

DESENVOLVIMENTO DE UM OBJETO VIRTUAL DE APRENDIZAGEM PARA O ENSINO DE QUÍMICA

Diogo Xavier de Noronha ¹; Gabriel da Silva ²; Vássia Carvalho Soares³;

1 Diogo Xavier de Noronha, Bolsista IFMG, Engenharia de Computação, IFMG Campus Bambuí, Bambuí - MG; dexisene@gmail.com

2 Gabriel da Silva: Pesquisador do IFMG, Campus Bambuí; gabriel.silva@ifmg.edu.br

3 Vássia Carvalho Soares: Pesquisador do IFMG, Campus Bambuí; vassia.soares@ifmg.edu.br

RESUMO

A dificuldade no aprendizado de química, assim como das demais ciências exatas no ensino médio, tem representado um obstáculo na formação dos alunos e se enquadra entre os principais motivos de evasão nos cursos superiores. Para tentar quebrar este paradigma, uma maneira que tem se mostrado viável é o uso de objetos de aprendizagem, físicos ou virtuais. Acredita-se que a aplicação, pedagogicamente adequada, dessas ferramentas possa ampliar a prática docente, promovendo mudanças significativas na ação do professor em sala de aula. Hoje existe uma série de objetos virtuais de aprendizagem com a finalidade de química, porém são bastante limitados no que diz respeito ao conteúdo abordado ou não são atrativos, uma vez que são complicados para se manusear ou não tem a capacidade de envolver os alunos pela falta de contextualização com o seu cotidiano. Este trabalho apresenta o desenvolvimento de um objeto virtual de aprendizagem para o ensino de química feito em Unity3D, em conjunto com os softwares de modelagem e animação Blender e MakeHuman. A modelagem do software foi feita utilizando UML, e a programação foi desenvolvida em C#, com o uso de MonoDevelop. Foi implementado um jogo de escape onde a personagem deve se mover pelos ambientes do jogo e solucionar quebra-cabeças relacionados ao conteúdo de química abordado, com a finalidade de estimular a aprendizagem do usuário. Como resultado este trabalho traz significativas contribuições para a sociedade uma vez que possibilita a inclusão digital e tem como base o despertar do interesse pela ciência, através da unificação de estudo e da diversão, utilizando dos jogos virtuais que são hoje tão presentes no cotidiano destes jovens. E com base nas ideias pedagógicas construtivistas construídas a partir das ideias do biólogo Jean Piaget, pode-se concluir que um jogo educativo, devidamente selecionado e monitorado por um professor apto, pode ser uma ótima forma de ensino e aprendizagem.

Palavras-chave: Lúdico no aprendizado; *Game engine*; Desenvolvimento de jogos.

INTRODUÇÃO:

Estudos feitos na década de 1990 já destacavam a dificuldade nas disciplinas iniciais dos cursos nas áreas de Ciências Exatas, e este é um fator considerável entre as causas de evasão no ensino superior que se estende até os dias atuais. Isso se dá principalmente pelo ensino deficitário das disciplinas de exatas nos níveis que antecedem o superior. Outro fator que deve ser mencionado é a forma tradicional com que essas disciplinas - matemática, física e química - são ministradas, deixando uma lacuna na aprendizagem dos alunos (GREGÓRIO et al., 2017).

Existe uma busca constante por novos métodos de ensino que tem como objetivo minimizar a dificuldade no aprendizado de Ciências Exatas, para eliminar a desigualdade com que os alunos ingressam no ensino superior, diminuir a evasão e democratizar a educação.

Com base na obra de Jean Piaget, biólogo e psicólogo que viveu entre 1896 e 1980 e dedicou sua vida a pesquisar sobre as formas de aquisição de conhecimento, criou-se a metodologia de ensino construtivista, onde se destacam uma menor interferência do professor na aprendizagem do aluno, ambientes diferentes de ensino não limitados somente à sala de aula, estímulo da participação dos alunos e um ensino menos rígido, mas que desenvolve nos alunos uma maior capacidade de autonomia e resolução de problemas (CÓRDULA, 2013).

