

FX Canvas2D: uma API de jogos bidimensionais para auxiliar na aprendizagem de programação

Gustavo Rissetti¹, Fhabiana Thieli Machado¹, Pauline Vielmo Miranda²

¹Instituto Federal Farroupilha – Campus São Vicente do Sul
São Vicente do Sul – RS – Brazil

²Universidade Federal de Santa Maria
Santa Maria – RS – Brazil

Abstract. *In computer science courses, in programming teaching, it is often necessary to use different strategies to aid in the understanding of the contents covered, such as game programming. In this sense, the work presents an API (Application Programming Interface) for programming 2D (two-dimensional) games using the Java language and JavaFX technology. This applies to computer graphics discipline, allowing primitive designs, rendering loop, geometric transformations, among others. This library aims to generate cognitive benefits for students by improving their performance in programming activities.*

Resumo. *Nos cursos da área da informática, quanto ao ensino de programação, muitas vezes é necessário utilizar diferentes estratégias para auxiliar na compreensão dos conteúdos abordados, como por exemplo, programação de jogos. Nesse sentido, o trabalho apresenta uma API (Application Programming Interface) para programação de Jogos 2D (bidimensionais) utilizando a linguagem Java e a tecnologia JavaFX. Esta se aplica a disciplina de computação gráfica, permitindo desenhos de primitivas, laço de renderização, transformações geométricas, entre outros. Esta biblioteca tem por objetivo, gerar benefícios cognitivos nos estudantes, aprimorando seu rendimento nas atividades de programação.*

1. Introdução

As novas tecnologias de informação e comunicação (TICs) estão atreladas à educação, principalmente nas áreas que envolvem informática, ocupando um espaço cada vez maior nas reflexões e práticas educativas. Nas últimas décadas do século XX, o avanço do conhecimento científico tornou possível o aparecimento dessas novas tecnologias, auxiliando no processo educativo e na difusão do conhecimento.

A escola, nesse contexto, torna-se um espaço de democratização da sociedade, podendo garantir o acesso à informação e desenvolver a capacidade de selecioná-la e utilizá-la, transformando-a em conhecimento. Há um papel fundamental da escola como agente de transformação social. [Valente 2003] considera que o computador pode contribuir para o aperfeiçoamento, complementação e possível mudança na qualidade do ensino, além de auxiliar na superação das situações de exclusão e desigualdade no âmbito social e de ser também uma ferramenta para promover a aprendizagem.

A informática na educação desejável seria construída gradativamente por profissionais de diferentes áreas, de forma interdisciplinar. Essa integração entre as áreas de

conhecimento realizada pelo professor colabora para a expansão e flexibilização de sua formação enquanto profissional. Segundo [Ramos 2011] estudar lógica, racionalizar os conceitos lógicos e, mais precisamente, a lógica de programação, assim como, ao criar algoritmos e conceitos computacionais, desenvolver aplicações e implementações na forma de software, tornou-se hoje uma área de conhecimento extremamente importante.

Para [Ramos 2011] a dificuldade de adaptação dos alunos para desenvolver o raciocínio lógico, está no fato de que eles estão acostumados a decorar os conteúdos. Isto gera a falta de motivação e desânimo, pois acreditam que a disciplina só se constitui num obstáculo extremamente difícil de ser superado. Os conceitos de lógica são utilizados em nosso dia a dia, porém, transformá-los em algoritmos lógicos ou em etapas de procedimentos lógicos não é algo trivial e, realmente, as pessoas têm dificuldade para transformar em signos abstratos aquilo que facilmente fazem quando tomam uma decisão.

Neste contexto de utilização da informática na educação, é cada vez mais frequente os casos de aplicação da Computação Gráfica [Azevedo and Conci 2003], que abrangem desde os mais simples desenhos vetoriais ou matriciais, como também, a preparação de projetos e elementos gráficos que contam com o suporte de tecnologias gráficas, tais como a indústria de publicidade e do entretenimento.

