construct2*⁵⁷, *GameSalad*⁵⁸ e *Stencyl*⁵⁹, que permitem a criação de jogos mais complexos. Estas ferramentas podem ser utilizadas no desenvolvimento de projetos mais robustos ou com públicos mais maduros.

2.5. Integrando os Jogos em Cenários Educacionais

As salas de aula são cenários de diversidade. Em um mesmo espaço de aprendizagem estão estudantes com estilos cognitivos de aprendizagem variados e com características pessoais, sociais e culturais distintas. Portanto, incorporar os jogos, sobretudo os digitais, em contextos educativos traz consigo ainda outros desafios: (i) como integrar os jogos nos cenários de aprendizagem? (ii) como escolher adequadamente os jogos considerando perfis e por conseguinte, necessidades de aprendizagem tão heterogêneos?

No intuito de oferecer caminhos possíveis para a integração dos jogos no ensino, Perrota *et al.* (2013) descrevem um conjunto de implicações para os professores e para as escolas no uso de jogos em cenários de aprendizagem. Inicialmente, é sugerido o balanceamento entre a diversão e aprendizagem ao inserir as atividades de aprendizagem dentro dos jogos e dos contextos de entretenimento. Um bom exemplo disso é integrar à prática pedagógica jogos que não foram projetados com fins educacionais (ou seja, aqueles que podem estar classificados no quadrante Jogos/Brinquedos).

Desde que haja um planejamento adequado, com objetivos educacionais bem definidos, estes jogos podem ser utilizados de maneira a desenvolver habilidades e competências relacionadas aos conteúdos que serão abordados durante a aula, como por exemplo, jogos que abordem a resolução de problemas, raciocínio lógico, o reconhecimento e o uso de padrões, dentre outros conteúdos. A utilização destes jogos com propósitos educacionais envolve o que se classifica como aprendizagem tangencial ou periférica, que diz respeito ao desenvolvimento de habilidades e competências nos jogadores para as quais estes jogos não foram projetados, fomentando experiências de aprendizagem espontâneas e realmente divertidas.

Neste sentido, Gomes e Melo (2013), por exemplo, relatam uma experiência de ensino de programação para estudantes do ensino médio através de uma abordagem híbrida, envolvendo uma rede social, jogos e uma linguagem visual de programação para dispositivos móveis e aulas presenciais. Na fase de uso dos jogos em sala de aula, os alunos foram convidados à experimentarem diversos jogos envolvendo raciocínio lógico, reconhecimento e uso de padrões, resolução de problemas, dentre outros conceitos, a fim de fomentar a discussão sobre algoritmos. Após as sessões de jogos, os alunos foram convidados a descrever algoritmos de resolução dos jogos em uma linguagem tão clara quanto possível, sob a condição de que para obter os pontos relacionados à atividade os seus colegas pudessem ler as instruções, jogar e obter a mesma pontuação.

Outra sugestão destacada por Perrota *et al.* (2013) é a incorporação do conteúdo no contexto de jogo, através de uma narrativa, por exemplo. Atividades específicas de

⁵⁷ <https://www.scirra.com/construct2>

⁵⁸ <http://gamesalad.com/>

⁵⁹ <http://www.stencyl.com/>

conteúdo funcionam melhor quando inseridas em um contexto ficcional e através de mecânicas de jogos. Um exemplo é a experiência introdutória de ensino de programação relatada por Giraffa, Muller e Moraes (2015), que alternava aulas presenciais e um ambiente virtual. O diferencial deu-se através de exercícios imersos em narrativas consideradas interessantes pelos alunos como a série *Game of Thrones*(GoT), o filme Os Vingadores, a série de filmes *Guerra nas Estrelas*, dentre outros. No caso da série GoT, por exemplo, os enunciados lançados no ambiente virtual eram relacionados aos acontecimentos do episódio exibido naquela semana.

O planejamento cuidadoso das funções a serem desempenhadas pelos aprendizes e pelos professores ao longo do uso do jogo ou de uma abordagem baseada em jogos é um outro aspecto a ser considerado [Perrotta *et al.* 2013]. Sugere-se que os professores desempenhem ações que lhes permitam atuar como mediadores, fornecendo instruções quando necessário, assegurando que as funções designadas para os aprendizes sejam seguidas e mantendo uma atmosfera respeitável.