Em entrevista não estruturada com uma professora de Química do Instituto Federal de Minas Gerais - *Campus Bambuí*, também se verificou essa dificuldade de aprendizagem em Química, representando uma demanda por métodos didáticos alternativos. Com base nestes termos, propôs-se criar um objeto virtual de aprendizagem com a finalidade de facilitar o aprendizado dos alunos.

O ensino de Química, assim como das demais ciências exatas, representa um obstáculo para muitos alunos devido às dificuldades de compreensão e absorção do conteúdo. Defende-se que o conhecimento da Química deve permitir a compreensão das transformações químicas no mundo físico

(NUNES e ADORNI, 2010). “No entanto, nem sempre o professor está preparado para atuar de forma interdisciplinar, relacionando o conteúdo com a realidade dos alunos” (ROCHA e VASCONCELOS, 2016).

Sabendo-se que o tema de ensino, objeto de estudo para este trabalho, é a química lecionada para o ensino médio e reconhecendo-se as vantagens do uso de objetos de aprendizado no ensino, tais como capacidade de reutilização, socialização e especialização, foi proposta a elaboração e o desenvolvimento de um objeto virtual de aprendizagem na forma de um jogo educativo de química.

Existem hoje vários objetos digitais de aprendizagem com foco no ensino e aprendizado de química, mas, em sua maioria, são apenas simuladores ou animações que pouco instigam o interesse por aprendizado nos alunos. Muitos pesquisadores, tais como Cunha (2012) e Soares (2004), constataram, de forma quantitativa, e comprovaram a eficácia do uso de jogos para auxiliar no ensino de química.

Como parte do resultado de sua pesquisa sobre os objetos de aprendizagem em estudo de química, Machado (2015) diz que existe uma enorme variedade de simuladores que abordam somente três áreas de química, sendo elas: ligações químicas, tabelas periódicas e concentração e dissolução, tornando-se um fator limitante no ensino da disciplina por meio de mídias, já que os outros conteúdos não são contemplados pela tecnologia.

Os aspectos pedagógicos mais importantes, quando se diz respeito a facilitar o trabalho de alunos e professores no processo de aquisição de conhecimento por meio de objetos virtuais de aprendizagem, são: a interatividade do estudante com o objeto de aprendizagem; a autonomia do aprendizado, tornando o processo de aprender independente de professores; a cooperação, no sentido de permitir aos aprendizes que façam trocas de informação durante o processo de aprendizagem; a cognição, ressaltando a capacidade do objeto de transmitir um conhecimento real e significativo ao aprendiz; a afetividade, que fica responsável pela motivação do aprendiz em estar ultrapassando novos limites e engrandecendo o seu nível de conhecimento (GALAFASSI et al., 2013).

Além destes aspectos, porém, devem também ser consideradas características técnicas de um objeto de aprendizagem, tais como: métodos de acesso; capacidade de agrupamento com outros conteúdos; autonomia do objeto de aprendizagem; classificação a fim de facilitar catalogação; durabilidade; interusabilidade ou compatibilidade com múltiplas plataformas; reusabilidade (GALAFASSI et al., 2013).

Longmire, citado por Grossi e Silva (2015), destaca como características técnicas essenciais para um objeto de aprendizagem: a flexibilidade, a facilidade para utilização, a possibilidade de customização, a compatibilidade com múltiplas plataformas, o aumento do conhecimento e a padronização para facilitar a indexação e a procura.

Savi e Ulbricht (2008) listam alguns benefícios que os jogos educacionais podem representar para o processo de ensino e aprendizagem. Em primeiro lugar, deve-se considerar o efeito motivador que os jogos têm, seja pela experiência visual ou pelos desafios oferecidos. O jogo pode ser visto como facilitador de aprendizado devido a sua capacidade de tornar visuais alguns conceitos, como moléculas ou células, e aquisição de conhecimento por meio de tentativa e erro. O jogo permite o desenvolvimento de habilidades cognitivas, uma vez que pode estimular a capacidade de resolver problemas e tomar decisões. O jogo possibilita o aprendizado por descoberta, favorece a socialização, o desenvolvimento da coordenação motora e até mesmo a especialização em um determinado conteúdo, pois um jogo pode ser jogado por várias vezes, com o objetivo de aperfeiçoar as capacidades do jogador.

