Animações interativas no ensino de algoritmos

André Gustavo Schaeffer

Universidade Federal da Fronteira Sul (UFFS)
Av. Dom João Hoffmann 313, bairro Fátima. CEP 99700-000 - Erechim - RS - Brasil
andre_schaeffer@uffs.edu.br

Abstract. *The present paper discusses the use of interactive animations in the teaching of algorithms supported by theories and past experiences regarding the use of multimediatic resources in terms of appropriation of scientific knowledge. It also presents and analyzes an animation in the form of a learning object, developed in the software Alice, that shows the operation of the Bubble Sort method for sorting arrays, proposing the use of that software as an appropriate tool for teachers be able to create animations for didactic purposes.*

Resumo. *O presente artigo aborda o uso de animações interativas no ensino de algoritmos amparado em teorias e experiências anteriores quanto a utilização de recursos multimidiáticos voltados à apropriação de conhecimentos científicos. Também apresenta e analisa uma animação na forma de um objeto de aprendizagem, desenvolvido no software Alice, que mostra o funcionamento do algoritmo de ordenação de vetores Bubble Sort, em defesa do uso desse software como uma ferramenta apropriada para que professores consigam criar animações com fins didáticos.*

1. Introdução

Pode parecer excessivamente recorrente o surgimento de novas alternativas para ensino de lógica da programação. Um forte argumento seria o fato de estarmos constatando um crescimento a passos largos da informática nas últimas décadas, voltando-se mais especificamente ao desenvolvimento de software, impulsionado por habilidosos programadores que, dependendo do tempo de experiência, foram envolvidos em processos de ensino e de aprendizagem ainda antes do advento e da disseminação de recursos multimidiáticos voltados à educação. Em outras palavras, muito se avança mesmo que se tenha lançado mão "somente" de métodos tradicionais de ensino, ou seja, sem a presença de recursos multimidiáticos avançados.

O presente artigo abordará o uso de recursos multimidiáticos voltados ao ensino e objetiva analisar, ao final, uma nova possibilidade de uso de animações para o ensino de lógica da programação (algoritmos), resgatando alguns estudos em que autores revelam que não necessariamente a simples presença de recursos multimidiáticos no ensino significa garantia de melhorias de aprendizagem (razão pela qual a palavra

somente aparece entre aspas no parágrafo anterior). Para esta análise, uma animação interativa na forma de um objeto de aprendizagem foi criada com vistas a demonstrar a movimentação de valores durante a execução de um algoritmo de ordenação de vetores (método Bubble Sort). Algumas características desse objeto de aprendizagem criado serão confrontadas com apontamentos de estudos recentes mais amplos sobre aprendizagem multimídia interativa objetivando incentivar a apropriação dessas novas ferramentas de desenvolvimento de animações amparando-se no conhecimento já produzido por esses estudos.

Sabemos que ainda é grande o entusiasmo e a expectativa que se deposita sobre conteúdo multimidiático voltado para o ensino, quando ponderação parece ser o melhor caminho, não caracterizando-se, necessariamente, o uso de vídeos e demais conteúdos multimídia como uma abordagem melhor do que as outras. Outro enfoque deste artigo é a defesa de que o surgimento de diferentes abordagens pedagógicas para o ensino da programação não precisa ser visto como excessivo e recorrente, uma vez que seria demasiadamente ingênuo pensar que lógica da programação tende a se limitar cada vez mais aos bancos escolares do ensino superior e somente em cursos de ciência da computação e afins. Pelo contrário. Expande-se cada vez mais nas diferentes ciências e na matemática, bem como pulveriza-se pelos currículos da educação básica. Se atualmente, para a formação em engenharias, a programação de computadores não passa de formação complementar ou disciplina optativa, o forte crescimento da modelagem matemática amparada em algoritmos e de nossa percepção de um mundo cada vez mais conectado através de sensores, microssensores e nanossensores, induzirá a uma nova reflexão quanto a possibilidades e necessidades de atuação dos futuros engenheiros. Além disso, o público-alvo que está entrando em contato com tais conceitos também está mudando. Mesmo que ficássemos restritos ao ensino superior, a expansão do acesso às universidades, experimentado nos últimos anos, tem colocado estudantes frente ao estudo da construção de algoritmos computacionais ainda que muitos deles tragam deficiências do ensino médio em conteúdos considerados fundamentais para o bom aproveitamento em uma disciplina de lógica da programação. A grande disponibilidade de conteúdos multimidiáticos na internet voltados a esse fim, nesse entendimento, não chega a surpreender, pois o público foi ampliado, houve diminuição de complexidade em termos de produção e publicação de conteúdos multimídia, e ganha força e importância o ensino de algoritmos em um mundo cada vez mais customizável, conectado e automatizado, às bases de revoluções que se iniciam com as impressoras 3D e a internet das coisas.

