

PROJETO TCC - BCC	ANO/SEMESTRE:	2018/2
-------------------	---------------	--------

REFATORANDO VISEDU-CG NO MOTOR DE JOGOS UNITY

Peterson Boni Buttenberg

Prof. Dalton Solano dos Reis – Orientador

1 INTRODUÇÃO

A tecnologia evolui constantemente e a cada dia, uma nova tecnologia é criada ou aperfeiçoada, contribuindo para o desenvolvimento de muitas áreas. Dentre as diversas áreas do conhecimento, há um grande avanço para que tecnologias interativas possam ajudar no ensino e aprendizagem dos acadêmicos no ensino superior, criando métodos híbridos de ensino (FERREIRA, 1998). Dados estatísticos comprovam que o desempenho dos alunos que fazem uso de várias metodologias de ensino-aprendizagem são melhores que os de alunos que continuam utilizando o método tradicional, repleta de aulas expositivas.

Dessa forma, o VisEdu-CG vêm ao encontro dessas ideias, criando novas metodologias ativas para esse processo. Conforme Reis (2018) "o VisEdu-CG é um projeto para desenvolver uma plataforma Web que permita os alunos da disciplina de Computação Gráfica do curso de Ciências da Computação praticarem os conceitos ministrados nesta disciplina". Essa aplicação contou com o desenvolvimento de vários módulos específicos, dentre eles pode-se citar o motor de jogos, matemática, estatística, processamento de imagens, realidade aumentada e simulação.

Para que o aplicativo tenha uma evolução e continue a ser aprimorado, deve haver uma mudança em sua linguagem. Segundo Barrozo, Vinhas e Reis (2012), a refatoração é uma técnica utilizada para desenvolver softwares de qualidade de fácil compreensão, bem estruturado e de fácil manutenibilidade. Utilizando essa técnica o software é alterado de uma forma a tornar-se mais fácil de ser modificado e com menos propensão a erros, porém, seu comportamento continua o mesmo.

Entretanto, existe uma carência maior em relação às funcionalidades do WebGL, o que torna o VisEdu-CG um sistema igualmente limitado. Em razão disso, este trabalho propõe migrar as tecnologias utilizadas atualmente no VisEdu-CG para o motor de jogos Unity, tornando-o um sistema mais acessível e com maior possibilidade de modificação e extensões futuras. Com isso, estima-se que seja possível entender o funcionamento das transformações geométricas, entre outras rotinas, e simultaneamente cria-se uma familiaridade com o ambiente gráfico Unity.

1.1 OBJETIVOS

O objetivo deste trabalho é refatorar o trabalho intitulado “VisEdu-CG 4.0: Visualizador de Material Educacional” (KOEHLER, 2015) para o motor de jogos Unity.

Os objetivos específicos do trabalho são:

- a) converter a ferramenta de visualização gráfica atual para o motor de jogos Unity;
- b) apresentar uma interface de ajuda em forma de tutorial informando os passos a serem seguidos;
- c) disponibilizar funcionalidades gráficas, como câmeras e texturas.

2 TRABALHOS CORRELATOS

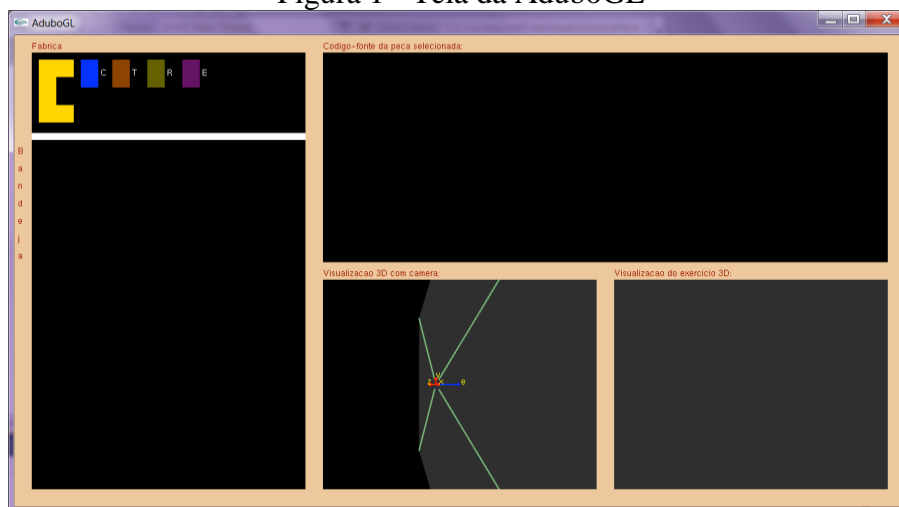
A seguir, são apresentados quatro trabalhos correlatos com características semelhantes aos principais objetivos do estudo proposto. A seção 2.1 detalha o AduboGL (ARAÚJO, 2012), uma aplicação didática, que colabora no aprendizado de computação gráfica. A seção 2.2 descreve o AduboGL 2.0 (SCHRAMM, (2012), sistema voltado para iPad e auxilia no aprendizado de conceitos de computação gráfica, e a seção 2.3 expõe o VisEdu-CG (MONTIBELER, 2014) e o VisEdu-CG 3.0 (NUNES, 2014), aplicações web direcionadas para o aprendizado de computação gráfica, com foco em conceitos de câmeras e transformações geométricas.

