

## Games for Programming as an Approach for First Programming Experiences

Original Title: Jogos de Programar: Abordagem para os Primeiros Contatos dos Estudantes com a Programação

Guilherme A. Zanchett<sup>1</sup>, Adilson Vahldick<sup>3,4</sup>, André Raabe<sup>1,2</sup>

<sup>1</sup> Mestrado em Computação Aplicada, Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

<sup>2</sup> Programa de Pós-Graduação em Educação, Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

<sup>3</sup> Departamento de Engenharia de Software, Universidade do Estado de Santa Catarina (UDESC) – Ibirama, SC – Brasil

<sup>4</sup> Departamento de Engenharia Informática – Universidade de Coimbra – Portugal

---

### ARTICLE INFO

#### Article history:

Received 23 June 2016

Accepted 25 October 2016

Available online 1 August 2017

#### Keywords:

Games for programming

Computational thinking

Programming Learning

ISSN: 2594-5602

DOI:

10.14210/ijcthink.v1.n1.p39

---

### ABSTRACT

**INTRODUCTION:** This work investigates the use of programming games as a way to establish the high school students first contacts with programming concepts. Programming games are puzzle style games in which the player must solve problems that require algorithmic logic using some programming notation. **OBJECTIVE:** The research sought to evaluate which characteristics of the programming games are perceived positively by the students. **METHOD:** The research was carried out with 14 students who used three different programming games (Code Combat, NoBug's Snack Bar and Lighbot). They answered a questionnaire with open and closed questions about (i) preferred game, (ii) Engagement and fun, (iii) tutorial style (way of proposing problems to players); And (iv) perception about programming concepts involved. **RESULTS:** The Code Combat and LightBot games tied the students' preference, with Lightbot achieving a better engagement and fun score. The Tutorial style of Code Combat and NoBug's Snack Bar received the best scores. The students reported having understood the concepts of function (92.9%), Loop (92.9%) and variable (71.2%) after using the games and their perception of the difficulty of programming was easy or average (71.4%). **CONCLUSION:** The use of programming games in students' initial contacts can bring beneficial results by creating a less intimidating environment in which the student perceives himself capable of achieving his goals while having fun. It aids to avoid obstacles that the perception of the difficulty of programming could generate. Concepts such as loops and functions were better assimilated by students than the concept of a variable, suggesting the need to rethink the traditional organization of textbooks and introductory programming courses.

---

### RESUMO

**INTRODUÇÃO:** Este trabalho investiga a utilização de jogos de programar como forma de estabelecer os primeiros contatos de estudantes de ensino médio com conceitos de programação. Jogos de programar são jogos do estilo puzzle em que o jogador deve solucionar problemas que requerem lógica algorítmica usando alguma notação de programação. **OBJETIVO:** A pesquisa buscou avaliar que características dos jogos de programar são percebidas de forma positiva pelos estudantes. **MÉTODO:** A pesquisa foi realizada com 14 estudantes que utilizaram três diferentes jogos de programar (Code Combat, NoBug's Snack Bar e Lighbot) e responderam um questionário com perguntas abertas e fechadas sobre: (i) jogo preferido, (ii) engajamento e diversão, (iii) estilo tutorial (forma de propor os problemas aos jogadores); e (iv) percepção sobre conceitos de programação envolvidos. **RESULTADOS:** Os jogos Code Combat e LightBot empataram na preferência dos estudantes, tendo o Lightbot atingido melhor escore de engajamento e diversão. O estilo Tutorial do Code Combat e do NoBug's Snack Bar foram os melhores pontuados. Os estudantes relataram terem compreendido os conceitos de função (92,9%), Repetição (92,9%) e variável (71,2%) após usarem os jogos e percepção sobre a dificuldade de programar de 71,4% deles foi fácil ou média. **CONCLUSÃO:** O uso de jogos de programar nos contatos iniciais dos estudantes pode trazer resultados benéficos ao criar um ambiente menos intimidador em que o estudante se percebe capaz de atingir seus objetivos enquanto se diverte evitando obstáculos que a percepção da dificuldade de programar poderia gerar. Conceitos como repetições e funções foram melhor assimilados pelos estudantes do que o conceito de variável, sugerindo a necessidade de repensar a organização tradicional dos livros didáticos e cursos de programação introdutória.

## 1. Introdução

O primeiro contato dos estudantes com conceitos de programação pode ser determinante na forma como eles perceberão os desafios e enfrentarão as dificuldades inerentes à aprendizagem da lógica de programação. Segundo Rogerson e Scott (2010), fatores psicológicos como motivação pessoal e autoconfiança exercem grande influência sobre alunos de programação pois estes estão diante da necessidade de dominar uma forma completamente nova de pensar. Kinnunen et al. (2007) ressaltam que programação requer um alto nível de abstração, prática e um esforço intenso. Gomes e Mendes (2007) destacam a motivação como um dos fatores principais para o sucesso, principalmente devido ao fato de que muitos alunos possuem uma visão negativa associada à programação e aos programadores, que são em boa parte vistos como pessoas antissociais ou introspectivas. Orehovački e Babić (2015) afirmam que 60% dos alunos de cursos de engenharia gostariam que disciplinas de programação fossem mais interativas, amigáveis e divertidas.