2. Recursos multimidiáticos no ensino

Com o advento dos recursos multimidiáticos em computadores pessoais, cresceu o interesse por parte dos professores da educação básica em utilizá-los a fim de facilitar o aprendizado dos estudantes. De certa forma, o alto custo e a alta complexidade de uso de aplicativos multimídia para o desenvolvimento de conteúdo, limitaram as possibilidades aos recursos disponíveis em aplicativos como editores de texto, editores

de vídeo, editores de planilhas eletrônicas e aqueles com ênfase em apresentação de slides, como o Microsoft Powerpoint e o Libre Office Impress. Isso fez com que animações mais detalhadas e complexas acabassem sendo desenvolvidas somente mediante a constituição de equipes multidisciplinares em grandes projetos com algum dinheiro disponível, ficando então aquele professor, sem conhecimento de linguagens de programação, limitado a esses editores e aos aplicativos de slides.

Uma das contribuições deste artigo será mostrar o resultado de pesquisas que apontam que ao se trabalhar com tais aplicativos não se está tão limitado assim, uma vez que a simples presença de animações ou vídeos não determina sempre vantagens em comparação a imagens estáticas, e que a abordagem pedagógica utilizada, o conteúdo trabalhado e habilidades metacognitivas de visualização têm um impacto ainda maior sobre o aprendizado do que a forma como o conteúdo é apresentado. Para Bidarra (2009) a investigação ainda não demonstrou categoricamente que as representações mais realistas e com maior resolução têm sempre vantagens sobre as representações diagramáticas mais simples, e que seria o mesmo que afirmar que um filme em cores é sempre melhor do que um filme em preto-e-branco.

No estudo comparativo entre apresentações estáticas versus dinâmicas, Hegarty (2004) mostra que pesquisas recentes falham em provar claras vantagens das dinâmicas em comparação com as estáticas, e que embora estudos apontem que animações têm efeito positivo em relação à motivação do estudante e ao aprendizado implícito, outros estudos apontam vantagens das apresentações estáticas em comparação às dinâmicas quando se trata de aprendizado conceitual. Vavra et al. (2011) examinaram estudos com resultados parecidos, em que os autores observaram que estudantes expostos a animações 3D, interativas ou não, demandaram mais tempo para aprender a tarefa que aqueles estudantes expostos a ilustrações 3D. O grupo de estudantes exposto às animações também demonstrou ter maiores dificuldades na construção de informações relevantes ao conteúdo já que aquelas animações avançavam muito rapidamente. Em contrapartida, esses autores afirmam que as mídias dinâmicas levam vantagem em relação às mídias estáticas no tocante ao conteúdo a ser estudado: quanto mais complexo for, maiores são as vantagens de seu uso. Ainda apontam que tais mídias dinâmicas somente seriam apropriadas e facilitariam o aprendizado quando representassem um modelo significativo de um processo ou sistema, levando em conta os limites do sistema cognitivo e sendo construídas tomando-se por base os conhecimentos prévios dos aprendizes.

Vavra et al. (2011) também sugerem que animações sejam usadas para explicar conceitos que não podem ser vistos, como colisões subatômicas de partículas. Se formos analisar, poderíamos enquadrar aqui tudo aquilo que escapa aos nossos sentidos. É claro que neste particular, ao se tratarem de animações em computador, restringimos nossa interface aos sentidos da visão e da audição. No viés do ensino de lógica da programação temos uma situação que se enquadra sobremaneira a esses dois últimos apontamentos. O primeiro enquadramento diz respeito a uma construção da animação levando-se em conta os conhecimentos prévios dos aprendizes. Então observemos que