Os recursos apresentados nas seções anteriores, como jogos digitais e plataformas gamificadas, oferecem oportunidades de aprender e praticar programação, alguns de maneira colaborativa, outros aplicados em um determinado contexto. A integração destes recursos em sala de aula pode ser realizada utilizando-os como ambientes de exploração e/ou prática dos conceitos aprendidos. Pode-se sugerir que os alunos explorem o artefato para posterior discussão e apresentação dos conceitos aprendidos, ou os conceitos podem ser explanados anteriormente e os recursos podem ser utilizados para reforçá-los. Os jogos desenvolvidos para dispositivos móveis, por exemplo, são uma possibilidade de explorar modalidades híbridas de ensino. Estes aplicativos podem ser integrados às aulas de diversas formas, permitindo que os aprendizes possam retomar ou explorar os conceitos em práticas lúdicas que vão além dos espaços escolares, em seu próprio ritmo e tempo.

No entanto, um outro desafio relacionado à integração dos jogos em contextos educativos refere-se a seleção dos artefatos, e de algum modo, a problemática envolve atender de maneira adequada a um público heterogêneo. Uma possibilidade é atentar para o que os mantém engajados enquanto jogam, considerando por exemplo, seus perfis psicológicos. Neste sentido, o longo das últimas décadas vários modelos psicológicos dos jogadores têm sido desenvolvidos e oferecem subsídios importantes para a compreensão do comportamento e da interação dos jogadores com os jogos.

Neste direcionamento, o pesquisador britânico Richard Bartle propôs um modelo tipológico de perfis de jogadores, mais conhecida como a teoria de Bartle [Bartle 1996]. Este modelo foi baseado na observação e análise dos resultados de um fórum de discussão entre jogadores MUD (do inglês *Multi-User Dungeon*) sobre o que eles consideravam divertido no jogo e o que pensavam que os outros acharam divertido no mesmo jogo. Os MUDs podem ser definidos como mundos virtuais multijogadores (RPG) totalmente baseados em texto (ou seja, não possuem nenhum gráfico). Nestes jogos, os jogadores assumem o papel de uma personagem e recebem informações textuais que descrevem salas, objetos, outras personagens e criaturas controladas pelo computador, também conhecidas como *non-player characters*(NPCs), em um mundo

virtual. Eles podem interagir com outros jogadores e personagens digitando comandos que, normalmente, se encontram em inglês⁶⁰.

Assim, tem-se quatro tipos diferentes de estilo de jogo (ou retratos psicológicos), os quais são: (i) Conquistadores, (ii) Exploradores, (iii) Predadores (ou Assassinos, na tradução literal), (iv) Socializadores. Ou seja, basicamente, os tipos de jogadores propostos por Bartle são quatro personas definidas por linhas perpendiculares que delineiam quatro quadrantes (Figura 2.19).



Figura 2.19. Diagrama com os tipos de Bartle, livremente traduzido e adaptado.

Os conquistadores são movidos por objetivos no jogo para se destacar dos outros jogadores, geralmente alguma forma de acumular pontos, sejam pontos de experiência, níveis ou mesmo cupons de desconto. Estes jogadores costumam ser atraídos por um inventário de *badges* ou troféus, por exemplo. Eles **agem** sobre o **mundo**.

Os exploradores são conduzidos pela vontade de descobrir o máximo possível sobre o jogo incluindo desde o mapeamento da área geográfica até mesmo a compreensão das mecânicas empregadas no jogo. Eles são curiosos e vão querer entender o porquê e como cumprir um desafio proposto. Eles **interagem** com o **mundo**.

Os predadores são os jogadores de perfil mais competitivo. Estes são movidos pela vontade de impor-se e ficam satisfeitos em proporcionar momentos de agonia e ansiedade nos outros jogadores. Para que eles ganhem, alguém que precisa perder, comumente são presenças frequentes no topo do ranking. Eles **agem** sobre outros **jogadores**.