Segundo Papastergiou (2009), Oblinger (2004) e Prensky (2001a) esta geração de jovens é nativa em relação aos jogos digitais, independente de jogarem casualmente ou com mais frequência. Prensky (2001a) e Oblinger (2004) afirmam que a motivação intrínseca dos jovens em relação aos jogos anda na contramão de seus interesses por assuntos relacionados ao currículo escolar, e além disso o fato de crescerem em constante exposição à internet, mudou completamente a forma como os jovens aprendem e descobrem conteúdos novos. Oblinger (2004) reforça o conceito de que jogos não são mais apenas para diversão, mas se tornaram poderosos ambientes de aprendizado, sendo utilizados por um número crescente de faculdades e escolas para aprimorar o sistema de ensino e o aprendizado dos alunos. Kiili (2005) afirma que jogos atendem todos os requisitos necessários para um ambiente de ensino, e podem ser ferramentas engajadoras poderosas.

Os jogos sérios tem sido cada vez mais usados por educadores, corporações, e governos, entre outros, para propósitos que vão além do entretenimento. Segundo Corti (2006) e Squire e Jenkins (2003) este gênero possui algumas vantagens, como permitir ao jogador experimentar situações que seriam impossíveis no mundo real, por razões de segurança, custos ou tempo. Corti (2006) afirma que os jogos sérios são meios de alavancar o poder dos jogos de computador para cativar e engajar usuários finais para um propósito específico, como a aquisição de algum conhecimento ou desenvolvimento de uma nova habilidade. Segundo Michael e Chen (2006), o ponto principal é conduzir o aprendizado de forma divertida. Entretanto, Prensky (2001b) resalta que a jogabilidade é um fator forte de envolvimento, intenso e apaixonante, bem como objetivos que motivam e regras que geram uma estrutura.

Segundo Long (2007), o uso de jogos como ferramenta de ensino de programação é mais satisfatório para os alunos, apresenta resultados de aprendizado mais rápidos e faz com que os alunos se lembrem melhor do conteúdo se comparado a uma aula expositiva. O uso de jogos para o ensino de programação com sua abordagem imaginativa, encoraja os alunos a desenvolverem suas habilidades, e eleva o interesse dos mesmos em buscar informações complementares sobre o assunto.

Neste contexto, o trabalho busca investigar a utilização de jogos de programar como forma de estabelecer os primeiros contatos de estudantes de ensino médio com conceitos de programação. Jogos de programar são jogos do estilo *puzzle* em que o jogador deve solucionar problemas que requerem lógica algorítmica usando alguma notação de programação (De Jesus & Raabe, 2010). À medida que soluciona os problemas avança para novas fases, onde os conceitos são reforçados, ou novos conceitos são apresentados.

Foram testados três jogos de programar com diferentes enredos e notações de programação, buscando conhecer a opinião dos estudantes sobre: (i) jogo preferido, (ii) engajamento e diversão, (iii) estilo tutorial (forma de propor os problemas aos jogadores); e (iv) percepção sobre conceitos de programação envolvidos. A compreensão sobre a forma como estes jogos são percebidos pelos estudantes auxilia no planejamento de atividades de introdução ao pensamento computacional, possibilitando também delinear diretrizes para a concepção e o projeto de novos jogos.

O artigo está organizado como segue: a seção 2 contextualiza o uso de jogos para o aprendizado de programação, a seção 3 apresenta os procedimentos metodológicos detalhando os jogos utilizados e o instrumento de mensuração construído (questionário). A seção 4 apresenta os resultados e a seção 5 relaciona as conclusões do trabalho.

## **2. Aprendizado Introdutório de Programação usando Jogos**

Para muitos alunos, o primeiro contato com programação é intimidador (Connolly, Murphy & Moore, 2009). A solução dos problemas em geral exigem do aluno habilidades e competências ainda não desenvolvidas. A criação de jogos pelos estudantes tem sido utilizado como forma de motivar o aprendizado de programação. Ambientes de autoria como Alice e Scratch possibilitam que os alunos iniciem a criar aplicações com poucos conhecimentos, iniciando com elementos concretos e gradativamente sendo conduzidos para a compreensão dos aspectos mais abstratos da computação (Dann & Cooper, 2009).

Nesses tipos de ambientes o aluno pratica e desenvolve a construção de soluções computacionais por meio de elementos concretos. Para alcançar essa solução é preciso desempenhar tarefas de resolução de problemas, e assim praticar e aprender abstração, decomposição e análise, que são competências essenciais para escrever programas de computador. Entretanto, a produção de jogos envolve a utilização de muitos conceitos e a gestão mental de inúmeras tarefas (Ke, 2014), que transformam o aprendizado introdutório de programação numa missão complexa.

Uma alternativa à produção de jogos é o próprio uso de jogos, seja para aprender conceitos novos, ou reforçar algum conhecimento. Aproveitando assim o aspecto motivacional envolvido sem no entanto envolver uma complexidade alta. Os jogos são experiências interativas fáceis e divertidas, e normalmente não exigem um treino ou ajuda externa (Prensky, 2001a), facilitando a sua adoção nas atividades em aula ou extra-classe.

Com pouco esforço prévio do professor em ensinar conceitos de programação, os alunos podem iniciar o seu aprendizado pois, conforme Whitton (2010), os jogos são uma alternativa em que os alunos podem aprender de forma personalizada (o jogo acompanha a evolução do jogador) e auto assistida (o jogador percebe os seus erros ao falhar nas missões, ou o jogo instrui o jogador em como realizar alguma ação).

