

Programação Criativa e Colaborativa no Ensino Médio: uma Experiência com Studio Sketchpad

Ana Luísa Veroneze Solórzano¹, Pedro Prestes Matiuzzi¹
Andrea Schwertner Charão¹

¹ Curso de Ciência da Computação
Departamento de Linguagens e Sistemas de Computação
Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brasil
{alsolorzano, ppmatiuzzi, andrea}@inf.ufsm.br

Resumo. *O desenvolvimento de habilidades de programação na educação básica tem sido alvo de muitas iniciativas, geralmente amparadas em ferramentas computacionais consolidadas e especialmente concebidas para este público. Neste trabalho, explora-se um recurso pouco conhecido neste meio, denominado Studio Sketchpad, que utiliza a programação textual baseada na linguagem Javascript. Dentre suas principais características, estão a facilidade de expressar criações com apelo visual e a possibilidade de criar programas colaborativamente. O Studio Sketchpad foi utilizado em oficinas oferecidas a um grupo de alunos de ensino médio, revelando pontos fortes e limitações que podem contribuir à replicação da experiência em contextos semelhantes.*

Abstract. *The development of programming skills in basic education has been the target of many initiatives, usually supported by consolidated computer tools specially designed for this public. In this work, we explore a lesser known resource in this medium, called Studio Sketchpad, which uses textual programming based on the Javascript language. Among its main characteristics are the ease of expressing creations with visual appeal and the possibility of creating programs collaboratively. We used Studio Sketchpad in workshops offered to a group of high school students, revealing strengths and limitations that can contribute to the replication of experience in similar contexts.*

1. Introdução

O ensino-aprendizagem de programação na educação básica é um tema desafiador, que tem sido alvo de muitos estudos e ações [da Silva et al. 2015, Aureliano e Tedesco 2012]. Abordagens para iniciantes têm combinado atividades lúdicas e criativas com programação visual, geralmente baseada em encaixe de blocos, para motivar e facilitar o processo. Para amparar tais atividades, utiliza-se ferramentas consolidadas, como Scratch¹, ou mais recentes, como os jogos e tutoriais do Code.org², que estão presentes em experiências no exterior e no Brasil [Wyffels et al. 2014, Alvarado 2014, Mota et al. 2014, Martins et al. 2016].

Atividades com programação com uma linguagem textual, como Python, também são exploradas em algumas experiências, focadas principalmente no público do ensino

¹<http://scratch.mit.edu>

²<http://code.org>

médio [Marques et al. 2011]. A programação textual tende a ser percebida como mais difícil que a programação visual [Weintrop e Wilensky 2015], porém é uma abordagem poderosa e flexível, amplamente usada profissionalmente para desenvolvimento dos mais variados tipos de software.

A linguagem Javascript, assim como Python, é uma linguagem de programação textual difundida em aplicações profissionais, mas que também suscita interesse como recurso para iniciantes em programação [Ward e Smith 1998]. Trata-se de uma linguagem versátil, que vem se popularizando em aplicações gráficas, como jogos e animações. Para iniciantes em programação, esta linguagem é abordada, por exemplo, em Codecademy³, Khan Academy⁴ e SACI⁵.

Numa linha semelhante, também baseada em programação textual em Javascript, existe uma ferramenta *online* denominada Studio Sketchpad [Bader-Natal 2010], que é voltada à criação de gráficos e animações interativas, servindo ao aprendizado de fundamentos de programação em comunidades de artistas e designers. Embora não seja um recurso difundido no meio de educação em computação, o Studio Sketchpad dispõe de funcionalidades que podem ser úteis neste meio, a saber: a facilidade de expressar criações com apelo visual e a possibilidade de criar programas colaborativamente. Além disso, sua abordagem distingue-se de outras alternativas pela proposta de criação livre, sem níveis avançados gradualmente, como em Code.org e sem objetivos predefinidos, como em Khan Academy.