as atribuições de valores a variáveis em um programa, em particular variáveis contadoras e acumuladoras, sempre levam em conta o valor atual da variável para então acrescer a este valor algum outro valor qualquer a ser acumulado ou o número 1 no caso de uma variável contadora. Para citar somente alguns exemplos, a "equação" do comando de atribuição tem o formato “x:=x+1” (linguagem Pascal), “x = x+1” (linguagem C) ou “x ← x+1” (linguagem algorítmica). O conhecimento prévio do aluno, herança da matemática do ensino médio, não aceita que x possa ser igual a ele mesmo mais 1. Deve-se ensinar a ler, então, que x recebe o valor atual de x mais 1. Finalmente, o aluno deve assimilar que o novo valor obtido à direita da igualdade (ou do sinal de atribuição) deve ser atribuído à variável à esquerda da igualdade, substituindo o valor anterior daquela variável. Este primeiro enquadramento leva em conta o conhecimento prévio do aluno mas, ao contrário, não constrói a partir disso, e sim, a partir, primeiramente, da desconstrução do conceito. O segundo enquadramento diz respeito a uma construção da animação para explicar conceitos que não podem ser vistos. Uma animação ancorada em uma situação familiar ao estudante pode facilitar a compreensão das sequências de comandos em estruturas de programação. De fato, não podemos ver como um laço de repetição é executado, ao menos em um nível de abstração mais didático. Em mais baixo nível (tecnicamente se falando - instruções de máquina), ainda que pudéssemos visualizar os acessos aos registradores, à pilha do sistema, à unidade lógico-aritmética e aos demais componentes da CPU envolvidos, tal visualização poderia gerar um pouco de confusão.

Hegarty (2004) escreve que, em geral, as pessoas assumem que quanto mais realística for uma animação dinâmica, mais efetiva ela será em termos de aprendizagem. Porém, nesse mesmo estudo, a autora contrapõe os resultados de vários autores que apontam que tal efetividade na aprendizagem terá relação com a capacidade da animação em abstrair a realidade, ou até mesmo distorcê-la de diferentes maneiras. As ferramentas de programação possuem opções de verificação de erros (opções de *debug*), que permitem inspeção do valor de variáveis e de endereços de memória, e que serviriam para esses propósitos, mas para um primeiro contato de um estudante com lógica de programação tais opções poderiam causar nele uma sobrecarga informacional. Além dele ter que focar na desconstrução de conceitos para poder se adequar à nova linguagem sendo utilizada, o estudante ainda teria que entender como fazer uso de tais ferramentas.

O uso de múltiplos meios no aprendizado também encontra amparo na Teoria da Codificação Dual, de Allan Paivio, que, conforme Vavra et al. (2011), tem seu foco voltado para a visualização como meio de compreender como a informação linguística (palavras e sentenças) e a informação visual (imagens) são codificadas por sistemas mentais independentes: um verbal e outro não-verbal. Sua premissa fundamental é, portanto, que as visualizações sejam acompanhadas por instruções relevantes. A informação armazenada em cada um desses sistemas pode ser acessada de forma também independente e a combinação da informação linguística e da informação visual dão melhor suporte ao aprendizado.

3. Animações para o ensino de algoritmos

Com vistas a facilitar o ensino de algoritmos amparado no que os próprios autores chamam de apresentações cognitivamente projetadas, o trabalho de Narayanan e Hegarty (2002) mostra os resultados da avaliação do Software Visualization System, programa de computador que faz parte de um conjunto de apresentações hipermídia criado e avaliado por eles. Em sua abordagem, comparam aspectos do ensino de algoritmos com o ensino do mundo mecânico (concreto, não abstrato, com forma física). Afirmam, então, que algoritmos são entidades abstratas, sem forma física, e que no domínio mecânico os sistemas são feitos de componentes com formas e características físicas familiares a quem os estuda. Complementam explicando que os componentes de uma máquina operam de acordo com as leis da física e da causalidade, e, portanto, até mesmo iniciantes possuem algum conhecimento intuitivo em assuntos relacionados a este domínio. De forma contrária, os algoritmos operam de acordo com leis matemáticas e lógicas sobre as quais iniciantes geralmente não possuem conhecimento intuitivo, o que inutiliza seus conhecimentos construídos de percepção acerca do mundo real. Nesse raciocínio, os autores finalizam comparando que o processo cognitivo correspondente à análise da ilustração de uma máquina e sua decomposição em partes é equivalente à análise do pseudocódigo algorítmico e sua decomposição em uma sequência de operações aplicada ao processamento de dados.