Os socializadores são os jogadores que buscam estabelecer relações com os outros jogadores, seja pela contação de histórias dentro dos jogos, seja por motivarem os desafios em time ou mesmo por simplesmente comentarem status. Segundo Vianna *et al.* (2013), para este tipo de jogador mais importante do que atingir os objetivos propostos ou concluir as tarefas designadas, é a ocasião do jogo em si e seu potencial de estimular vínculos sociais que os interessam. Os socializadores costumam preferir jogos cooperativos, que demandam trabalho conjunto e evidenciam personalidades

⁶⁰Fonte: https://pt.wikipedia.org/wiki/Multi-user_dungeon

colaborativas. De acordo com a pesquisa, representam algo em torno de 80% da totalidade dos jogadores existentes. Eles **interagem com outros jogadores**.

Os tipos psicológicos de Bartle são traçados a partir do teste de Bartle⁶¹, definindo elementos de interação que mais interessam a cada categoria que ele descreve. A tabela⁶² a seguir (Tabela 2.2) relaciona variados gêneros de jogos (e respectivos títulos de jogo como exemplo) considerando os tipos psicológicos de jogadores proposto por Bartle.

Tabela 2.2. Gêneros de Jogos versus Tipos de Jogadores

Gênero	Jogos típicos	Tipos de Jogadores
FPS	<i>Halo, Call of Duty, Half-Life, Crysis</i>	Predadores, Conquistadores
MMORPG	<i>World of Warcraft, EVE Online, Guild Wars</i>	Conquistadores, Exploradores, Socializadores
MMOG	<i>Unreal Tournament, Team Fortress, any FPS multiplayer mode</i>	Predadores, Conquistadores
Aventura	<i>King's Quest, Myst, The Longest Journey</i>	Socializadores, Exploradores
Ação	<i>Tomb Raider, Uncharted, Angry Birds</i>	Predadores, Conquistadores
Horror	<i>Resident Evil, Dead Space, Amnesia</i>	Predadores, Conquistadores
Estratégia Baseada em Turnos	<i>Civilization, Master of Orion, Galactic Civilizations</i>	Exploradores
<i>PhysicsPuzzler</i>	<i>Half-Life 2, Portal, World of Goo</i>	Predadores, Exploradores
Estratégia em Tempo Real	<i>Age of Empires, StarCraft, Supreme Commander</i>	Conquistadores, Predadores
Simulador de voo	<i>Falcon 4.0, Microsoft Flight Simulator X</i>	Predadores, Exploradores
Atiradores Espaciais	<i>WingCommander, Freelancer</i>	Conquistadores
Música	<i>Rock Band, Guitar Hero, Audiosurf</i>	Predadores, Socializadores
Simulação	<i>SimCity, Balance of Power, Railroad Tycoon</i>	Exploradores
Social	<i>FarmVille, Mafia Wars</i>	Socializadores, Conquistadores
Apostas Online	Blackjack, Texas Hold-Em Poker	Predadores, Conquistadores

Um modelo eficaz deveria ser capaz de explicar como games em particular satisfazem determinados interesses de estilos de jogo. De um modo geral, estes modelos são incorporados ao desenvolvimento de jogos, mas podem ser úteis para identificar os

⁶¹Nenhum questionário oficial do teste de Bartle foi publicado, mas há diversas versões disponíveis na internet.

⁶² Fonte: http://www.gamasutra.com/view/feature/6474/personality_and_play_styles_a_.php?print=1

perfis de jogadores em sala de aula e nortear a escolha dos jogos mais adequados conforme os tipos de perfis encontrados.

Outra perspectiva a ser considerada na escolha dos jogos são as habilidades e competências a serem desenvolvidas e os estilos de jogos possíveis a serem adotados. No Quadro 2.2, livremente adaptado e traduzido de Wassila e Tahar (2012), são descritas algumas habilidades a serem desenvolvidas e as técnicas de aprendizagem mais adequadas e os possíveis gêneros de jogos que poderiam ser incorporados.

Quadro 2.2. Relação entre habilidades, técnicas de aprendizagem e gêneros de jogos, livremente adaptada e traduzida de Wassila e Tahar (2012)

Habilidades	Técnicas de Aprendizagem	Gêneros de Jogos
<ul style="list-style-type: none">- Raciocínio- Resolução de problemas- Relações de causa e efeito	<ul style="list-style-type: none">- Aprendizagem pela prática- Sistemas behavioristas- Prática e feedback- Exploração- Micromundos	<ul style="list-style-type: none">- Ação- Jogos de reflexo- Simulações- <i>Puzzles</i>- Estratégia- Aventura

Por exemplo, considerando que as práticas introdutórias de programação essencialmente envolvem o desenvolvimento do raciocínio lógico e habilidades de resolução de problemas é recomendável empregar técnicas de aprendizagem que fomentem a aprendizagem pela prática, especialmente envolvendo a prática e o feedback. Estilos de jogos que proveem e incorporam estas características em geral são os jogos de ação, de reflexo, *puzzles* (desafios).

