

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
( X ) PRÉ-PROJETO	( ) PROJETO	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 vem se desenvolvendo a passos largos. A cada dia que passa, 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 de encontro à essas 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.

Chega-se então a um ponto crucial para a evolução dessa aplicação, a necessidade de mudança da linguagem para que possa continuar se aprimorando. 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.

Portanto, pode-se observar uma carência maior em relação às funcionalidades do WebGL, o que torna o VisEdu-CG um sistema igualmente limitado. Dessa forma, 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, será possível entender o funcionamento das transformações geométricas e simultaneamente criar familiaridade com o ambiente gráfico Unity.

## 1.1 OBJETIVOS

O objetivo deste trabalho é refatorar o sistema atual para o motor de jogos Unity e estender o trabalho intitulado “VisEdu-CG 4.0: Visualizador de Material Educacional” desenvolvido por Koehler (2015).

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

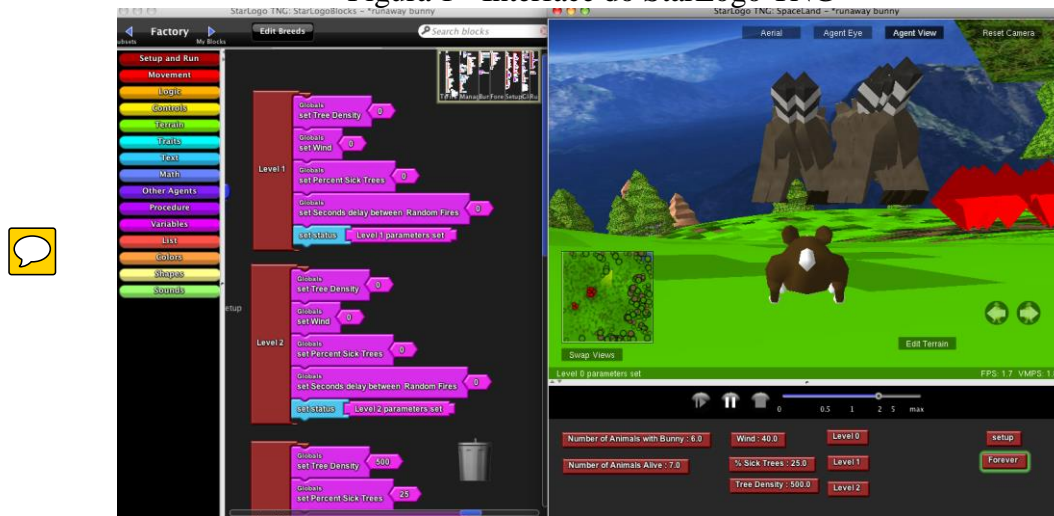
São apresentados três trabalhos com características semelhantes aos principais objetivos do estudo proposto. O primeiro é uma linguagem de simulação, educacional, procedural, baseada em agentes conhecida por StarLogo TNG (STEP, 2018), o segundo é uma plataforma web de criação de cenas 3D, chamada de WebGL Studio (AGENJO, 2016) e o terceiro é um *framework* que ensina programação de uma forma fácil, denominado por FURBOT (VAHLDIECK; MATTOS, 2009).

### 2.1 STARLOGO TNG

O StarLogo TNG é um ambiente de programação que possibilita a criação de jogos 3D e simulações para o entendimento de sistemas complexos. Foi projetada pela Massachusetts Institute of Technology (MIT) Scheller Teacher Education Program (STEP) para educação com o intuito de aumentar a facilidade de uso, através de uma linguagem gráfica utilizando blocos, unindo-os como se fosse um quebra-cabeça. Em sua interface existe uma área para a visualização dos objetos gráficos que estão sendo utilizados usando gráficos OpenGL.

Na versão 1.5 do StarLogo TNG é possível utilizar recursos como terreno texturizado e permite também desenhar no terreno ou até mesmo importar imagens de arquivos, importar modelos do Google Earth para seus projetos para uso como personagens ou cenário, melhoria no rastreamento da câmera por cima do ombro para uma melhor experiência de jogo, modelo de execução paralela remodelado para melhor desempenho de simulação e resultados mais precisos (STEP, 2018). A Figura 1 demonstra a interface de programação do StarLogo TNG.