Com a ascensão da aprendizagem baseada em jogos na última década, vários jogos vêm sendo desenvolvidos (Malliarakis, Satratzemi & Xinogalos, 2014; Vahldick, Mendes & Marcelino, 2014), para diferentes plataformas, públicos, mecânicas e abrangência de conteúdo. Contudo, existe a necessidade de ampliar o conhecimento existente sobre em que grau os jogos auxiliam na aprendizagem de programação e que características destes jogos podem favorecer esta aprendizagem.

### 3. Procedimentos Metodológicos

Os sujeitos participantes da pesquisa foram 14 estudantes (6 meninas e 8 meninos) que fazem parte de um projeto de extensão voltado à introdução do pensamento computacional envolvendo ensino de programação e robótica. Os estudantes são do 1º e 2º anos do ensino médio de uma escola pública e foram selecionados pela direção da escola devido ao bom desempenho acadêmico.

Três jogos de programar foram utilizados como primeira atividade de contato dos estudantes com conceitos de programação. Foram selecionados os seguintes jogos que usam diferentes notações de programação: (i) Code Combat<sup>1</sup> que utiliza notação de código textual; (ii) Lightbot<sup>2</sup> que utiliza sequenciamento de blocos de instruções; e (iii) NoBug's Snack Bar<sup>3</sup> que utiliza a notação de blocos similares ao da ferramenta Scratch. Os jogos são detalhados na seção 2.1.

Os estudantes foram divididos em duas turmas de sete indivíduos que participaram do projeto em dias intercalados. Durante três semanas cada turma jogou por aproximadamente 4 horas cada um dos jogos na seguinte sequência: Light Bot na primeira semana, Code Combat na segunda semana e NoBug's na terceira semana. Explicações complementares foram feitas sempre que os estudantes solicitavam ou que a turma como um todo demonstrava alguma dificuldade de compreensão. Não houveram explicações prévias sobre conceitos de programação de forma que o uso dos jogos representou um primeiro contato dos estudantes com o tema e a partir deste uso as explicações eram fornecidas conforme as dúvidas surgiam. Após as três semanas de experimentação com os três jogos, foi aplicado um questionário com perguntas abertas e fechadas que é detalhado na seção 3.2.

#### 3.1. Jogos Selecionados

A escolha dos jogos foi feita considerando os seguintes critérios: (i) abordar conceitos de programação como variáveis, repetições, desvios condicionais e funções; (ii) possuir diferentes notações de programação (ícones, blocos, código); (iii) diferentes enredos e formas de propor os problemas aos jogadores (que foram denominados de estilos tutoriais); (iv) que fossem executados diretamente no navegador. Como o estudo foi realizado com estudantes do ensino médio, jogos com linguagem visual voltada ao público infantil foram excluídos. Da mesma forma, jogos que tinham custo de aquisição também foram excluídos. Os jogos foram identificados a partir do trabalho de Vahldick, Mendes e Marcelino (2014). Os três jogos selecionados são detalhados a seguir.

##### 3.1.1. Code Combat

O Code Combat é um jogo que aborda conceitos de programação através de uma aventura medieval, onde o jogador controla heróis, e ao longo de sua jornada vai coletando cristais, e derrotando diversos inimigos, como ogros, bandidos, entre outros. É um jogo gratuito, que também disponibiliza fases pagas. Estas fases não são obrigatórias e não influenciam diretamente no progresso do jogo. O jogo disponibiliza uma IDE (Integrated Development Environment) no próprio navegador para que o jogador implemente seus códigos. Esta IDE permite que o jogador selecione entre 6 linguagens de programação: Python, JavaScript, CoffeeScript (Experimental), Clojure (Experimental), Lua e IO (Experimental). O jogador pode escolher entre 23 idiomas na página inicial do site do jogo. O jogo era dividido em 6 mundos, dentre eles 5 disponíveis com 165 fases gratuitas.

<sup>1</sup> <http://www.codecombat.com>

<sup>2</sup> <http://www.lightbot.com>

<sup>3</sup> <http://nobugssnackbar.dei.uc.pt/>

Cada mundo fornece uma estimativa de tempo necessário para sua conclusão. Os conceitos abordados e as informações a respeito de cada mundo estão dispostos no Quadro 1.

Mundo	Quantidade de Fases	Tempo de Conclusão Estimado (horas)	Conceitos Abordados
1 Kithgard Dungeon	34	1 - 3	Sintaxe, métodos, parâmetros, strings, estruturas de repetição e variáveis
2 Backwoods Forest	45	2 - 6	Desvios condicionais, operadores relacionais, propriedades de objetos e manuseio de inputs
3 Sarven Desert	37	4 - 11	Operadores aritméticos, <i>while-loops</i> , arrays, comparação de strings, encontrar min/max
4 Cloudrip Mountain	41	N/D	Object-literals, invocação remota de métodos, <i>for-loops</i> , funções, desenho e módulo
5 Kelvinthap Glacier	12	N/D	Técnicas avançadas

**Quadro 1.** Informações sobre os mundos do Code Combat.

Os dois últimos mundos ainda estão em desenvolvimento e abordam o que o jogo denomina de técnicas avançadas, entretanto não deixa claro quais são estas técnicas. . A Figura 1 ilustra a interface do jogo.



**Figura 1.** Interface do Code Combat.

A interface do jogo, ilustrada na figura 1, mantém a IDE para digitar o código das soluções sempre no lado direito da tela. Os objetivos e dicas estão sempre escritos em azul no início do código, em forma de comentário. Abaixo da IDE, estão dispostos todos os métodos que podem ser utilizados pelo jogador. Em caso de dúvida, basta o jogador passar o mouse sobre os métodos e uma caixa de diálogo se abrirá com uma

explicação. O jogo possui um botão de ajuda, que leva o jogador a uma tela com tutoriais em vídeo sobre a fase em questão.

