

Pensamento Computacional Praticado com um Jogo Casual SériO no Ensino Superior

Adilson Vahldick^{1,2}, António José Mendes², Maria José Marcelino², Paulo Roberto Farah¹

¹Centro de Educação do Alto Vale do Itajaí (CEAVI)
Universidade do Estado de Santa Catarina (UDESC) – Ibirama, SC – Brasil

²Centro de Informática e Sistemas da Universidade de Coimbra (CISUC)
Universidade de Coimbra – Portugal
{adilson.vahldick, paulo.farah}@udesc.br, {toze, zemar}@dei.uc.pt

Resumo. Nesse artigo é discutida a importância para os alunos das disciplinas introdutórias de programação de cursos de graduação iniciarem o seu aprendizado com o desenvolvimento em habilidades de Pensamento Computacional (PC). Devido a restrições de tempo no semestre letivo, é difícil para os professores oferecerem essa oportunidade através de ambientes de autoria específicos, como Alice e Scratch. Por essa razão, foi criado um jogo casual sério para apoiar no desenvolvimento dessas habilidades. Jogos casuais são caracterizados pelas missões curtas e rápidas, e isso pode ser favorável como atividades extraclasse. Um ciclo de avaliação já foi concluído em um curso de graduação e outro está em andamento noutro curso. Os detalhes do jogo, da sua utilização, a opinião dos alunos e o parecer do professor são apresentados nesse artigo. A principal contribuição desse artigo é elencar alguns princípios que possam auxiliar no desenvolvimento de jogos para o aprendizado de programação através de tarefas com PC.

Abstract. This paper discusses the importance to develop Computational Thinking (CT) skills in undergraduate introductory programming courses. Due to time constraints, it is hard for teachers provide this opportunity through specific authoring environments, like Alice and Scratch. For this reason, it was created a serious casual game to support the development of these skills. Casual games are characterized by fast and short missions, and this can be favorable as extra classes activities. An evaluation cycle was finished in an undergraduate course and another is underway. The details of the game, the opinion of students and the teacher are presented in this article. The main contribution of this work is to list some principles that supports the development of games for programming learning through tasks with CT.

1. Introdução

Os jogos são experiências interativas fáceis e divertidas. Normalmente não exigem um treino ou ajuda externa, a não ser pelos comentários das pessoas que os apreciaram jogando (Prensky, 2001). A aprendizagem baseada em jogos (*Game Based Learning* – GBL) é 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) (Whitton, 2010). Os jogos sérios (ou educacionais) oferecem um contexto relevante que

ajuda a transformar conceitos abstratos em mais concretos (Eagle & Barnes, 2009). Usar um jogo envolve naturalmente atividades de resolução de problemas (Schell, 2008). É justamente a falha na apropriação da habilidade de resolução de problemas que resulta no maior fracasso na aprendizagem de programação (Robins et al., 2003).

O aprendizado em programação é um processo em que se exigem transformações cognitivas profundas e desenvolvimento no domínio de habilidades complexas (Gomes & Mendes, 2007). Escrever programas envolve compreender o problema e abstrair um modelo, dividir o problema em partes menores, decidir pela melhor estratégia para resolver cada uma das partes, e aplicar ou adaptar soluções já conhecidas (Winslow, 1996). As habilidades e competências (abstração, decomposição, análise, etc) para desempenhar esse processo é descrito como Pensamento Computacional (PC) (Wing, 2008). A investigação para o aprendizado do PC tem sido focada na educação básica (França & Tedesco, 2015) devido a possibilidade de agregá-lo com outras disciplinas como matemática e biologia.

Apesar de haverem vários jogos disponíveis para a prática e aprendizagem de programação e PC (Vahldick et al., 2014), existem lacunas para a sustentabilidade do processo ensino-aprendizagem. Esses jogos não abrangem com profundidade os assuntos na disciplina, e nem oferecem um sistema de acompanhamento do progresso dos alunos. Este artigo expõe as vantagens no aprendizado do PC no ensino superior nas disciplinas introdutórias de programação e apresenta a experiência no uso de um jogo casual sério para promover esse aprendizado. Jogos casuais exigem pouquíssimas instruções para aprender a jogá-los (Kadle, 2009). Para fins educacionais, essa característica favorece para que o aluno se foque mais nas tarefas instrucionais do que gastar seu tempo em aprender as mecânicas e demais elementos do jogo (Landers & Callan, 2011). O artigo descreve o jogo, quanto aos seus elementos e design instrucional. O trabalho está estruturado da seguinte forma: na segunda seção é justificada a importância no uso de blocos para o aprendizado de programação no ensino superior; a terceira seção fundamenta o uso de jogos no aprendizado de programação e PC; na quarta seção é apresentado o jogo desenvolvido na investigação; na quinta são descritos os experimentos com o jogo; na sexta os resultados dessas experiências; e por fim são realizadas as conclusões sobre os resultados.

