

## Relato sobre o Uso de uma Ferramenta de Desenvolvimento de Jogos para o Ensino Introdutório de Lógica de Programação

Emanuel F. Coutinho<sup>1,2</sup>, Mara F. Bonates<sup>1</sup>, Leonardo O. Moreira<sup>1,2</sup>

<sup>1</sup>Instituto Universidade Virtual (UFC Virtual)

Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

<sup>2</sup>Grupo de Pesquisa em Computação em Nuvem e Sistemas (IBITURUNA)

Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

{emanuel, marabonates, leoomoreira}@virtual.ufc.br

**Abstract.** *Teaching programming logic and consequently coding are tasks that require a substantial amount of effort from the professors. In addition, helping students to associate theory with practice is a complex task, since students usually enter undergraduate courses with the most diverse capabilities and talents. A common difficulty among teachers is how to teach programming logic so that the student is motivated to learn, while getting prepared to use a coding tool and applying the concepts in practice. The objective of this work is to report the experience of using a specific tool for the development of digital games (Game Maker) in the teaching of programming logic in an undergraduate course.*

**Resumo.** *O ensino de lógica de programação e consequentemente a codificação são tarefas que requerem muito esforço por parte dos docentes. Além disso, auxiliar o aluno a associar a teoria com a prática é uma tarefa complexa, pois os alunos geralmente ingressam nos cursos de graduação possuindo os mais diversos perfis. Uma dificuldade comum é como ensinar lógica de programação de maneira que o aluno se veja motivado a aprender, ao mesmo tempo em que eles se preparam para utilizar uma ferramenta de codificação e aplicar os conceitos na prática. O objetivo deste trabalho é relatar a experiência da utilização de uma ferramenta específica para o desenvolvimento de jogos digitais (Game Maker) para o ensino de lógica de programação em um curso de graduação.*

### 1. Introdução

O ensino de qualquer disciplina técnica constitui um conjunto de atividades desafiantes para o docente, tanto pelo constante surgimento de novas tecnologias e ferramentas, quanto pela busca por estratégias para se ajustar às novas práticas de aquisição de conhecimento adotadas pelos aprendizes atuais. Em particular, o ensino de lógica de programação e consequentemente a codificação são tarefas complexas que requerem muito esforço por parte dos docentes. Além disso, existe um desafio que é auxiliar o aluno a relacionar a teoria com a prática, pois geralmente eles ingressam nos cursos de graduação com os mais variados perfis e níveis de conhecimento.

Nesse contexto, disciplinas que pretendem alinhar teoria com prática são normalmente muito difíceis de se conduzir devido à necessidade de se tratar fatores técnicos (línguas de programação, componentes, ferramentas etc) e humanos (disponibilidade,

comunicação, gestão etc), tudo ao mesmo tempo [Coutinho et al. 2016]. Adicionalmente, a tarefa de motivar os alunos e os próprios professores não é uma atividade fácil, se verificando particularmente em disciplinas de primeiro ano de cursos de graduação. Isto se deve ao fato que os conteúdos abordados são, na maioria das vezes, novidade para os alunos, constituindo, assim, um desafio para eles.

O aprendizado em linguagens de programação necessita de conhecimentos prévios em lógica e matemática [Ribas et al. 2016]. Mesmo que o raciocínio lógico não seja trabalhado exclusivamente em disciplinas de Matemática, é nessa área que geralmente esses conhecimentos são mais explorados nas escolas. Nesse sentido, seria possível afirmar que cada vez mais os estudantes chegam com menor conhecimento prévio sobre matemática e lógica nos cursos de graduação. Vários trabalhos na literatura discutem diferentes relatos do ensino de lógica de programação e codificação em semestres iniciais de cursos de graduação [Ferreira et al. 2016][Marinho et al. 2016][Coutinho et al. 2017]. Uma dificuldade comumente discutida é como ensinar lógica de programação de maneira que o aluno se motive a aprender, ao mesmo tempo em que se preparam para a utilização de ferramentas de codificação mais avançadas e aplicar conceitos de lógica de programação na prática.