2.1 ADUBOGL

Conforme apresentado por Araújo (2012), o AduboGL, é uma aplicação voltada ao aprendizado da computação gráfica com foco nas transformações geométricas. Ele utiliza a biblioteca OpenGL para montar o cenário 2D presente na aplicação e tem seu resultado em um espaço 3D.

A Figura 1 apresenta a tela da AduboGL, ela se divide em 4 sub janelas. A janela da esquerda contém duas partes, sendo a parte da fábrica que possui as peças a serem usadas nos exercícios e a parte da bandeja onde serão montados os exercícios utilizando as peças. A segunda janela é a de código-fonte da peça selecionada e as outras duas telas restantes apresentam em 3D o resultado do exercício montado.

Figura 1 - Tela da AduboGL



Fonte: Araújo (2012).

A aplicação foi desenvolvida utilizando a linguagem de programação C++ no ambiente Microsoft Visual C++ 2010 Express. A biblioteca OpenGL 4.2 utilizada para o desenvolvimento de aplicações gráficas 2D e 3D e a biblioteca V-ART que é um *framework* utilizado para facilitar a criação de ambientes em 3D.

2.2 ADUBOGL 2.0

Segundo Schramm (2012), o AduboGL ES 2.0 é uma aplicação desenvolvida com intuito de auxiliar no aprendizado de conceitos relacionados a computação gráfica direcionado para o iPad. A aplicação tem como foco principal a utilização da biblioteca OpenGL ES na versão 2.0. A biblioteca OpenGL ES 2.0 é baseada em um módulo chamado *shaders*, que são pedaços de códigos divididos entre *vertex shaders* e *fragment shaders*. No *vertex shader* são processados os vértices com um propósito geral, como computar equações de luz, transformar matrizes de pontos e gerar ou transformar coordenadas de texturas (MUNSHI; GINSBURG; SHREINER, 2009, p. 38). O *fragment shader* é fundamental na produção de efeitos em texturas, luz por pixel e sombras (MUNSHI; GINSBURG; SHREINER, 2009, p. 181).

O AduboGL ES 2.0 disponibiliza uma área chamada de fábrica, onde ficam as peças coloridas a serem encaixadas e ao lado uma área para que esses encaixes sejam feitos, chamada de bandeja. Segundo Schramm (2012), quando um dos blocos *GLClearColor* ou *GLDepthTest*, é selecionado, exibe-se um botão chamado parâmetros, que ao ser selecionado exibe uma nova janela com opções para o bloco selecionado. A janela de parâmetros do bloco *GLClearColor* tem opções relacionadas a cor de fundo, já o parâmetro *GLDepthTest*, habilita ou desabilita a profundidade do objeto.

A Figura 2 demonstra um exercício feito no AduboGL ES 2.0. Na figura são apresentada duas sub janelas. Na janela da esquerda (fábrica) estão os blocos a serem utilizados e na janela da direita (bandeja) os blocos já encaixados após terem sido arrastados para suas devidas posições.

Figura 2 - Tela do AduboGL ES 2.0



Fonte: Schramm (2012).

A plataforma foi implementada utilizando a linguagem de programação Objective-C 2.0. Para o desenvolvimento das telas principais foram utilizados os *frameworks Foundation, UIKit e CoreGraphics*. A visualização gráfica tornou-se possível através da utilização dos *frameworks QuartzCore e OpenGL ES*. A ferramenta de desenho gráfico vetorial bidimensional CorelDRAW X4 foi utilizada para desenhar as imagens dos blocos e calcular os pontos de encaixe. (SHRAMM, 2012).