### 3.1.2. Lightbot

O Lightbot é um jogo de programar que utiliza uma notação de ícones representando blocos de comandos. O jogo aborda os conceitos de lógica e funções. Com o objetivo de programar um robô para acender a luz de todas as plataformas de cor azul disponíveis no nível, o jogador tem à sua disposição uma interface com poucas informações, conforme ilustrada na Figura 2.

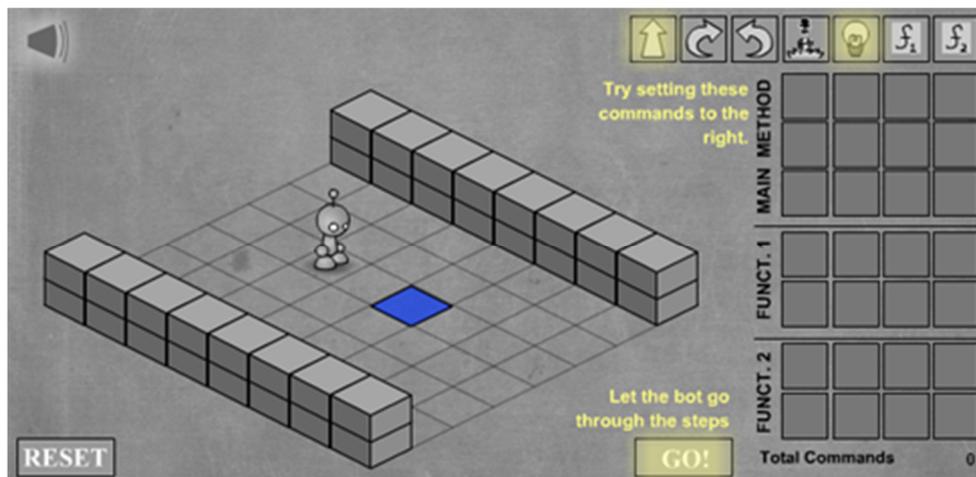


Figura 2. Interface do Lightbot.

Os espaços (*slots*) do lado direito servem para o jogador arrastar os blocos de comando localizados na parte superior da interface, posicionando-os de forma a solucionar os problemas. A única forma de tutorial que o jogo fornece é destacando com uma luz amarela novos comandos, que ainda não haviam sido utilizados. Juntamente com o destaque de comandos, o jogo fornece uma frase de auxílio nas fases iniciais que fica junto da grade de comandos. Os comandos são representados por símbolos. Estão à disposição do jogador os seguintes comandos: andar para frente, virar para a esquerda, virar para a direita, acender, invocar a função 1, invocar a função 2, pular para frente.

Os desafios do jogo se baseiam em acender todas as luzes, criando um algoritmo, que possui um tamanho máximo estipulado pelos *slots* disponíveis. O jogador dispõe de três conjuntos de *slots* para onde é possível arrastar os blocos de comando. O primeiro é o método principal, que pode receber até doze blocos de comandos. O segundo e terceiro são denominados de função 1 e função 2, que podem receber até oito blocos de comando cada um.

### 3.1.3. NoBug's Snack Bar

O NoBug's Snack Bar é um jogo que utiliza a notação de blocos de comandos como no Scratch. É um jogo em desenvolvimento que adota a mesma tecnologia do projeto Code.org do governo norte-americano (Vahldick et. al, 2015). Para ter acesso ao jogo, é preciso entrar em contato com os desenvolvedores e solicitar o registro de cada aluno.

O jogo tinha 16 missões com assuntos como sequenciamento de comandos, manipulação de variáveis e condicionais. O jogador assume o papel do atendente de uma lanchonete e programa o comportamento do seu personagem para satisfazer os pedidos dos clientes. Os comandos fazem o personagem ir ao cliente, obter informações sobre o pedido, ir aos locais onde se preparam ou pegam as comidas e bebidas, e entregar os pedidos.

