

Descrição e Comparação de Jogos Digitais para Auxiliar no Ensino de Programação

Henrique Artur Cordeiro Gomes¹, José Monteiro Melo¹, Hyuan Peixoto Farrapo¹
Mara Franklin Bonates¹, Emanuel Ferreira Coutinho¹

¹ Instituto Universidade Virtual (UFC Virtual)
Universidade Federal do Ceará (UFC) – Fortaleza – CE – Brasil

{henriqueacg, monteiromelojose, hyuanpeixoto}@gmail.com

{marabonates, emanuel}@virtual.ufc.br

Abstract. *Computer Programming courses are usually taught in undergraduate curricula with the purpose of developing logical reasoning skills, ability to solve problems using ideas association, generalization and modularization. However, such courses are intrinsically difficult, usually associated with high failure rates and school dropout. One of the reasons for this is the lack of interest shown by the students in the course's related topics. One of the instructional strategies to facilitate and encourage learning is to use games for teaching. In this context, games have been used to aid in computer programming teaching. The objective of this work is to present some digital games used in programming teaching, consisting of a literature review of papers that reported programming teaching experiences.*

Resumo. *Disciplinas de programação de computadores são normalmente ministradas em cursos de graduação com o objetivo de desenvolver habilidades de raciocínio lógico, capacidade de resolver problemas utilizando associação, generalização e modularização de ideias. Entretanto, tais disciplinas são intrinsecamente difíceis, normalmente associadas a altos índices de reprovação e de evasão. Um dos motivos deste fato se deve ao desinteresse por parte dos alunos em relação ao conteúdo da disciplina. Uma maneira de facilitar e incentivar o aprendizado é a utilização de jogos para o ensino. Nesse contexto, jogos têm sido utilizados para auxiliar no ensino de programação. O objetivo deste trabalho é apresentar alguns jogos digitais utilizados no ensino de programação, consistindo de uma revisão documental de trabalhos que relatam experiências de ensino de programação.*

1. Introdução

Disciplinas que possuem a intenção de adequar teoria com prática são normalmente muito difíceis de se conduzir devido à necessidade de se tratar fatores técnicos, como linguagens de programação e ferramentas, e humanos, como comunicação e planejamento, tudo ao mesmo tempo [Coutinho et al. 2016]. Cada vez mais, profissionais da área de tecnologia da informação devem estar capacitados, e assim, faz-se necessário que o ensino de diversas disciplinas seja priorizado e transmitido mais eficientemente, não apenas em cursos de computação [Jorge et al. 2015].

Disciplinas de introdução à programação são usualmente ministradas para cursos de graduação em Engenharia e Ciências Exatas com o propósito de desenvolver nos aprendizes algumas habilidades e competências, tais como: raciocínio lógico, capacidade de resolver problemas utilizando associação, generalização, modularização, entre outras [Carvalho et al. 2016].

Disciplinas introdutórias de programação são intrinsecamente difíceis, comumente associadas a altos índices de reprovação e de evasão, sendo necessário lidar com tais dificuldades [Aureliano et al. 2016]. Desse modo, é tarefa do professor de programação organizar atividades de ensino e de aprendizagem que auxiliem os estudantes na aquisição do conhecimento necessário. Além disso, linguagens de programação são intrinsecamente complexas, devido à variabilidade, generalidade e completude de suas construções. Isso justifica a adoção de pseudolinguagens de alto nível no ensino de Lógica de Programação e Algoritmos [Barbosa et al. 2016].

Diversos casos na literatura relatam experiências em disciplinas de programação [Charão et al. 2016] [da Silva et al. 2016] [Raeder et al. 2016] [Francisco et al. 2017] [Netto et al. 2017]. Muitas situações são relacionadas a desmotivação e problemas no aprendizado. Em um caso real analisado em uma disciplina inicial de programação, do curso de graduação em Sistemas e Mídias Digitais da Universidade Federal do Ceará, observou-se que ela contém um dos maiores índices de reprovação do curso, tornando-se um motivo para a evasão dos alunos da graduação [de Oliveira et al. 2017]. Uma das causas deste acontecimento deve-se ao problema de desinteresse por parte dos alunos em relação ao conteúdo da disciplina, tornando o aprendizado ainda mais difícil.

Uma maneira de facilitar e incentivar o aprendizado é a utilização de jogos para o ensino. Nesse contexto, jogos têm sido utilizados para ajudar no ensino de diversas áreas do conhecimento [Sousa et al. 2012]. Assim, com o objetivo de auxiliar a simulação de conceitos teóricos, jogos têm sido propostos na literatura [Bezerra e Coutinho 2013]. E assim, a utilização de jogos digitais se torna uma boa estratégia para o ensino de lógica e linguagens de programação. Além disso, neste cenário, jogos educativos de maneira geral são interessantes para auxiliar na aprendizagem devido ao fato que estes simulam situações reais de projetos de software [Bezerra et al. 2014].

A utilização de recursos interativos no andamento da disciplina pode servir tanto como um método de aprendizado, quanto para fixação de conteúdo, tornando-se um objeto de acompanhamento interessante, percebendo-se a progressão do aluno durante as etapas do recurso escolhido. Segundo [Rapkiewicz et al. 2016], o uso de jogos de forma lúdica, propicia flexibilidade e criatividade, fazendo o aluno explorar, pesquisar e encorajar o pensamento criativo.