Como colocado anteriormente neste artigo, é comum professores de diferentes áreas de conhecimento interessarem-se pela criação de animações para integrá-las aos seus planos de aula, com vistas a facilitar o aprendizado dos alunos. A incipiente percepção da necessidade de proporcionar contato com lógica da programação desde cedo fez com que programas de computador para este fim se desenvolvessem bastante nos últimos anos. Uma característica comum entre esses programas é o processo *drag and drop* (arrastar e soltar) para inserção de comandos de programação, o que evita a ocorrência de erros de escrita. Como o mesmo ocorre para a inserção de estruturas de programação, como *loops* e testes *if-then-else*, erros sintáticos também são evitados. A dificuldade de corrigir erros decorrentes da análise léxico-sintática dos compiladores é um dos maiores fatores causadores de frustração entre os estudantes de programação. Um desses programas é o Alice, disponível gratuitamente para download em http://www.alice.org , em versões disponíveis para os sistemas operacionais Windows, Linux e MacOs, e que em relação a outros com o mesmo propósito, possui como diferencial a construção de ambientes de programação na forma de cenas ou mundos tridimensionais. Para criação desses mundos, um vasto leque de objetos está disponível, desde pequenos utensílios até prédios, aviões, animais, etc. Com um foco voltado à programação orientada a objetos, declaradamente à linguagem de programação Java, os objetos instanciados no mundo 3D possuem propriedades, métodos, são agrupados em classes, respondem a eventos e até a fluxos de execução paralela (multithreading). Porém, são estáticos assim que instanciados. Só executam ações se programados para isso.

Uma vez que permite um primeiro contato da programação inclusive para crianças, o Alice torna-se um programa encorajador para aqueles professores com interesse em criar objetos de aprendizagem e animações no intuito de facilitar o aprendizado. Em outras palavras, pode-se utilizá-lo alinhado com seu objetivo principal, o de ensinar programação, mas pode-se também utilizá-lo como ferramenta de criação de animações para ensino de conteúdos, já que o produto final, a animação, pode ser transformada em um vídeo ou mesmo interagir com o estudante na medida em que o professor julgar necessário.

Em termos de complexidade, é muito mais fácil, portanto, que um professor interessado na criação de um objeto de aprendizagem o faça no Alice, pois os conhecimentos necessários para a criação de um programa como o Software Visualization System (NARAYANAN e HEGARTY, 2002) citado anteriormente, demandariam um aprofundamento e um tempo dispendido muito maior. Também em termos estéticos o Alice levaria vantagem, pela presença de objetos tridimensionais de alta resolução. Porém, há nele também algumas limitações quando comparado aos recursos de uma linguagem de programação atual. O Alice possui um número bastante reduzido de funções matemáticas e o posicionamento dos objetos 3D no mundo virtual não é tão simples de ser feito se comparada à tarefa de posicionar uma lista de valores ou um botão na interface gráfica de uma linguagem de programação visual. Também poderíamos mencionar o menor número de eventos aos quais seus objetos são sensíveis se comparados aos eventos que podem ser programados na interface visual de uma linguagem de programação moderna. Não percamos de vista, no entanto, que o mesmo não foi feito para este fim. Foi feito para ensinar programação orientada a objetos para crianças e adolescentes.

A seção seguinte do artigo abordará um objeto de aprendizagem criado no Alice para o ensino de algoritmos, no intuito de servir de exemplo e de apresentar os recursos disponíveis, assim como os tipos de animação e de interação que o mesmo disponibiliza.

4. Uma animação no Alice para ensino de lógica da programação

A animação criada no Alice para demonstração didática do funcionamento do algoritmo de ordenação de vetores Bubble Sort está disponível, na forma de vídeo, em https://www.youtube.com/watch?v=R_0hTehoZVo e seu arquivo (formato .a3p) pode ser baixado em https://sites.google.com/site/andregustavoschaeffer/home/uffs/roteiros-didaticos. Ela foi feita usando-se um cenário onde

- os números desordenados do vetor aparecem na parte inferior da tela
- à esquerda encontram-se os espaços reservados às variáveis envolvidas na forma de blocos de neve
- ao centro aparece o algoritmo transcrito para a linguagem de programação Pascal

- à direita, há um tutor (um boneco de neve) de onde partem os comentários escritos (na forma de balões de diálogo)