METODOLOGIA:

Para a escolha das ferramentas utilizadas, foram observadas características como: valor de aquisição, compatibilidade, facilidade de uso e riqueza de recursos. As principais ferramentas selecionadas e aqui listadas são as seguintes:

UML

UML (*Unified Modeling Language*, ou Linguagem de Modelagem Unificada) é definida por Guedes (2004) como “uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos”. O mesmo autor afirma que, por menor que seja, todo software deve ser modelado, pois isso possibilita eventuais mudanças que ele possa vir a sofrer.

Para Booch et al. (2006), a importância da modelagem se dá porque, através dela, podemos visualizar o sistema como é ou como se espera que ele seja, podemos prever a estrutura ou o comportamento de um recurso, podemos fazer dela um guia para a construção de um sistema e documentar as decisões tomadas. O autor considera a UML uma linguagem adequada, pois é destinada a estes quatro objetivos: visualizar, especificar, construir e documentar.

Unity3D e MonoDevelop

Antes de falar em Unity3D, é preciso entender o que é um motor de jogo. “Um motor de jogo é uma ferramenta que tem como objetivo simular a física do mundo real dentro de um ambiente de jogo” (BRITO, 2012). Ele existe para facilitar a criação de jogos, pois remove a demanda de todos os cálculos matemáticos que envolvem as simulações dos jogos.

Teotônio e Araújo (2017) afirmam que “antes de existirem os *game engines*, os jogos eram escritos por completo, sem a separação de gráfico e cálculos físicos, por exemplo. Os primeiros motores de jogos surgiram em 1985, após o crash do videogame, quando houve o aumento do número de computadores domésticos”.

Unity3D é um motor de jogo, ou game engine, que fornece funcionalidades como renderização de gráficos, detecção de colisão, suporte para criação de animações, efeitos sonoros e suporte a linguagens como JavaScript e C#. Neste trabalho, optou-se pelo uso de C# devido à facilidade oferecida pelas características da linguagem orientada a objetos e pelo material de estudos, que é mais completo e abrangente.

O Unity 3D tem uma interface intuitiva que integra as principais ferramentas que devem estar presentes em um motor de jogo e tem a capacidade para gerar jogos para as mais diversas plataformas, como: Iphone, Xbox, Nintendo wii, jogos web, android, além dos sistemas operacionais convencionais. Apesar de ser pago, ele conta com versão pessoal gratuita bastante adequada para protótipos ou pequenos projetos sem fins lucrativos (LEMES, 2009).

Geralmente, os scripts da programação dos jogos são escritos na ferramenta MonoDevelop, que também será o caso deste trabalho. Todos os objetos criados dentro do Unity3D são instâncias da classe GameObject, que pertencem à biblioteca UnityEngine. Essa classe possui, invariavelmente, dados como nome, posição, tamanho do objeto, camada em que se encontra, cena em que se encontra, se está ou não ativo na cena, entre outros. Também possui funções de comparação, se autodestruir, se autoinstanciar, procurar por outros objetos do mesmo tipo na cena, listar objetos filhos ou listar objetos pais. A biblioteca UnityEngine contém tudo o que é necessário para a integração do código C# editado em MonoDevelop e interface Unity.

Blender

Blender diz respeito a um software gratuito e de código aberto que pode ser executado em Linux, Windows e Mac OS X. Trata-se de uma ferramenta completa para desenvolvimento de jogos 3D, que permite modelagem, texturização, animação e que conta com um motor de jogo integrado. Para a realização deste trabalho, o Blender foi utilizado somente para modelagem, texturização e animação (BRITO, 2007).