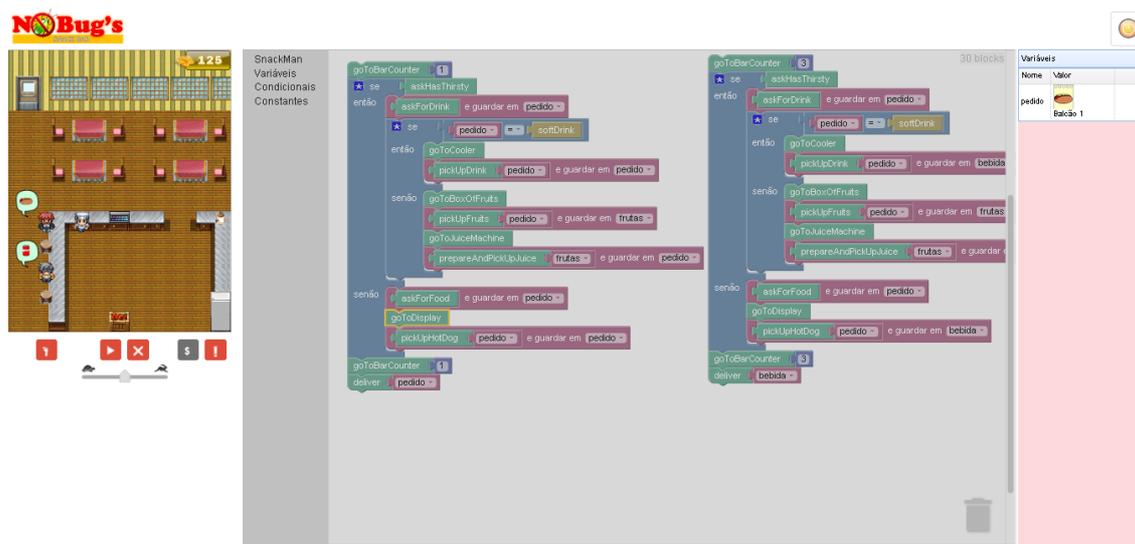


Figura 3. Interface do NoBug's Snack Bar.

Existem missões em que o jogador constrói a solução toda, outras em que são introduzidos alguns blocos que serão utilizados na solução final, e ainda missões em que são disponibilizados todos os blocos da solução para o jogador ordená-los. Algumas missões têm restrições quanto à quantidade de blocos e variáveis a serem usadas na solução. Outras fornecem bônus de pontuação se resolver o problema dentro de um limite de tempo. À medida que o jogador avança, o jogo fornece novos comandos. Se o jogador ficar inativo por alguns segundos, principalmente nas primeiras missões, o jogo apresenta dicas de uso de suas ferramentas tal como em um tutorial. O jogo possui recursos de depuração, permitindo que o jogador execute um comando por vez e visualize o estado das variáveis.

O jogo também apresenta três tipos diferentes de classificação: por quantidade de pontos, pelo tempo total gasto em todas as missões, e pelo total de tentativas para finalizar as missões. Isso cria um ambiente de motivação entre os jogadores através da competição pela mera comparação, pois esses recursos não fornecem pontuações extras ou qualquer outro tipo de bonificação. Como são três tipos de classificação, os jogadores podem ajustar a sua estratégia de resolução de acordo com aquela que pretendem melhorar. Por exemplo, a classificação pela quantidade de tentativas promove o jogador que tenta resolver através de testes de mesa com papel e caneta.

### 3.2. Mensuração

Para mensurar as opiniões dos estudantes a respeito dos jogos, foi elaborado um questionário com perguntas gerais sobre o perfil dos estudantes e sobre a percepção da aprendizagem e da dificuldade em aprender a programar. As perguntas específicas foram respondidas para cada um dos três jogos e referem-se ao estilo tutorial e à percepção de diversão, além de três perguntas abertas. Dois tipos de escala de resposta foram usados nas questões objetivas: (i) escala dicotômica (sim/não); e (ii) escala de diferencial semântico (intervalos de 1 a 5), onde em alguns casos o valor 1 representou sempre uma percepção mais negativa e o valor 5 uma percepção mais positiva conforme exemplifica a Figura 4.

**Figura 4.** Exemplo de escala utilizada.

Outra interpretação para a escala de 5 níveis também foi adotada para questões em que o nível intermediário (3) era o desejável. Por exemplo, uma questão sobre o ritmo em que os desafios evoluem teve a opção intermediária como sendo a mais positiva (ritmo adequado) e os níveis 5-muito rápido ou 1-muito lento como sendo negativos.

As escalas foram todas equilibradas contendo o mesmo número de itens ligados à percepção positiva, negativa e um item neutro. O Quadro 2 ilustra as questões do questionário e suas respectivas escalas.

Perguntas Gerais	
Perfil	Sexo (M ou F) Possuía algum contato anterior com programação? (sim/não)
Conceitos de programação e percepção sobre programar	Você acha que programar é difícil? (escala de 1-5) Você entendeu o que é uma "função"? (sim/não) Você entendeu o que é um "loop"? (sim/não) Você entendeu o que é uma "variável"? (sim/não) Dos jogos testados, qual achou mais divertido? Baseado na experiência com os jogos, você gostaria de estudar programação? (sim/não)
Perguntas Específicas (respondidas para cada jogo)	
Estilo Tutorial	Facilidade para aprender a jogar (escala de 1-5) Ritmo em que os desafios evoluem (escala de 1-5) Quantidade de explicações que o jogo fornece (escala de 1-5) Os objetivos das missões estavam claros? (escala de 1-5) Feedback do jogo em relação ao acertos e erros? (escala de 1-5)
Diversão	Nível de diversão (escala de 1-5) Nível de envolvimento com o jogo (escala de 1-5)
Perguntas Abertas	Qual foi sua principal dificuldade durante o jogo? (aberta) O que mais gostou no jogo? (aberta) O que menos gostou no jogo? (aberta)

**Quadro 2.** Questões do questionário aplicado na pesquisa.

O questionário foi implementado na ferramenta Google Forms e foi respondido pelos estudantes via web no mesmo ambiente onde as atividades do projeto de extensão foram realizadas. O tempo médio de preenchimento do questionário foi 25 minutos.

## 4. Resultados

As questões fechadas foram tabuladas e sintetizadas em gráficos e tabelas. E as questões abertas foram analisadas principalmente para auxiliar a compreensão dos escores numéricos.