Ou seja, atualmente a computação gráfica está presente em diversas áreas do conhecimento humano, desde a engenharia que utiliza ferramentas CAD (Computer-Aided Design), até a medicina que trabalha com técnicas de visualização para auxiliar o diagnóstico por imagens. Outros tipos de simuladores são usados para treinamento em centro de formação de condutores, antes das aulas práticas em automóveis, entre outros. Levando-se em consideração a grande gama de utilização da computação gráfica, esta foi incluída em muitos cursos da área da informática, justificando a participação da disciplina de computação gráfica no currículo de referência da SBC - Sociedade Brasileira de Computação [SBC 2016].

O processo de ensino e aprendizagem de programação é complexo e constituído por muitas e diferentes variáveis, as quais se relacionam e se interligam de maneira muito dinâmica. Esse sistema ativo e intenso de busca de metodologias e práticas de como ensinar e como aprender é formado por vários obstáculos. Há diversas propostas para solucioná-los, como apresentado em [Medeiros et al. 2013]. Com relação a ferramentas gráficas, existem algumas propostas para ensino de estrutura de dados como em [dos Santos and Costa 2005] e [Silva and Júnior 2013]. Do mesmo modo, [de Souza 2016] que é baseado em pseudo-código e gera uma visualização. Com base no exposto, o presente trabalho diferencia-se por desenvolver uma API para ensino de programação aplicada à disciplina de computação gráfica.

Nesse sentido, este trabalho tem o objetivo de utilizar a computação gráfica para auxiliar no processo de desenvolvimento de raciocínio lógico de estudantes da área da informática, através do desenvolvimento de uma API (Application Programming Interface) para programação de Jogos 2D (Bidimensionais). Jogos digitais, assim como sua programação, podem ser elementos motivadores para o processo de ensino e aprendizagem de programação, uma vez usando tais elementos possibilita-se oferecer ao aprendiz momentos lúdicos e interativos como etapas do processo de aprendizagem [Sá et al. 2007]. Assim, a programação de pequenos jogos se torna atraente para os alu-

nos, e com isso, eles podem se motivar mais e se desenvolver mais quanto à programação de computadores.

2. Desenvolvimento da API para Jogos 2D

A computação gráfica é um tema que atrai muitos estudantes pelo fato de ser relacionada muitas vezes com jogos. Porém, outro motivo de interesse para a área é a questão da programação, onde o estudante tem a oportunidade de codificar e visualizar em uma tela o resultado do código, muitas vezes criando pequenos “joguinhos” para exercitar atividades básicas como desvios condicionais e laços de repetição.

[Rapkiewicz et al. 2006] afirma que se observam as dificuldades dos alunos em assimilar as abstrações envolvidas no processo de ensino-aprendizagem de programação computacional. Alguns típicos exemplos de atividades sugeridas para disciplinas de programação baseiam-se em conceitos matemáticos, tais como porcentagem e fatorial, que não motivam a maioria dos alunos. Devido a isso, os alunos não conseguem desenvolver o raciocínio lógico necessário para desenvolvimento dos programas e, assim, o foco do problema é a falta de capacidade de abstração e de transformação do problema numa sequência lógica para desenvolvimento dos algoritmos de programação.

Logo, o problema é de compreensão textual e de transformação desse conteúdo em uma sequência lógica, por isso, buscamos o “fazer” partindo do interesse dos alunos. A partir da interatividade dos *games*, encontramos formas inovadoras de participação que podem constituir vantagem para os processos educacionais.

Normalmente, em disciplinas de computação gráfica se trabalham com bibliotecas como OpenGL (*Open Graphics Library*), que é uma biblioteca de rotinas gráficas e de modelagem, bidimensional e tridimensional ([Cohen and Manssour 2006], [Shreiner et al. 2004], [Wright Jr et al. 2010]). Ela permite desenvolver aplicações interativas e gerar imagens de cenas bidimensionais e tridimensionais, com um alto grau de realismo, tendo como vantagem a alta velocidade de renderização de imagens, uma vez que incorpora vários algoritmos otimizados, incluindo o desenho de primitivas gráficas, o mapeamento de textura e outros efeitos especiais.