O Blender permite modelar e animar, entre outros, o formato FBX, que é perfeitamente compatível com Unity3D. Permite até mesmo modificá-lo depois de já importado no motor de jogo. Visto que o Unity3D não fornece o mesmo nível de edição de objetos, essa compatibilidade entre as ferramentas tornou o Blender a melhor opção para o trabalho.

Gimp

Para o desenvolvimento de um jogo convencional, é necessário um conjunto de imagens e texturas para a compreensão, entretenimento e, como se espera, a imersão do jogador. Existe uma grande quantidade de ferramentas de edição de imagem no mercado, mas, por motivo de acessibilidade e custos, para este trabalho, foi escolhido o GIMP. Este software usa a biblioteca GTK, tem código aberto e, portanto, conta com centenas de desenvolvedores e tem licença pública, o que representa uma redução de custos significativa para o desenvolvedor de jogos (MILANI, 2008).

Tenaglia et al. (2011) contam que o GIMP foi criado em 1995, na University of California, Berkeley, como projeto de faculdade de Spencer Kimball e Peter Mattis. Era um programa de edição de imagens para sistemas Unix que, na época, se fazia necessário. Hoje, o GIMP conta com uma extensa equipe de desenvolvimento e pode ser executado em outros sistemas operacionais, como Windows e Mac OS.

Elaboração da aplicação segundo os conceitos de desenvolvimento de jogos

O processo de desenvolvimento teve início com uma pesquisa pelo conteúdo ensinado nas escolas e cobrado no Exame Nacional de Ensino Médio e vestibulares e além disso foi coletada a opinião de um profissional da área de química. Após um levantamento geral, os temas foram filtrados e selecionados para um resultado de mais simples adaptação no desenvolvimento do objeto de aprendizagem.

Atualmente, existem diversos processos de software, e algumas atividades são comuns entre eles, como, por exemplo: determinação das especificações, produção de um software que as atenda, validação do software através de análise de riscos e evolução do software por meio de atualizações para atender às necessidades mutáveis do cliente (RIBEIRO, 2013). Para este trabalho, foi selecionado o modelo em espiral devido às suas características cíclicas, que facilitam estimativas mais realísticas com o progresso do trabalho, para que problemas importantes sejam descobertos mais cedo (MACÊDO, 2012).

Empregando-se a linguagem C# e a biblioteca *UnityEngine*, a programação foi bastante facilitada. Cada bloco de código, chamado de *Script*, é uma extensão da classe *MonoBehavior*, que conta com dois métodos principais chamados *Start* e *Update*. O método *Start* é executado assim que o objeto é instanciado em uma cena. Neste momento, atributos são declarados e inicializados, de modo que o objeto tenha suas características iniciais estabelecidas no momento de sua instanciação. O método *Update* é executado a cada frame da execução e tem como finalidade atualizar os atributos do objeto em função do tempo decorrido e em função de eventos que possam executar alterações sobre ele. O *Script* deve estar sempre anexado a um objeto de jogo que se encontra em cena ou a um objeto que será instanciado, o que significa ser inserido em cena. Quando a cena é executada, todos os objetos de jogo têm seus *Scripts* executados.

Além dos métodos *Start* e *Update*, destacam-se *Invoke*, usado para chamar um método em um determinado tempo; *FixedUpdate*, que é parecido com o *Update*, mas tem uma frequência de execução fixa em relação ao tempo, e não de acordo com o número de frames por segundo; *OnCollisionEnter*, *OnCollisionExit*, métodos usados no reconhecimento de colisão de objetos; *OnMouseDown*, *OnMouseDrag*, *OnMouseExit*, que são métodos utilizados para o reconhecimento do mouse. Por fim, métodos essenciais, como *Instantiate*, usado para instanciar um objeto de jogo, e *Destroy*, empregado para destruir um objeto e removê-lo de cena. Estes são apenas alguns dos métodos mais utilizados.

Como alguns dos atributos mais usados, também podem ser citados: *Enabled*, atributo booleano que indica se o objeto está habilitado ou não. Somente em caso positivo, o método *Update* é executado a cada *frame*. O objeto *transform*, uma instância da classe *Transform* que armazena posição, direção e escala do objeto, outros objetos em que o objeto está contido e aqueles que ele contém dentro de si, além de funções que permitem deslocar, girar e procurar outros objetos, entre várias outras. *Name*, o nome do objeto; e *Tag*, um rótulo que permite classificar um objeto e compará-lo com outros.