O objetivo deste trabalho é apresentar alguns jogos digitais, com breves comentários e características, para facilitar a seleção ou simplesmente para o conhecimento de ferramentas de jogos digitais que ajudam no ensino de programação. O trabalho consistiu de uma revisão documental de trabalhos que relataram experiências de ensino de programação. Deixamos claro que neste trabalho apenas nove ferramentas foram apresentadas, mas a gama de opções é muito maior e a cada dia que passa novos jogos para esse fim são produzidos, para as mais variadas plataformas.

Este trabalho está dividido nas seguintes seções, além desta introdução: trabalhos relacionados, com alguns relatos e pesquisas sobre o ensino de programação com jogos; a metodologia aplicada à pesquisa; uma breve descrição de alguns jogos digitais utilizados no ensino de programação; e por fim, as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Diversos trabalhos na literatura já discutiram sobre o ensino de programação, tanto em níveis de ensino médio, para crianças e adolescentes, quanto para o ensino superior nos mais diversos cursos de graduação. Alguns propuseram e descreveram ferramentas, outros compararam. Entretanto, percebe-se de maneira geral uma dificuldade na maneira de ensinar, no real aprendizado e como melhorar aulas.

Um trabalho relacionado à aprendizagem de programação por meio de jogos foi o descrito por [Ouahbi et al. 2015]. Basicamente, ele consistiu em separar 3 grupos de alunos do ensino médio de uma escola. Cada grupo seguiria uma metodologia diferente. O primeiro grupo usaria a ferramenta Scratch¹ para aprender programação, juntamente com computadores e projetor em sala de aula. O segundo grupo utilizaria métodos convencionais de aprendizagem em programação com a linguagem Pascal. Por fim, o terceiro grupo também utilizaria métodos convencionais, mas usaria computadores e projetor em sala. O resultado foi que 85% dos alunos do primeiro grupo não acharam desinteressante o aprendizado e 90% pensaram ou tentaram fazer um jogo. Em comparação aos outros dois grupos, apenas 20.7% dos alunos gostaram de aprender programação sendo que 37.9% do total de alunos dos sub-grupos que aprenderam de forma convencional tentaram criar seus próprios programas. Outro dado bastante importante é que 65% alunos do primeiro grupo disseram que continuariam os estudos de programação, enquanto que apenas 10.3% dos outros dois grupos disseram que continuariam.

[Aureliano et al. 2016] discutiram alguns dos desafios enfrentados pelos professores no planejamento e na execução de atividades de ensino e de aprendizagem, tais como: professor deve propor atividades de ensino e professor também deve propor atividades de aprendizagem. Além disso, também apresentaram algumas oportunidades de pesquisa na área de educação em programação para alunos iniciantes e uma proposta de abordagem para minimizar alguns dos desafios discutidos.

[Astolfi e Junior 2016] relataram uma intervenção, com ênfase na prática docente, que prioriza uma abordagem de ensino conduzida por meio da aprendizagem significativa, na qual o estudante desenvolve o processo de aprendizagem tendo como base seu próprio conhecimento prévio. Na abordagem, foi proposto identificar os conhecimentos prévios dos estudantes, em relação ao tema linguagem de programação de computadores, analisá-los, amenizar as deficiências conceituais e ao final introduzir o novo conhecimento por meio de desenvolvimento de um projeto de software ao longo da unidade curricular. Os resultados apresentados sugeriram a necessidade de uma constante reflexão e revisões das práticas pedagógicas e suas implicações, nas quais tanto professores quanto alunos compartilham o protagonismo do processo de ensino e aprendizagem.

[Carvalho et al. 2016] trabalharam a dificuldade que alunos possuem em não se motivarem na programação. Para isso, uma equipe de professores desenvolveu uma metodologia de ensino-aprendizagem que considera o conjunto de disciplinas que estão sendo

¹Scratch - MIT Media Lab - <https://scratch.mit.edu/>

ministradas em um mesmo período letivo. A metodologia foi aplicada em quatro turmas de cursos e professores diferentes, no primeiro período letivo de 2015. Os resultados preliminares indicaram um índice de aprovação maior do que em turmas lecionadas no mesmo período onde a metodologia não foi aplicada. A metodologia foi avaliada pelos discentes onde em sua maioria aprovaram a dinâmica de aplicação e as atividades propostas.

[Charão et al. 2016] propuseram um padrão pedagógico dedicado ao ensino de programação, para cursos de nível superior relacionados a Computação. Este padrão é orientado na exposição e discussão de bons e maus exemplos de códigos produzidos pelos alunos, gerando o que foi denominado respectivamente de “*Hall of Fame*” e “*Hall of Shame*”. Os resultados indicaram uma avaliação positiva do padrão pelos alunos e revelaram sua eficácia no incentivo de boas práticas de programação.

[Raeder et al. 2016] relataram uma experiência diferenciada de ensino de lógica, programação e matemática para Computação, onde os sujeitos envolvidos são alunos da graduação. Formou-se um grupo de estudos denominado L2PM, que teve três professores como mediadores e ocorreu em período extraclasse. Os encontros do grupo contemplaram em um mesmo ambiente aulas integradas das disciplinas em questão, realizadas em sessões de 15 minutos, alternadas entre os professores e temas, e finalizando com uma sessão que serve de integração, na qual se reuniram os conteúdos tratados no grupo. O resultado desta experiência apontou que a dinâmica adotada no L2PM possibilitou outras condições para a construção de conhecimento.