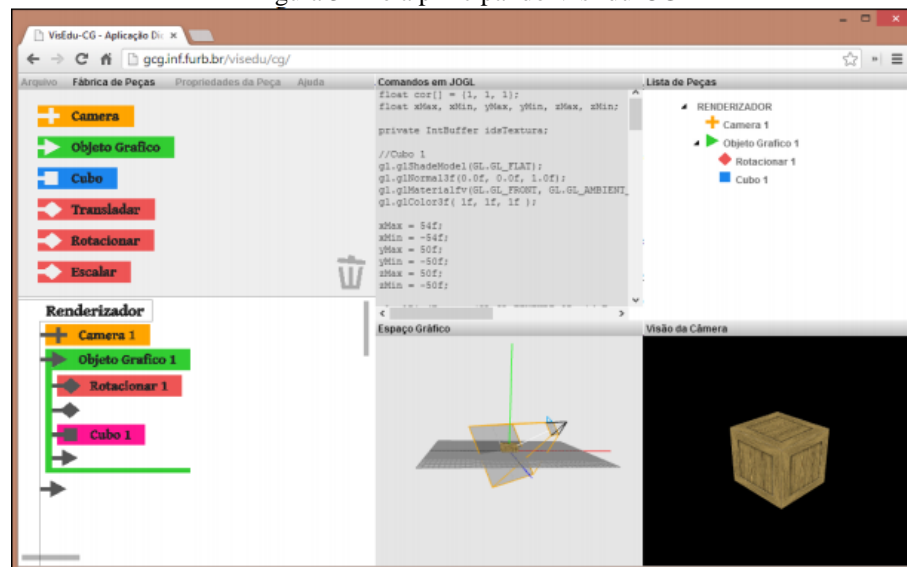
2.3 VISEDUCG E VISEDUCG 3.0

O VisEdu-CG é um projeto desenvolvido em uma plataforma web que possibilita aos alunos de computação gráfica, utilizarem os conceitos aprendidos na disciplina. De acordo com Montibeler (2014) é uma aplicação web direcionada ao aprendizado de computação gráfica, focada em conceitos de câmera sintética, grafo de cena e transformação geométrica em um ambiente tridimensional. Nela existem blocos que, se encaixados corretamente, permitem a visualização de formas geométricas. Se uma das peças forem selecionadas o código referente a mesma é demonstrado, aumentando a compreensão e o entendimento da ordem lógica dos comandos.

Na Figura 3 é apresentado a tela inicial do VisEdu-CG, que é dividida em seis subjanelas. A Fábrica de Peças é onde ficam as peças a serem utilizadas. O Renderizador,

onde são arrastadas as peças até o local determinado. Comando em JOGL, é a subjanela onde é mostrado o código em Java ao selecionar uma das peças. Espaço gráfico e a Visão da Câmera são as sub telas onde é exibido o resultado gerado a partir do encaixe das peças no Renderizador. E a subjanela de Lista de Peças, que lista todas as peças encaixadas no painel de montagem.

Figura 3 - Tela principal do VisEdu-CG



Fonte: Montibeler (2014).

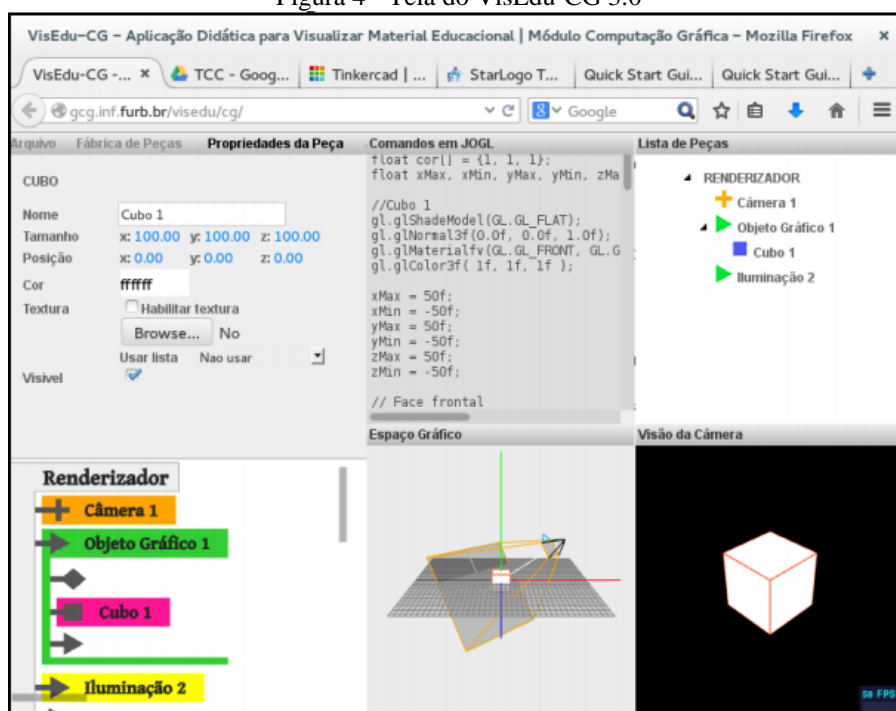
A linguagem utilizada é Java, com a biblioteca JOGL. O uso da biblioteca Three.js foi essencial para a visualização dos objetos no espaço gráfico 3D da aplicação. Essa biblioteca é voltada para renderização em WebGL (MONTIBELER, 2014).