Figura 1 - Interface do StarLogo TNG



O StarLogo TNG executa em qualquer sistema operacional porque ele roda sobre a plataforma Java Virtual Machine (JVM). O StarLogo TNG somente pode ser executado fazendo o download do aplicativo e instalando em algum computador pois não há uma versão web (STEP, 2018).

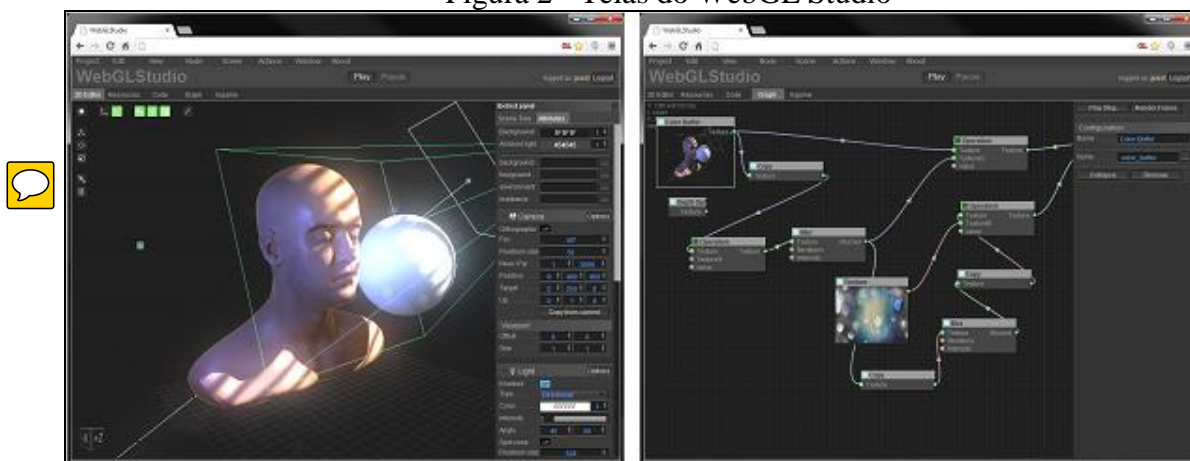
## 2.2 WEBGL STUDIO

O WebGL Studio é uma plataforma que permite a criar e editar cenas e materiais, efeitos de design e *shaders*, o funcionamento do código, utilizando tecnologias web a partir browser (AGENJO, 2016).

De acordo com Agenjo (2016) a plataforma contém varias bibliotecas gráficas que podem ser utilizadas para aumentar suas possibilidades. Um exemplo de biblioteca é a Litescene, onde seu funcionamento é dado a partir de um sistema de nós hierárquicos baseados em componentes. Ela vem com um pipeline de renderização realista e vários componentes que auxiliam e facilitam a criação e o comportamento de cenas, um sistema que calcula automaticamente o melhor sombreador, um gerenciador de recursos para carregar qualquer tipo de recurso. O Litescene também possui métodos de serialização para converter qualquer cena em JSON.

A Figura 2 demonstra duas imagens da plataforma WebGL Studio. A imagem do lado esquerdo mostra a interface da aplicação, contendo uma sub janela para a manipulação dos objetos gráficos, um menu superior com todas as opções que a interface disponibiliza e ao lado as configurações mais usadas. A imagem da direita apresenta uma forma de criar cenas em 3D sem o uso de código, apenas fazendo uma ligação através de nós de cena.

Figura 2 - Telas do WebGL Studio



Conforme afirma Agenjo (2016), a aplicação disponibiliza um editor para que torne-se possível codificar o comportamento de cada item. Assim como, utilizar nós de cena para especificar as características de cada item, apenas ligando os nós uns aos outros de forma adequada e os projetos ou *assets* desenvolvidos no sistema podem ser salvos em um servidor.

### 2.3 FURBOT

O *framework* denominado de Furbot foi desenvolvido devido a necessidade de um ambiente de aprendizagem que motivasse o aluno através da curiosidade e desafio. O FURBOT ensina introdução a programação de uma forma fácil, por meio de um jogo, onde o aluno deve passar por alguns obstáculos utilizando raciocínio lógico para chegar até a sua recompensa (VAHLIDIECK; MATTOS, 2009).

Os requisitos definidos para a criação do Furbot, de acordo com Vahldieck e Mattos (2009), foram: Um código simples com sequência lógica expressa, a codificação deve ser feita em qualquer ambiente de programação (IDE) Java, a criação das atividades é de responsabilidade do professor e deveria ser desenvolvido em JAVA.