[da Silva et al. 2016] apresentaram a elaboração de um jogo sério para o ensino de programação orientada a objetos, denominado POOGame. O desenvolvimento do jogo foi baseado em jogos de RPG (*Role-playing game*), de modo que o jogador controla um personagem, e a partir de batalhas é possível se invocar criaturas e controlar suas ações utilizando comandos baseados na linguagem JAVA. O jogo foi elaborado e avaliado no contexto da disciplina de Introdução ao Desenvolvimento de Jogos de um curso de Engenharia de Software, adotando uma metodologia para o desenvolvimento de jogos. Os resultados da avaliação do POOGame indicaram evidências de que o jogo contribui para o aprendizado de programação orientada a objetos.

[Ferreira et al. 2016] descreveram a experiência de um curso de programação, voltado para jovens e adultos sem contato prévio com programação, objetivando despertar o raciocínio lógico e computacional. Utilizou-se como recursos para apoiar o estudo o Scratch, o VisuAlg, o ambiente virtual de aprendizagem Moodle e o Facebook e WhatsApp para comunicação. Considerando a avaliação do curso, a utilização de dinâmicas e atividades práticas associadas à ludicidade colaboraram à aprendizagem dos estudantes, ao passo que tornam os estudantes mais confiantes e confortáveis no decorrer do curso. Destaca-se a adição de um projeto final para construção de um jogo motivou os estudantes e autônomos na construção do aprendizado.

[Godinho et al. 2017] apresentaram o projeto Aprenda a Programar Jogando e a experiência resultante das atividades nele desenvolvidas. O projeto tem como objetivo introduzir o universo da programação computacional a crianças e jovens, além de incentivá-los a serem criadores de tecnologia. Nesse trabalho, ferramentas selecionadas para o ensino da programação foram apresentadas. Também apresentou-se a estrutura de ativida-

des realizadas tanto nas oficinas quanto nos minicursos, e o plano de atividades realizados nos minicursos. Como resultado, esse projeto já beneficiou 298 estudantes e aproximadamente 90% classificaram a experiência com a programação como excelente ou ótima.

[da Silva Junior e França 2017] analisaram e compararam algumas ferramentas propostas em trabalhos anteriores e que são utilizadas no ensino de programação, buscando observar se essas ferramentas atendem a realidade do ensino inicial de programação para crianças e jovens. Dentre os softwares analisados tem-se: Feeper, WH-IDE, TutorICC, Portugol Studio, AAPW, TSTView, The Huxley, SOAP e Scratch. Observou-se que, embora alguns dos projetos encontrados na literatura sejam extremamente exitosos, ainda se pode notar algumas deficiências. Este trabalho apresentou ferramentas indicadas para iniciantes em programação, e dentre as ferramentas discutidas, a que mais se pareceu adequada para uma aprendizagem inicial de programação foi a Portugol Studio.

[Silva et al. 2017] apresentaram o ProgressCode, uma ferramenta desenvolvida com a intenção de facilitar o registro e apresentação de observações realizadas por tutores durante atividades de ensino não-formal de programação. A ferramenta é um aplicativo móvel para coleta de dados e um sistema *web* que possibilita a visualização de informações sobre o progresso de cada participante. O ProgressCode foi testado durante oficinas de programação, nas quais a ferramenta mostrou cumprir seus objetivos.

[Netto et al. 2017] desenvolveram um trabalho para auxiliar na estruturação do pensamento cognitivo aliado à resistência implícita do aprendizado de algoritmos. Para isso, apresentou-se o Game Logic, um jogo voltado para o ensino-aprendizado de algoritmos por meio da programação em blocos, partindo do princípio cognitivo de que imagens são melhores memorizadas, tendo seu uso devidamente comprovado. Neste trabalho verificou-se que aspectos fundamentais para a construção de jogos educacionais devem ser seguidos, e a correta aplicação dos conceitos durante o desenvolvimento de qualquer ferramenta voltada público gera resultados positivos.

Na pesquisa proposta neste artigo, não nos propomos a comparar jogos digitais, nem pesquisas de outros autores, sendo esta análise considerada para trabalho futuro. Identificamos como foco desta pesquisa a identificação de trabalhos que utilizaram jogos digitais, e apresentamos alguns desses jogos ou ferramentas, para facilitar o ensino de programação.

3. Metodologia

Este artigo é uma pesquisa documental de natureza aplicada e objetivo descritivo. O material utilizado foi resultado da procura pelas palavras chave “Ensino”, “Programação” e “Jogos” no Google Scholar² e Google Search³. Como resultado dessa busca, algumas listas de jogos que ensinam programação e ferramentas relacionadas foram encontradas. Destas listas, os jogos que apareciam com maior frequência e que passavam maior quantidade de conteúdo foram selecionados para compor esse trabalho.