Existem muitas ferramentas para o apoiar ensino de lógica de programação. Algumas delas são praticamente ambientes de desenvolvimento, como o Scratch. Outros ambientes são mais específicos para o ensino de algoritmos, como o VisuAlg. Além disso, pequenos jogos também podem ser utilizados para o ensino de lógica, desenvolvendo o raciocínio lógico e matemático, que muitas vezes precisa ser desenvolvido nos estudantes, a depender de seu perfil. Esses jogos muitas vezes estão disponíveis na forma de quebra-cabeças ou pequenas aplicações para que o usuário construa soluções para resolver algum desafio ou objetivo específico, e assim utilizar conceitos de lógica. Um exemplo de jogo que estimula o raciocínio lógico é o LightBot.

Nesse contexto, a utilização de uma ferramenta que: (i) possibilite a aplicação prática dos conceitos de lógica de programação; (ii) que contenha elementos motivantes para os alunos (por exemplo jogos digitais); e (iii) que permita a criação de um produto completo, nas devidas proporções do conhecimento adquirido ao longo do semestre, seria de grande ajuda para o ensino de lógica de programação.

O objetivo deste trabalho é relatar a utilização de uma ferramenta específica para o desenvolvimento de jogos digitais (Game Maker) para o ensino de lógica de programação em um curso de graduação, para alunos iniciantes na área. Ao final do trabalho, uma avaliação da utilização da ferramenta por parte dos alunos e professores é discutida. O restante do artigo está dividido na seguinte estrutura: Seção 2, uma breve descrição da ferramenta Game Maker e trabalhos relacionados; Seção 3 a contextualização do trabalho para o curso de graduação e disciplina; Seção 4 a metodologia aplicada; Seção 5 os resultados da pesquisa e discussão; e por fim, na Seção 6, a conclusão e trabalhos futuros.

## **2. Ferramenta e Trabalhos Relacionados**

### **2.1. Game Maker**

O Game Maker é um ambiente de desenvolvimento de jogos que permite aos desenvolvedores criarem jogos de forma ágil por meio de recursos de arrastar e soltar (*drag-and-drop*), além de possuir suporte a codificação por meio de uma linguagem de *script*

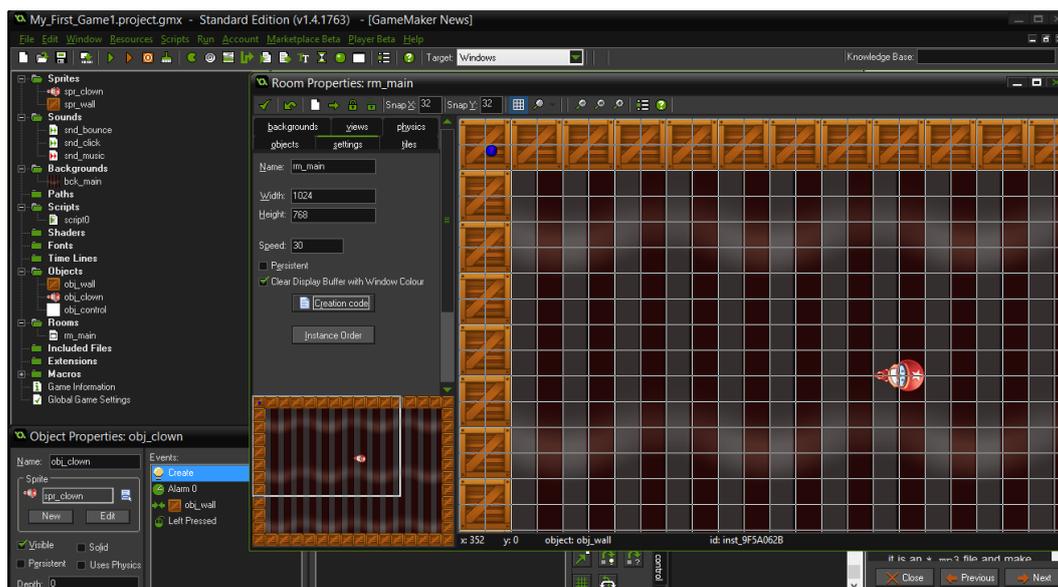


Figura 1. Visão geral do ambiente de trabalho do Game Maker

[Zhang et al. 2014a]. Além disso, o Game Maker possui um motor de jogo (*game engine*) e um conjunto de bibliotecas que auxilia e facilita a abstração no decorrer do desenvolvimento de um jogo [Morais e Silva 2009]. A Figura 1 exibe uma visão geral do ambiente de desenvolvimento do Game Maker.