Apesar de a biblioteca OpenGL ser extensamente utilizada, para muitos iniciantes em computação gráfica ela pode parecer difícil de se compreender e utilizar. Tendo isso em mente, este trabalho consiste na criação de uma biblioteca chamada FX Canvas2D API, que consiste em rotinas para desenhos de primitivas gráficas bidimensionais em Java, utilizando a tecnologia JavaFX [Oracle 2016b]. Essa biblioteca foi desenvolvida com o intuito de facilitar a aprendizagem de conceitos básicos da computação gráfica, tais como desenhos de primitivas, laço de renderização e transformações geométricas. Com isso, pode-se retomar conhecimentos já adquiridos em disciplinas de algoritmos e programação e aprimorar o raciocínio lógico, uma vez que seu uso para a criação de pequenos jogos tende a entusiasmar e chamar a atenção dos estudantes.

Deste modo, pode-se considerar que o motivo que leva alguém a querer criar um jogo é, na maior parte das vezes, a curiosidade. Porém, as etapas de projeto e implementação de jogos podem ser bastante complexas, envolvendo conhecimentos de diversas áreas da computação, uma vez que normalmente deseja-se que um jogo completo apresente características como simulações consistentes, ambientes de jogo de fácil

compreensão, soluções plausíveis para o contexto do jogo, sensação de imersão, chances justas de vitória, participação do usuário no jogo e não ser um mero espectador, entre outras [Rouse III 2010].

O mais importante durante a criação de jogos é ter a capacidade de desenhar na tela, sendo que esta deve ser atualizada dinamicamente para atualizações de cena de acordo com o tempo, ou seja, um laço de repetição, para assim poder redesenhar a tela constantemente. Além disso, deve ser permitida alguma forma de interatividade do jogador para redesenhar a cena de acordo com a interação do usuário (movimentar o personagem para frente, para os lados, etc). Outro aspecto de interesse na criação de jogos é o fato de detectar colisões e tomar ações além de permitir de forma fácil a realização de transformações dos elementos do jogo.

Assim, para a aprendizagem de conceitos básicos, é possível trabalhar e criar pequenos jogos ou simulações mais simples, que não exijam tantos conhecimentos. Ou seja, é possível, apenas com conhecimentos básicos de programação, tais como programação estruturada (ou orientada a objetos), noção espacial (plano cartesiano), desvios condicionais, e laços de repetição, programar a lógica de um pequeno jogo. Para facilitar ainda mais esse processo, a parte de renderização e geração de primitivas e tratamento de colisão pode ser realizada por uma biblioteca gráfica, nesse caso, com a API proposta e desenvolvida neste trabalho.

O desenvolvimento desta proposta é centrado no uso da classe Canvas [Oracle 2016a] disponível no JavaFX. A partir dela, é possível desenhar primitivas, textos, figuras e transformações geométricas. A API é totalmente em português e com comentários nos códigos, que facilita o entendimento para o estudante. Com o uso dela, os estudantes podem desenvolver grande parte dos trabalhos da disciplina de computação gráfica (bidimensionais), sem se preocupar em utilizar a biblioteca OpenGL. Além disso, com ela é possível programar jogos 2D como Space Invaders [Taito 1978], Tetris [Nintendo 1989], e muitos outros. Também é possível exercitar conceitos básicos de laços de repetição e desvio condicionais, fazendo desenhos de figuras geométricas estáticas ou animadas e até mesmo tratando colisão entre elas, e exibir imagens de qualquer formato de arquivo. A API também suporta a utilização de um controlador de jogo (*gamepad*) para controlar as ações dos personagens criados.