De acordo com Vahldieck e Mattos (2009), o Furbot foi experimentado em quatro turmas da matéria de Programação de Computadores, primeiro semestre, do curso de Ciências da Computação da FURB. A metodologia e a base dos exercícios foram os mesmos da forma tradicional feitas nos semestres anteriores, mas com o **furbot** foi mais estimulante e desafiador. As turmas desenvolveram os jogos "Bomberman", "Come-come" e "Invasores do espaço". Com esse trabalho os alunos ganham experiência no desenvolvimento de um programa com maior complexidade, que necessita de várias classes. O objetivo do semestre como matéria era envolver mais o aluno e permitir que ele experimente o máximo de programação, se sentindo motivado nas atividades. Acredita-se que o objetivo foi alcançado ao longo do semestre, pois no semestre seguinte, onde os alunos aprovados estão na disciplina

de Programação Orientada a Objetos I, o professor titular dessa disciplina, que não esteve envolvido na disciplina do primeiro semestre, notou maior participação em debates dos alunos pois eles sentiam que tinham mais conhecimento sobre o assunto, quando comparados com aqueles dos semestres anteriores. O resultado esperado com a aplicação do Furbot é de ajudar o aluno a vencer as dificuldades iniciais de programação através de um maior envolvimento e assim diminuir a quantidade de desistências e o aumento de aprovações na disciplina.

### **3 SISTEMA ATUAL**

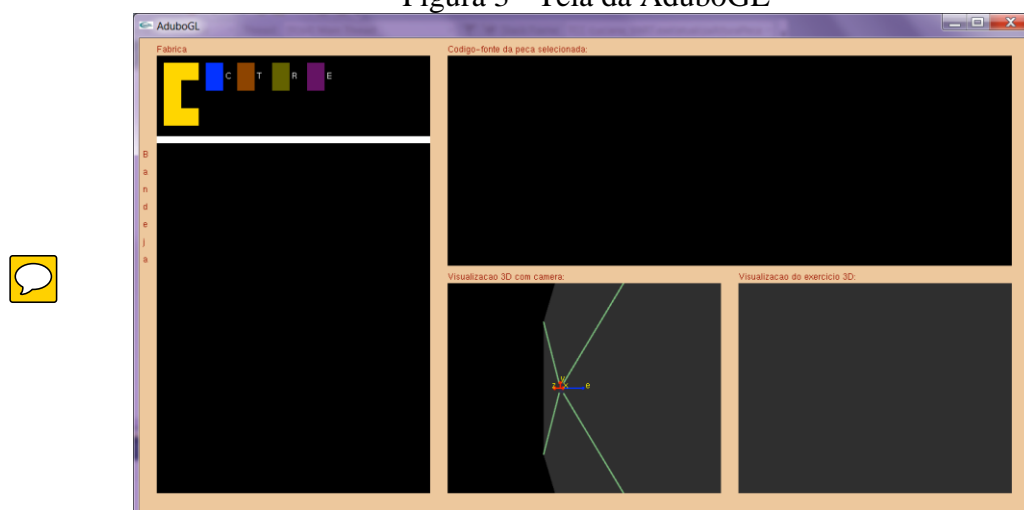
São apresentados cinco trabalhos onde foram dado continuidade até chegar ao sistema atual. A seção 3.1 descreve o AduboGL – Aplicação didática usando a biblioteca Open GL (ARAUJO, 2012), uma aplicação voltada ao aprendizado da computação gráfica com o foco nas transformações geométricas e o AduboGL ES 2.0 - Aplicação didática usando a biblioteca Open GL ES 2.0 no iOS (SCHRAMM, 2012), uma aplicação para iPad que auxilia no aprendizado de conceitos relacionados à computação gráfica. A seção 3.2 aborda o VisEdu-CG: Aplicação didática para visualizar material educacional, módulo de computação gráfica (MANTIBELER, 2014), apresenta também o VisEdu-CG 3.0: Aplicação didática para visualizar material educacional – Módulo de Computação Gráfica (NUNES, 2014) e por fim o VisEdu-CG 4.0: Visualizador de material educacional (KOEHLER, 2015), são aplicações utilizadas para auxiliar o aprendizado de estudantes de computação gráfica.