Nunes (2014), teve como objetivo estender o VisEdu-CG, incluindo novas funcionalidades à aplicação, deixando-a mais completa e aumentando ainda mais o entendimento dos conceitos de computação gráfica. Ele incrementou no projeto o tipo de objeto polígono e spline 3D, adicionar na representação gráfica o uso de iluminação.

A Figura 4 demonstra a tela principal do VisEdu-CG 3.0. Nota-se que não houve mudanças perceptíveis na visualização ou forma de apresentação e disposição dos objetos gráficos nas telas.

Além da inclusão de novos objetos gráficos e a iluminação, a aplicação simula uma visualização de um espaço 2D mesmo sendo implementada em 3D, através de um posicionamento fixo da câmera e desabilitando a manipulação dos objetos pelo *mouse* (NUNES, 2014).

Figura 4 - Tela do VisEdu-CG 3.0



Fonte: Nunes (2014).

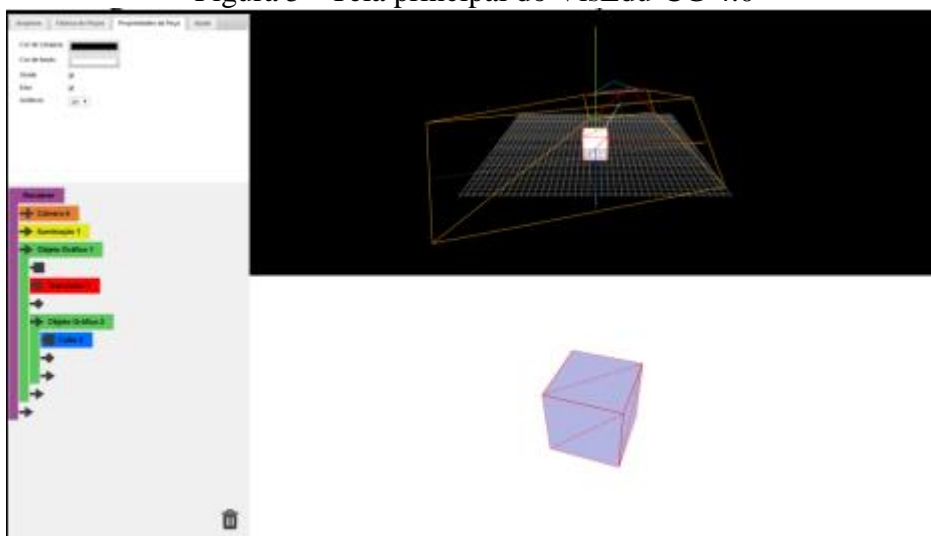
3 SISTEMA ATUAL

Como afirma Koehler (2015), o VisEdu-CG 4.0 teve um aumento significativo em sua programação para que fosse concretizado a proposta de reestruturar o visualizador de material educacional e com isso permitiu que a integração entre o motor de jogos fosse mais fácil. A visualização do objeto gráfico e o espaço gráfico foram melhores com a remoção do painel de Lista de Peças e Comandos e JOGL. A cor preta foi atribuída a esse espaço para melhor a visualização do objeto na cena.