A Game Maker Language (GML) é a linguagem de *script* proprietária e utilizada no Game Maker para codificação dos jogos. A GML é considerada uma linguagem de sintaxe simples e de fácil aprendizagem. A gramática e as estruturas de programação da GML são semelhantes às adotadas nas linguagens de programação C e Java. Além disso, a GML é considerada uma linguagem orientada a objetos (OO), flexível e extensível ao ponto de possibilitar a inclusão de novas funções que não estão agregadas na linguagem [Morais e Silva 2009]. Atualmente, o Game Maker possui suporte ao desenvolvimento de jogos para diversas plataformas, tais como: Windows, Mac, Ubuntu, HTML 5, Microsoft Universal Windows Platform (UWP), Android, iOS, PS4 e Xbox One.

## 2.2. Trabalhos que Relatam Experiências na Utilização com o GameMaker

Edirisinghe (2008) destacou que um fator que agrava o desafio enfrentado pelos alunos está relacionado à identificação de erros nos programas escritos, levando alunos a não adotarem medidas de prevenção [Edirisinghe 2008]. O trabalho relatou as lições aprendidas de um projeto piloto para o desenvolvimento de um jogo, usando o Game Maker, no intuito de ajudar os alunos a identificarem erros comuns de sintaxe em seus programas.

Yulia e Adipranata (2010) tentaram minimizar as dificuldades no aprendizado da programação orientada a objetos no contexto em que muitos alunos principiantes já estão familiarizados com o paradigma de linguagem procedimental, por meio de uma abordagem de aprendizagem cooperativa baseada no projeto de um jogo [Yulia e Adipranata 2010]. O Game Maker foi usado para ensinar conceitos de objetos por meio da criação de um jogo. O estudo foi aplicado em uma disciplina que foi dividida em grupos de alunos, e baseado nas notas finais a proposta mostrou que o percentual de aprovação dos alunos melhorou.

Hoganson (2010) usou o Game Maker para atrair interesses e experiências culturais dos alunos para a programação [Hoganson 2010]. O trabalho destacou o problema da transição da programação por ferramentas/interfaces gráficas de desenvolvimento para a programação em uma linguagem de programação tradicional. A GML permite que o aluno interaja, por código-fonte, com o jogo que ele está desenvolvendo no Game Maker.

Panitz et al. (2010) criaram um curso de curta duração para alunos que tradicionalmente não cursariam Ciências da Computação, para programação de jogos básicos utilizando o Game Maker e em seguida o C# [Panitz et al. 2010]. A avaliação conduzida verificou o nível de conhecimento do aluno antes e após a metodologia utilizada no trabalho. Conforme os resultados, os alunos aprenderam uma quantidade substancial do conteúdo e demonstraram uma maior satisfação na abordagem adotada no trabalho, em contraste com a abordagem que utiliza uma linguagem única de programação.

Zhang et al. (2014a) discutiram sobre as dificuldades que muitos alunos possuem na aprendizagem das técnicas de recursão para resolução de problemas em Ciências da Computação [Zhang et al. 2014a]. O trabalho proposto consistiu em apresentar um jogo projetado no Game Maker que visa reforçar a compreensão da recursão. O jogo proporcionou aos alunos um ambiente divertido para visualizar e acompanhar o fluxo de execução de funções recursivas e foi aplicado em turmas de duas disciplinas. Os resultados da avaliação se mostram promissores e os alunos relataram que a experiência foi positiva.

Zhang et al. (2014b) apresentaram um jogo para ensinar estruturas de decisão *if*, *if-else*, *if-else-if* e *switch* em uma disciplina introdutória de programação de computadores [Zhang et al. 2014b]. No jogo, desenvolvido no Game Maker, os participantes passeiam pelo ambiente e precisam parar em todas as cabines de informações para aprender ou rever os conceitos de programação. Com isso, o intuito do jogo é proporcionar aos participantes um ambiente divertido para avaliar e praticar os conceitos de programação com foco nas estruturas de decisão. Diante dos resultados, a utilização do jogo teve resultados promissores e a avaliação do jogo pelos os alunos foi em geral positiva.

### **3. Contextualização do Curso de Graduação e Disciplina de Programação**

O curso de graduação em Sistemas e Mídias Digitais (SMD), da Universidade Federal do Ceará (UFC), tem como objetivo formar profissionais com conhecimentos especializados em duas grandes áreas principais: Sistemas Multimídia e Mídias Digitais, contribuindo para perfis de profissionais para o desenvolvimento de mídias digitais e sistemas multimídia como: sistemas web, dispositivos móveis, jogos digitais e animações gráficas.