Para a criação de pequenos jogos, ou até mesmo pequenos desenhos estáticos, a API conta com métodos para desenhos linhas e de primitivas que podem ser desenhadas apenas com seu contorno, ou de forma sólidas (preenchidas com uma cor predefinida), além de métodos para exibir imagens e textos sobre o canvas. Alguns dos principais métodos da biblioteca são apresentados na Tabela 1.

Para mudar a cor de contorno dos textos ou primitivas, é disponibilizado o método `contorno`, que pode receber como parâmetros a cor desejada e a espessura da linha a ser usada. Para mudar a cor de preenchimento de uma forma, é disponibilizado o método `preenchimento`, que recebe como argumento a cor a ser utilizada.

Os métodos para renderizar formas e textos no canvas disponibilizados pela API são: `circulo`, `retangulo`, `linha`, `texto` e `imagem`. Além disso, um método chamado `colisão` é fornecido, para tratar colisão entre formas retangulares, muito utilizado em jogos. Ao ser fornecida aos alunos, juntamente com a API, são disponibilizados três exemplos de

uso da mesma, que aplicam gradativamente todas as funcionalidades codificadas.

Tabela 1. Exemplo de algumas funções da FX Canvas2D API

FX_CG_2D_API(int fps, float w, float h)	Construtor da classe principal da API, responsável por inicializar altura, largura e fps da mesma, assim como definir as call-backs a serem usadas.
ponto(double x, double y, double tam)	Método para desenho de pontos 2D nas coordenadas X e Y com tamanho TAM
circulo(int x, int y, int l, int a, Estilo estilo)	Método para desenho de círculos sólidos ou apenas com a linha de contorno
contorno(double espessura, Color cor)	Método para mudar a cor e espessura do contorno de um objeto.
linha(double xi, double yi, double xf, double yf)	Método para desenhar uma linha que vai de [xi,yi] até [xf,yf]
texto(String texto, double x, double y, int tam, FontWeight tipo)	Método para escrever um texto na tela, nas coordenadas [x,y], com a fonte de tamanho tam e tipo tipo.
transladar(double x, double y)	Método para transladar um objeto para as coordenadas [x,y].

Para as transformações geométricas, são disponibilizados três métodos: rotacionar, transladar e escalar, que aplicam as transformações de rotação, translação e escala, respectivamente. Com esses métodos é possível definir uma entidade base no plano cartesiano e transformá-la para a renderização da cena, conforme necessário, como é feito normalmente em computação gráfica em aplicações que utilizam OpenGL. Além disso, utilizou-se o conceito das pilhas de matrizes de transformação em OpenGL, que permitem empilhar e desempilhar transformações para gerar cenas e movimentos fracionados. Para isso, a API conta com os métodos empilhar e desempilhar, que possibilitam aplicar diferentes transformações para cada trecho de código a ser renderizado.

Os métodos que devem ser codificados para a criação de uma aplicação com a FX Canvas2D são definidos como métodos abstratos, que o aluno deve codificar. Existem dois métodos principais que são o método desenhar e o método atualizar. O método desenhar é responsável por renderizar as formas, linhas e textos no canvas (na tela definida). Assim, esse método é chamado continuamente, atualizando a tela da aplicação em um determinado FPS (*frame-per-second*) definido pelo programador. Por outro lado, o método atualizar é um método auxiliar, onde variáveis usadas na programação podem ser atualizadas, onde cálculos podem ser realizados para atualizações de movimentos e posições, entre outros.

Os outros métodos abstratos da API são as *call-backs*, ou seja, métodos responsáveis por tratar ações de movimento, como por exemplo, cliques do mouse e teclas pressionadas no teclado. Esses métodos são, respectivamente: cliqueDoMouse, movimentoDoMouse, teclaPressionada, teclaLiberada, e teclaDigitada. Com a codificação de algum desses métodos, é possível fazer a interação do usuário com o jogo, como é demonstrando em exemplos nos resultados. Caso o estudante queira apenas realizar

desenhos estáticos para exercitar formas geométricas ou fazer animações autônomas, é necessário apenas a codificação do método desenhar.