A versão 4.0 da plataforma VisEdu-CG passa a ter como foco na tela principal os painéis de Visão da Câmera e Espaço Gráfico. A Visão da Câmera demonstra o ângulo de visão a partir de onde o objeto gráfico será visualizado. Em uma cena 3D, a câmera pode ser movimentada para que seja possível visualizar todos os lados do objeto, já em uma cena 2D a câmera permanece estática. O Espaço Gráfico concebe a visualização do objeto gráfico escolhido que pode ser modificado de acordo com a posição da câmera. Na Figura 5 é possível ver a interface atual da aplicação.

O visualizador de material educacional teve um aumento significativo no volume de programação para concluir o trabalho, o que permitiu uma fácil integração com o motor de jogos e neutralizou o risco de incompatibilidade entre os programas. Para a criação e manutenção das abas, foram usadas tecnologias como HTML, CSS, Javascript e JQuery (KOEHLER, 2015).

Figura 5 - Tela principal do VisEdu-CG 4.0



Fonte: Koehler (2015).

Koehler (2015), realizou uma integração de um visualizador de material educacional com um motor de jogos HTML5, no qual foi proposto como objetivo do trabalho. A biblioteca Three.js foi incluída no projeto e, diante disso, o motor de jogos teve que ser adaptado para suportar a mesma e assim fornecer a capacidade, competência e eficácia necessária para uma visualização tridimensional adequada. A biblioteca Three.js se manteve estável e performática mesmo inserindo um grande número de elementos gráficos.

4 PROPOSTA DA FERRAMENTA

Este capítulo tem como objetivo apresentar a justificativa para elaboração deste trabalho, assim como os requisitos e metodologia para o desenvolvimento.

4.1 JUSTIFICATIVA

No Quadro 1 é demonstrado um comparativo e as características entre os trabalhos correlatos utilizados para estender o trabalho proposto. Uma das características descritas é a transformação geométrica (translação, escala e rotação). Essa característica torna-se ainda mais visível no VisEdu-CG, pois na plataforma, os conceitos de transformação geométrica são essenciais para a construção de objetos gráficos e cenas 3D. Todas ferramentas possuem uma forma de programação a partir de blocos de encaixe, indicando previamente e ensinando o usuário onde deve ser conectada determinada peça.

Quanto ao tipo de visualização, todos os sistemas permitem que sejam visualizados os objetos gráficos em formato tridimensional, porém o AduboGL disponibiliza uma visualização bem simples e pouco detalhada em comparação com o VisEdu-CG e VisEdu-CG 3.0, que possibilitam uma visualização muito mais avançada e bem apresentável.

O AduboGL 2.0 não permite modificar a visão de câmera, ou seja, o objeto gráfico da cena não pode ser visualizado de ângulos diferenciados, ao invés disso ele possui uma câmera fixa. Já os demais sistemas possuem visão de câmera e permitem a visualização do objeto gráfico de qualquer ângulo de visão.

As plataformas VisEdu-CG e VisEdu-CG 3.0 podem somente ser executadas no *browser*, ou seja, elas necessitam de acesso à internet para sua utilização, diferentemente das demais ferramentas que podem ser executadas no Desktop (AduboGL) ou tablet (AduboGL 2.0). O VisEdu-CG e VisEdu-CG 3.0 possuem uma forma de ajuda ou documentação para auxiliar o usuário na sua utilização, mas nenhuma delas disponibiliza tutoriais interativos em sua interface, a qual é um dos objetivos para estender o trabalho proposto.

Quadro 1 – Comparativo entre trabalhos correlatos

Característica / Correlatos	AduboGL (2012)	AduboGL 2.0 (2012)	VisEdu-CG (2014) VisEdu-CG 3.0 (2009)
transformação geométrica	Sim	Sim	Sim
programação visual	Sim	Sim	Sim
tipo de visualização 3D	Sim	Sim	Sim
visão de câmera	Sim	Não	Sim
trabalha on-line	Não	Não	Sim
tutoriais interativos	Não	Não	Não

Fonte: elaborado pelo autor.

Nota-se uma necessidade de modificação no sistema atual para torna-lo mais acessível, pois utilizará uma ferramenta gratuita (podendo ser obtida uma versão “paga” mais completa) com todas as funcionalidades necessárias para o desenvolvimento da aplicação proposta. Em função disso, propõe-se também converter a linguagem de programação atual (javascript) para a linguagem C#, tendo como pressuposto, fazer com que a aplicação seja de fácil manutenção.

Conclui-se que o ambiente VisEdu-CG, tem grande importância no aprendizado de alunos da disciplina de computação gráfica. Com o passar do tempo, foram feitas varias modificações e extensões que melhoraram a forma de exibição dos objetos gráficos, consequentemente melhorando o entendimento dos alunos.