O SMD possui em sua grade curricular diversas disciplinas de programação. Um problema identificado no curso é a alta taxa de reprovação nas disciplinas iniciais de programação. Este problema prejudica o desempenho dos alunos durante seu percurso acadêmico, afetando a todas as áreas do SMD, mas principalmente impacta na área de Desenvolvimento de Sistemas. Desta forma, observou-se que a aplicação de metodologias de ensino tradicionais, em especial, no ensino de programação de computadores, não funciona de maneira satisfatória para um corpo discente tão heterogêneo. Como consequência, tem-se que o aluno que não obtém uma base sólida de programação nas disciplinas dos semestres iniciais encontra sérias dificuldades nas demais disciplinas ofertadas ao longo do curso, que envolvam conceitos e técnicas de programação.

Uma das disciplinas iniciais do curso é Programação I, obrigatória para todos os

alunos, possuindo 64 horas-aula de carga horária, sendo na maior parte do tempo prática. Seus objetivos são apresentar e demonstrar a importância da programação no curso de graduação, assim como familiarizar o aluno com a lógica para o desenvolvimento de soluções de problemas. A metodologia de ensino atual adotada no curso inclui a utilização eventual de apresentação em slides, utilização de lousa para desenvolvimento de atividades em aulas expositivas, mas com a maior parte das atividades sendo de fato realizadas de maneira prática, diante do computador. Compõem sua forma de avaliação provas escritas, provas práticas com questões de desenvolvimento de programas em computador e trabalhos práticos. A disciplina conta, ainda, com o apoio de bolsistas monitores que auxiliam tanto os professores quanto os alunos.

Ainda no primeiro semestre há um projeto integrado com outras disciplinas do semestre: Desenho I, História do Design e Autoração Multimídia I. Neste projeto, os alunos devem produzir um jogo digital aplicando os conhecimentos adquiridos em todas as disciplinas do semestre. Todos os professores dessas disciplinas avaliam o produto final sob o ponto de vista de suas disciplinas e de integração com as demais.

#### 4. Metodologia

Muitos alunos gostam de jogos e a ideia de poder desenvolver um pode ser atrativa. Além disso, o desenvolvimento de jogos envolve aspectos de computação, artes, ciências sociais e psicologia [Overmars 2004], sendo, portanto, uma atividade que se alinha com a proposta interdisciplinar de uma formação em Sistemas e Mídias Digitais. Com esta motivação, esta pesquisa tem como objetivo analisar se a ferramenta de desenvolvimento Game Maker seria adequada para o ensino de lógica de programação prioritariamente e como consequência um pouco de codificação. Para isto, um questionário *online* foi projetado com perguntas objetivas e de texto livre. As questões objetivas foram formuladas em formatos de afirmações com as quais os alunos deveriam dizer o nível de concordância. Para tanto, foi utilizada a escala de Likert [Likert 1932] onde o aluno deveria selecionar um valor em uma escala de 1 a 5, onde 1 = Discordo totalmente, 2 = Discordo, 3 = Não concordo nem discordo, 4 = De acordo e 5 = Totalmente de acordo. Sendo assim, as afirmações com maior número de respostas 4 e 5 são aquelas com as quais os alunos mais concordaram. A Tabela 1 lista as questões aplicadas. Os alunos que poderiam responder à pesquisa eram ingressos no curso de graduação e estavam divididos em duas turmas,

**Tabela 1. Questões aplicadas ao questionário**

Q1	A utilização do GameMaker foi fácil?
Q2	As aulas de algoritmos apoiaram a utilização do GameMaker?
Q3	As aulas de teoria de lógica de programação apoiaram a utilização do GameMaker?
Q4	A utilização do GameMaker colaborou para o aprendizado de lógica de programação?
Q5	A utilização do GameMaker colaborou para o aprendizado de codificação?
Q6	Você gostaria de continuar utilizando o GameMaker em outras disciplinas, para conhecimentos mais aprofundados em codificação?
Q7	Você acha que a carga horária dispendida com aulas teóricas foi suficiente para apoiar a utilização do GameMaker?
Q8	Você acha que a carga horária dispendida com aulas da ferramenta foi suficiente para apoiar a utilização do GameMaker?
Q9	Você acha que o suporte dos bolsistas de programação foi suficiente para apoiar a utilização do GameMaker?
Q10	Você acha que o GameMaker lhe motivou a aprender lógica de programação e codificação?
Q11	Quais os pontos fortes da abordagem aplicada na disciplina?
Q12	Quais os pontos fracos da abordagem aplicada na disciplina?
Q13	Que melhorias você sugeriria aos professores da disciplina?
Q14	Quais os problemas que você teve com o GameMaker?