2. Blocos para o aprendizado de programação no ensino superior

Tem-se observado a omissão na capacidade de resolver problemas em disciplinas como matemática e ciências nos jovens do ensino médio (Grover & Pea, 2013). Ensinar habilidades computacionais na educação básica brasileira tem sido o desafio e a investigação para tentar amenizar essa situação (França & Tedesco, 2015). Em relação a abordagem do ensino de PC nos cursos de graduação, existem propostas para as disciplinas de matemática visando atenuar os altos índices de evasão e reprovação em tais disciplinas (Barcelos et al., 2015).

Essas propostas ainda são esforços recentes e isolados. Várias gerações de ingressantes nunca tiveram contato com esse conhecimento, o que pode ser apreciado na Tabela 1 onde 66,7% corroboram com esse argumento. Essa tabela apresenta os dados demográficos de dois experimentos, descritos mais adiante, que envolveram 96 alunos de primeiro semestre dos cursos de graduação. As reprovações e evasões são altas nas disciplinas introdutórias de programação. Um dos problemas graves dos alunos é abstraírem a noção de como a máquina interpreta um programa (Du Boulay, 1986). Para apoiar no desenvolvimento dessa noção de máquina abstrata, a estratégia adotada por

ambientes educacionais de programação como Alice e Scratch é que os alunos iniciem com elementos concretos e gradativamente sejam conduzidos para a compreensão dos aspectos 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. Esses ambientes utilizam uma linguagem visual baseada na construção dos programas usando blocos (*Block-Based Programming* – BBP). Esses blocos representam fisicamente os comandos das linguagens de programação o que torna mais fácil a iniciação dos alunos principiantes e mantê-los atraídos à tarefa (Weintrop & Wilensky, 2015). Além disso, as animações funcionam como uma ferramenta natural de depuração, mostrando passo-a-passo a execução do programa e o estado das variáveis.

3. Jogos como meios para praticar o pensamento computacional

Alice e Scratch são ambientes de autoria, ou seja, permitem que o aluno livremente construa sua solução. Esses ambientes são ricos de recursos e oportunidades, porém, demandam tempo para o aluno desenvolver os programas. A sua adoção na disciplina de introdução de programação exige que se consumam horas/aula da ementa que foram planejadas no curso para o ensino de resolução de problemas com uma linguagem de programação real.

Jogos vem sendo usados na educação para introduzir e reforçar conceitos, e para praticar como uma alternativa aos exercícios tradicionais (Whitton, 2010). O maior desafio é a manutenção da autoestima e motivação para que os alunos insistam em tentar e vencer o jogo. Entretanto, a diversão permite que o aluno realize as suas tarefas de forma mais fácil, mesmo quando exige mais esforço (Prensky, 2001).

O principal fator para promover a aprendizagem da programação é que o aluno disciplinadamente pratique intensivamente (Robins et al., 2003). Entretanto, muitas vezes falta-lhes motivação para se envolverem nessas tarefas. A prática em resolver os problemas nos jogos, que são mais motivadores do que os exercícios tradicionais, promove a confiança e a experiência, construindo um conjunto estruturado de padrões de soluções, o que lhe será muito útil quando se deparar com problemas usando linguagens de programação reais (Barnes et al., 2007).

Os jogos casuais compartilham essa ideia de terem poucos objetivos nas tarefas. O jogador é estimulado a vencer rapidamente as missões e permanecer envolvido com o jogo. As recompensas com as vitórias podem ser convertidas em novos poderes, personalizações do seu personagem, ou em pontos para classificá-lo num ranking entre outros jogadores (Trefry, 2010). Para a educação, os jogos casuais podem ser úteis, porque, além da facilidade de uso, e poucas tarefas para manter o jogador sempre motivado, eles fornecem *feedback* imediato, a rejogabilidade de missões permite que se aprimorem e testem novas possibilidades, e são fáceis de serem instalados ou são executados em navegador (Landers & Callan, 2011).