Notadamente, há poucos jogos destinados exclusivamente ao ensino de programação para iniciantes em que os conteúdos sejam parte integrante da estrutura do jogo, sobretudo considerando experiências de aprendizagem na educação superior, conforme visto em seções anteriores. Os jogos disponíveis atualmente são, em sua maioria, baseados em linguagens visuais de programação, cujas narrativas são pouco profundas ou inexistentes. Deste modo, uma estratégia possível é a apropriar-se da aprendizagem tangencial e utilizar jogos que não foram desenvolvidos exclusivamente para este fim, conforme mencionado no início desta seção. No entanto, os artefatos gamificados apresentados na seção 1.4.2 podem ser recursos interessantes a serem considerados, uma vez que são intrinsecamente sistemas behavioristas, possibilitam o aprendizado pela prática (inicialmente pela imitação), além de oferecerem feedback contínuo.

Os jogos e abordagens baseadas em elementos de jogos podem ser integradas aos contextos de ensino e aprendizagem de programação de diversas formas, desde que haja um planejamento adequado para as atividades a serem realizadas e os objetivos didático-pedagógicos estejam bem definidos.

Além disso, é necessário balancear a aprendizagem e a diversão, compreendendo que a mera inserção dos jogos no cenário de aprendizado não necessariamente promove o engajamento. Os modelos de perfis dos jogadores e tipos psicológicos, por sua vez, podem ser elementos auxiliares para a seleção de jogos adequados ou na criação de abordagens baseadas em jogos, considerando a heterogeneidade do público atingido.

2.6. Considerações Finais

Os desafios encontrados nas experiências introdutórias de programação, amplamente discutidos na literatura, despertam a necessidade de buscar compreender quais são as barreiras enfrentadas pelos alunos e como mitigá-las, sobretudo considerando a crescente necessidade de aprendizado destas habilidades elencadas como requisitos essenciais para os profissionais das próximas décadas. No entanto, neste processo estão envolvidos diversos elementos: os sujeitos aluno e professor - e seus respectivos conhecimentos e experiências prévias, modelos mentais, crença implícitas, assim como as próprias práticas didático-pedagógicas em si, tornando complexo definir quais a(s) origem(ns)/ causa(s) dos problemas associados ao ensino de programação.

Nesta perspectiva, têm surgido diversas hipóteses que apontam alguns fatores como possíveis causas: a falta de aptidão, a combinação inadequada e a utilização inapropriada dos conceitos básicos, conhecimentos prévios insuficientes, crenças implícitas, dentre outros. E embora não haja um consenso sobre quais as causas das barreiras iniciais enfrentadas em experiências introdutórias de programação, parece haver o consenso de que são estas barreiras a principal razão para a desmotivação, a falta de engajamento, o desinteresse e não raro, os elevados índices de reprovação nas disciplinas introdutórias de programação.

Neste sentido, desponta o questionamento que é o cerne do presente trabalho: *Como os educadores podem criar ambientes de aprendizagem mais engajadores cuja pedagogia fomente uma mentalidade incremental, encorajando a prática, valorizando o esforço e incorporando o erro como parte do processo de aprendizado?*

Há estudos que contemplam a apropriação ou o desenvolvimento de artefatos educacionais, e a proposição de métodos e práticas pedagógicas visando prover um ambiente de aprendizado mais efetivo, dentre estes considera-se que os jogos são os artefatos que mais se aproximam de fomentar uma mentalidade incremental, estimulando a prática e valorizando o esforço e o erro como parte do aprendizado, aspectos que são muito importantes em experiências introdutórias de programação.

No mais, muitos jogos possuem características consideradas necessárias para uma experiência de aprendizagem mais eficaz: os jogos iniciam fáceis, na maioria das vezes ensinando aos jogadores as habilidades necessárias para continuar e então tornam-se progressivamente mais complexos e desafiadores, provendo ao aprendiz a experimentação sob diferentes formas e níveis de complexidades crescentes, conforme sugere Guerra (2010).