Neste cenário (figura 1), há um alinhamento ao previamente exposto a partir de Hegarty (2004) já que, podemos nos perguntar, o que mais seriam os blocos de gelo que correspondem a espaços reservados às variáveis se não abstrações de endereços de memória reservados ao programa em execução? A presença de um tutor, no caso um boneco de neve, de fácil inserção no Alice por se tratar de um objeto pronto, ampara-se na observação do trabalho de Falloon (2013) em que os mais efetivos exemplos de aplicativos criados e que obtiveram melhores respostas indicando aprendizado curiosamente estavam apresentados no formato tradicional de ensino, frequentemente envolvendo um personagem ensinando conhecimentos particulares ou habilidades. André e Rist (2002, apud Reategui, 2007 p. 8), em estudo anterior, demonstraram que estudantes consideram um tópico em estudo menos difícil e com uma apresentação mais lúdica na presença de um personagem virtual.

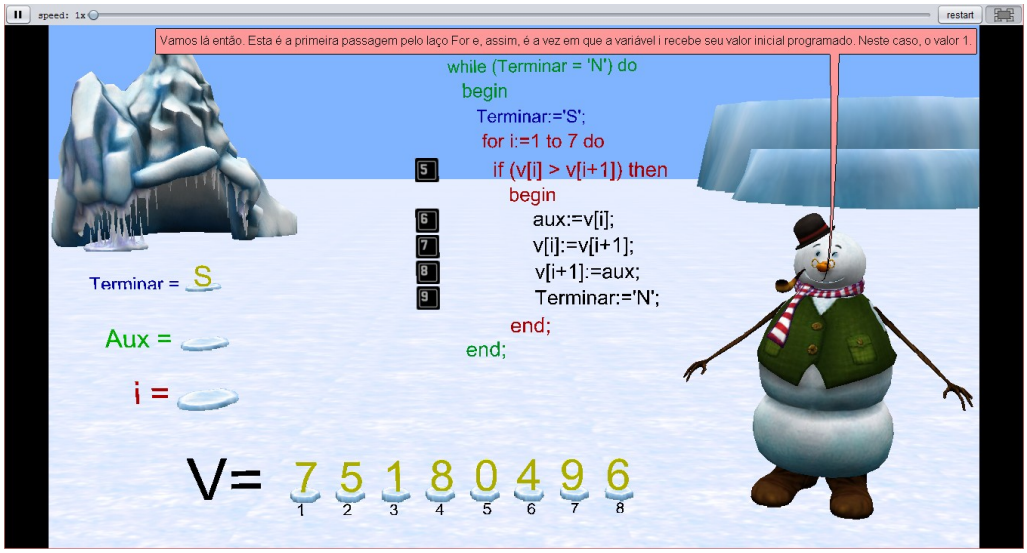

Figura 1. Cenário da animação de ordenação do vetor no Alice

Frente a outras recomendações teóricas do uso de animações, o Alice, independentemente de a animação interagir ou não com o estudante (por caixas de diálogo ou cliques do mouse sobre os objetos da cena), provê um botão de pausa acessível a qualquer momento (visível na barra de controle do canto superior esquerdo da figura 1). Através dessa barra de controle, é possível paralisar o tutorial sempre que necessário, recomeçar a partir do ponto em que foi paralisado, sair da animação sempre que desejado e reiniciá-la sem ter que fechá-la. Isto é particularmente importante pois atende a uma questão levantada por Ainsworth e Van Labeke (2004, apud Hegarty, 2004) argumentando que animações e vídeos avançando sem o controle do estudante sobrecarregam sua memória de trabalho se a informação apresentada em instantes

anteriores precisa ser integrada com alguma informação apresentada posteriormente. Hegarty (2004) complementa apontando que pesquisas voltadas para a análise do movimento dos olhos de estudantes ao observarem imagens estáticas comprovam que eles visualizam repetidas vezes as mesmas partes dessas imagens na frequência que julgarem necessário para que as possam compreender. Segundo Hegarty, uma possível razão para isso seria a possibilidade de aliviar a memória de trabalho do estudante que acaba por externalizá-la na forma da imagem sendo visualizada (2004, p.4). Ou seja, a própria imagem atuaria por alguns instantes como extensão da memória de trabalho respeitando de maneira mais adequada a velocidade do estudante na construção daquele conhecimento. A figura 2 mostra uma das possíveis formas de interação do Alice com o estudante, na forma de caixas de diálogo.