4. Jogo NoBug's Snack Bar

Considerando os benefícios e oportunidades expostas nas seções 2 e 3, foi desenvolvido um jogo para o aprendizado e prática das técnicas do PC. A mecânica do jogo é inspirada nos jogos casuais de gestão de tempo. Neste tipo de jogo, as tarefas são divididas em passos sequenciais em que o jogador precisa cumpri-las dentro de um limite de tempo (Trefry, 2010). O contexto do jogo é baseado em uma lanchonete. O jogador controla o

funcionário e o jogo controla os clientes que efetuam pedidos. As ações do funcionário são: perguntar ao cliente o que ele deseja comer e/ou beber, ir às máquinas preparar ou pegar o alimento e/ou bebida, cobrar a conta do cliente, devolver-lhe o troco, e falar (imprimir mensagens na tela). Esse pequeno conjunto de ações permite desenvolver uma gama imensa de missões. Algumas missões são inspiradas em exercícios tradicionais como cálculo de totais, médias, menor e maior valores.

Todas as ações do funcionário são programadas com blocos. Para exemplificar uma missão, a Figura 1a ilustra a solução e a Figura 1b apresenta a área da lanchonete para essa missão. Essa é uma missão em que ainda não foram aprendidas as estruturas de repetição, por isso aparecem blocos duplicados. Seguindo passo-a-passo os blocos da Figura 1a: o atendente vai até o cliente da posição 1 e se ele estiver com sede então executa um conjunto de blocos que pergunta o que deseja beber, vai até a geladeira, pega a bebida, volta ao cliente e entrega a bebida. Depois o atendente vai até a posição 3 e realiza as mesmas ações para esse cliente.

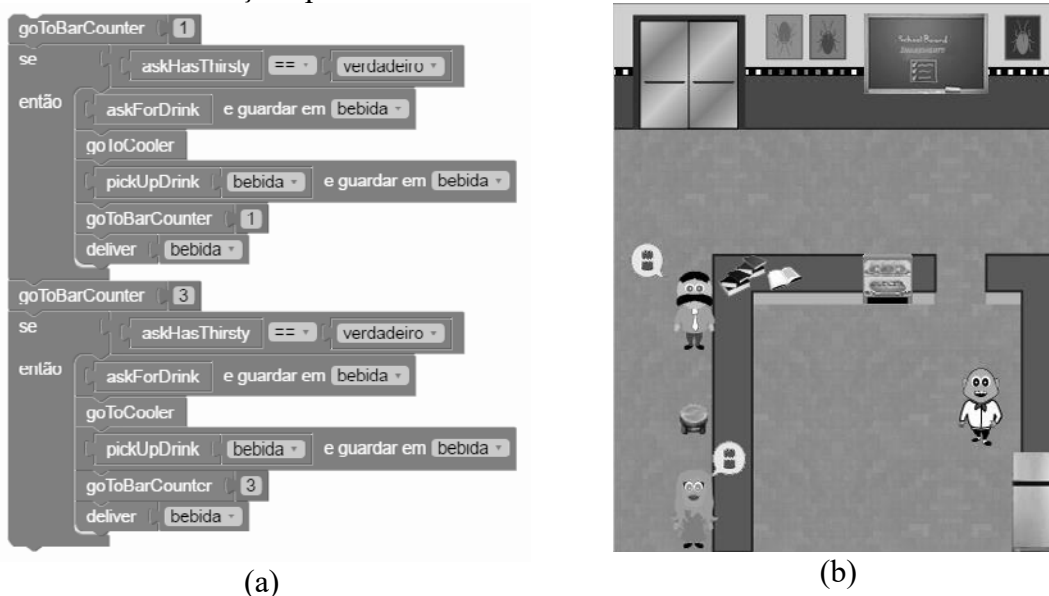


Figura 1. (a) blocos para resolver a missão; (b) área da lanchonete

O ambiente do jogo é apresentado na Figura 2. Ao centro está a área de trabalho, onde o aluno constrói a sua solução; à esquerda está a área de animação, com os botões para executar, depurar, parar e controlar a velocidade de animação, além da área da animação da lanchonete que é refletida conforme a solução vai sendo executada; à direita são apresentadas as variáveis quando o aluno depura a sua solução. Na parte superior existe um quadro com três estrelas, que se refere ao sistema de pontuação na missão. O jogador inicia com três estrelas quando entra a primeira vez em uma missão. Cada estrela corresponde a uma quantidade de pontos. A figura ilustra que cada estrela vale 30 pontos e o jogador já perdeu uma estrela. O consumo das estrelas pode variar entre as missões, mas existem duas formas: (1) por tempo ou (2) por tentativas. Mesmo que todas as estrelas tenham sido consumidas, ainda existe uma pontuação de motivação, que é bem menor, mas é garantida ao jogador quando finalizar a missão. Nas missões por tempo (1) é determinado um prazo para concluí-la. Cada terço desse prazo equivale a uma estrela. Quando encerra o tempo, todas as estrelas estão consumidas, mas o jogador continua a missão até finalizar, para então receber a pontuação de motivação. Nas missões por tentativas (2), como ilustrado na Figura 2, existe um contador de tentativas ao lado das estrelas. Cada vez que o aluno inicia uma execução ou depuração, é consumida uma

tentativa. Pode-se observar que o jogador já consumiu quatro tentativas: três para a primeira estrela e mais uma tentativa na segunda estrela.