Deste modo, o presente capítulo buscou, inicialmente, apresentar uma visão introdutória sobre os jogos enquanto parte de um fenômeno mais amplo: o pensamento de jogo ('game thinking') e como os jogos se manifestam de variadas formas e em diversos contextos. Os jogos e seus principais elementos estruturantes foram descritos visando oferecer ao leitor deste capítulo uma melhor compreensão do que são os jogos, quais são as suas principais características e como eles podem ser úteis em experiências educacionais, especialmente no processo de ensino e aprendizagem de programação.

Além disso, buscou-se traçar um panorama não exaustivo de possibilidades de integração dos jogos e abordagens baseadas em jogos no *design* de experiências de programação a partir do conceito de pensamento de jogo, assim como uma variedade de ferramentas e recursos de apoio ao professor que possam ser utilizados ou mesmo que

possam nortear a estruturação, ou ainda, o aperfeiçoamento de suas próprias iniciativas para o ensino de programação.

Finalmente, foram apresentados direcionamentos e discutidas algumas implicações sobre a incorporação dos jogos em contextos educacionais, no intuito de delimitar os benefícios e limitações oferecidas por esses recursos a fim de um melhor aproveitamento nas práticas pedagógicas.

Referências

- Alvarez, J., e Michaud, L. (2008). Serious games. Advergaming, edugaming, training and more. Montpellier, France: IDATE, 2008.
- Alves, F. (2014). “Gamification: Como criar experiências de aprendizagem engajadoras. Um guia completo: do conceito à prática”. DVS Editora.
- Andrade, D., Carvalho, T., Silveira, J., Cavalheiro, S., Foss, L. Fleischmann, A. M., Aguiar, M. e Reiser, R. Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental. In: Anais do Workshop de Informática na Escola. 2013.
- Aureliano, V. C. O. e Tedesco, P. C. A. R. (2012) “Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação”. In: XX Workshop sobre Educação em Computação, 2012, Curitiba. XXXII CSBC.
- Baeza-Yates, R. A. (1995). Teaching algorithms. ACM SIGACT News, v. 26, n. 4, p. 51-59.
- Barros, L., Ribeiro, S. P. S. e Oeiras, J. (2009). Projeto de Extensão Universitária para apoio e realização da Olimpíada Brasileira de Informática em Escolas. In: XXIX Congresso da SBC - XVII Workshop de Ensino de Computação, Bento Gonçalves.
- Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. Journal of MUD research, v. 1, n. 1, p. 19, 1996.
- Bezerra, F. e Dias, K. (2014) Programação de Computadores no Ensino Fundamental: Experiências com Logo e Scratch em Escola Pública. In: XXXIV Congresso da SBC - XXII Workshop de Ensino de Computação, Brasília.
- Bingimlas, K. A. Barriers to the successful integration of ICT in teaching and learning environments: A review of the literature. Eurasia Journal of Mathematics, Science & Technology Education, v. 5, n. 3, p. 235-245, 2009.
- Buckley, K. E., e Anderson, C. A. (2006) “A theoretical model of the effects and consequences of playing video games”. Playing video games: Motives, responses, and consequences, p. 363-378.
- Byrne, P., e Lyons, G. The effect of student attributes on success in programming. In: ACM SIGCSE Bulletin. ACM, 2001. p. 49-52.
- CAS - Computing At School Working Group (2012). Computer Science: A curriculum for schools. Disponível em: <http://bit.ly/1MNe9Mi>. Acessado em 27/07/2015.
- Caspersen, M. E. e Kölling, M. (2009) STREAM: A First Programming. Journal ACM Transactions on Computing Education (TOCE), v.9, n.1.