com 29 e 27 alunos respectivamente, ambas do primeiro semestre de 2017. Apenas 29 alunos responderam à pesquisa. Acredita-se que essa quantidade foi devido ao fato que a pesquisa não era obrigatória e foi no final do semestre onde os alunos estão mais sobrecarregados. A partir dos resultados obtidos, gerou-se um gráfico de barras com todas as questões de escala e seus níveis de resposta.

Por fim, um questionário com professores de disciplinas iniciais de programação sobre os resultados do uso do Game Maker foi aplicado. Cinco professores responderam.

## 5. Apresentação e Discussão dos Resultados

### 5.1. Questões de Escala

A Figura 2 exibe um gráfico de barras representando o resultado das questões objetivas segundo a escala de Likert. A maioria das questões teve uma avaliação boa, ou seja, com maior quantidade de alunos selecionando as duas maiores opções (de acordo e totalmente de acordo). Esta é uma primeira indicação de que a abordagem foi bem vista pelos alunos, considerando que as questões envolviam aspectos teóricos e práticos.

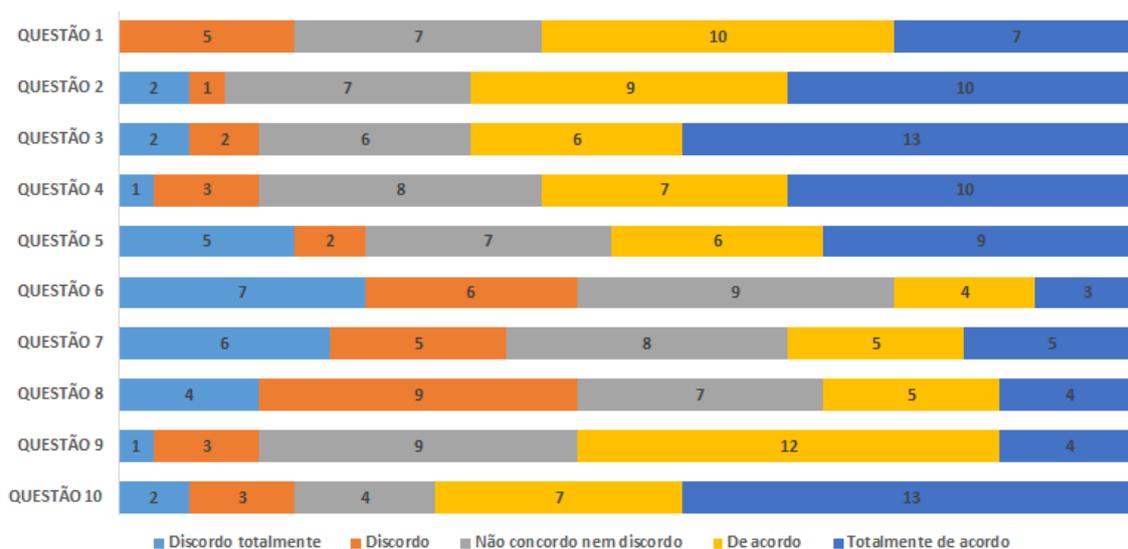


Figura 2. Gráfico de barras resultante da aplicação do questionário

Especificamente Q5, Q6, Q7 e Q8 obtiveram as piores avaliações. Q5 indicou que o Game Maker não colaborou tanto quanto se esperava para o aprendizado de codificação, talvez por possuir muitos comandos pré-determinados do tipo componentes gráficos já prontos, bastando configurar. Q6 destacou que os alunos não gostariam de utilizar o Game Maker em outras disciplinas para aprofundar codificação. Q7 e Q8 estão diretamente relacionadas à carga horária dispendida com aulas teóricas para apoiar a ferramenta e com aulas práticas na ferramenta para o conhecimento das funções e pleno uso.