4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação desenvolvida deve:

- a) permitir a escolha entre trabalhar em um espaço bidimensional ou tridimensional (Requisito Funcional - RF);
- b) desenhar componente do tipo Cubo (RF);
- c) disponibilizar um tutorial de ajuda inicial ao usuário (RF);
- d) permitir utilizar o conceito de câmera (RF);
- e) ser desenvolvido na linguagem C# (Requisito Não Funcional - RNF);
- f) utilizar o motor de jogos Unity (RNF).

4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: realizar levantamento bibliográfico sobre refatoração para o motor de jogos Unity e inclusão de tutoriais para melhor aprendizado de Computação Gráfica para os acadêmicos do curso de Ciências da Computação;
- b) elicitação de requisitos: detalhar e reavaliar os requisitos e, se necessário, especificar outros a partir das necessidades observadas durante a revisão bibliográfica;
- c) especificação e análise: utilizar as funcionalidades da ferramenta Star UML, para elaborar diagramas de classe e de atividades da Unified Modeling Language (UML);
- d) implementação da ferramenta: implementar a aplicação proposta, que visa migrar as funcionalidades existentes no trabalho de Koehler (2015), utilizando a linguagem de programação C# no ambiente de desenvolvimento Visual Studio em conjunto com o ambiente gráfico Unity que será responsável pela parte de visualização e usabilidade da aplicação.
- e) testes: elaborar testes de usabilidade da aplicação e com isso colher informações necessárias sobre seu processo de execução e buscar melhorias no intuito de desenvolver um sistema bem estruturado.

Quadro 2 – Cronograma de atividades a serem realizadas

etapas / quinzenas	2019									
	ago.		set.		out.		nov.		dez.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
especificação e análise										
implementação da ferramenta										
testes										

Fonte: elaborado pelo autor.

5 REVISÃO BIBLIOGRÁFICA

Este capítulo descreve brevemente os assuntos que fundamentarão o estudo a ser realizado: refatoração de sistemas, o motor de jogos Unity e inclusão de tutoriais para melhor aprendizado de Computação Gráfica para os acadêmicos do curso de Ciências da Computação.

5.1 REFATORAÇÃO DE SISTEMAS

A refatoração de códigos possibilita ao desenvolvedor a inclusão de novas funcionalidades à aplicação que antes não eram suportadas ou estavam limitadas por questões tecnológicas. Tem por objetivo principal permitir que as aplicações continuem em evolução constante com a liberação de novas versões aos usuários. Quando o código é refatorado ele se torna mais legível e eficiente promovendo a reutilização do código e facilitando a manutenibilidade, isso faz com que o processo de desenvolvimento se torne mais ágil e eficaz (BARROZO; VINHAS; REIS, 2012).

Segundo Maia (2015), conforme o software evolui, a complexidade do mesmo aumenta. Para reduzir a complexidade do código é necessária a utilização da refatoração, tornando-o mais extensível e reutilizável. As técnicas de transformação de código, como reestruturação e reengenharia, refatorações são mais facilmente realizadas utilizando ferramentas automatizadas, ocasionando uma redução no risco de introduzir novos erros devido a intervenções manuais, e faz com que a quantidade de testes diminua a cada vez que o código é modificado. Existem muitas ferramentas de refatoração disponíveis para praticamente todas as linguagens de programação. Ademais, a maior parte das ferramentas disponíveis tem uma customização não tão facilitada, aumentando muito a dificuldade para um desenvolvedor criar novas operações de refatoração, ou mesmo adaptar as existentes para sua realidade.