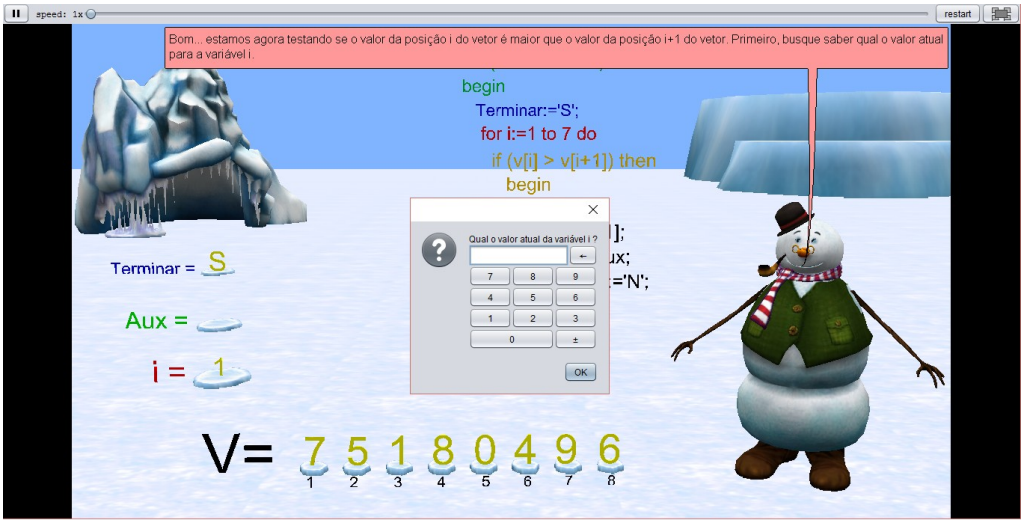

Figura 2. Uma das formas de interação do Alice, através de caixas de diálogo

5. Reflexões e análises

Com base em teorias e estudos anteriores, podemos analisar a animação criada e o próprio Alice como ferramenta de criação com certa criticidade, já que, resgatando o objetivo deste trabalho exposto anteriormente, ele aborda uma nova possibilidade de uso de animações para o ensino de lógica da programação e demais conteúdos, com relativamente fácil apropriação por parte daqueles professores interessados em criá-las. Neste olhar crítico, cabe analisá-lo sob alguns enfoques:

Acessibilidade: o uso de balões e caixas de diálogo no Alice, ao menos até a versão 3.2 utilizada, não é customizável no sentido de se poder aumentar o tamanho das fontes ou posicionar os textos em determinadas áreas da tela. A importância da questão da acessibilidade é estudada por vários autores (BEHAR et al. 2008; ESTABEL et al. 2005; SILVEIRA et al. 2007) e constitui-se como um valoroso impulso em direção à inclusão digital, à inclusão social, e à redução de desigualdades. Mas, no Alice, ficamos restritos às configurações de cores e ao tempo em que cada mensagem é exibida na tela.

É possível se fazer uso, também, com facilidade, de recursos de áudio como uma narração, por exemplo, e personalizar a animação para que o estudante escolha se quer ler ou ouvir as instruções. Neste particular, cabe ao menos mencionar o estudo de Alessi e Trollip (2001) em relação aos conflitos resultantes do uso impróprio de mensagens duais. Segundos eles, formas complementares de informação, como, por exemplo, no caso de uma imagem sendo exibida e uma fala descrevendo-a, facilitam o aprendizado. Porém, o uso simultâneo de um texto descrevendo algo e de uma fala idêntica ao texto, segundo os autores, pode dificultar o aprendizado. Finalmente, o uso de um texto instrucional escrito e de uma narração quase idêntica ao que está escrito, teria consequências piores.

Feedback: segundo Bidarra (2009, p. 7), no viés motivacional da aprendizagem multimídia interativa, "receber informação sobre a nossa própria performance aumenta a probabilidade de sucesso e de progressão". O Alice não deixa a desejar nesse aspecto, pois permite interatividade, seja na forma de entrada de informação por parte do estudante ou *feedback* no sentido oposto, de diferentes maneiras: cliques do mouse sobre objetos da cena, balões de diálogo, áudio, *loops* que podem impedir o avanço caso a resposta certa não tenha sido inserida, testes condicionais para validar a resposta do estudante e inserção de *timers* para execução de eventos de programação que envolvam interação ou qualquer outro controle.