Cada jogo selecionado foi testado. As fases iniciais de cada jogo foram verificadas de modo a entender suas mecânicas principais e como a ferramenta introduzia o usuário iniciante no ambiente. Após o teste de cada jogo, foram levadas em consideração a facili-

²Google Scholar - <https://scholar.google.com.br/>

³Google - <https://www.google.com.br/>

Tabela 1. Jogos com foco no ensino de programação

Jogo	Língua Portuguesa	Idade Indicada	Plataformas	Gratuidade
Algo Bot	Não	+9	Windows	Não
Blockly Games	Sim	+9	Web	Sim
CodeCombat	Sim	+13	Web	Apenas as primeiras fases são gratuitas
CodinGame	Não	+13	Web	Sim
CodeMonkey	Parcialmente	+9	Web	Não
Humans Resource Machine	Sim	+10	Windows, Linux, OS X, Android, iOS, Wii U, Nintendo Switch	Não
Lightbot	Sim	+9	Web, iOS, Android	Web: Sim, iOS e Android: Não
Scratch	Sim	+5	Web, Android, iOS	Sim
Tynker	Não	+3	Web, iOS	Não

dade de compreensão, de aprendizado e intuitividade presente neles, destacando os jogos que fossem melhores para iniciantes. Os jogos resultantes foram separados e descritos de acordo com suas características: idioma, idade indicada, plataformas e gratuidade. A Tabela 1 exibe os dados das características identificados. Depois de analisados e avaliados, breves resumos do funcionamento e particularidades de cada jogo foram escritos, com a extração de algumas telas para se ter uma visão geral do jogo.

4. Jogos Digitais para o Ensino de Programação

Nesta seção, nove jogos para o ensino de programação serão brevemente descritos, com algumas características funcionais e técnicas.

4.1. Algo Bot

O Algo Bot⁴, pertencente a Fishing Cactus, é um jogo de quebra-cabeças onde o jogador assume o papel do operador e utiliza uma linguagem de programação visual para emitir uma sequência de comandos. Mesmo não sendo um jogo com fins educativos, possui mecânicas que treinam o raciocínio em lógica de programação. O usuário controla um robô que deve executar comando em forma de quebra-cabeças em uma sala, assim ganhando acesso às próximas. Conceitos básicos de programação, como gerenciamento de variáveis, funções e recursividade são alguns dos assuntos abordados. Um exemplo de tela do jogo pode ser visto na Figura 1.

4.2. Blockly Games

O Blockly Games⁵ é ligado a uma tecnologia do Google para programação visual, tornando-a mais simplificada para pessoas que são novas na área da programação, com foco em educação. Possui sete dinâmicas, todas utilizando a mesma mecânica de blocos, porém com regras diferentes em cada um, evidenciadas por mudanças nas instruções. Há uma progressão entre a dificuldade dos jogos, aumentando a complexidade deles na medida que o usuário vai completando as fases e a sequência sugerida pelo *site*. Ele possui um tutorial em cada jogo para guiar o usuário sobre novos comandos e a lógica que deve ser aplicada entre eles. Suas linguagens suportadas são: Linguagem Visual e Javascript. A Figura 2 exibe uma tela do jogo.

⁴Algo Bot - Fishing Cactus - <http://www.algo-bot.com>

⁵Blockly Games - Google - <https://blockly-games.appspot.com>



Figura 1. Algo Bot, Fishing Cactus

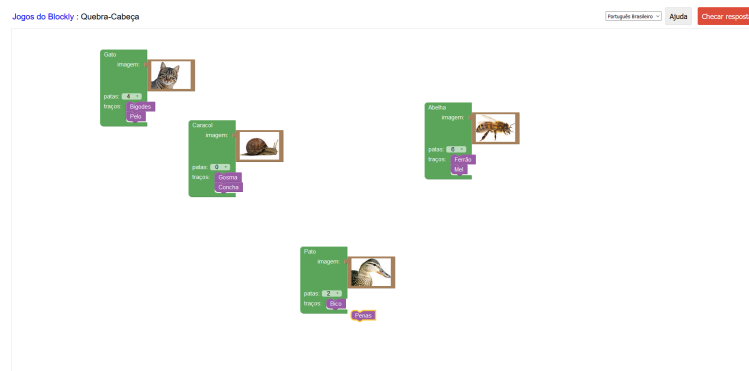


Figura 2. Blockly Games, Google

4.3. CodeCombat

O CodeCombat⁶ funciona com elementos de um RPG, em que o usuário escolhe uma classe que possui características únicas, como vida, velocidade, ataque e dificuldade de usar o personagem. É uma plataforma educativa com foco aprendizagem de lógica e conceitos de programação e codificação com ênfase no público infantil. As mecânicas vão de atravessar uma área a derrotar inimigos usando alguma das 4 linguagens disponíveis, ou até mesmo enfrentar amigos em uma batalha de programadores. Alguns equipamentos, que podem ser adquiridos comprando com as moedas do jogo, aumentam os status do seu personagem e outros te dão mais possibilidades de funções e procedimentos. O jogo é dividido em ilhas, sendo que cada uma possui um foco em algum aspecto de programação. Existem três tipos de perfis no jogo: Professor - esse que cria uma turma, onde alunos podem se juntar para seu progresso ser acompanhado; Aluno - suas informações de progresso são repassadas ao professor; e o Avulso - no qual o usuário joga sem vínculo algum. Possui fórum de discussão para a troca de conhecimento e tirar dúvidas. Possui suporte para as seguintes linguagens: Python, Javascript, CoffeeScript e Lua. Uma tela do jogo pode ser visualizada na Figura 3.