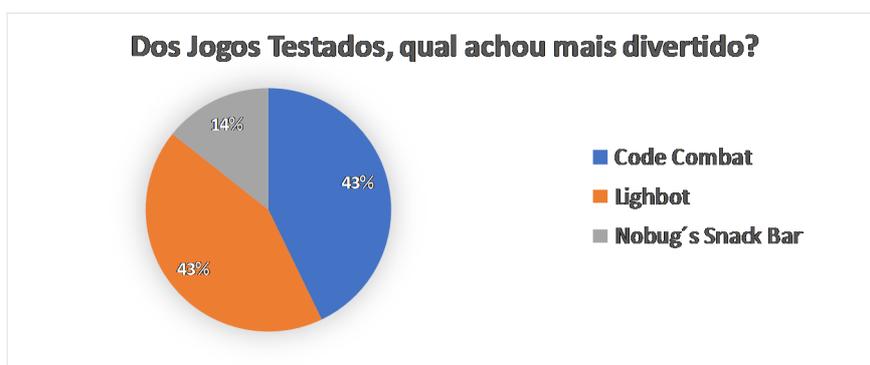
Dos quatorze estudantes que participaram da pesquisa apenas um tinha contato prévio com programação. As questões referentes ao estilo tutorial e à diversão que estavam em uma mesma escala foram somadas para formar um escore. Já a questão referente ao ritmo que os desafios evoluem foi codificada da seguinte forma: 2 pontos para a opção “3-ritmo adequado”; 1 ponto para as opções “4-ritmo acelerado” e “2-ritmo lento”; e 0 pontos para as opções “5- ritmo muito acelerado” e “1-ritmo muito

lento”. O escore calculado permite comparar os três jogos utilizados, conforme apresenta a Tabela 1.

Jogo	Estilo tutorial	Diversão
Code Combat	186	106
Lightbot	164	116
NoBug’s Snack Bar	183	85

**Tabela 1.** Escores de estilo tutorial e diversão.

Nota-se que o estilo tutorial dos jogos Code Combat e NoBug’s Snack Bar agradou mais aos estudantes, pois ambos fornecem explicações textuais e dicas durante a realização dos desafios. O Lightbot não é muito explicativo. O resultado da Tabela 1 é corroborado pela pergunta específica que pedia aos estudantes para indicarem o jogo mais divertido. A Figura 5 ilustra que não houve diferença entre Code Combat e Lightbot e que o NoBug’s Snack Bar foi apontado como o menos divertido.



**Figura 5.** Gráfico sobre qual jogo os estudantes julgaram mais divertido.

Com base nas respostas das perguntas abertas foi possível compreender melhor os escores indicados. A facilidade de uso do Code Combat deve-se às suas mecânicas que conduzem o jogador, deixando a dificuldade para os desafios propostos. Por outro lado, a repetição de objetivos acaba tornando o processo monótono para alguns, códigos de difícil compreensão acabaram impactando no nível de divertimento e no ritmo em que os desafios evoluem, assim como, dificuldades com o idioma inglês também foram relatadas.

Com relação ao Lightbot, a análise das respostas abertas mostrou que a maior dificuldade foi compreender inicialmente a forma de organizar os comandos, se na vertical, ou na horizontal, e quando usar funções ou apenas o método principal, visto que o jogo não apresenta um tutorial muito claro a respeito. Este foi o único jogo que obteve dos alunos um *feedback* negativo em relação à quantidade de explicações fornecidas ao jogador. A velocidade em que os desafios evoluem também foi identificada como a mais alta entre os três jogos testados.

Nas respostas do NoBug’s Snack Bar, a facilidade para aprender a jogar ficou abaixo do Code Combat. Entretanto, o resultado que mais se diferenciou dos demais, foi o nível de diversão, e do envolvimento com o jogo, onde o NoBug’s Snack Bar obteve as piores notas entre os três jogos. Os estudantes mencionaram que as fases se tornavam repetitivas, e por consequência o jogo ficava monótono. Esse jogo insere um conceito por vez em cada missão, se tornando o jogo mais lento quanto ao ritmo em que os desafios são apresentados. Assim como no Code Combat, também foram citados problemas com os comandos serem em idioma inglês. O ambiente de edição do

programa necessita de melhorias pois em certas missões, com a exigência de mais código, torna-se difícil a visualização do programa.

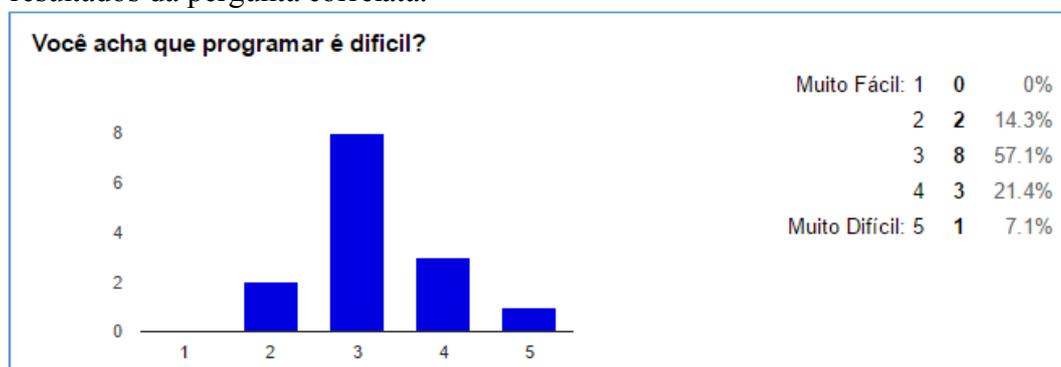
As questões gerais possibilitaram analisar o processo de compreensão dos conceitos de programação pelos estudantes, após terem jogado os três jogos. Ainda que alguns jogos não apresentassem estes conceitos explicitamente, o intuito não era comparar os jogos, mas sim avaliar a abordagem de introdução à programação por meio de jogos de programar. Com relação aos conceitos de programação envolvidos a Tabela 2 sintetiza as respostas.