Assim, tomando por hipótese que o Studio Sketchpad poderia trazer benefícios no ensino-aprendizagem de programação para iniciantes, respeitando as suas preferências individuais durante a criação de animações, este trabalho apresenta uma experiência de utilização deste recurso em oficinas voltadas a um grupo de alunos de uma escola de ensino médio. Os objetivos foram proporcionar uma experiência original aos alunos e, ao mesmo tempo, coletar observações sobre pontos fortes e limitações deste recurso, que possam beneficiar outros públicos com características semelhantes.

O restante deste artigo está organizado como segue. Na seção 2, apresenta-se o Studio Sketchpad, suas versões e suas funcionalidades. Na seção 3, descreve-se a metodologia empregada no trabalho, caracterizando o público-alvo, a delimitação e o planejamento das atividades. Na seção 4, apresenta-se uma avaliação qualitativa da experiência, elencando e discutindo aspectos levantados. Na seção 5, por fim, apresentam-se considerações finais e perspectivas de trabalhos futuros.

2. Studio Sketchpad

Studio Sketchpad (<http://sketchpad.cc>) é um *site* para criação e compartilhamento de programas na linguagem Processing, permitindo expressar animações interativas e visualizações de dados [Bader-Natal 2010]. A linguagem Processing [Reas e Fry 2003], que por sua vez baseia-se em Java, é voltada para comunidades de artistas e designers, servindo ao aprendizado de fundamentos de programação de computadores para criação de *sketches* (desenhos, visualizações, animações) digitais.

³<http://www.codecademy.com/>

⁴<http://www.khanacademy.org/>

⁵<http://olimpiada.ic.unicamp.br/saci>

Toda a interação com Studio Sketchpad se dá diretamente via navegador web, sem necessidade de instalação de software adicional. Desde 2015, está disponível uma versão do Studio Sketchpad (<http://p5js.sketchpad.cc>) para autoria de *sketches* em linguagem Javascript, com a biblioteca P5.js⁶. Esta biblioteca tem funcionalidades inspiradas e parcialmente compatíveis com Processing, porém utiliza recursos suportados por navegadores modernos. A P5.js é empregada em outras ferramentas voltadas para iniciantes em programação, como por exemplo os tutoriais sobre programação de animações da Khan Academy [Khan Academy 2017].

Ambas as versões do Studio Sketchpad possuem interface semelhante e suportam códigos-fonte que seguem os mesmos princípios, com diferenças em recursos avançados, como por exemplo a manipulação de mídias de fontes externas (captura de vídeos, importação de imagens, etc.). No restante deste artigo, o termo Studio Sketchpad será usado para designar a versão mais recente, com Javascript, que foi a versão empregada nas oficinas desenvolvidas.

No Studio Sketchpad, a programação em Javascript geralmente requer poucas linhas de código, pois as funções da biblioteca P5.js abstraem detalhes do sistema gráfico. Em muitos casos, basta que o usuário preencha duas funções que são invocadas implicitamente pelo ambiente: `setup`, que é chamada no início de uma execução para configurar propriedades da tela, e `draw`, que é executada continuamente, ou seja, num laço implícito, que controla a apresentação do *sketch* a cada passo da animação resultante. Na Figura 1, apresenta-se um exemplo de código com essas funções, juntamente com uma visualização resultante. Pode-se observar que, além de definir `setup` e `draw`, o código usa outras funções definidas na P5.js ou pelo usuário (no caso, `click`).