⁶CodeCombat - CodeCombat Inc. - <https://codecombat.com/>

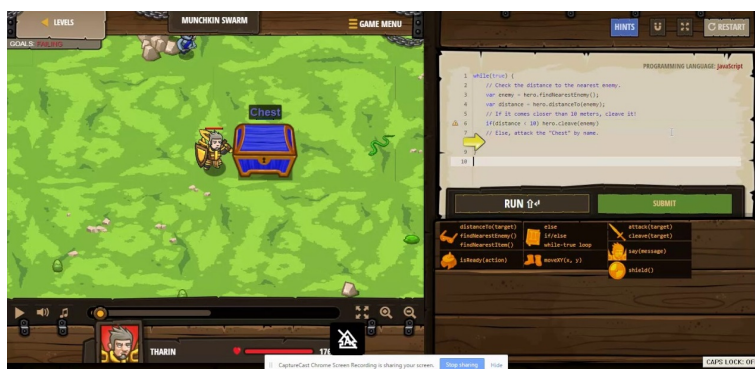


Figura 3. CodeCombat, CodeCombat Inc.

4.4. CodinGame

O CodinGame⁷ é uma plataforma gamificada que pode ajudar bastante a quem está começando a aprender programação, ainda mais se tiver uma orientação de um professor ou um tutor. Essa plataforma de estudos tem como base treinar os conceitos básicos de programação já em código. Uma ferramenta que auxilia são seus *chats* em tempo real com os outros jogadores, inclusive com uma área para português brasileiro, incentivando a troca de conhecimentos. Outra ferramenta que ajuda bastante é uma espécie de *debug*, que mostra as ações que estão sendo executadas, assim facilitando a compreensão do código criado pelo jogador. A ação de “debugar” é o processo de encontrar e reduzir defeitos em um programa. Há também um modo que simula uma batalha entre os jogadores, com o ganhador sendo o que conseguir cumprir o maior número de tarefas no tempo estabelecido ou acabar as tarefas primeiro. O jogo suporta as seguintes linguagens: Java, Python, Kotlin, Scala, Javascript, Swift, Ruby, Rust, C#, C++, C, Groovy, Clojure, Perl, Haskell, Go, PHP, Bash, Lua, Objective C, Dart, VB.Net, F#, OCaml e Pascal. A Figura 4 exibe uma tela do jogo.

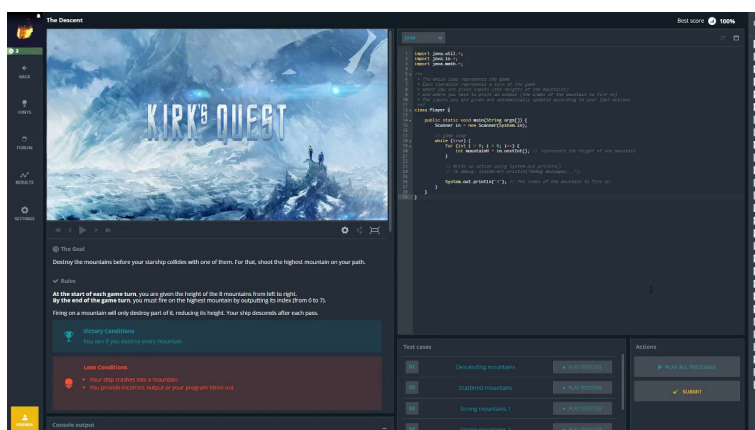


Figura 4. CodinGame, CodinGame

⁷CodinGame - CodinGame - <https://www.codingame.com/>

4.5. CodeMonkey

CodeMonkey⁸ consiste em fazer um macaquinho ou outros animais pegar bananas, enfrentando desafios lógicos, dando comando por meio de codificação. O jogo é uma plataforma que tem foco na aprendizagem educacional de lógica e conceitos de programação e codificação com ênfase no público infantil. Existem três tipos de Perfis no jogo, assim como no CodeCombat: Professor - esse que cria uma turma, onde alunos podem se juntar para seu progresso ser acompanhado; Aluno - suas informações de progresso são passadas ao professor; e Avulso - que o usuário joga sem vínculo nenhum. A Figura 5 exhibe uma tela do jogo.

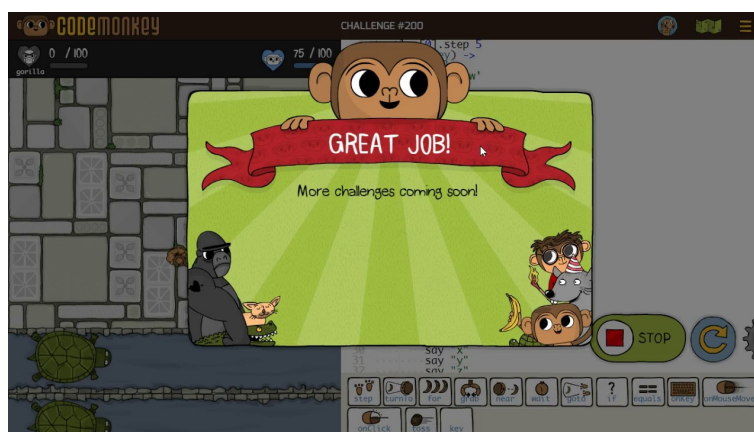


Figura 5. CodeMonkey, CodeMonkey Studios inc.