Pergunta	Sim	Não
Você entendeu o que é uma função?	13 (92.9%)	1 (7.1%)
Você entendeu o que é uma repetição?	13 (92.9%)	1 (7.1%)
Você entendeu o que é uma variável?	10 (71.4%)	4 (28.6%)

**Tabela 2.** Compreensão dos conceitos de programação.

Pode-se verificar que conceitos normalmente mais avançados no contexto de disciplinas introdutórias, como repetições e funções são conceitos em que os estudantes apresentam mais segurança em comparação com um conceito geralmente inicial que é o de variáveis. Este aspecto está ligado à dinâmica dos jogos que reduzem parte dos desafios à construção de soluções com variáveis pré-estipuladas ou então implícitas (como por exemplo o passo do robô no Lightbot), desta forma reduzindo o contato do estudante com este conceito. Também representa a viabilidade de abordar estes conceitos sem necessariamente ver outros que são com frequência considerados pré-requisitos como operações relacionais e lógicas.

Com relação à percepção quanto a dificuldade de programar a Figura 6 apresenta os resultados da pergunta correlata.



**Figura 6.** Percepção sobre dificuldade de programar.

Nota-se que predomina uma percepção de que a dificuldade é média. Esta resposta fornece um indício interessante de que a abordagem com jogos pode auxiliar os estudantes a perderem o temor tradicionalmente relatado em pesquisas de aprendizagem de programação. A percepção da dificuldade está fortemente relacionada à ansiedade, que produz efeitos negativos na aprendizagem de programação (Connolly et al., 2009; Chang, 2004).

Dos 14 estudantes, 13 (92,9%) indicaram que gostariam de estudar programação futuramente. Este resultado fornece outro indício de que o uso da abordagem de jogos de programar pode ser positiva em relação a percepção da dificuldade e em reduzir a aversão à programação.

## 5. Considerações Finais

Ao analisar os resultados obtidos nesta pesquisa, pôde-se identificar que, apesar de utilizar a notação textual considerada por diversos pesquisadores como uma notação intimidadora à primeira vista, o Code Combat teve o melhor escore com relação à facilidade para aprender a jogar e teve um alto escore de estilo tutorial. Por ser um resultado com uma pesquisa de amostra pequena e enviesada (estudantes foram escolhidos pela escola por desempenho), não se pode afirmar que esta característica ocorra em outros contextos, mas é um indício que motiva a realização de uma futura investigação mais direcionada a avaliar o impacto proporcionado pelas diferentes notações de programação nos contatos iniciais dos estudantes.

A diversão é um elemento importante, pois possibilita ampliar o engajamento e a motivação que são elementos fundamentais para facilitar a aprendizagem. O Lightbot obteve as maiores pontuações para diversão e para o nível de envolvimento com o jogo, sendo que na última ficou empatado com o Code Combat.

Observou-se também que o NoBug's Snack Bar obteve o pior resultado em relação ao divertimento e ao nível de envolvimento com o jogo, indicando que o estilo tutorial utilizado (bastante explicativo) pode criar um ambiente menos atrativo. O equilíbrio entre diversão e instrução é um desafio permanente no projeto de software educacional e o mesmo vale para os jogos de programar.

Os estudantes relataram terem compreendido os conceitos de funções e repetição de forma mais sólida do que o conceito de variável. Este resultado também é um indício que necessita ser investido em amostras maiores e que pode indicar que a ordem geralmente sugerida pelos livros didáticos de algoritmos e programação é passível de revisão. Isto reflete também a natureza dos jogos que buscam dar algum significado imediato à solução que o estudante está criando no contexto do jogo, elemento que muitas vezes está ausente nas explicações feitas por professores em sala de aula e que pode ser fonte de desinteresse (Zanini e Raabe, 2012).

A tarefa de programar não foi vista como difícil, mas como uma tarefa de média dificuldade. A grande maioria das respostas foi favorável quando perguntado aos alunos se gostariam de estudar programação. Estes resultados indicam que a abordagem de uso de jogos de programar nos contatos iniciais dos estudantes pode trazer resultados benéficos ao criar um ambiente menos intimidador em que o estudante se percebe capaz de atingir seus objetivos educacionais enquanto se diverte.

Essa investigação apresenta uma limitação quanto ao procedimento da experimentação. A mesma amostra usou os mesmos jogos na mesma sequência, o que proporcionou um efeito de aprendizagem (Lazar, Feng e Hochheiser, 2010), em que os mesmos conceitos aprendidos no primeiro jogo são repetidos no segundo e terceiro jogos, e nada de novo efetivamente é exercitado, podendo levar os alunos a sensação de aborrecimento e fadiga, podendo comprometer, por exemplo, a avaliação da diversão.

As perspectivas futuras desta pesquisa incluem ampliar a coleta das opiniões dos estudantes com a realização de entrevistas e grupos focais. O foco estará em elucidar mais profundamente as características dos jogos que favoreceram ou dificultaram o aprendizado dos conceitos iniciais de Computação e impactaram na motivação para aprender a programar.