#### **3.1 ADUBOGL 1.0 E 2.0**

Conforme defendido por Araújo (2012), o AduboGL – Aplicação didática usando a biblioteca Open GL, é uma aplicação voltada ao aprendizado da computação gráfica com o foco nas transformações geométricas, utilizando a biblioteca OpenGL para montar o cenário 2D presente na aplicação e tendo seu resultado em um espaço 3D.

A Figura 3 apresenta a tela da AduboGL, ela se divide em 4 sub janelas. A janela da esquerda contem duas partes, a parte da fábrica que possui as peças a serem usada 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 3 - Tela da AduboGL



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.

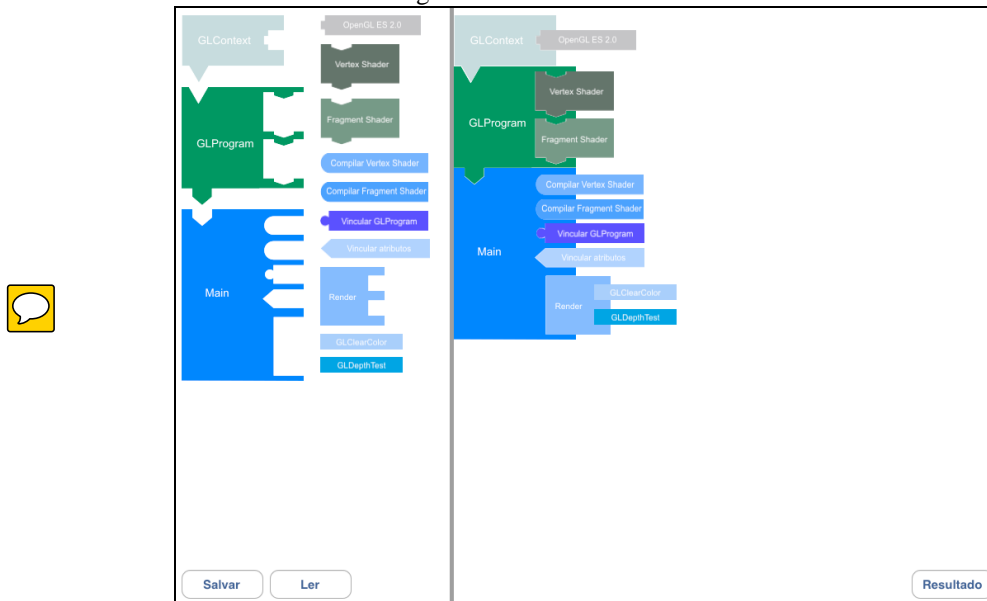
Segundo Schramm (2012), o AduboGL ES 2.0 é uma aplicação desenvolvida com intuito de auxiliar no aprendizado de conceitos relacionados a computação gráfica direcionado para o iPad. A aplicação tem como foco principal a biblioteca OpenGL ES na versão 2.0. A biblioteca OpenGL ES 2.0 é baseada em um módulo 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), os blocos *GLClearColor* e *GLDepthTest*, quando um deles 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 4 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 4 - Tela do AduboGL ES 2.0



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).

### 3.2 VISEDUCG, 3.0 E 4.0

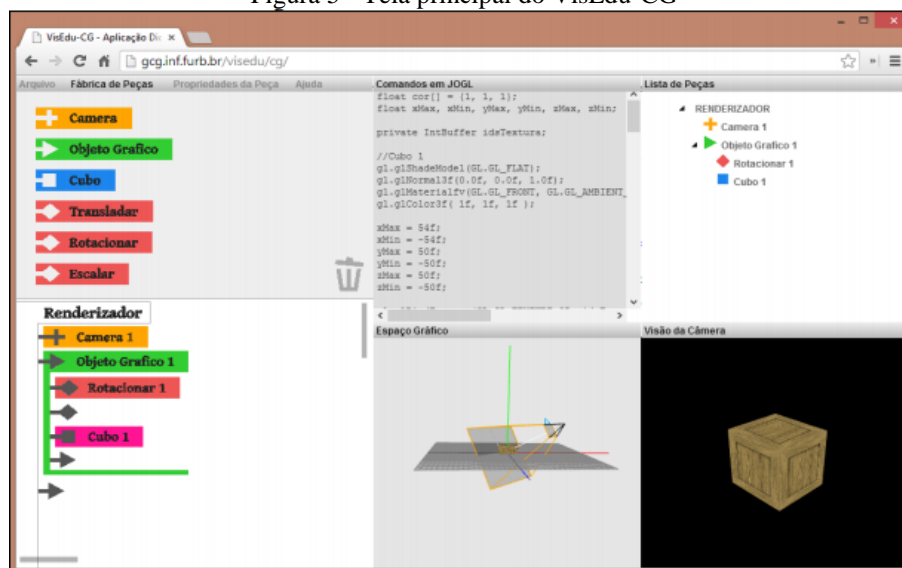
O VisEdu-CG é um projeto desenvolvido em uma plataforma web que possibilita aos alunos de computação gráfica, utilizarem os conceitos aprendidos na disciplina. De acordo com Montibeler (2014) é uma aplicação web direcionada ao aprendizado de computação gráfica, focada em conceitos de câmera sintética, grafo de cena, 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.