A construção do jogo se desenvolve basicamente por montar as cenas de jogo, ao colocar os objetos em cada cena, em posições adequadas, formando, assim, um cenário, e ao atribuir a estes objetos funções por meio dos *Scripts*. Depois de compiladas várias cenas, a primeira é instanciada e, através dos *Scripts* contidos nela, as seguintes também podem ser instanciadas, de acordo com as ações do jogador. Toda a sequência de cenas tem por objetivo promover o aprendizado de química para o usuário.

A personagem chamada Samara, em homenagem ao elemento Samário, de número atômico 62, representa uma cientista e foi modelada utilizando *MakeHuman*, um software cuja função é a modelagem de personagens humanoides em 3D. Após esta modelagem, foi usado o *Blender* para criar animações à personagem, as quais são executadas durante o jogo, quando a personagem se move.

RESULTADOS E DISCUSSÕES:

Samara recebeu um *Script* de controle de personagem que permite que ela, controlada pelo jogador, possa se movimentar, pular, abaixar, coletar objetos e usá-los. Espera-se, desta maneira, que o jogador encontre um meio de tirar Samara do laboratório no qual está presa.

O cenário idealizado foi montado no próprio *Unity3D* e se refere a um laboratório de química integrado, com almoxarifado, escritório, banheiro e cozinha. Nos ambientes, estão disponíveis vidrarias, balança de precisão, estufa, agitador magnético, bico de Bunsen, condensador, centrífuga, capela de exaustão de gases, termômetro, condutivímetro, pHmetro, entre outros. O cenário é baseado em um laboratório real, buscando familiarizar o jogador com o ambiente. Além disso, traz nome e função de cada objeto em jogo, mesmo que alguns não sejam realmente utilizáveis. Os objetos do cenário também têm a

finalidade de ensinar um pouco sobre os equipamentos de um laboratório de química. Alguns podem ser coletados pela personagem, enquanto outros são utilizados apenas onde estão.

Quando os equipamentos encontrados no cenário são utilizados, uma nova cena é iniciada, onde um microjogo deve ser completado para que, com uma recompensa, se possa dar continuidade ao jogo como um todo. Estes microjogos, também chamados de quebra-cabeças, são testes para o conhecimento de química e têm como finalidade fazer o jogador aprender sobre a matéria enquanto joga. Desta forma, iniciam-se os microjogos - jogos menores, no estilo de quebra-cabeças, quando dentro do macrojogo – o jogo como um todo, onde são usados objetos encontrados no cenário. Estes podem ser objetos variados, como uma balança de precisão ou um armário trancado com uma senha.

Nas primeiras execuções do jogo, foi necessária, em média, uma hora para ele ser completado. Notou-se que o jogo estimula não somente o aprendizado em relação à química, mas também o raciocínio e a capacidade de interação com o meio digital.

CONCLUSÕES:

Tendo como premissa as ideias pedagógicas construtivistas, pode-se concluir que um jogo educativo com abrangente conteúdo, sendo este devidamente selecionado e monitorado por um professor apto, pode ser uma ótima forma de ensino e aprendizagem.

Para este trabalho, várias ferramentas foram utilizadas, sendo comprovada a eficácia de cada uma, desde o formalismo dos padrões de projetos utilizando UML, a modelagem facilitada utilizando *MakeHuman*, a animação facilitada utilizando Blender e arquivos BVH até a programação utilizando Unity3D e a linguagem C#.

Considera-se que os objetivos foram alcançados, com a elaboração de um objeto virtual de aprendizagem e sua implementação na forma de um jogo educativo, compatível com Windows, utilizando ferramentas gratuitas que permitiram um orçamento de valor nulo.