4.6. Human Resource Machine

O Human Resource Machine⁹, pertencente a Tomorrow Corporation, mesmo que sem um foco educativo, ensina conceitos básicos de programação ao usuário por meio de uma linguagem visual de arrastar blocos, como comandos de entrada e saída, alguns operadores lógicos, importância de laços de repetição. Ele é um jogo de quebra-cabeças, onde em cada nível seu chefe lhe dá um emprego e o jogador deve automatizá-lo programando. Ao obter sucesso, será promovido para o próximo nível. O jogo acaba pecando em fases mais complexas, pois possui uma estrutura similar às das primeiras linguagens de programação como “go to line” o que pode acabar sendo muito frustrante pra quem está começando a aprender. A Figura 6 exhibe uma tela do jogo.

4.7. Lightbot

O LightBot¹⁰ é um jogo de quebra-cabeça baseado em codificação, que ensina lógica de programação enquanto se joga. Neste jogo, o jogador é um robô que deve executar diversas operações. O objetivo do Lightbot é comandar um pequeno robô para navegar em um labirinto e acender lâmpadas. O jogador deve organizar símbolos disponíveis na tela para comandar o robô a andar, girar, pular, acender uma lâmpada e assim por diante. O labirinto e a lista de símbolos se tornam mais complicados à medida que as lições progridem. Ao usar esses comandos, os jogadores aprendem conceitos de programação como laços de

⁸CodeMonkey - CodeMonkey Studios inc. - <https://www.playcodemonkey.com>

⁹Human Resource Machine - <https://tomorrowcorporation.com/humanresourcemachine>

¹⁰Lightbot - Lightbot Inc. - <http://lightbot.com>

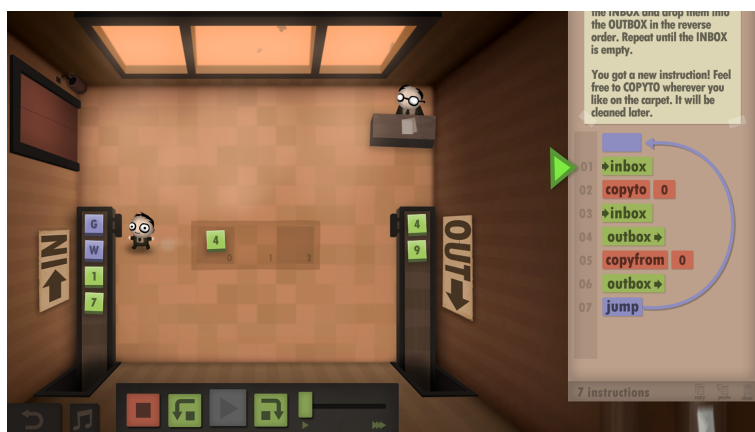


Figura 6. Humans Resources Machine, Tomorrow Corporation

repetição e procedimentos, sem inserir código em nenhuma linguagem de programação. Ele é uma plataforma com foco em aprendizagem de lógica de programação com linguagem visual para crianças, possuindo alguns conceitos básicos, como procedimentos e laços de repetição. Uma tela do jogo pode ser visualizada na Figura 7.

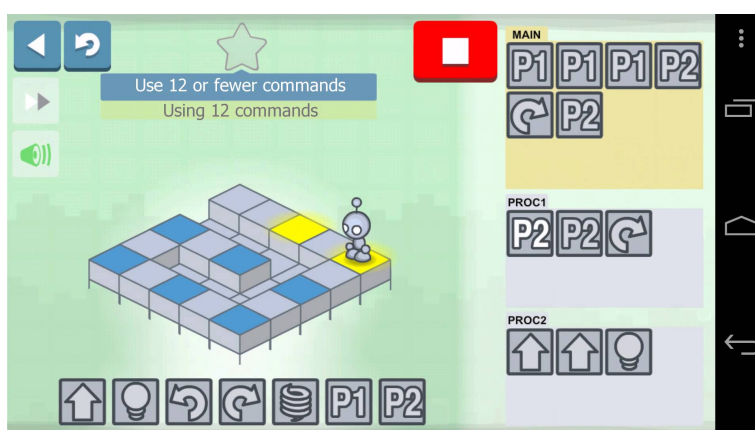


Figura 7. Lightbot, Lightbot Inc.

4.8. Scratch

O Scratch a princípio não é um jogo, mas uma aplicação que gera jogos que é amplamente utilizado para ensino de programação em várias partes do mundo. A Ferramenta utiliza uma diagramação de maneira similar ao Blockly, sendo um pouco mais lúdico que o mesmo e mais voltado a crianças a partir dos 8 anos de idade. É uma ferramenta utilizada para elaborar diversos elementos como animações, jogos, artefatos visuais interativos, voltada para quem está iniciando no mundo da programação por meio de uma linguagem visual. A plataforma contempla laços de repetição, procedimentos entre outros conteúdos base de programação. Há uma versão voltada para crianças de 5 a 7 anos de idade chamada Scratch Jr. A Figura 8 exibe uma tela do jogo.