Conhecendo-se as funcionalidades da API, para iniciar o desenvolvimento de uma aplicação que a utilize, o aluno deve criar uma classe padrão de JavaFX (classe que estende da classe *Application*), contendo um método *main* e o método *start* (método sobrescrito da classe *Application*). Dentro do método *start* é onde o aluno deverá instanciar um objeto da classe *Canvas2D_FX_API*, que é a API descrita e desenvolvida neste trabalho. No momento da instanciação desse objeto, os métodos abstratos a serem codificados são disponibilizados e o desenvolvimento tem início. Um exemplo para primeira aplicação é disponibilizado junto à API com um processo descrito e documentado, onde apenas são realizados desenhos estáticos de linhas, círculos, retângulos e textos.

3. Materiais e métodos

Para esta iniciativa de criar uma API para programação de Jogos 2D, foi pensado em um projeto com a finalidade de utilizá-la para aprimorar o raciocínio lógico dos estudantes, uma vez que os mesmos podem visualizar de forma gráfica o resultado da execução de seus códigos, criando pequenos “jogos”. Inicialmente foi desenvolvida a API através da linguagem Java, com as bibliotecas do JavaFX. Posteriormente, foram elaborados alguns exemplos de uso da mesma para disponibilização aos alunos, além da documentação de cada método e forma de utilização.

A duração total da utilização da API pelos estudantes foi de um semestre, em uma disciplina com carga horária de 36 horas (Computação Gráfica) no curso superior em Análise e Desenvolvimento de Sistemas de uma instituição federal. Esta foi utilizada em todas as aulas para estudo e produção de pequenos aplicativos. As aulas foram realizadas em laboratórios de informática, semanalmente. A primeira aula consistiu na apresentação e explanação da API, apresentando os exemplos criados, e as demais para propor desafios para criação de pequenos jogos, tendo auxílio do professor.

As atividades propostas tiveram como objetivo lembrar a utilização de conteúdos vistos nas aulas de algoritmos e de programação, como sequência de instruções, uso correto de métodos, funções e laços de repetição. Com as atividades propostas e por ser uma API de programação gráfica, os alunos conseguem ver e entender o que está aparecendo na tela do computador enquanto são digitados os comandos. Assim, os discentes são instigados a pensar e elaborar procedimentos para a realização de atividades específicas, facilitando assim, a compreensão da lógica envolvida na programação de computadores.

Após a implementação do projeto, se deu início ao processo investigativo das percepções discentes sobre a API utilizada em sala de aula. Para tanto, ocorreu um levantamento dos acadêmicos que estavam interessados em responder os questionamentos. Para a investigação das percepções discentes, optou-se pela técnica de entrevista, o que permitiu uma conversação entre as partes e proporcionou a informação necessária com vistas a consistência dos dados da pesquisa.

Segundo [May 2004], “as entrevistas geram compreensões ricas das experiências, opiniões, valores, aspirações, atitudes e sentimentos das pessoas”. Buscou-se com a entrevista a averiguação de fatos e obtenção de informações a utilização da API proposta

permitiram a melhor compreensão de como os alunos perceberam a formação construída com a prática em sala de aula. Para atingir estes objetivos, optou-se pela entrevista estruturada, seguindo um roteiro previamente estabelecido e com perguntas pré-determinadas.

Na entrevista estruturada, segundo [Marconi and Lakatos 2003], deve-se “obter dos entrevistados, respostas às mesmas perguntas, permitindo ao pesquisador que as respostas sejam comparadas e que se reflita com as diferenças entre os respondentes e não nas diferenças de perguntas”. [May 2004, p. 146] conclui que “a mesma pergunta é feita a cada pessoa da mesma forma”.