Locus de controle: na animação criada, o locus de controle permanece a maior parte do tempo com o objeto multimídia (ou poderíamos até dizer, com o tutor), e menos com o estudante. Estudos levantados por Alessi e Trollip (2001, p. 51) tocam no assunto, e o apresentam como ainda controverso. Citam o psicólogo americano Jerome Bruner em defesa de que os estudantes tomariam melhores decisões quanto ao sequenciamento de suas próprias atividades de aprendizado em comparação aos seus professores. Em contrapartida, apresentam estudos que defendem exatamente o contrário e outros que defendem o locus de controle no estudante desde que recebam *feedbacks* específicos quanto ao progresso a ao sucesso de suas decisões.

Textos explicativos: igualmente simples de serem inseridos no Alice, os textos explicativos podem complementar o que a animação apresenta. Caberá ao professor ponderar a quantidade de texto explicativo conforme julgar necessário. Rieber et al. (2004, apud Hegarty, 2004, p. 348) descrevem um estudo no qual estudantes decidiam se queriam ou não ter acesso a mensagens conceituais durante uma simulação em questão acerca do estudo das leis de Newton, e, embora sua pesquisa tenha mostrado que os estudantes aprendiam mais a partir de simulações que incluíam mensagens conceituais, eles costumavam optar por somente focar na simulação de forma isolada, sem a presença delas.

Enquanto aguardamos a emersão de um novo fato científico a partir das controvérsias atuais da neurociência, da psicologia e da pedagogia, parece razoável concluir que abordagens pedagógicas que se utilizam de múltiplos meios, quando comparadas a outras que fazem uso de uma forma única de representação, podem

facilitar o aprendizado ao explorarem diferentes domínios cognitivos do estudante, propiciando um melhor estabelecimento de conexões e uma consequente melhora na construção de conhecimentos e no entendimento das formas de aplicá-los.

6. Referências

Alessi, S. M.; Trollip, S. R. Multimedia for learning: methods and development (3rd ed.). Boston: Allyn & Bacon, 2001.

Behar, p. A.; Souza, e. K.; Góes, c. G. G.; Lima, e. M. A importância da acessibilidade digital na construção de objetos de aprendizagem. RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 6, N. 2, dez. 2008.

Bidarra, j. Aprendizagem multimédia interactiva. In: Ensino Online e Aprendizagem Multimédia. Lisboa: Relógio d'Água, 2009. ISBN 978-989-641-141-1. p. 352-382

Estabel, L. B.; Moro, E. L.; Santarosa, L. M. C. O acesso às tecnologias de informação e de comunicação e a superação das limitações dos PNEES com limitação visual incluindo-os em um ambiente de aprendizagem mediado por computador. RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 3, N. 1, mai. 2005.

Falloon, G. Young students using iPads: App design and content influences on their learning pathways. Computers & Education, v. 68, p. 505-521, 2013.

Hegarty, M. Dynamic visualizations and learning: Getting to the difficult questions. Learning and Instruction, v. 14, n. 3, p. 343-351, 2004.

Narayanan, N. H.; Hegarty, M. Multimedia design for communication of dynamic information. International journal of human-computer studies, v. 57, n. 4, p. 279-315, 2002.

Reategui, E. Interfaces para softwares educativos. RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 5, N. 1, jul. 2007.

Schaeffer, A. G. Computer programming in public spaces for digital inclusion using Alice: challenges and opportunities. Proceedings of the 2013 Alice Symposium, 2013. Duke University, Durham(NC), EUA. Disponível em: <http://dl.acm.org/citation.cfm?doid=2532333.2532343> . Acesso em: 1 nov. 2016.

Silveira, C.; Reidrich, R.; Bassani, P. Avaliação das tecnologias de softwares existentes para a inclusão digital de deficientes visuais através da utilização de requisitos de qualidade. RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 5, N. 1, jul. 2007.

Vavra, K.; Watrich, V.; Loerke, K.; Phillips, L.; Norris, S.; Macnab, J. Visualization in science education. Alberta Science Education Journal, v. 41, n. 1, p. 22-30, 2011.