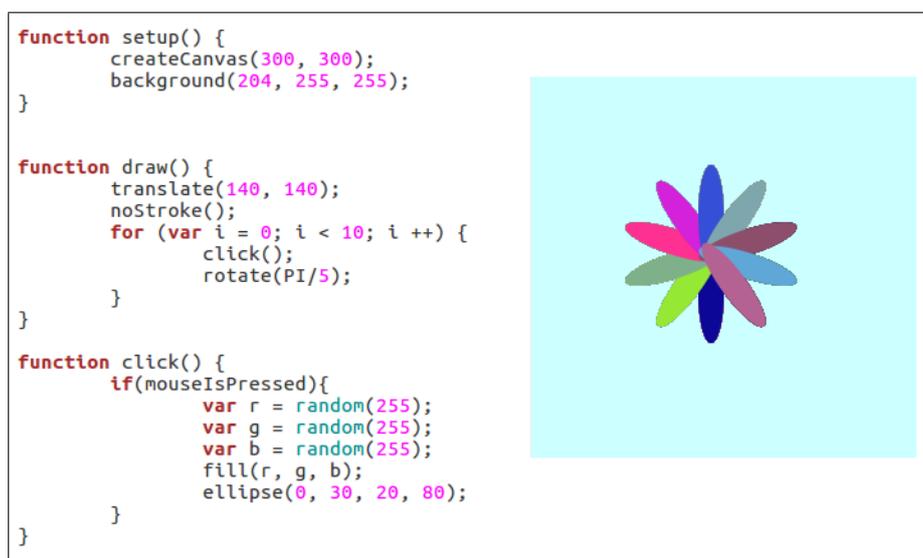


Figura 1. Exemplo de código de uma animação que cria elipses de cores aleatórias a cada clique do mouse, formando um desenho.

Usuários do Studio Sketchpad podem criar *sketches* que são compartilhados e disponibilizados publicamente. Usuários não-cadastrados no *site* podem visualizar e

⁶<http://p5js.org>

executar *sketches* públicos, mas para acessar recursos de edição e colaboração é necessário efetuar um cadastro, fornecendo nome, e-mail e senha.

Um importante diferencial do Studio Sketchpad em relação ao ambiente da linguagem Processing é um editor *online* colaborativo, baseado no editor EtherPad⁷, que salva automaticamente cada revisão do código-fonte de um *sketch*. A interface do Studio Sketchpad (Fig. 2) apresenta o histórico de revisões do código-fonte de um *sketch* ao longo do tempo e, para cada revisão, permite visualizar o código e sua respectiva apresentação gráfica.

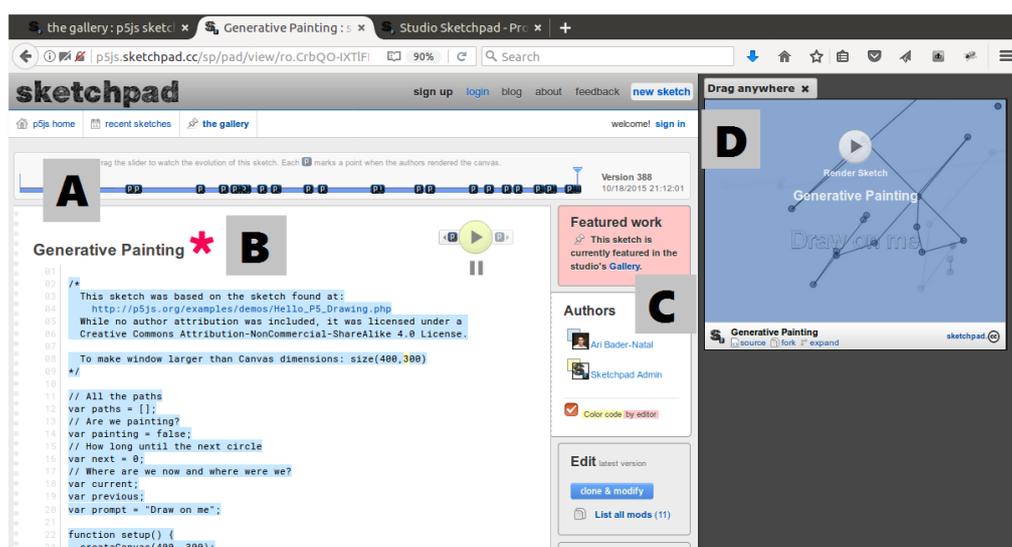


Figura 2. Interface de um *sketch*: (A) Linha do tempo de revisões (B) Título e código da animação (C) Definição dos autores (D) Apresentação gráfica

Outro recurso do Studio Sketchpad é uma galeria que lista os *sketches* criados e publicados no ambiente (Fig. 3), ordenados por data de criação, número de edições, cópias modificadas ou número de visualizações. Este tipo de recurso é oferecido em alguns ambientes concebidos para educação em computação, como em Scratch por exemplo, servindo de inspiração para novas criações. Usuários do Studio Sketchpad podem criar cópias (clones) de qualquer revisão de um *sketch*, para modificação. O ambiente mantém registros das cópias e oferece um espaço que apresenta visualmente as múltiplas cópias modificadas de um *sketch*.