## Agradecimentos

AV agradece a bolsa de doutorado apoiada pelo CNPq/CAPES – Programa Ciência sem Fronteiras – CsF (6392-13-0) e autorização de afastamento da UDESC (688/13). GZ agradece pela bolsa de Taxas Escolares concedida pela CAPES, através do Programa de Suporte à Pós-Graduação de Instituições de Ensino Particulares (PROSUP). Os autores também agradecem a disponibilidade e participação dos alunos envolvidos no experimento.

## Referências

- CHANG, S.E. Computer anxiety and perception of task complexity in learning programming-related skills. *Computers in Human Behaviour*. 21, pp 713-728, 2004.
- CONNOLLY, C.; MURPHY, E. and MOORE, S. Programming Anxiety amongst Computing Student - A Key in Retention Debate? *IEEE Transactions on Education*, vol. 52, no. 1, pp 52 -56, 2009.
- CORTI, K. Games-based Learning: a serious business application. *Informe de PixelLearning*, v. 34, n. 6, pp 1-20, 2006.
- DE JESUS, E.; RAABE, A. Avaliação Empírica da Utilização de um Jogo para Auxiliar a Aprendizagem de Programação. *Anais do Simpósio Brasileiro de Informática na Educação*, 2010. v. 1. p. 10-20.
- DANN, B. W.; COOPER, S. Education Alice 3: Concrete to Abstract. *Communications of the ACM*, 52(8), 27–29, 2009.
- GOMES, A.; MENDES, J. A. Learning to program - difficulties and solutions. *CompSysTech International Conference on Engineering Education – ICEE*, 2007.
- KE, F. An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, pp. 26-39, 2014.
- KIILI, K. Digital game-based learning: Towards an experiential gaming model. *Internet and Higher Education*. v. 8, pp 13-24, 2005.
- KINNUNEM, P.; MC CARTNEY, R.; MURPHY, L.; THOMAS, L. Through the eyes of instructors: a phenomenographic investigation of student success. *Proceedings of the third international workshop on Computing education research*. pp 61-72, 2007.
- LAZAR, J.; FENG, J.H.; HOCHHEISER, H. *Research Methods in Human-Computer Interaction*. West Sussex, UK: John Willey & Sons, 2010.
- LONG, J. Just For Fun: Using Programming Games in Software Programming Training and Education — A Field Study of IBM Robocode Community. v.6, pp 279-290, 2007.
- MALLIARAKIS, C.; SATRATZEMI, M.; XINOGALOS, S. Educational games for teaching computer programming. In C. Karagiannidis, P. Politis, & I. Karasavvidis (Eds.), *Research on e-Learning and ICT in Education: Technological, pedagogical and instructional perspectives*, pp 87–98, New York: Springer, 2014.
- MICHAEL, D. R.; CHEN, S. L. *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade, 2006.

- OBLINGER, G. The next generation of educational engagement. *Journal of interactive media in education*. pp 1-18, 2004.
- OREHOVAČKI, T.; BABIĆ, S. Inspecting Quality of Games Designed for Learning Programming. *Lecture Notes in Computer Science - 17th International Conference on Human-Computer Interaction*, 2015.
- ORMEROD, T. Human cognition and programming. *Psychology of Programming*, Hoc, J.M.; Green, T.R.G.; Samurçay, R.; Gilmore, D.J. (Eds), pp. 63-82, 1990.
- PAPASTERGIOU, M. Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. v. 52, pp.1-12, 2009.
- PEREIRA, G.; BRISSON, A.; PRADA, R.; PAIVA, A.; BELLOTI, F.; KRAVICK, M.; KLAMMA, R. Serious Games for Personal and Social Learning & Ethics: Status and Trends. *Procedia Computer Science*, v. 15, p. 53-65, 2012.
- PRENSKY, M. *Digital Game-Based Learning*. New York: McGraw-Hill, , 2001a.
- PRENSKY, M. *Digital Natives, Digital Immigrants*. On the Horizon - NCB University Press, Vol. 9 No. 5, 2001b.
- ROGERSON, C.; SCOTT E. The Fear Factor: How It Affects Students Learning to Program in a Tertiary Environment. *Journal of Information Technology Education* v 9, 2010.
- SQUIRE, K.; JENKINS, H. Harnessing the power of games in education. *Insight*, v. 3, n. 1, pp 5-33, 2003.
- VAHLDICK, A.; MENDES, A. J.; Marcelino, M.J.P.; HOGENN, M.; SCHOEFFEL, P. Testando a Diversão em um Jogo Sério para o Aprendizado Introdutório de Programação, in 23° WEI - Workshop sobre Educação em Computação, (XXXV Congresso da Sociedade Brasileira de Computação), 2015.
- VAHLDICK, A.; MENDES, A.J.; MARCELINO, M.J., A review of games designed to improve introductory computer programming competencies, *Frontiers in Education Conference (FIE)*, 2014 IEEE , vol., no., pp 1-7, 22-25 Oct. 2014.
- WHITTON, N. *Learning with digital games: A practical guide to engaging students in higher education*. New York: Taylor & Francis. 2010.
- WING, J. M. Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society A*, 366 (July), pp 3717–3725, 2008.
- ZANINI, Adriana S.; RAABE, André L.A. . Análise dos enunciados utilizados nos problemas de programação introdutória em cursos de Ciência da Computação no Brasil. In: WEI - XX Workshop sobre Educação em Computação, 2012, Curitiba. XXX Congresso da Sociedade Brasileira de Computação. Curitiba: SBC, 2012.