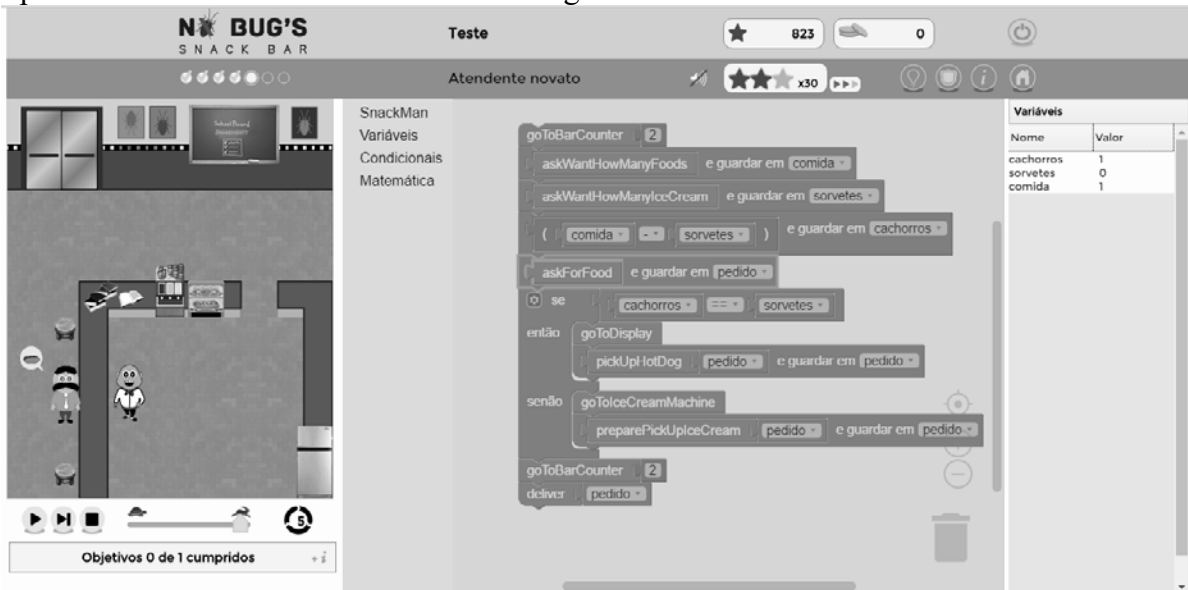


Figura 2. Ambiente do jogo

O jogo apresenta três rankings ordenando os alunos pela pontuação (1), pelo tempo total gasto para cumprir as missões (2) e pela quantidade total de tentativas (3). Além disso, conforme o aluno vai acumulando pontos são liberadas opções para personalizar seu *avatar*: cor dos olhos, estilo e cor dos cabelos/bigode, cor da pele, estilo e cor do chapéu e cor das roupas.

O professor pode acompanhar o desempenho dos alunos de duas formas: consultando as suas conquistas, ou a situação dos alunos nas missões. O jogo tem um sistema de conquistas, ou recompensas, em que fornece medalhas ao aluno em algumas situações definidas pelo professor como, por exemplo, finalizar todas as missões de uma fase dentro de um prazo. A Figura 3 apresenta a tela usada pelo professor para consultar a situação dos alunos. Cada quadrado representa uma missão e indica quantos alunos estão nela. Ao clicar na missão são listados os alunos com suas quantidades de tentativas e tempo consumido na missão. O professor pode consultar a resposta de cada tentativa e contribuir com o aluno enviando-lhe um e-mail ou mesmo discutindo o problema em sala de aula.



Figura 3. Área de administração do professor

5. Métodos, materiais e participantes

O processo investigativo no desenvolvimento do jogo é constituído principalmente de dois ciclos de avaliação em cursos de graduação em informática. No primeiro participaram 60 alunos durante dois meses numa universidade portuguesa. O segundo ciclo iniciou no dia 26/02/2016 e está em andamento com 36 alunos numa universidade brasileira. A amostra é descrita na Tabela 1. A maior parte da amostra portuguesa é do sexo feminino entre 18 e 19 anos sem hábito de jogar. A maior parte da amostra brasileira é do sexo masculino abaixo dos 18. Dois extremos empataram com a maior frequência nessa amostra em relação aos hábitos de jogar: os que jogam diariamente e os que não costumam jogar. Além disso, como já discutido na seção 2, a maior parte da população nunca teve contato prévio com algoritmos ou programação.