As entrevistas foram durante as últimas aulas do semestre, e tiveram duração de dez a quinze minutos dependendo do entrevistado. Com a análise das entrevistas foi reconhecida a relevância da API e as contribuições para a sua prática no ensino, estabelecendo um debate teórico-prático da temática.

4. Resultados e Discussões

A utilização de jogos é bastante antiga, uma vez que existem referências arqueológicas de jogos que datam de mais de 5000 anos [Fox and Verhovsek 2002]. Normalmente, estudantes da área da computação apresentam grande interesse por jogos, e se tratando disso, o ato de programar seu próprio jogo, por mais simples que seja, traz satisfação ao aluno e seu conhecimento e raciocínio lógico é reforçado. Segundo [Ramos 2011] utilizar jogos no contexto educativo, com a possibilidade de desenvolver competências relacionadas a uma área de conhecimento, pode tornar -se um recurso importante para o professor, mas este deve possuir metodologia adequada e uma intencionalidade justificada por um projeto.

Como resultado de desenvolvimento deste projeto, foi concebida a FX Canvas2D API, que permite ao aluno exercitar seus conhecimentos de programação e raciocínio lógico através da programação de pequenos jogos bidimensionais. Junto da API são fornecidos alguns exemplos de uso, conforme Figura 1 e Figura 2, para que os usuários se familiarizem com a utilização dos comandos.

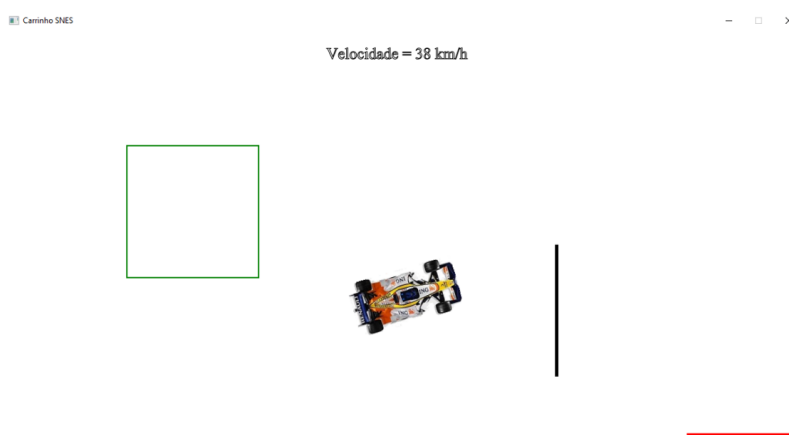


Figura 1. Exemplo de jogo de carro feito com a FX Canvas2D API

A Figura 1 mostra um exemplo de jogo de carro desenvolvido com a API, onde são exemplificados os conceitos de transformações geométricas como translação e rotação, além de aceleração e colisão.

A Figura 2 mostra um exemplo de jogo no estilo plataforma, também desenvolvido inteiramente com a FX Canvas2D API. Nesse jogo são explorados os conceitos de colisão e movimentação de personagem pelo cenário, mostrando tais funcionalidades aos estudantes.

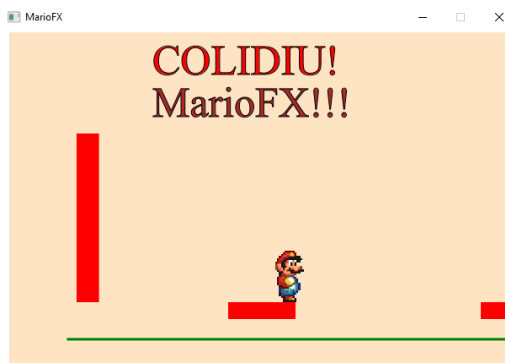


Figura 2. Exemplo de jogo estilo plataforma feito com a FX Canvas2D API

A biblioteca desenvolvida neste trabalho foi aplicada na disciplina de computação gráfica, conforme mencionado nos materiais e métodos, com o objetivo de aplicar na prática os fundamentos teóricos estudados, tais como, desenhos de primitivas gráficas, cálculos de distância entre pontes, utilização de vetores de posição e aceleração, detecção e tratamento de colisões.