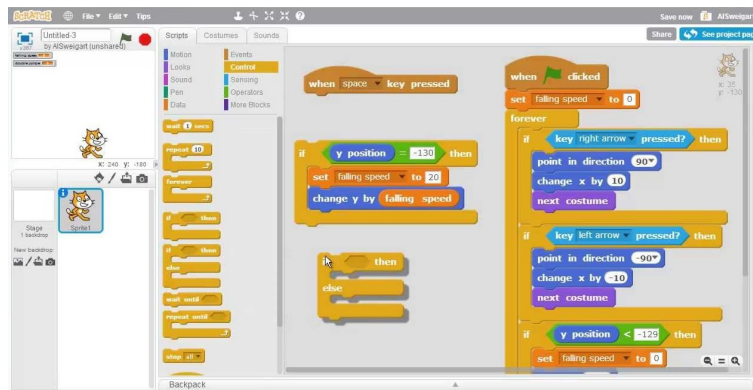


Figura 8. Scratch, MIT Media Lab Lifelong Kindergarten Group

4.9. Tynker

O Tynker¹¹ é uma plataforma educativa de computação criativa na qual milhões de crianças aprenderam a programar e criar jogos e aplicativos, como também oferecem jogos que passam conhecimentos de programação. Ele oferece cursos *on-line* individualizados para as crianças aprenderem codificação em casa, bem como um programa de programação para as escolas. Ele também possui módulos de ensino usando programação para jogos, drones, e para o jogo Minecraft. Além disso, os usuários podem postar seus jogos na plataforma. Focado em transmitir de forma lúdica conceitos de programação para crianças por meio de *mini-games*, sendo alguns dos personagens a Barbie, Hot Wheels, entre outros. Existem três tipos de perfis no jogo: Professor - esse que cria uma turma, onde alunos podem se reunir para seu progresso ser acompanhado; Aluno - suas informações de progresso são passadas ao professor; e Pais - por meio do qual os pais acompanham o progresso dos filhos. Linguagem Suportadas: Linguagem Visual, Python e Javascript. Uma tela do jogo pode ser visualizada na Figura 9.

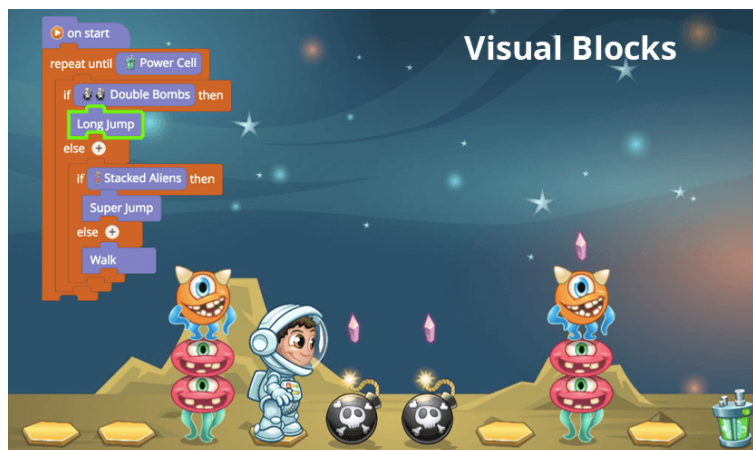


Figura 9. Tynker, Neuron Fuel, Inc.

5. Conclusão

Programar para quem está iniciando é muitas vezes complicado. Em vista dessa dificuldade, a utilização de mecanismos que facilitem a comunicação e o aprendizado sempre

¹¹Tynker - Neuron Fuel, Inc. - <https://www.tynker.com/>

são úteis tanto para professores quanto para os alunos. Este trabalho procurou apresentar alguns jogos digitais para o ensino de programação, fazendo um levantamento de diversas ferramentas e selecionando nove delas para apresentar e descrever. Desta forma, é importante ressaltar que a lista que apresentamos não é exaustiva, dado que a quantidade existente de jogos para o ensino de programação é muito maior, e a cada dia um novo jogo é produzido.

Uma das maiores dificuldades do trabalho foi, após jogar e ler sobre todos os jogos testados, decidir quais jogos deveriam entrar primeiramente para a lista que este trabalho apresenta, como também decidir quais aspectos deveríamos selecionar em nossa análise.

Um aspecto que vale a pena ressaltar é a correta identificação da ferramenta a ser utilizada em sala de aula, e como ela deve ser aplicada. Não necessariamente só por ser um jogo o aprendizado ocorrerá. A ferramenta também deve ser adequada ao nível escolar e cultural dos alunos. Alunos de ensino fundamental, médio e de graduação possuem perfis diferentes, então faz sentido selecionar ferramentas ou jogos adequadas ao nível escolar.

Percebe-se que ensinar programação mesmo com ferramentas lúdicas e variadas ainda é uma tarefa difícil. Muitos aspectos educacionais estão envolvidos, como ritmo, experiência, metodologia, aprendizado, dentre outros, e todos devem ser cuidadosamente planejados para que as aulas tenham efetividade.

A partir deste trabalho, com a descrição e comparação de algumas ferramentas para o ensino de programação com jogos, espera-se que professores possam identificar e selecionar com mais facilidade alguma ferramenta para aplicação em sala de aula. Além disso, com a descrição breve de cada jogo, também espera-se que este trabalho possa divulgar a quem se interessar por conhecer alguma ferramenta para o ensino de programação com jogos.

Como trabalhos futuros que essa pesquisa pode proporcionar, uma ideia seria a identificação e inclusão de mais jogos digitais para o ensino de programação, e sua descrição, para a criação de um repositório de jogos, facilitando ainda mais a busca e seleção. Outro trabalho seria avaliar a efetividade do aprendizado dos alunos em programação por meio de jogos digitais, e comparar com turmas onde jogos não foram utilizados. Mais um trabalho seria após uma análise das características dos jogos apresentados e o comportamento dos alunos que os utilizaram em sala, desenvolver um jogo educacional que melhore mais essas qualidades e auxilie ainda mais o professor em sala de aula.