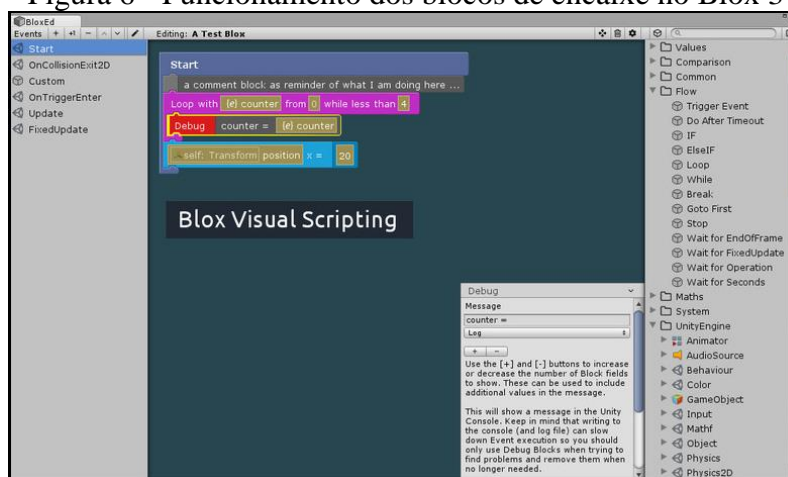
Uma forma de manter um sistema orientado a objetos com código limpo e de fácil entendimento é mudar sua arquitetura utilizando padrões de projeto, que por sua vez, oferecem soluções eficientes para problemas recorrentes encontrados em todos os tipos de sistemas. Os padrões de projetos facilitam a reusabilidade e manutenibilidade até mesmo em sistema que não são projetados para isso (RAPELI, 2006).

5.2 MOTOR DE JOGOS UNITY

O motor de jogos Unity 3D é uma ferramenta que permite a compilação de jogos e aplicativos em 2D e 3D. A *engine* oferece recursos de renderização, iluminação, terrenos, partículas, física, áudio, programação e *networking*. A renderização oferecida pela Unity 3D proporciona a possibilidade de criar jogos com gráficos com realismo, suportando vários efeitos de iluminação e texturas. O Unity 3D apresenta na criação de terrenos a implantação de vegetação e sua personalização dependendo do cenário a ser desenvolvido (WATKINS, 2011).

A plataforma possibilita a alteração e edição de objetos do jogo em tempo de execução, é possível testar o jogo e ao mesmo tempo alterar códigos de programação e editar objetos que estejam no cenário, facilitando a correção de eventuais problemas. Na *engine* também é possível, importar *assets*, que são objetos gráficos ou materiais prontos, para dentro da aplicação. Esses *assets* podem ser baixados na loja do Unity chamada *Assets Store*, onde alguns itens são gratuitos e outros pagos e são criados pelos próprios desenvolvedores do Unity. Um dos *assets* existentes na *Assets Store* é o Blox 3, que é um gerador de scripts onde é possível adicionar comportamentos a objetos através de uma interface *drag-and-drop* com blocos que são interconectados (ASSETSTORE UNITY, 2018). Na Figura 6 é possível entender o funcionamento dos blocos de encaixe presentes no Blox 3.

Figura 6 - Funcionamento dos blocos de encaixe no Blox 3



Fonte: AssetStore Unity (2018).

As linguagens de programação suportadas pelo Unity 3D eram: *JavaScript*, C# e Boo, onde era possível integrar no mesmo projeto arquivos diferentes escritos nas três linguagens, por exemplo, um script feito em JavaScript consegue comunicar-se com outro feito em C#, ou seja, seria possível desenvolver um sistema paralelamente em linguagens diferentes e posteriormente fazer a integração entre as tecnologias (WATKINS, 2011). Atualmente, o Unity padronizou a programação da plataforma somente para a linguagem C#.

5.3 TUTORIAS INTERATIVOS

Aplicações com fins pedagógico-didáticos estão sendo aplicadas em várias áreas do conhecimento. Pesquisas mostram que métodos híbridos de ensino-aprendizagem tornam a experiência dos alunos mais prazerosa e traz mais benefícios a curto e a longo prazo, além do incentivo à inovação e a melhora significativa apontada em diversos métodos de avaliação. Conforme aponta Moran (2017), essas novas tecnologias não só auxiliam em todo esse processo, como também incentiva os alunos a se tornarem produtores, não apenas receptores de conteúdo. Tutoriais vêm a contribuir para a aquisição de novas práticas, introduzindo de forma objetiva a utilização de uma nova ferramenta.

Ter um acompanhamento individual nas tarefas exigidas por uma ferramenta se torna indispensável para a boa utilização de uma aplicação e para que possa ser aproveitado de toda a capacidade do sistema, para isso, um recurso de assistência visual é de grande ajuda no processo de aprendizagem e conhecimento na questão de usabilidade em sistema de software. Isso faz com que o usuário sinta-se mais próximo e tenha uma rápida compreensão e ambientação em qualquer ferramenta (SANTOS, 2014).

REFERÊNCIAS

ARAÚJO, Luciana. **AduboGL – Aplicação didática usando a biblioteca Open GL:** documentação. Blumenau, 2012. Disponível em: < http://www.bc.furb.br/docs/MO/2012/350348_1_1.pdf >. Acesso em: 03 jul. 2012.