Tabela 1. Dados demográficos dos participantes

	Sexo		Idade			Frequência que joga				Conhecimento prévio			
	M	F	<=17	18 ou 19	>=20	Diariamente	Semanalmente	Mensalmente	Não tem hábito de jogar	Nenhum	Fiz umas cadeiras	Fiz um curso técn./superior	Trabalho com programação
Portugal	22 (36,7%)	38 (63,3%)	9 (15,0%)	31 (51,7%)	20 (33,3%)	19 (31,7%)	11 (18,3%)	8 (13,3%)	22 (37,7%)	36 (60,0%)	23 (38,3%)	1 (1,7%)	0 (0,0%)
Brasil	31 (86,1%)	5 (13,9%)	17 (47,2%)	10 (27,8%)	9 (25,0%)	12 (33,3%)	10 (27,8%)	2 (5,6%)	12 (33,3%)	28 (77,8%)	7 (19,4%)	1 (2,8%)	0 (0,0%)
Total	53 (55,2%)	43 (44,8%)	26 (27,1%)	41 (42,7%)	29 (30,2%)	31 (32,3%)	21 (21,9%)	10 (10,4%)	34 (35,4%)	64 (66,7%)	30 (31,2%)	2 (2,1%)	0 (0,0%)

O jogo está disponível em um servidor web (<http://nobugssnackbar.dei.uc.pt/>) e os alunos podiam jogá-lo a qualquer momento diretamente no seu navegador. Em ambos os ciclos o jogo foi usado como uma atividade extraclasse: os alunos jogavam fora do horário de aula, e aconselhados a jogarem antes do assunto relacionado em sala.

No ciclo em Portugal foram desenvolvidas 55 missões que eram jogadas em série agrupadas em fases. Cada fase era relacionada a um assunto: sequência de comandos, manipulação de variáveis, condicionais, ciclos, *arrays* e funções. Não havia conexão nas aulas com o jogo e nem qualquer tipo de atribuição de notas em relação ao cumprimento das missões.

As observações e conclusões do ciclo em Portugal fomentaram mudanças nos elementos do jogo, em aspectos instrucionais e na integração com o currículo. Quanto ao ciclo no Brasil, até o momento foram desenvolvidas 48 missões (algumas reaproveitadas do experimento anterior) com os assuntos sequência de comandos, manipulação de variáveis, estruturas condicionais e ciclos. A sequência instrucional das fases (Figura 4) seguiu um padrão que oferece mais oportunidades ao aluno.

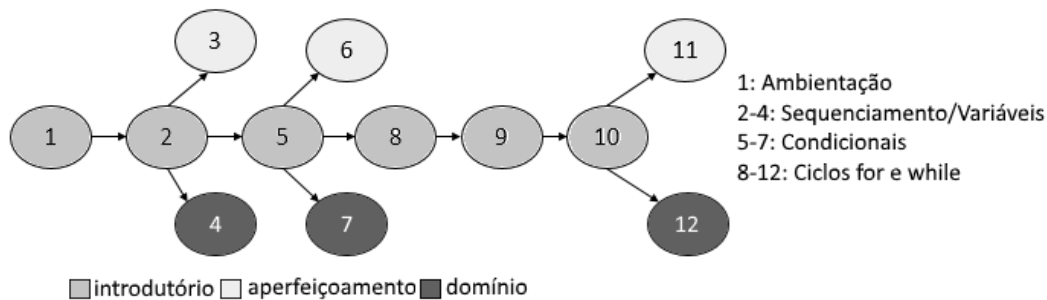


Figura 4. Sequência das fases no segundo ciclo

As fases foram classificadas em três níveis de dificuldade: (1) introdutório, que abrange de forma superficial todo o conteúdo básico de um assunto; (2) aperfeiçoamento em que se aprofunda no assunto; e (3) domínio que é formada por missões que desafiam com situações complexas e restritivas (por exemplo, limite de comandos e/ou variáveis). As fases introdutórias são pré-requisitos umas das outras, e das fases de aperfeiçoamento e domínio. Por exemplo, se o aluno concluir a fase 2, ele pode decidir jogar livremente entre as fases 3, 4 ou 5.