Já Q3 e Q10 obtiveram as maiores notas. Q3 analisa se as aulas de teoria de lógica de programação apoiaram a utilização do Game Maker, o que foi surpreendentemente bem avaliada. Atribui-se essa nota elevada à utilização prévia de jogos ou pequenas aplicações, o que motiva o aluno que está iniciando os conceitos de lógica de programação, e a correlação com situações de outras disciplinas, como, por exemplo,

com a disciplina de Autoração Multimídia I, abordando exemplos de algoritmos para operações reais aplicadas a imagens digitais. Q10 é relacionada à motivação em usar o Game Maker no aprendizado de lógica de programação e codificação. Essa questão obteve a nota mais alta, o que indica que a ferramenta ajudou a tratar um dos problemas mais comuns no ensino de lógica de programação, que é a desmotivação.

## 5.2. Questões Abertas

Quatro questões de texto livre fizeram parte do questionário, relacionadas a pontos fortes, pontos fracos, sugestões e problemas com a abordagem e a ferramenta Game Maker. O objetivo dessas questões foi possibilitar ao aluno indicar de forma livre os pontos fortes, fracos e melhorias sobre a disciplina. Além de identificar problemas na utilização da ferramenta, seja por defeitos, seja por mal uso por parte dos alunos ou professores.

Na questão **“Quais os pontos fortes da abordagem aplicada na disciplina?”**, em geral os alunos acharam a metodologia interessante e comentaram sobre diversos aspectos: explicação clara e direta com prática constante; boa ferramenta utilizada para quem está iniciando a programar, de linguagem fácil; explicação inicial sobre algoritmos (atribuição, estruturas de condição e repetição); correções de questões das listas de exercícios em sala; início dinâmico da disciplina; atenção dos professores com os alunos e com dúvidas em geral; monitorias; aplicação de conceitos básicos de lógica em um ambiente de fácil codificação; ensino inicial por meio de jogos; dinamismo dos professores.

Já a questão **“Quais os pontos fracos da abordagem aplicada na disciplina?”** apontou mais problemas com relação ao projeto integrado (descrito na Seção 3), onde os alunos precisam desenvolver um jogo digital envolvendo conceitos estudados em todas as disciplinas do primeiro semestre em um trabalho de projeto integrado, revelando os seguintes pontos fracos: não aprofundamento do Game Maker voltado para o projeto integrado; foco mais em programar e codificar no âmbito geral, deixando de lado como criar o jogo; ferramenta não é recomendada para explicar os conceitos de programação; falta de apoio para o projeto integrado; não explicação de algumas funções essenciais para o projeto; não detalhamento dos eventos da ferramenta; avaliações pesadas; algumas funções do Game Maker são inúteis em outras linguagens; desconexão com o trabalho do projeto final; Game Maker não é intuitivo; pouco material que explicasse a ferramenta de maneira didática; falta de atividades valendo pontos; melhorar a resolução de problemas com algoritmos; necessidade de consultar tutoriais na internet.

A questão **“Que melhorias você sugeriria aos professores da disciplina?”** destacou as seguintes melhorias para a disciplina: ensinar na disciplina mecânicas de jogos (correr, pular, atacar etc); propor desafios semanais para os alunos; explicar melhor conceitos de funções e vetores; repassar material relacionado a jogos para ajudar no projeto integrado; práticas da sala com dificuldade próxima das listas de exercícios; mais aulas teóricas; destacar funções que serão importantes para o projeto; ensinar a lógica pelo uso de blocos por um período um pouco mais extenso antes da codificação em evento; sistema de avaliação mais focado em atividades, incentivando os alunos a praticarem os conceitos aprendidos em sala; mais aulas específicas para conhecer a ferramenta; utilização de um programa próprio para programação; ensinar programação com *“drag-and-drop”*; aumentar a quantidade de funções do Game Maker apresentadas; aumentar a carga horária de uso da ferramenta; criação de atividades extra para melhorar notas; resolver mais questões de algoritmos em sala; direcionar mais aulas para o projeto integrado.

Por fim, a questão “**Quais os problemas que você teve com o Game Maker?**” pontuou problemas específicos da ferramenta Game Maker que os alunos tiveram, tais como: manipulação de *scripts*; vetores e matrizes; estruturação de códigos; programa não ser gratuito; falta de compreensão das funções próprias do Game Maker; utilização de *scripts* no evento draw; saber como e quando usar os diferentes tipos de eventos (create, draw, step etc); entrada de informação do usuário; problemas com máquinas defeituosas.