A seguir, na Figura 5 é apresentado a tela inicial do VisEdu-CG, que é dividida em seis sub janela. A fábrica de peças onde ficam as peças a serem utilizadas. O renderizador, onde são arrastadas as peças até o local determinado. Comando em JOGL, é a sub janela onde é



mostrado o código em Java ao selecionar uma das peças. Espaço gráfico e a Visão da Câmera são as telas onde é exibido o resultado gerado a partir do encaixe das peças no **Renderizador e o tela** de Lista de Peças, lista todas a peças encaixadas no painel de montagem.

Figura 5 - Tela principal do VisEdu-CG



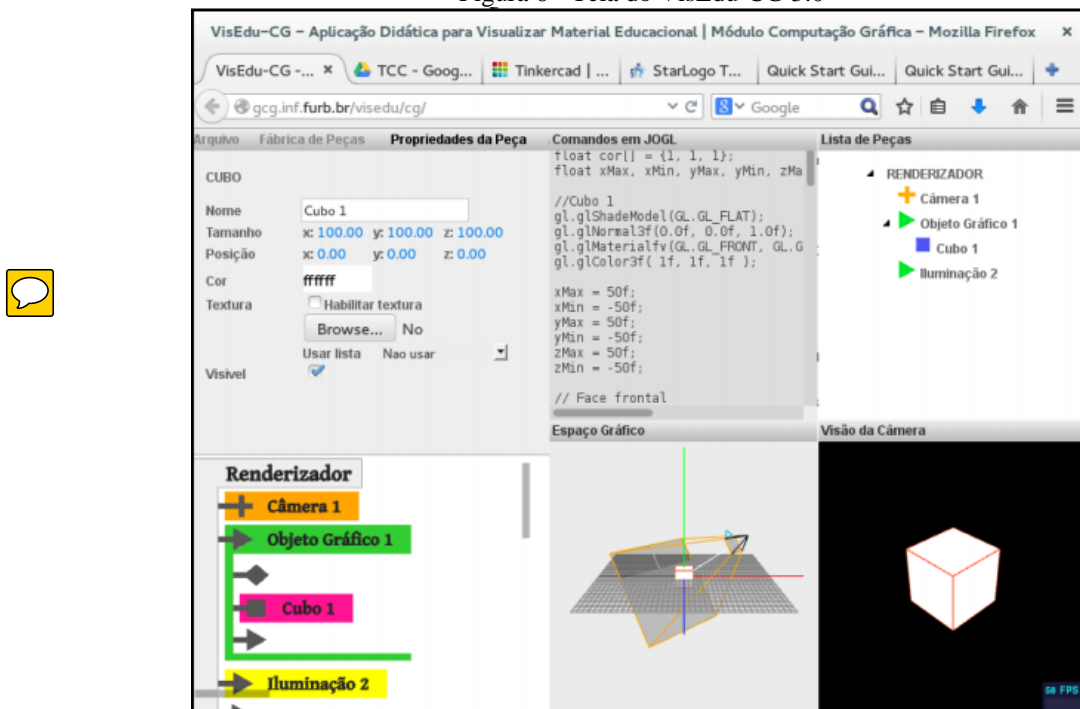
A linguagem utilizada é Java, com a biblioteca JOGL. O uso da biblioteca Three.js foi essencial para a visualização dos objetos no espaço gráfico 3D da aplicação. Essa biblioteca é voltada para renderização em WebGL (MONTIBELER, 2014).