Referências

- Astolfi, G. e Junior, D. L. (2016). Ensino de linguagem de programação com Ênfase na aprendizagem significativa. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Aureliano, V. C. O., de Azevedo Restelli Tedesco, P. C., e Giraffa, L. M. M. (2016). Desafios e oportunidades aos processos de ensino e de aprendizagem de programação para iniciantes. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Barbosa, L. L., Couto, C. M. S., e Terra, R. (2016). Portucol: uma pseudolinguagem inspirada em c++ para o ensino de lógica de programação e algoritmos. In *XXIV*

Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação.

- Bezerra, C. I. M. e Coutinho, E. (2013). Avaliação do jogo itestlearning: Um jogo para o ensino de planejamento de testes de software. In *XXI Workshop sobre Educação em Computação (WEI2013)*.
- Bezerra, C. I. M., Coutinho, E. F., Santos, I. S., Monteiro, J. M., e Andrade, R. M. C. (2014). Evolução do jogo itestlearning para o ensino de testes de software: Do planejamento ao projeto. In *XIX Conferência Internacional sobre Informática na Educação (TISE2014)*, Fortaleza.
- Carvalho, L. S. G., Gadelha, B. F., Nakamura, F. G., Oliveira, D. B. F., e Oliveira, E. H. T. (2016). Ensino de programação para futuros não-programadores: Contextualizando os exercícios com as demais disciplinas de mesmo período letivo. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Charão, A. S., Neto, A. F. K., de O. Stein, B., e de A. Barcelos, P. P. (2016). Hall of fame/shame: um padrão pedagógico para o ensino de programação. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Coutinho, E. F., Gomes, G. A. M., e Júnior, A. J. M. L. (2016). Applying design thinking in disciplines of systems development. In *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, pages 1–8.
- da Silva, L. M. P., Bonfim, B. C., Silva, R. C., da Silva, J. B., nad Carla I. M. Bezerra, W. L. M., e Jucá, P. M. (2016). Poogame: Um jogo sério para o ensino de programação orientada a objetos. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- da Silva Junior, S. M. e França, S. V. A. (2017). Programação para todos: Análise comparativa de ferramentas utilizadas no ensino de programação. In *XXV Workshop sobre Educação em Computação (WEI2017) - XXXVII Congresso da Sociedade Brasileira de Computação*.
- de Oliveira, B. P., Balan, G. R., Leitao, P. R. M. B., e Coutinho, E. F. (2017). Identificação e discussão de problemas nas disciplinas iniciais de programação do curso de graduação sistemas e mídias digitais. *Revista Sistemas e Mídias Digitais (RSMD)*, 2(1).
- Ferreira, A. C., Santos, J., Silva, R., Oliveira, A. T., Zabot, D., Santos, D. A., e de Souza Matos, E. (2016). Hello world: relato de experiência de um curso de iniciação à programação. In *II Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg2016) - Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (CBIE 2016)*.
- Francisco, R. E., Júnior, C. X. P., e Ambrósio, A. P. (2017). Grau de dificuldade de problemas de programação introdutória: Uma revisão sistemática da literatura. In *XXV Workshop sobre Educação em Computação (WEI2017) - XXXVII Congresso da Sociedade Brasileira de Computação*.

- Godinho, J., Torres, K., Batista, G., Andrade, E., e Gomide, J. (2017). Projeto aprenda a programar jogando: Divulgando a programação de computadores para crianças e jovens. In *XXV Workshop sobre Educação em Computação (WEI2017) - XXXVII Congresso da Sociedade Brasileira de Computação*.
- Jorge, F. F., Bezerra, C. I. M., Coutinho, E. F., Monteiro, J. M., e Andrade, R. M. C. (2015). A evolução do jogo itestlearning para o ensino das atividades de execução de testes de software. In *XX Conferência Internacional sobre Informática na Educação (TISE 2015) - Nuevas Ideas en Informática Educativa TISE 2015*.
- Netto, D., Medeiros, L. M., de Pontes, D., e de Moraes, E. (2017). Game logic: Um jogo para auxiliar na aprendizagem de lógica de programação. In *XXV Workshop sobre Educação em Computação (WEI2017) - XXXVII Congresso da Sociedade Brasileira de Computação*.
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., e Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia - Social and Behavioral Sciences*, 191:1479 – 1482. The Proceedings of 6th World Conference on educational Sciences.
- Raeder, M., Py, M., Rigo, S., e Pinheiro, J. (2016). L2pm: relato de uma experiência sobre o ensino integrado de lógica, programação e matemática para computação. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Rapkiewicz, C. E., Falkembach, G., Seixas, L., dos Santos Rosa, N., da Cunha, V. V., e Klemann, M. (2016). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *Revista Renote*, 4(2).
- Silva, L. F., Corrêa, I. C., Charão, A. S., e Lima, J. V. F. (2017). Progresscode: uma ferramenta para monitorar o progresso de alunos no ensino não-formal de programação. In *XXV Workshop sobre Educação em Computação (WEI2017) - XXXVII Congresso da Sociedade Brasileira de Computação*.
- Sousa, V., Bezerra, C. I. M., Coutinho, E., e Santos, I. S. (2012). itest learning: Um jogo para o ensino do planejamento de testes de software. In *V Fórum de Educação em Engenharia de Software (FEES 2012)*, Natal - RN.