### 5.3. Resultados Iniciais da Utilização da Ferramenta sob a Visão dos Professores

Como uma avaliação complementar, alguns professores do semestre seguinte à aplicação da ferramenta foram contatados para que eles repassassem suas primeiras impressões sobre o desempenho dos alunos (alunos que cursaram a disciplina com o Game Maker). Para guiar a conversa, algumas questões foram disponibilizadas aos professores: (1) De maneira geral, em relação aos semestres passados o nível melhorou ou piorou? (2) Em relação a entender conceitos melhorou ou piorou? (3) Em relação a entender exercícios melhorou ou piorou? E (4) Em relação a provas / trabalhos melhorou ou piorou?

De maneira geral, o relato dos professores foi que no início os alunos apresentaram um aproveitamento um pouco melhor que os das turmas anteriores. A princípio aparentavam ter melhorado, entretanto o nível caiu. Não se pode afirmar se foi novamente por falta de motivação em programação ou deficiência conceitual e técnica, pois as notas finais foram variadas, e a plataforma de desenvolvimento não foi o Game Maker.

Em relação aos conceitos, os professores apontaram que os alunos compreenderam bem conceitos relacionados a exemplos com gráficos. Isso pode ser em decorrência da adoção de exercícios práticos orientados a aplicações gráficas ou de manipulação de elementos gráficos na ferramenta. Os alunos também compreendem mais facilmente exercícios relacionados a jogos, que envolvem conceitos como colisões e movimentações. Mesmo após a finalização das avaliações e do projeto integrado, ainda não se pode afirmar se houve melhoria em relação às notas finais. É necessário se obter parâmetros objetivos do rendimento dos alunos, tais como a quantidade de aprovações e reprovações, média geral da turma e frequência, para comparação com semestres anteriores, em que o uso da ferramenta Game Maker não tinha sido adotado, e relacionar com elementos qualitativos.

Ao que tudo indica, o medo de que os alunos não compreendessem codificação devido ao estilo de programação baseada em blocos (estilo do Game Maker) não se confirmou. Isso é bom, pois é necessário que os alunos entendam e exercitem diversos estilos de programação e tipos de ferramentas. Eles também reconheceram melhor bem a diferença entre um algoritmo e um programa em si, o que era um problema conhecido.

### 5.4. Ameaças à Validade

Esta pesquisa foi o primeiro ensaio para analisar a eficiência da aplicação da ferramenta Game Maker no ensino de lógica de programação. Entretanto, alguns aspectos foram considerados ameaças à validade do trabalho. Para esta análise, considerou-se alguns itens da classificação de Wohlin [Wohlin et al. 2012].

**Validade de construção:** apesar da ferramenta ter sido aplicada a duas turmas (56 alunos), essa população é pouca. Além disso, apenas 29 (cerca da metade dos alunos) responderam ao questionário. As questões elaboradas no questionário também podem não ter sido refinadas o suficiente para atenderem devidamente os objetivos da pesquisa.

**Validade interna:** não necessariamente todos os alunos utilizaram a ferramenta profundamente no desenvolvimento de suas atividades, conduzindo a respostas com fraca credibilidade. Além disso, o mecanismo de coleta de dados da pesquisa não foi bem planejado.

**Validade externa:** o ambiente do Game Maker possui algumas diferenças de outros ambientes (e.g. Stencyl e Scratch), e esta análise considerou apenas o Game Maker.

**Validade de conclusão:** poucos alunos responderam, impossibilitando generalizações. Além disso, não é possível realmente chegar à conclusão que a utilização da ferramenta melhorou o entendimento e a didática, em relação às turmas anteriores. O retorno dado pelos professores de programação do semestre seguinte é um bom indício ou um direcionamento para futuros experimentos, mas ainda não é possível se concluir que a abordagem com a ferramenta foi realmente efetiva na aprendizagem dos alunos, até mesmo porque foram apenas três professores e em um curto tempo de observação.

## 6. Conclusão

Ensinar lógica de programação é uma atividade que ainda instiga nos docentes a necessidade de se pesquisar estratégias e metodologias que auxiliem na sua efetividade. Aprender lógica de programação requer dedicação por parte do aluno, pois necessita de habilidades adquiridas e melhoradas por meio da prática constante. Um problema comum é identificar quais ferramentas são apropriadas para se aplicar conceitos. Nesta análise não estão sendo consideradas apenas aspectos didáticos de uma ferramenta, e sim, a inclusão de outros fatores que possam favorecer o aumento da motivação dos alunos, adequando as atividades práticas propostas em aulas à exploração de aspectos que envolvam multimídia.