Nunes (2014), que teve como objetivo estender o VisEdu-CG, incluiu novas funcionalidade à aplicação, deixando-a mais completa e aumentando ainda mais o entendimento dos conceitos de computação gráfica. Ele se propôs a incluir o tipo de objeto polígono e spline 3D, adicionar na representação gráfica o uso de iluminação.

A Figura 6 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.



Figura 6 - Tela do VisEdu-CG 3.0

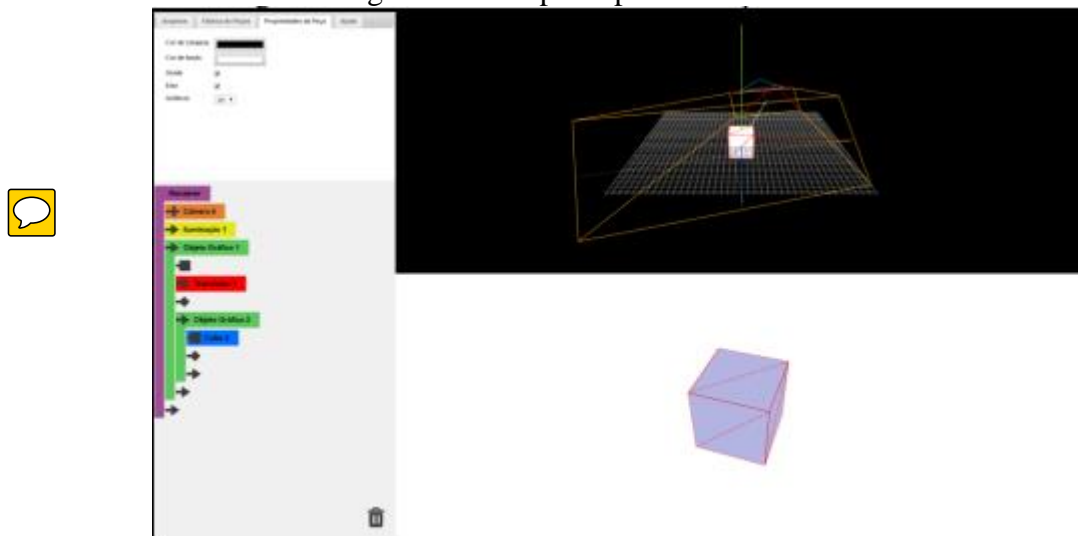


Além da inclusão de novos objetos gráficos e a iluminação, a aplicação simula uma visualização de um espaço 2D mesmo sendo implementada em 3D, através de um posicionamento fixo da câmera e desabilitando a manipulação dos objetos pelo *mouse* (NUNES, 2014).

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 versão 4.0 da plataforma VisEdu-CG passa a ter como foco na tela principal os painéis de Visão da câmera e Espaço gráfico. Na Figura 7 é possível ver a interface atual da aplicação.

Figura 7 - Tela principal do VisEdu-CG 4.0



Fonte:

Esses cinco projetos mostram a evolução do sistema VisEdu-CG até a sua versão atual, que por sua vez, são projetos que tendo cada um sua particularidade, têm participação na plataforma. Com o objetivo de, segundo Koehler (2015), melhorar o ensino e contribuir com um melhor entendimento de conceitos da computação gráfica.

## 4 PROPOSTA DA FERRAMENTA

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

### 4.1 JUSTIFICATIVA

### 4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação desenvolvida deve:

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

### 4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- levantamento **bibliográfico**;
- elicitação de requisitos: detalhar e reavaliar os requisitos e, se necessário, especificar outros a partir das necessidades observadas durante a revisão bibliográfica;
- especificação e análise: utilizar as funcionalidades da ferramenta Star UML, para elaborar diagramas de classe e de atividades da Unified Modeling Language (UML);
- 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.
- 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 1 – 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**:

### REFERÊNCIAS

BARROZO, Gracielle 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: 1< <http://revistas.unifenas.br/index.php/RE3C/article/view/12/10>>. Acesso em: 21 set. 2018.



<http://www.scielo.br/pdf/qn/v21n6/2913>

[http://gcg.inf.furb.br/?page\\_id=1147](http://gcg.inf.furb.br/?page_id=1147)

STEP. StarLogo TNG. [S.l.], 2013. Disponível em:  
< [https://education.mit.edu/portfolio\\_page/starlogo-tng](https://education.mit.edu/portfolio_page/starlogo-tng) > . Acesso em: 9 set. 2018.