3. Metodologia

A experiência com Studio Sketchpad inseriu-se em um projeto de escopo mais amplo, em execução desde 2014, visando desenvolver o raciocínio lógico e habilidades de programação junto ao público de educação básica [Charão et al. 2015]. Este projeto é coordenado por professores universitários e conta com vários tutores, que são graduandos de cursos superiores da área de Computação. Desde a sua criação, o projeto já envolveu um público de alunos e professores de diversas escolas, públicas e privadas, tendo empregado diferentes ferramentas concebidas para o ensino-aprendizado de fundamentos de computação e programação, como Scratch, jogos do Code.org e MIT App Inventor⁸.

⁷<http://etherpad.org>

⁸<http://appinventor.mit.edu>

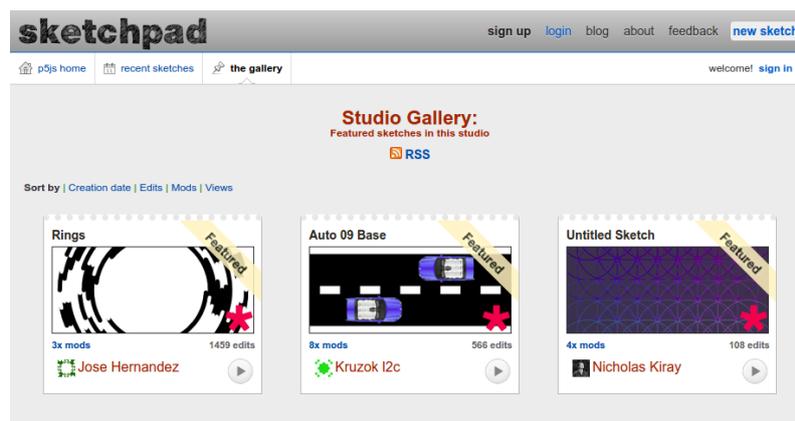


Figura 3. Galeria com *sketches* criados por diferentes autores

3.1. Participantes, delimitação e planejamento da experiência

No segundo semestre de 2016, o projeto atendeu um grupo de 12 alunos do primeiro ao terceiro ano do ensino médio de uma escola pública, que se inscreveram voluntariamente em uma oficina gratuita de 8 horas de duração, no turno oposto ao das aulas, distribuída em 4 encontros semanais de 2 horas, realizadas num laboratório da própria escola. Essa oficina abordou apenas a programação visual com encaixe de blocos, iniciando com jogos do Code.org e culminando com a criação de 2 aplicativos simples com MIT App Inventor. As atividades foram predominantemente práticas e realizadas individualmente, no ritmo de cada aluno. Os encontros foram sempre apoiados por no mínimo 4 graduandos de cursos da área de Computação, que indicaram recursos e forneceram orientações individualizadas.

Deste grupo do ensino médio, 7 alunos manifestaram interesse em programação textual e disponibilidade para participar de mais alguns encontros, em um laboratório localizado na sede do projeto. Tais alunos revelaram-se talentosos e curiosos por novidades, sendo que alguns, por conta própria, já tinham buscado informações sobre linguagens como Python, Java ou Javascript. No planejamento de novos encontros, procurou-se alimentar esta curiosidade e, ao mesmo tempo, acomodar uma limitação de horários, pois todos os envolvidos (público-alvo e tutores) tinham agenda com poucos horários livres.

Assim, planejou-se um percurso de atividades, todas na linguagem Javascript, que permitissem, em poucos encontros presenciais, promover avanços e manter a motivação dos alunos, estimulando sua curiosidade e criatividade. Esse percurso sequencial baseou-se em 3 recursos disponíveis publicamente, que poderiam ser facilmente acessados pelos alunos fora do período dos encontros.