O processo de aprendizagem dentro das fases também foi revisto. A quantidade de missões foi reduzida em cada fase. Os tipos de tarefas a serem realizadas em cada missão foram inspiradas no desenvolvimento cognitivo proposto na Taxonomia de Bloom (Ferraz & Belhot, 2010). A Tabela 2 apresenta a relação entre cada categoria do domínio cognitivo com o tipo de tarefa nas missões. As fases introdutórias e de aperfeiçoamento possuem aproximadamente oito missões, obedecendo a sequência das categorias, mas variando na sua distribuição. Por exemplo, uma fase pode iniciar com duas missões de conhecimento, e uma de compreensão. Em seguida alterna entre duas de aplicação e análise. Essas fases sempre finalizam com uma missão da categoria síntese. As fases de domínio possuem menos missões (em torno de quatro) e não seguem essa sequência.

Tabela 2. Relação entre as categorias dos domínios cognitivos da Taxonomia de Bloom e os tipos de tarefas das missões

Categorias	Tipos de Tarefas
1. Conhecimento	Corrigir erros: é fornecida a solução com erros. O aluno precisa alterar a variável que está sendo referenciada, trocar operadores de comparações ou lógicos, trocar blocos de posição ou inserir novos blocos.
2. Compreensão	Organizar blocos: todos os blocos já estão disponíveis espalhados na área de trabalho e o aluno precisa organizá-los na sequência correta.
3. Aplicação	Construir iniciando com sugestões: são fornecidos alguns blocos que podem ser o início, o fim ou uma parte central da solução. O aluno cria a solução, inserindo novos blocos, para completar a parte que falta.
4. Análise	Construir: o aluno cria sua solução a partir do zero.
5. Síntese	Construir com restrições: as tarefas têm restrições quanto a quantidade de blocos usados, quantidade de vezes que pode ser usado um determinado bloco, e/ou a quantidade de variáveis que podem ser usadas.

Para melhorar a integração curricular, foi desenvolvido um sistema de conquistas: o aluno recebe um distintivo quando finaliza uma fase até uma data definida pelo professor. Em compensação, o aluno recebe bonificações na nota de uma avaliação (teste/prova). Além disso, o professor utiliza exemplos do jogo (desenho dos blocos ou soluções) nas aulas introdutórias dos assuntos da disciplina.

Ainda existem outras três diferenças significativas entre os experimentos: foram melhoradas a interface com o usuário e a qualidade das imagens (as Figuras 1 e 2 são do segundo experimento), a forma de oferecer *feedback* e de consumir estrelas. No primeiro experimento, durante a missão, o jogo fornecia muitas dicas para o aluno ultrapassar as suas dificuldades e resolver os erros. O consumo das estrelas nas missões era na totalidade pelo tempo. No segundo experimento, o consumo das estrelas é frequentemente por tentativas. O jogo praticamente não fornece dicas. Em contrapartida, foi desenvolvido um novo recurso que mostra a resposta em português estruturado nas missões do tipo “construir” das fases introdutórias e de aperfeiçoamento. O custo para usar esse recurso é consumir todas as estrelas da missão.

6. Resultados

O primeiro aspecto avaliado é quanto à dedicação e interesse dos alunos no jogo. Para isso, foram comparadas as quantidades de alunos que concluíram as missões. A Figura 5 apresenta os resultados entre as duas experiências respectivamente. A Figura 5a apresenta somente as primeiras 19 missões porque a partir dessa somente dois alunos seguiram

adiante e concluíram o jogo. A Figura 5b apresenta a frequência de alunos em todas as missões em que algum aluno já tenha concluído. As duas últimas missões do gráfico correspondem a fase 8 disponibilizada na véspera da redação desse artigo. Como existe uma certa liberdade nas fases em que jogam, é possível notar que a missão 23 tem menos alunos concluintes que as posteriores. A missão 23 é a última da fase 4, que é a fase de domínio no assunto de manipulação de variáveis.