ASSETSTORE UNITY. **Blox Game Systems**. [S.l.], [2018]. Disponível em: < <https://assetstore.unity.com/packages/templates/systems/blox-game-systems-113619> >. Acesso em: 03 nov. 2018.

BARROZO, Graciele Castilho; VINHAS, Hingridi Marques; REIS, José Cláudio de Sousa. **Refatoração: Aperfeiçoando um código existente**. Refatoração, Alfenas, v. 7, n. 1, p.1-15, 2012. Disponível em: < <http://revistas.unifenas.br/index.php/RE3C/article/view/12/10> >. Acesso em: 21 set. 2018.

FERREIRA, Vítor F. **As técnicas interativas no ensino**. Niterói, p. 780-786, mar. 1998. Disponível em: <<http://www.scielo.br/pdf/qn/v21n6/2913>>. Acesso em: 19 set. 2018.

KOEHLER, William Fernandes. **VisEdu-CG 4.0**: visualizador de material educacional. 2015. 90 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Fundação Universidade Regional de Blumenau, Blumenau, 2015.

MAIA, Paulo H. M. **Um Ambiente para Refatoração de Código Java Utilizando Tecnologias XML**. Fortaleza, p. 1-3, [2015?]. Disponível em: <https://www.researchgate.net/profile/Nabor_Mendonca/publication/268341050_Um_Ambiente_para_Refatoracao_de_Codigo_Java_Utilizando_Tecnologias_XML/links/54f917e30cf210398e974c3a.pdf>. Acesso em: 31 out. 2018.

MONTIBELER, James Perkison. **VisEdu-CG**: Aplicação didática para visualizar material educacional, módulo de computação gráfica. 2014. 106 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Fundação Universidade Regional de Blumenau, Blumenau, 2014.

MORAN, J. **Metodologias ativas para uma aprendizagem mais profunda**. Educatrix – Dossiê Currículo, São Paulo: Moderna, a. 7, n. 12, p. 66-69, 2017. Disponível em: <http://www2.eca.usp.br/moran/wp-content/uploads/2013/12/metodologias_moran1.pdf>. Acesso em: 18 set. 2018.

MUNSHI, Aaftab; GINSBURG, Dan; SHREINER, Dave. **OpenGL ES 2.0 programming guide**. Boston: Pearson Education, 2009.

NUNES, Samuel Anderson. **VisEdu-CG 3.0**: Aplicação didática para visualizar material educacional, módulo de computação gráfica. 2014. 89 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Fundação Universidade Regional de Blumenau, Blumenau.

RAPELI, Leide Rachel Chiusi. **Refatoração de sistemas Java utilizando padrões de projeto**: um estudo de caso. 2006. 130 p. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Universidade Federal de São Carlos, São Carlos. Disponível em: <<https://repositorio.ufscar.br/bitstream/handle/ufscar/626/DissLRCR.pdf?sequence=1&isAllowed=y>>. Acesso em: 02 nov. 2018.

SANTOS, Rafael S. et al. eTutor: Um ambiente de aprendizagem interativo. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, [S.l.], p. 476-485, 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2976/2709>>. Acesso em: 03 nov. 2018.

SCHRAMM, Elizandro. **AduboGL ES 2.0**: Aplicação didática usando a biblioteca OpenGL ES 2.0 no IOS. 2012. 79 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Fundação Universidade Regional de Blumenau, Blumenau.

WATKINS, Adam. **Creating Games With Unity And Maya**. Londres: Focal Press, 2011.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO (PROJETO) – PROFESSOR TCC I

Acadêmico(a): _____

Avaliador(a): _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
ASPECTOS METODOLÓGICOS	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			
	9. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?			
	10. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?			
	11. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?			
	As citações obedecem às normas da ABNT?			
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?			

PARECER – PROFESSOR DE TCC I OU COORDENADOR DE TCC:

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 4 (quatro) itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos 4 (quatro) itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: () APROVADO () REPROVADO

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

FORMULÁRIO DE AVALIAÇÃO (PROJETO) – PROFESSOR AVALIADOR

Acadêmico(a): _____

Avaliador(a): _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			

PARECER – PROFESSOR AVALIADOR:

O projeto de TCC será reprovado, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **5 (cinco)** tiverem resposta ATENDE PARCIALMENTE.

PARECER: () APROVADO () REPROVADO

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.