- Chiesa, B. della (2013) "Um ABC do cérebro". In: A Ciência do Aprendizado. Revista Educação. Edição Especial Neuroeducação. Vol 1. Editora Segmento.
- Chiesa, B. della (2014) "O cérebro em contínua construção". In: Vida Longa a seu Cérebro. Revista Educação. Edição Especial Neuroeducação. Vol 2. Editora Segmento.
- Davy, J., e Jenkins, T. Research-led innovation in teaching and learning programming. ACM SIGCSE Bulletin, v. 31, n. 3, p. 5-8, 1999.
- Deterding, S., Dixon, D., Khaled, R., e Nacke, L. From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments. ACM, 2011. p. 9-15.
- Dickey, M. D. Game design and learning: A conjectural analysis of how massively multiple online role-playing games (MMORPGs) foster intrinsic motivation. Educational Technology Research and Development, v. 55, n. 3, p. 253-273, 2007.
- Dweck, C.S.: Messages that motivate: How praise molds students' beliefs, motivation, and performance (in surprising ways). In: Improving academic achievement: Impact of psychological factors on education, pp. 37–60. Academic Press (2002).
- Frome, J. (2007) Eight Ways Videogames Generate Emotion. In: Proceedings of the 2007 DiGRA International Conference: Situated Play.
- Giraffa, L., Muller, L., e Moraes, M. C.. Ensino Programação apoiada por um ambiente virtual e exercícios associados a cotidiano dos alunos: compartilhando alternativas e lições aprendidas. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2015. p. 1330.
- Gomes, T. C. e Melo, J. C. App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2013a.
- Gomes, T. e Melo, J. O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended-Learning. In: Anais do XXI Workshop sobre Educação em Computação–XXXIII Congresso da Sociedade Brasileira de Computação. Maceió, AL–Brasil. 2013b.
- Goulart, F. (2016) "Ensino de programação e linguagens digitais ganha espaço e adeptos no Brasil", <http://cbn.globoradio.globo.com/editorias/tecnologia/2016/02/27/ENSINO-DE-PROGRAMACAO-ELINGUAGENS-DIGITAIS-GANHA-ESPACO-E-ADEPTOS-NO-BRASIL.htm>, Maio.
- Gove, M. (2012) Digital Literacy and the Future of ICT in Schools. BETT Show, Departamento de Educação.
- Henderson, P. (1987) Modern introductory computer science. In: ACM SIGCSE Bulletin. ACM, 1987. p. 183-190.
- Huggard, M. (2004). Programming trauma: can it be avoided. Proceedings of the BCS Grand Challenges in Computing: Education, 50-51.

- Huizinga, J. (2004) “Homo-ludens: o jogo como elemento da cultura”. 5 ed. São Paulo: Perspectiva.
- IFTF - Institute for the Future (2009). Everyone is a programmer: Making the World a Control System. When everything is programmable: Life in a Computational Age.
- Jenkins, T. On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences. 2002. p. 53-58.
- Juul, J. (2003) The game, the player, the world: looking for a heart of gameness. In Level up: digital games research conference proceedings. Utrecht University, <http://www.jesperjuul.net/text/gameplayerworld/>, Abril.
- Koliver, C., Dorneles, R. V., & Casa, M. E. (2004). Das (muitas) dúvidas e (poucas) certezas do ensino de algoritmos. In: XII Workshop de Educação em Computação.
- Kozma, R. B. Learning with media. Review of educational research, v. 61, n. 2, p. 179-211, 1991.
- Krendl, K. A., e Lieberman, D. A. Computers and learning: A review of recent research. Journal of Educational Computing Research, v. 4, n. 4, p. 367-389, 1988.
- Lahtinen, E.; Ala-Mutka, K. e Järvinen, H.-M. (2005) A Study of the Difficulties of Novice Programmers. Proceedings do 10th ITiCSE, Monte de Caparica, Portugal, p. 14-18.
- Logo Foundation (2015) "Logo History", http://el.media.mit.edu/logo-foundation/what_is_logo/history.html, Maio.
- Machado, E. Z. A., Vasconcelos, I. R., e Malta, K. (2010) Uma Experiência em Escolas de Ensino Médio e Fundamental para a Descoberta de Jovens Talentos em Computação. In: XXX Congresso da SBC - XVIII Workshop de Ensino de Computação, Bento Gonçalves.
- Malan, D. J.; Leitner, H. H. (2007) Scratch for budding computer scientists. Proceedings do 38th SIGCSE'07, Kentucky, USA, p. 223–227.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B. e Resnick, M. Scratch: A Sneak Preview. In: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing. IEEE Computer Society, 2004. p. 104-109.
- Marczewski A. (2014) “How to use Game Thinking”, <http://www.gamified.uk/2014/10/08/game-thinking-gettingtools/>, Maio.
- McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., e Mander, K. (2005). Grand challenges in computing: Education—a summary. The Computer Journal, v. 48, n. 1, p. 42-48.
- McGonigal, J.(2011) Reality is broken: Why games make us better and how they can change the world. Penguin.
- Melissinos, C. (2015) “Are Video Games Art? Video Games Are One of the Most Important Forms in History”. Question Everything, TIME. Disponível em: <http://time.com/4038820/chris-melissinos-are-video-games-art/>, Julho, 2015.