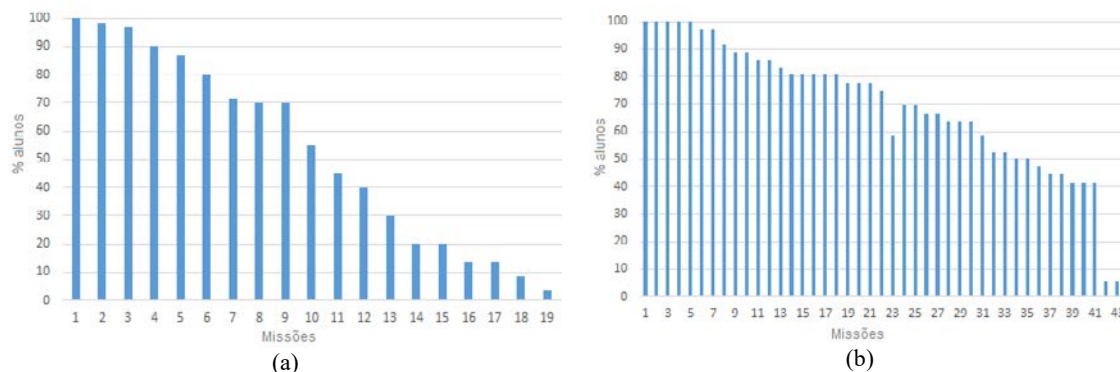


Figura 5. Relação entre alunos e missões concluídas na experiência (a) portuguesa e (b) brasileira

O segundo aspecto avaliado na experiência brasileira foi quanto a opinião dos alunos na aprendizagem percebida por eles. Após os alunos concluírem três fases, o jogo fornece um inquérito com seis questões na escala *Likert* (1-discordo completamente a 5-concordo completamente). A Figura 6 apresenta a distribuição das respostas dos 26 alunos (72,2%) que responderam ao inquérito.

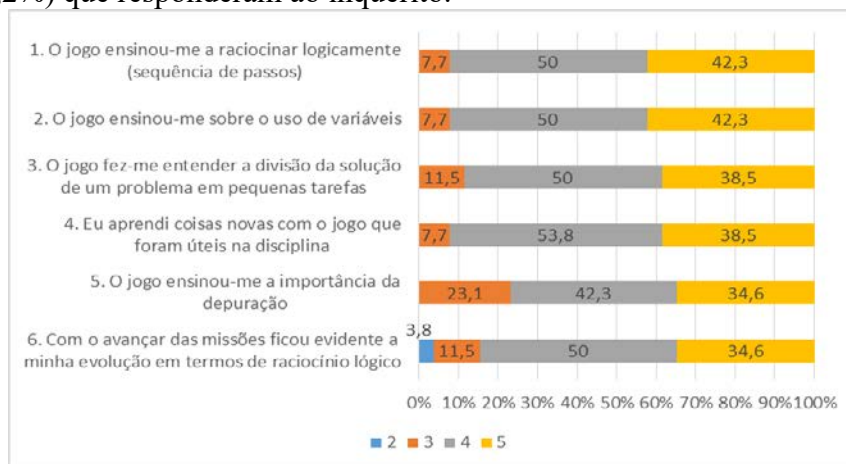


Figura 6. Aprendizagem percebida pelos alunos

O terceiro aspecto avaliado verificou quanto o recurso para mostrar o resultado em português estruturado foi intensamente usado, podendo representar alguma deficiência no processo instrucional. 15 missões (31,3%) têm esse recurso. Esse conjunto de missões foi concluído 380 vezes e o recurso foi usado 40 vezes (10,5%).

O último aspecto avaliado foi a percepção do professor em relação ao comportamento dos alunos. Segundo ele, “os alunos conectam rapidamente o conteúdo que estamos vendo em sala com os conceitos vistos anteriormente no jogo. Como professor tive a impressão que a maioria dos alunos já tinham noção do que eu estava explicando e até se arriscavam com perguntas mais avançadas”.

7. Discussão dos resultados

Existem várias investigações sobre o desenvolvimento de habilidades de PC na educação básica. Contudo, há uma lacuna em pesquisas consoantes ao público-alvo do ensino superior, e como apresentado nesse trabalho, a maior parte desse público não é familiarizado com o assunto. Nesse artigo foi apresentado um jogo casual para a prática dos conceitos de PC. O jogo foi desenvolvido e evoluído através de uma experimentação durante dois meses com 60 alunos de graduação numa universidade portuguesa. Os resultados dessa experiência propuseram mudanças no jogo e na sua forma de usar. Um novo experimento em andamento está sendo conduzido em um curso de graduação numa universidade brasileira. As três semanas de experimento mostraram avanços na motivação em jogar e nas atitudes e comportamentos dos alunos em sala de aula.