3.2. Iniciação à programação textual

O primeiro recurso foi o ambiente SACI [Anido 2015], que provê vídeos, textos e exercícios para iniciação à programação com linguagem textual (Javascript). Com SACI, os alunos têm contato com noções básicas de programação, aprendendo comandos de leitura e de escrita, declaração e tipos de variáveis e exercitando estruturas de sequência, condicionais e de repetição, que já eram exploradas na programação visual com blocos. Neste ambiente, os alunos também podem editar e executar seus programas para resolver problemas propostos nos exercícios. Para isso, utilizam somente o navegador, sem necessidade de instalação de software adicional. Erros e acertos são indicados pelo ambiente a cada execução. Os

programas ficam restritos a uma interface de entradas e saídas textuais, sem possibilidade de usar recursos gráficos.

O segundo recurso foi o tutorial *online* intitulado “Introdução a JavaScript: Desenho & Animação” [Khan Academy 2017], o primeiro de 8 tutoriais sobre programação da Khan Academy. Este tutorial aborda conceitos básicos de programação por meio de vídeos, textos e exercícios de criação de animações simples. Assim como no ambiente SACI, os usuários editam e executam seus códigos-fonte no próprio ambiente da Khan Academy, usando funções da biblioteca ProcessingJS que, assim como P5.js, oferece funcionalidades gráficas da linguagem Processing em Javascript. A maioria dos exercícios deste tutorial são trechos de códigos que o usuário deve modificar seguindo um enunciado, a fim de obter um resultado esperado. O ambiente também indica erros e acertos, mas de forma diferente do SACI, usando diálogos contextualizados no exercício em que o usuário se encontra.

O Studio Sketchpad foi introduzido como um terceiro recurso, após os alunos terem contato com os ambientes precedentes. Durante 5 encontros presenciais realizados semanalmente, foram dedicadas cerca de 2 horas para o ambiente SACI, 2 horas para o Khan Academy e 8 horas para o Studio Sketchpad. Entre um encontro e outro, os alunos foram incentivados a acessar os ambientes para completar os tutoriais. A seção seguinte detalha as atividades com o Studio Sketchpad, que ocuparam a maior carga horária.

3.3. Atividades com Studio Sketchpad

As atividades com Studio Sketchpad não obedeceram um roteiro pré-estabelecido, sendo construídas ao longo dos encontros, como fruto da interação entre alunos e tutores. Com a possibilidade de compartilhamento dos *sketches*, os alunos trabalharam em duplas ou trios, cada um em um computador, gerando múltiplas revisões de código a partir de exemplos escolhidos conforme os interesses individuais.

Como material de apoio às atividades, os alunos utilizaram o manual de referência da biblioteca P5.js, que apresenta detalhadamente as funções disponíveis, e a página de exemplos desse manual⁹. Os exemplos mostram, sob forma de pequenos tutoriais para edição e execução, desde como trabalhar com as coordenadas e posicionamento na tela até como criar formas tridimensionais.

A seguir encontra-se um registro das principais atividades, feito posteriormente. Ressalta-se, no entanto, que alguns alunos exploraram recursos não listados abaixo, e nem todos os alunos realizaram todas as atividades.

Modificação de animação: Esta atividade consistiu na modificação de uma animação simples, na qual os alunos pudessem explorar os recursos de animação em si, mudando as dimensões da imagem, o número de quadros por segundo, o posicionamento da imagem na tela, a inserção de outras imagens, etc. Nesta atividade, contabilizou-se 126 revisões na animação feitas pelos alunos em um encontro. Isto pode ser visualizado na Figura 4, na última coleção de pontos na linha do tempo (as revisões anteriores foram feitas pelos graduandos tutores durante a preparação da atividade).

Figuras geométricas: Nesta atividade, foi trabalhada a criação de figuras geométricas planas (círculo, quadrado, triângulo), avançando para a criação de figuras mais

⁹<https://p5js.org/examples/>

elaboradas, como uma estrela, por exemplo. Os códigos-fonte de base foram os exemplos *Simple Shapes*, *Regular Polygon* e *Star*, disponíveis no manual da biblioteca P5.js. Ao criar novas revisões dos códigos, os alunos colaboraram usando noções básicas de trigonometria (ângulos, seno e cosseno), de forma a modificar propriedades das figuras e animá-las na tela.