<https://webglstudio.org/>

<https://github.com/jagenjo/litescene.js/tree/master>

<http://dsc.inf.furb.br/noticias/ler/projeto-da-furb-participa-em-concurso-latino-americano>

[https://www.researchgate.net/profile/Adilson\\_Vahldick/publication/280879314\\_Ensinando\\_Programacao\\_Atraves\\_de\\_Dispositivos\\_Moveis-Mobile-Furbot\\_e\\_iFurbot/links/55c9ebbb08aeb97567496416/Ensinando-Programacao-Atraves-de-Dispositivos-Moveis-Mobile-Furbot-e-iFurbot.pdf](https://www.researchgate.net/profile/Adilson_Vahldick/publication/280879314_Ensinando_Programacao_Atraves_de_Dispositivos_Moveis-Mobile-Furbot_e_iFurbot/links/55c9ebbb08aeb97567496416/Ensinando-Programacao-Atraves-de-Dispositivos-Moveis-Mobile-Furbot-e-iFurbot.pdf)

[https://www.researchgate.net/profile/Adilson\\_Vahldick/publication/273763127\\_Aprendendo\\_Programacao\\_de\\_Computadores\\_com\\_Experiencias\\_Ludicas/links/550b0a500cf265693cef6401/Aprendendo-Programacao-de-Computadores-com-Experiencias-Ludicas.pdf](https://www.researchgate.net/profile/Adilson_Vahldick/publication/273763127_Aprendendo_Programacao_de_Computadores_com_Experiencias_Ludicas/links/550b0a500cf265693cef6401/Aprendendo-Programacao-de-Computadores-com-Experiencias-Ludicas.pdf)

SCHRAMM, Elizandro. **AduboGL ES 2.0**: documentação. Blumenau, 2012. Disponível em:  
<[http://www.inf.furb.br/gcg/gcg/download/TCC2012-1-11-ElizandroSchramm\\_documentacao.zip](http://www.inf.furb.br/gcg/gcg/download/TCC2012-1-11-ElizandroSchramm_documentacao.zip)>. Acesso em: 03 jul. 2012.

ARAÚJO, Luciana. **AduboGL – Aplicação didática usando a biblioteca Open GL**: documentação. Blumenau, 2012. Disponível em:  
<[http://www.inf.furb.br/gcg/gcg/download/TCC2012-1-11-ElizandroSchramm\\_documentacao.zip](http://www.inf.furb.br/gcg/gcg/download/TCC2012-1-11-ElizandroSchramm_documentacao.zip)>. Acesso em: 03 jul. 2012.

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

[http://dsc.inf.furb.br/arquivos/tccs/monografias/2014\\_1\\_james-perkison-montibeler\\_monografia.pdf](http://dsc.inf.furb.br/arquivos/tccs/monografias/2014_1_james-perkison-montibeler_monografia.pdf)

[http://dsc.inf.furb.br/arquivos/tccs/monografias/2014\\_1\\_samuel-anderson-nunes\\_monografia.pdf](http://dsc.inf.furb.br/arquivos/tccs/monografias/2014_1_samuel-anderson-nunes_monografia.pdf)

[http://dsc.inf.furb.br/arquivos/tccs/monografias/2015\\_1\\_william-fernandes-koehler\\_monografia.pdf](http://dsc.inf.furb.br/arquivos/tccs/monografias/2015_1_william-fernandes-koehler_monografia.pdf)

**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 – PROFESSOR TCC I

Acadêmico(a): \_\_\_\_\_

Avaliador(a): \_\_\_\_\_

ASPECTOS AVALIADOS <sup>1</sup>		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 (atenção para a diferença de conteúdo entre projeto e pré-projeto) 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 (PREENCHER APENAS NO PROJETO):

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: \_\_\_\_\_

<sup>1</sup> 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 – PROFESSOR AVALIADOR

Acadêmico(a): \_\_\_\_\_

Avaliador(a): \_\_\_\_\_

ASPECTOS AVALIADOS <sup>1</sup>		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 (atenção para a diferença de conteúdo entre projeto e pré-projeto) 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: (PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

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

**PARECER:** (    ) APROVADO (    ) REPROVADO

Assinatura: \_\_\_\_\_ Data: \_\_\_\_\_

<sup>1</sup> 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.