- Mendonza, M. (2015) “Australia Replaces History and Geography With Coding in New Primary School Curriculum”, <http://www.techtimes.com/articles/86669/20150921/australia-replaces-history-and-geography-with-coding-in-new-primary-school-curriculum.htm>, Março.
- Michael, D. R., e Chen, S. L.. Serious games: Games that educate, train, and inform. Muska&Lipman/Premier-Trade, 2005.
- Mishra, G. (2010) “Why Game Thinking is a Bigger Idea Than Game Mechanics”, <https://humancapitalleague.com/why-game-thinking-is-a-bigger-idea-than-game-mechanics/>, Maio.
- Monsalve, E. S. (2014) Uma Abordagem para Transparência Pedagógica usando Aprendizagem Baseada em Jogos. Tese de Doutorado. PUC-Rio.
- NRC - National Research Council (1999). Being Fluent with Information Technology. NationalAcademy Press: Washington DC.
- Oliveira, M. L. S., Souza, A. A., Ferreira, A. e Barbosa, E. F. S. B. (2014) Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In: XXXIV Congresso da SBC - XXII Workshop de Ensino de Computação, Brasília.
- Opusphere. (2013) “Os 4 perfis de jogadores segundo Richard Bartle”, <http://www.opusphere.com/os-4-perfis-de-jogadores-segundo-richard-bartle/>, Agosto, 2016.
- Papert, S., Watt, D., diSessa, A., e Weir, S. (1979). Final Report of the Brookline LOGO Project. Part III: Profiles of Individual Student's Work.
- Pereira Júnior, J. C. R. e Rapkiewicz, C. E. (2004). O processo de ensino-aprendizagem de fundamentos de Programação: uma visão crítica da pesquisa no Brasil. In: Anais do XII Workshop sobre Educação em Computação (SBC).
- Petry, L. C. “O conceito ontológico de jogo”. In: Jogos digitais e aprendizagem: Fundamentos para uma prática baseada em evidências/ Lynn Alves e Isa de Jesus Coutinho (orgs.) Campinas, SP: Papirus, 2016.
- Prensky, M. Aprendizagem baseada em jogos digitais. São Paulo: SENAC, p. 575, 2012.
- Raabe, A., Rodrigues, A. J., Santana, A. M., Vieira, M. V., Rosário, T. e Carneiro, A. C. R.. Brinquedos de Programar na Educação Infantil: Um estudo de Caso. In: Anais do Workshop de Informática na Escola. 2015. p. 42.
- Ranhel, J. (2009) “O conceito de jogos e jogos computacionais”, O mapa do jogo: a diversidade cultural dos games, L. Santaella e M. Feitoza, Brasil, Cengage-Learning, p.3-22.
- Sales, K.,Zimmerman, E. (2012) “Regras do Jogo: fundamentos do design de jogos: principais conceitos”. vol.1. São Paulo: Blucher.
- Scott, M. J., e Ghinea, G.(2013) Implicit theories of programming aptitude as a barrier to learning to code: are they distinct from intelligence? In: Proceedings of the 18th ACM conference on Innovation and technology in computer science education. ACM, 2013. p. 347-347.