Interação via mouse e teclado: Esta atividade consistiu na criação de animações interativas, capazes de reagir a ações do usuário via mouse e teclado. Os alunos usaram o exemplo *Interactivity 1* como base, modificando-o a partir de exemplos do manual de referência com funções tais como `mousePressed()` e `keyPressed()`. Nesta atividade, os alunos colaboraram entre si exercitando estruturas de controle condicionais, para provocar alterações na tela conforme a ação do usuário.

Transformações interativas: Nesta atividade, os alunos combinaram recursos das atividades anteriores para criar animações interativas, com transformações de figuras (translação, rotação) guiadas por ações do usuário no teclado e mouse. Alguns alunos aproveitaram a atividade para animar sólidos em 3D (exemplo na seção *3D Geometries* do manual), enquanto outros estenderam a animação de uma estrela criada anteriormente. Outros ainda optaram por criar um jogo no qual pequenos círculos animados eram mostrados na tela e reagem a ações do usuário, contabilizando pontos a cada vez que um círculo fosse atingido.

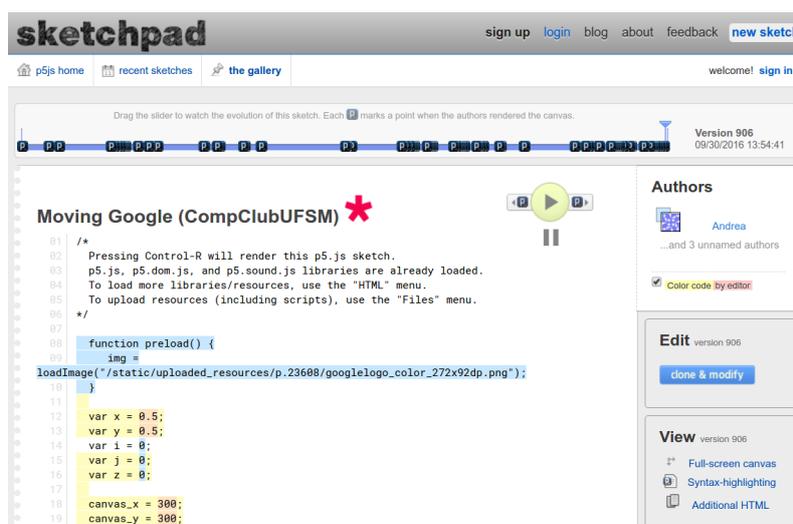


Figura 4. Atividade com várias revisões de código realizadas colaborativamente

4. Resultados e Discussão

Dados os objetivos do trabalho e o pequeno número de alunos envolvidos, optou-se por uma abordagem qualitativa de avaliação da experiência, por meio de observações e conversas informais entre tutores e alunos. A lista a seguir resume os pontos fortes do Studio Sketchpad levantados junto a alunos e tutores:

1. **Ambiente online:** esta característica, também encontrada em outras ferramentas educacionais atuais, mostrou-se vantajosa por não exigir um tempo prévio de preparação do laboratório e por facilitar o uso pelos alunos em outros momentos e espaços.

2. **Feedback imediato:** a visualização das criações e modificações do código-fonte permitem que o aluno receba *feedback* imediato e frequente sobre os recursos de programação utilizados. Comportamentos inesperados das animações são percebidos facilmente e instigam os alunos a revisar e alterar seus códigos para obter o resultado desejado.
3. **Documentação do P5.js:** o manual de funções e de exemplos do P5.js, acessível via Studio Sketchpad, mostrou-se um recurso útil e instigante, de fácil navegação e entendimento, mesmo tratando-se de uma documentação em inglês. Mais do que isso, os alunos exercitaram uma prática usada por programadores profissionais, que também precisam consultar documentação de referência sobre funções que utilizam.
4. **Liberdade de criação:** em comparação com os recursos anteriores (SACI e tutorial da Khan Academy), o Studio Sketchpad ofereceu maior liberdade de criação, o que motivou os alunos a buscar novas funções e exercitar diferentes maneiras de programar para obter um resultado desejado.
5. **Colaboração:** a possibilidade de editar uma animação entre colegas, em tempo real, foi bem aceita pelos alunos e percebida como original, já que os ambientes usados anteriormente não dispunham dessa funcionalidade. Alunos que já se conheciam aproveitaram melhor esta possibilidade, desafiando-se mutuamente a explorar novas funções e, com isso, mantendo-se claramente motivados.
6. **Histórico de modificações:** esta funcionalidade mostrou-se útil tanto para os alunos, que puderam explorar e comparar diferentes versões de animações produzidas colaborativamente, como para os tutores, que puderam observar registros de progressos realizados pelos alunos.