Acredita-se que a principal contribuição do presente trabalho para a sociedade está relacionada à possível inclusão digital e ao possível despertar do gosto pela ciência nos jovens, uma vez que o jogo permite que eles estudem e se divirtam de maneira simultânea.

Para trabalhos futuros, poderão ser implementados outros jogos para outras áreas de ensino, utilizando o mesmo método ou o aperfeiçoando. Espera-se que, através da reutilização dos códigos desenvolvidos, possa-se reduzir o tempo de implementação de novos softwares significativamente.

REFERÊNCIAS BIBLIOGRÁFICAS:

BOOCH, G.; RUMBAUGHT, J. & Jacobson, I. UML: guia do usuário Elsevier Brasil, 2006.

BRITO, A. Blender 3D. Novatec Editora, 2017.

CÓRDULA, E. B. de L. Construtivismo e o ensino-aprendizagem na sala de aula; Educação Publica. Disponível em <<http://www.educacaopublica.rj.gov.br/biblioteca/>

CUNHA, M. B. da Jogos no ensino de química: considerações teóricas para sua utilização em sala de aula. Química Nova na Escola, São Paulo, [s. L.], 2012.

GALAFASSI, F. P.; GLUZ, J. C. & GALAFASSI, C. Análise crítica das pesquisas recentes sobre as tecnologias de objetos de aprendizagem e ambientes virtuais de aprendizagem Revista Brasileira de Informática na Educação, 2013.

GREGÓRIO, José Ribeiro et al. O Programa de Apoio à Graduação em Química (PAG-Química) e sua contribuição para a democratização e permanência dos estudantes no ensino superior. Revista Electrónica de Enseñanza de las Ciencias. Pontevedra, Espanha. Vol. 16, no. 3 (2017), p. 540-558, 2017.

GROSSI, M. G. R. & SILVA, M. A. Diretrizes básicas para a construção de interface gráfica para objetos de aprendizagem. Educação & Tecnologia, 2015.

LEMES, D. de. O. Games independentes: fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais. Pontifícia Universidade Católica de São Paulo, 2009.

MACEDO, D. Modelos de Ciclo de Vida. 2012 Disponível em: <https://www.diegomacedo.com.br/modelos-de-ciclo-de-vida/> Acesso em: 02 de junho de 2018.

MACHADO, A. S. Uso de softwares educacionais, objetos de aprendizagem e simulações no ensino de química. Revista Química Nova na Escola, 2015.

MILANI, A. GIMP-Guia do usuário. 2ª Edição. Novatec Editora, 2008.

NUNES, A. S. ; ADORNI, D.S . O ensino de química nas escolas da rede pública de ensino fundamental e médio do município de Itapetinga-BA: O olhar dos alunos.. In: Encontro Dialógico Transdisciplinar - Enditrans, 2010, Vitória da Conquista, BA. - Educação e conhecimento científico, 2010

RIBEIRO, C. Artigo publicado na Edição Nº 101 da Revista .Net Magazine 18 de janeiro de 2013 Disponível em: <https://www.devmedia.com.br/modelos-de-desenvolvimento-de-software-revista-net-magazine-101/26747> acesso em 24 de jun 2018.

ROCHA, J. S.; VASCONCELOS, T.C. Dificuldades de aprendizagem no ensino de química: algumas reflexões. XVIII Encontro nacional de ensino de química. VIII ENEQ, Florianópolis, SC, v. 25, 2016.

SAVI, R. & ULBRICHT, V. R. Jogos digitais educacionais: benefícios e desafios. RENOUE, 2008.

SOARES, M. H. F. B. O lúdico em Química: jogos e atividades aplicados ao ensino de Química. Universidade Federal de São Carlos, 2004.

TENAGLIA, L. F. L. C.; Bucek, G. T. U.; Filgueiras, T. M. & Mello, L. A. Utilizando o GIMP para edição de imagens. Anais do Congresso Nacional Universidade, EAD e Software Livre, 2011.

TEOTÔNIO, W. & ARAÚJO, A. Desenvolvimento de Jogos para Dispositivos Móveis utilizando o Motor de Jogo Unity 3D: um Estudo de Caso. 2017.