Os alunos se mostraram motivados a utilizar a ferramenta e a classificaram como de fácil utilização visto que todos os comandos e funções são em português. Como trabalho final da disciplina foi proposto o desenvolvimento de um clássico jogo bidimensional Pac-Man [Namco 1983], onde cada aluno o desenvolveria individualmente. Diante do desafio proposto, todos os alunos conseguiram com sucesso obter uma versão muito parecida com a original, utilizando imagens no formato “gif” para os personagens (*pac-man* e fantasmas) e primitivas como linhas, retângulos e círculos para o desenho do labirinto do cenário. Um exemplo de trabalho desenvolvido por um dos alunos pode ser observado na Figura 3.

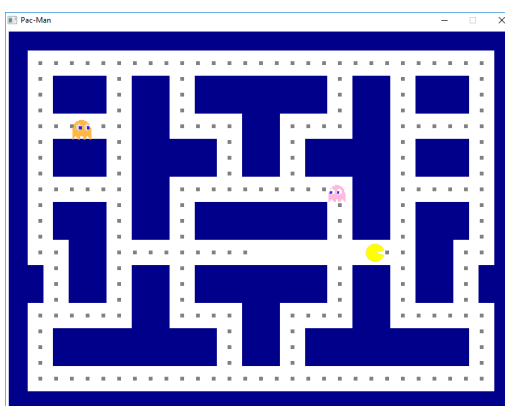


Figura 3. Exemplo de jogo Pac-Man desenvolvido por um aluno

A avaliação dos alunos quanto à utilização da API foi positiva, e os mesmos, relataram que seria interessante o uso das ferramentas em outras disciplinas também,

como em estrutura de dados, por exemplo, para poderem programar e visualizar de forma gráfica as estruturas.

Conforme apresentado, o jogo passa a ser um recurso importante para ser utilizado em sala de aula, pois favorece a ação do aluno, exige que ele coordene diferentes pontos de vista, torna o erro um fato observável, propõe uma situação-problema e gera dados para análise e reflexão. Assim, o uso da API proposta pode proporcionar uma melhor experiência de aprendizagem para os alunos, onde eles poderão desenvolver melhor seu raciocínio lógico e melhorar suas práticas de programação através da criação de pequenos jogos.

5. Conclusões

A proposta deste trabalho foi de desenvolver uma forma alternativa e lúdica para a aprendizagem de programação e raciocínio lógico aplicada na disciplina de computação gráfica. A FX Canvas2D gera possibilidades do ensino de alguns conceitos relacionados a computação gráfica e tem o objetivo principal reforçar o raciocínio lógico do estudante, na medida em que o mesmo se sente atraído a desenvolver pequenos jogos e utilizar as funcionalidades disponibilizadas pela API. Ressalta-se que todos os comandos foram desenvolvidos utilizando-se a língua portuguesa, tendo comentários no formato JavaDoc e métodos com nomenclatura condizente com sua funcionalidade, facilitando a compreensão dos mesmos.

Além do reforço na aprendizagem e retomadas de conceitos de programação como decisões e estratégias de programação, o uso desta API proposta pode resultar em uma introdução a conceitos mais complexos de computação gráfica, tais como transformações e noções de posicionamento espacial, que podem ser mais aprofundados utilizando-se a biblioteca OpenGL. É preciso ensinar as pessoas a utilizar o computador em benefício próprio e coletivo, como uma ferramenta de serviço ao homem, e assim fazer parte da educação.