Na experiência realizada, também foram observadas algumas limitações do Studio Sketchpad, bem como dificuldades percebidas pelos alunos:

1. **Laços de repetição implícitos:** conforme mencionado anteriormente, animações em P5.js executam um laço de repetição implícito, que faz chamadas a funções do código do usuário. Embora esta característica vise facilitar a criação de animações, notou-se que isso causou alguma confusão inicial, principalmente em alunos que estavam à vontade com a programação em Javascript nas atividades anteriores e já dominavam estruturas de repetição. Estes alunos tentaram expressar a repetição explicitamente no código, obtendo um resultado inesperado em suas primeiras animações.
2. **Versões do ambiente:** conforme mencionado anteriormente, há duas versões do *site* (studio.sketchpad.cc e p5js.sketchpad.cc), apresentando interfaces e linguagens muito semelhantes, porém não idênticas. Isso às vezes causou confusão quando os alunos buscaram o recurso na internet e escolheram uma das opções sem prestar atenção à URL, pois não estavam cadastrados na outra versão do Studio. O ideal seria que os alunos sempre acessassem a URL indicada pelos tutores, mas notou-se que isso nem sempre ocorreu.
3. **Erros no código:** o ambiente realiza verificação e indicação de erros, porém com limitação a erros de sintaxe. Erros estruturais ou de lógica no programa não são acusados, o que pode fazer com que uma animação não execute, sem emitir nenhum alerta. Esta limitação foi agravada, em alguns casos, pelo fato do *site* mostrar-se lento em alguns acessos. O tratamento de erros foi percebido como o principal

problema do Studio pelos alunos, que se habituaram, em outros ambientes, a receber alertas e explicações apontando precisamente onde estavam os erros.

4. **Idioma:** a interface do Studio Sketchpad é oferecida somente com interface em inglês, mas nesta experiência os alunos não tiveram dificuldades quanto a isso. De fato, eles possuíam algum conhecimento do idioma e até se mostraram estimulados com a documentação em inglês, que representou uma oportunidade de aumentar o vocabulário. No entanto, com base em experiências dos tutores com alunos menos habituados a outro idioma, entende-se que a falta de opção em português possa ser um obstáculo significativo, dependendo do público.

De forma geral, pode-se dizer que, no presente trabalho, as vantagens do Studio Sketchpad superaram suas limitações. Os alunos mantiveram-se motivados, frequentando os encontros voluntariamente e demonstrando avanço em suas habilidades, à medida que modificavam e criavam animações de complexidade crescente.

5. Considerações Finais

Neste trabalho, apresentou-se uma experiência original de exploração dos recursos do Studio Sketchpad com um grupo de alunos do ensino médio. Com esta ferramenta, os alunos puderam expressar sua criatividade na criação de animações, atuando de forma colaborativa, adicionando incrementos sucessivos a cada *sketch* compartilhado entre os colegas do grupo.

A experiência realizada utilizou uma abordagem qualitativa e não teve a ambição de produzir resultados estatisticamente relevantes. Mesmo assim, pôde-se verificar que o Studio Sketchpad, embora não tenha sido concebido para explorar noções de computação na educação básica, tem funcionalidades que o tornam atraente nesse contexto, especialmente para estimular jovens com interesse na área. De fato, a liberdade de programação textual colaborativa instiga os alunos a explorar os recursos da P5.js e, com isso, reforçar e aprofundar seus conhecimentos e habilidades em programação.