Este trabalho relatou o uso da ferramenta Game Maker para o ensino de lógica de programação na graduação. Neste estudo, o foco foi nas impressões dos estudantes em relação ao aprendizado do conteúdo abordado e como alguns professores perceberam essa mudança. Como dito previamente, o perfil do aluno é híbrido e faz-se necessário encontrar uma abordagem (metodologias e escolhas de ferramentas) que se adéquem aos interesses dos alunos e que contemplem os conceitos e fundamentos importantes para o aprendizado de programação. Em geral, os alunos gostaram da estratégia de utilizar o Game Maker como ferramenta, mas com algumas ressalvas. Percebeu-se, nos comentários apresentados pelos alunos que a expectativa de que a disciplina de Programação I se concentrasse em auxiliar o aluno no desenvolvimento de jogos, e não como uma disciplina para ensinar Fundamentos de Programação, que os preparariam para o desenvolvimento de aplicações de propósito geral, por exemplo, aplicações web e para dispositivos móveis.

Como trabalhos futuros, pretende-se adequar a abordagem dos professores conforme as sugestões dos alunos e refinar a maneira de ministrar conceitos. Além disso, um acompanhamento dos alunos ao longo dos demais semestres e das disciplinas subsequentes de programação traria uma visão mais real dos efeitos da utilização do Game Maker como ferramenta de apoio ao ensino de programação. Por fim, alguns aspectos gerais que merecem ser trabalhados para edições futuras da utilização da estratégia seriam: produção de conteúdo, alinhamento pedagógico e interoperabilidade entre diferentes plataformas.

## Referências

Coutinho, E. F., de Lima, E. T., e Santos, C. C. (2017). Um panorama sobre o desempenho de uma disciplina inicial de programação em um curso de graduação. *Revista*

*Tecnologias na Educação*, 19(9).

- Coutinho, E. F., Gomes, G. A. M., e José, M. L. A. (2016). Applying design thinking in disciplines of systems development. In *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, pages 1–8.
- Edirisinghe, E. M. N. S. (2008). Teaching students to identify common programming errors using a game. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education, SIGITE '08*, pages 95–98, New York, NY, USA. ACM.
- Ferreira, A. C., Santos, J., Silva, R., Oliveira, A. T. R., Zabot, D., Abdalla, D., e 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 (WAlgProg)*, pages 1306–1315. SBC.
- Hoganson, K. (2010). Teaching programming concepts with gamemaker. *J. Comput. Sci. Coll.*, 26(2):181–188.
- Likert, R. (1932). *Technique for the Measurement of Attitudes*. Number 140 in Archives of Psychology. New York: The Science Press.
- Marinho, C. S. S., Moreira, L. O., Coutinho, E. F., Paillard, G. A. L., e Neto, E. T. L. (2016). Experiências no uso da metodologia coding dojo nas disciplinas básicas de programação de computadores em um curso interdisciplinar do ensino superior. In *II Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg)*, pages 1097–1106.
- Morais, F. C. e Silva, C. M. (2009). Desenvolvimento de jogos eletrônicos. *Revista e-Xacta*, 2(2).
- Overmars, M. (2004). Teaching computer science through game design. *Computer*, 37(4):81–83.
- Panitz, M., Sung, K., e Rosenberg, R. (2010). Game programming in cs0: A scaffolded approach. *J. Comput. Sci. Coll.*, 26(1):126–132.
- Ribas, E., Bianco, G. D., e Lahm, R. A. (2016). Um curso de programação a distância com metodologias ativas e análise de aprendizagem por métricas de software. *RENOTE - Revista Novas Tecnologias na Educação*, 14(2).
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., e Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer-Verlag Berlin Heidelberg, 1 edition.
- Yulia e Adipranata, R. (2010). Teaching object oriented programming course using cooperative learning method based on game design and visual object oriented environment. In *2010 2nd International Conference on Education Technology and Computer*, volume 2, pages 355–359.
- Zhang, J., Atay, M., Smith, E., Caldwell, E. R., e Jones, E. J. (2014a). Using a game-like module to reinforce student understanding of recursion. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–7.
- Zhang, J., Smith, E., Caldwell, E. R., e Perkins, M. (2014b). Learning and practicing decision structures in a game. *J. Comput. Sci. Coll.*, 29(4):60–67.