Referências

- Azevedo, E. and Conci, A. (2003). *Computação gráfica: teoria e prática*. Elsevier.
- Cohen, M. and Manssour, I. H. (2006). *OpenGL: uma abordagem prática e objetiva*. São Paulo: Novatec editora. 486 p.
- de Souza, C. M. (2016). Visualg-ferramenta de apoio ao ensino de programação. *Revista Eletrônica TECCEN*, 2(2):01–09.
- dos Santos, R. P. and Costa, H. A. X. (2005). Tbc-aed: Um software gráfico para apresentação de algoritmos e estruturas de dados aos iniciantes em computação e informática. Disponível em: <http://www.cos.ufrj.br/~rps/pub/completos/2005/COMPSULMT.pdf>. Acesso em agosto de 2016.
- Fox, D. and Verhovsek, R. (2002). *Micro Java game development*. Addison-Wesley Professional.
- Marconi, M. d. A. and Lakatos, E. M. (2003). *Fundamentos de metodologia científica*. 5. ed.-São Paulo: Atlas.
- May, T. (2004). *Pesquisa social: questões, métodos e processos*. Porto Alegre: Artmed, 3 edition.

- Medeiros, T. J., da Silva, T. R., and da Silva Aranha, E. H. (2013). Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. *RENOTE*, 11(3).
- Namco (1983). Pac-man. Disponível em: <http://www.ign.com/games/pac-man/arcade-9644>. Acesso em: 10 agosto de 2016.
- Nintendo (1989). Tetris. Disponível em: <http://www.ign.com/games/tetris-nintendo/nes-7298>. Acesso em: 10 agosto de 2016.
- Oracle (2016a). Canvas do javafx documentação. Disponível em: <http://docs.oracle.com/javafx/2/canvas/jfxpub-canvas.htm>. Acesso em: 25 julho de 2016.
- Oracle (2016b). Javafx - documentação oficial. Disponível em: <http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>. Acesso em: 20 julho de 2016.
- Ramos, R. A. d. O. (2011). O uso de mídias interativas na compreensão de conceitos da lógica computacional. 90 f. Dissertação (Mestrado em Mídias Digitais) - Pontifícia Universidade Católica de São Paulo.
- Rapkiewicz, C. E., Falkembach, G. A. M., Seixas, L. M. J. d., Santos, N. d. S. R. S. d., Cunha, V. V. d., Klemann, M., et al. (2006). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *VIII Ciclo de Palestras Novas Tecnologias na Educação. RS - Porto Alegre: CINTED-UFRGS, V. 4 N. 2, Dezembro*.
- Rouse III, R. (2010). *Game design: Theory and practice*. Jones & Bartlett Learning.
- Sá, E., Teixeira, J., and Fernandes, C. T. (2007). Design de atividades de aprendizagem que usam jogos como princípio para cooperação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1, pages 539–549.
- SBC (2016). Currículo de referência da sociedade brasileira de computação. Disponível em: <http://www.sbc.org.br/documentos-da-sbc/category/131-curriculos-de-referencia>. Acesso em: agosto de 2016.
- Shreiner, D., Woo, M., and Neider, J. (2004). *OpenGL: programming guide: the official guide to learning opengl, version 1.4*. 4rd Edition. Reading, Massachusetts: Addison-Wesley.
- Silva, M. and Júnior, P. A. P. (2013). Aesda: Ferramenta educacional gráfica extensível para ensino de algoritmos de ordenação e pesquisa com ênfase na análise da eficiência de algoritmos. <http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0032.pdf>.
- Taito (1978). Space invaders. Disponível em: <http://www.ign.com/games/space-invaders/arcade-14048>. Acesso em: agosto de 2016.
- Valente, J. A. (2003). *Formação de educadores para o uso da informática na escola*. Campinas: UNICAMP/NIED.
- Wright Jr, R. S., Haemel, N., Sellers, G. M., and Lipchak, B. (2010). *OpenGL SuperBible: comprehensive tutorial and reference*. Pearson Education.