Acredita-se que esta experiência possa ser replicada com resultados similares, em outros grupos com características semelhantes. Há que se considerar, no entanto, que o ambiente pode oferecer obstáculos demasiados a alunos que não tenham alguma familiaridade prévia com a programação textual, e que a carga horária de encontros pode se tornar pequena para criações avançadas. Por isso, sugere-se apresentar o Studio Sketchpad como uma alternativa após atividades de iniciação e exibir exemplos de animações complexas antes e durante o uso da ferramenta, buscando fomentar a curiosidade dos participantes nos períodos entre encontros.

A título de trabalhos futuros, há uma funcionalidade a explorar que não estava ativa no Studio Sketchpad à época da experiência: a possibilidade de criar *sites* privados, customizados para um grupo (turma ou curso) que deseje criar e compartilhar *sketches*. Este serviço oferece recursos como, por exemplo: página inicial e modelo de código-fonte customizados para o grupo, galeria exclusiva para as criações do grupo, entre outros. Acredita-se que tais recursos possam ser especialmente úteis para experiências com maior número de alunos, facilitando o acompanhamento pelos tutores.

Referências

Alvarado, C. (2014). Cs ed week 2013: The hour of code. *SIGCSE Bull.*, 46(1):2–4.

- Anido, R. (2015). SACI – ainda outro ambiente para o ensino de programação. In *Anais do Workshop sobre Educação em Computação (WEI)*.
- Aureliano, V. e Tedesco, P. (2012). Ensino-aprendizagem de programação para iniciantes: uma revisão sistemática da literatura focada no SBIE e WIE. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*.
- Bader-Natal, A. (2010). Studio Sketchpad – Processing on EtherPad. Disponível em: <https://aribadernatal.com/projects/Sketchpad/>.
- Charão, A. S., de A. Barcelos, P. P., de Azevedo, B. R., e Lima, J. C. D. (2015). Clube de computação para alunos de ensino médio: um relato de experiência. In *Anais do XXIII Workshop sobre Educação em Informática (WEI 2015)*.
- da Silva, T. R., Medeiros, T., Medeiros, H., Lopes, R., e Aranha, E. (2015). Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, 23(1):182–196.
- Khan Academy (2017). Introdução a JavaScript: Desenho & animação. Disponível em: <https://pt.khanacademy.org/computing/computer-programming/programming>.
- Marques, D. L., Costa, L. F. S., de Azevedo Silva, M. A., e Rebouças, A. D. D. S. (2011). Atraindo alunos do ensino médio para a computação: Uma experiência prática de introdução a programação utilizando jogos e Python. In *Anais do XVII Workshop de Informática na Escola (WIE 2011)*, pages 1138 – 1147.
- Martins, R., Reis, R., e Marques, A. B. (2016). Inserção da programação no ensino fundamental: Uma análise do jogo Labirinto Clássico da Code.org através de um modelo de avaliação de jogos educacionais. In *Anais do XXII Workshop de Informática na Escola (WIE 2016)*, pages 121 – 130.
- Mota, F. P., Ribeiro, N. F. A., Emmendorfer, L., Butzen, P., Machado, K. S., e Adamatti, D. F. (2014). Desenvolvendo o raciocínio lógico no ensino médio: uma proposta utilizando a ferramenta Scratch. In *Anais do XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014)*, pages 377 – 381.
- Reas, C. e Fry, B. (2003). Processing: A learning environment for creating interactive web graphics. In *ACM SIGGRAPH 2003 Web Graphics*, SIGGRAPH '03, pages 1–1, New York, NY, USA. ACM.
- Ward, R. e Smith, M. (1998). Javascript as a first programming language for multimedia students. *SIGCSE Bull.*, 30(3):249–253.
- Weintrop, D. e Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children, IDC '15*, pages 199–208, New York, NY, USA. ACM.
- Wyffels, F., Martens, B., e Lemmens, S. (2014). Starting from scratch: Experimenting with computer science in flemish secondary education. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education, WiPSCE '14*, pages 12–15, New York, NY, USA. ACM.