- Silva, T. R., Medeiros, T. J., e Aranha, E. H. D. S. (2014) Jogos Digitais para Ensino e Aprendizagem de Programação: uma Revisão Sistemática da Literatura. In: Anais do Simpósio Brasileiro de Informática na Educação.
- Souza, S. M., Rios, M. S., Rodrigues, C. A., Santos, D. M. B. e Bittencourt, R. A. (2015) Oficinas de Programação com Ambientes Lúdicos para Meninas do Ensino Fundamental. In: XXXV Congresso da SBC - XXIII Workshop de Ensino de Computação, Recife.
- Stephenson, C., Cooper, S., Boucher Owens, B., e Gal-Ezer, J.. The new CSTA K--12 computer science standards. In: Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education. ACM, 2012.
- Tang, S., Hanneghan, M. and El Rhalibi, A. (2009) Introduction to Games-Based Learning, In Games-based Learning Advancement for Multisensory Human Computer Interfaces: Techniques and Effective Practices (Eds: T.M. Connolly, M.H. Stansfield and E. Boyle). Idea-GroupPublishing: Hershey.
- Teixeira, A., Martins, J. R., Batistela, F., Pazinato, A., e Oro, N. (2015). Programação de computadores para alunos do ensino fundamental: A Escola de Hackers. In: Anais do Workshop de Informática na Escola. 2015. p. 112.
- Teixeira, V. A. ReviewJourney. Tech Tudo. Disponível em: <http://www.techtudo.com.br/review/journey-ps4.html>, Julho.
- Veen, W., e Vrakking, B. Homo Zappiens: educando na era digital. Artmed Editora, 2009.
- Vianna, Y., Vianna, M., Medina, B. e Tanaka, S. Gamification, Inc - Como reinventar as empresas a partir dos jogos. 1ª Ed. Rio de Janeiro: MVJ Press, 2013. 116p.; e-book.
- Wangenheim, C. G. von, e Wangenheim, A. von. (2012). “Ensinando Computação com Jogos”.
- Wassila, D., e Tahar, B.. Using serious game to simplify algorithm learning. In: International Conference on Education and e-Learning Innovations (ICEELI). IEEE, 2012. p. 1-5.
- Wing, J. M. Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, v. 366, n. 1881, p. 3717- 3725, 2008.
- Wing, J. M. Computational thinking. Communications of the ACM, v. 49, n. 3, p. 33-35, 2006.
- Winslow, L. E. Programming pedagogy—a psychological overview. ACM SIGCSE Bulletin, v. 28, n. 3, p. 17-22, 1996.

Sobre os autores

Tancicleide Carina Simões Gomes

Mestranda em Ciência da Computação pela Universidade Federal de Pernambuco, bacharela em Sistemas de Informação pela Universidade Federal Rural de Pernambuco (2015). Educadora na escola Geração do Futuro, atua com o ensino de programação para alunos da Educação Infantil e Ensino Fundamental I usando jogos digitais. De 2011 a 2015, ensinou programação para crianças, adolescentes e jovens utilizando jogos digitais e linguagens visuais de programação através de projetos de extensão universitária. Neste mesmo período, participou do Programa de Iniciação Científica (PIC/CNPq), investigando a relação entre a criação de jogos educativos por usuários finais e a aprendizagem de conteúdos. Tem experiência com a concepção de artefatos educacionais para suporte à formação docente e na implantação de tecnologias em cenários de aprendizagem.

Patricia Cabral de Azevedo Restelli Tedesco

Possui graduação em Ciência da Computação pela Universidade Federal de Pernambuco (1994), mestrado em Ciência da Computação pela Universidade Federal de Pernambuco, na área de Inteligência Artificial Aplicada à Educação (1997) e doutorado em Ciência da Computação - University Of Leeds Computer Based Learning Unit (2001). Atualmente é professora adjunta do Centro de Informática - UFPE. Tem experiência na área de Ciência da Computação, com ênfase em Inteligência Artificial, atuando principalmente nos seguintes temas: Sistemas Multiagentes e Atores Sintéticos, Trabalho Colaborativo Apoiado por Computador, Contexto Computacional e Educação a Distância.

Jeane Cecília Bezerra de Melo

Bacharel em Matemática pela Universidade Federal de Pernambuco, mestre em Ciência da Computação pela Universidade Federal de Pernambuco e doutora em Ciência da Computação pela Universidade Federal de Pernambuco (2005). Professor Associado da Universidade Federal Rural de Pernambuco (UFRPE), tem experiência na área de Ciência da Computação, com ênfase em Análise de Algoritmos e Complexidade, atuando principalmente nos seguintes temas: Biologia Computacional, Estruturas de Proteínas e Informática na Educação. Desde junho de 2009, coordena a área de Computação no Programa Institucional de Bolsas de Iniciação à Docência (PIBID/UFRPE). Coordenou o projeto de extensão Aplicativos Educacionais para Dispositivos Móveis em 2012, e as duas edições do projeto de extensão Educação Sem Fronteiras ao longo de 2013 e 2014