A principal contribuição desse artigo para o desenvolvimento de jogos sérios no ensino de programação é quanto ao design instrucional e integração curricular. A forma como estava organizada a sequência de fases proposta na segunda experiência disponibilizava ao aluno um maior conjunto de missões para jogar. Isso influenciou para que o aluno não parasse de jogar caso estivesse com dificuldade em vencer uma missão. No intuito de direcionar de forma coerente, clara e concisa o processo de ensino, a sequência instrucional dentro das fases seguiu a Taxonomia de Bloom que definiu os tipos de tarefas de cada missão. Segundo a opinião dos alunos, eles acreditavam que estavam aprendendo com o jogo (Figura 6). Esse sentimento vem a ser confirmado com o julgamento do professor que observou maior confiança e autoestima nos alunos, a ponto de relacionar essas atitudes com alunos do segundo semestre. Acreditamos que um componente que ajudou na motivação em continuar jogando é a integração do jogo com a disciplina, seja na questão didática ao utilizar os blocos e problemas do jogo nas aulas introdutórias, seja na relação de serem bonificados com pontos extras na próxima avaliação. Outro indício de que os alunos estejam aprendendo é a utilização de um recurso para mostrar o resultado em português estruturado. O recurso consome toda a pontuação da missão. Em 31,3% das missões é oferecido esse recurso. Entretanto, ele foi utilizado somente em 10,5% das vezes que os alunos concluíram essas missões.

O uso do jogo para praticar o PC na disciplina introdutória de programação permitiu de forma mais concreta que os alunos aprendessem conceitos abstratos de programação. Desenvolver a solução com blocos evitou problemas com sintaxe e facilitou a composição de estruturas mais complexas, como os condicionais e repetições. A animação integrada à solução permitiu que os alunos observassem o resultado de cada comando e percebessem a noção de máquina sequencial. A maior parte dos alunos ainda chega no curso superior sem nenhum contato com PC. O jogo facilitou aos alunos aprenderem conceitos de PC antes de alcançarem o mesmo assunto em aula. A vantagem no Ensino Superior em usar o jogo ao invés de um ambiente como Scratch ou Alice é que permitiu mais oportunidades de aprendizado aos alunos sem dispender aulas específicas para isso, e assim o professor continuar cumprindo com a ementa original da disciplina.

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). Os autores também agradecem a disponibilidade de professores e alunos envolvidos no experimento.

Referências

Barcelos, T., Muñoz, R., Villarroel, R., & Silveira, I. (2015). Relações entre o Pensamento Computacional e a Matemática através da construção de Jogos Digitais. In *IV Congresso*

- Brasileiro de Informática na Educação* (pp. 1369–1378).
- Barnes, T., Powell, E., Chaffin, A., Godwin, A., & Richter, H. (2007). Game2Learn: Building CS1 learning games for retention. In *12th SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 121–125). Dundee, Scotland.
- Dann, B. W., & Cooper, S. (2009). Education Alice 3: Concrete to Abstract. *Communications of the ACM*, 52(8), 27–29.
- Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57–73.
- Eagle, M., & Barnes, T. (2009). Experimental evaluation of an educational game for improved learning in introductory computing. In *40th ACM Technical Symposium on Computer Science Education* (pp. 321–325). Chattanooga, TN.
- Ferraz, A. P. D. C. M., & Belhot, R. V. (2010). Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gestão & Produção*, 17(2), 421–431.
- França, R., & Tedesco, P. (2015). Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil. In *IV Congresso Brasileiro de Informática na Educação* (pp. 1464–1473).
- Gomes, A., & Mendes, A. J. N. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education* (pp. 1–5). Coimbra, Portugal.
- Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.
- Kadle, A. (2009). Casual and Serious Digital Games for Learning – Some Considerations. Retrieved December 27, 2015, from <https://www.upsidelearning.com/blog/index.php/2009/04/18/casual-and-serious-digital-games-for-learning-some-considerations/>
- Landers, R. N., & Callan, R. C. (2011). Casual social games as serious games: The psychology of gamification in undergraduate education and employee training. In M. Ma, A. Oikonomou, & L. C. Jain (Eds.), *Serious Games and Edutainment Applications* (pp. 399–423). Springer London.
- Prensky, M. (2001). *Digital game-based learning*. New York: McGraw-Hill.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.
- Schell, J. (2008). *The art of game design: A book of lenses*. Burlington, MA: Elsevier Inc.
- Trefry, G. (2010). *Casual game design: Designing play for the gamer in all of Us*. Burlington, MA: Elsevier.
- Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. In *44th Annual Frontiers in Education Conference* (pp. 781–787). Madrid, Spain.
- Weintrop, D., & Wilensky, U. (2015). To block or not to block , that is the question : Students’ perceptions of blocks-based programming. In *ACM SIGCHI Interaction Design and Children* (pp. 199–208). Medford, MA.
- Whitton, N. (2010). *Learning with digital games: A practical guide to engaging students in higher education*. New York: Taylor & Francis.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society A*, 366(July), 3717–3725.
- Winslow, L. E. (1996). Programming pedagogy: A psychological overview. *ACM SIGCSE Bulletin*, 28(3), 17–22.