**DREAM TRAVEL: Desenvolvimento de jogos com ferramentas**

**Open Source**

Danilo Silva Santos[[1]](#footnote-1)

João Vitor Alves[[2]](#footnote-2)

Lucas Fernando Yoshioka[[3]](#footnote-3)

Rafael Yamaguti[[4]](#footnote-4)

Antônio Rodrigues Carvalho5

**RESUMO**

O desenvolvimento de jogos digitais está atraindo cada vez mais interesse da sociedade brasileira, muitos buscando uma nova maneira de gerar renda, outros trabalhando para produzir seu *hobby* favorito. O principal objetivo desta pesquisa é demonstrar um jogo digital produzido utilizando *software freeware* e *open source*. A pesquisa atual é baseada em artigos, livros, sites e um questionário com empresas de jogos. O objetivo secundário é analisar a experiência e listar as vantagens e desvantagens do uso dessas ferramentas usadas durante o desenvolvimento do jogo. Um dos resultados mais importantes foi a produção do jogo com *software* livre e gratuito, com custo mínimo em seu desenvolvimento. Portanto, é possível desenvolver um jogo com custo mínimo e apresentar *software* livre que pode ser uma alternativa ao *software* pago.

**Palavras-chave:** Custo. Desenvolvimento. *Freeware*. Jogos Digitais. *Open Source*.

***ABSTRACT***

*The development of digital games is increasingly attracting more interest from Brazilian society, many seeking a new way to generate income, others from working to produce their favorite hobby. The main objective of this research is demonstrate a digital game produced using freeware and open source software. The current research is based on articles, books, websites and a questionnaire with game companies. The secondary objective is to analyze the experience and list the advantages and disadvantages of the use of these tools used during the development of the game. One of the most important results was the production of the game with free and free software with minimal cost in its development. Therefore, it is possible to develop a game with minimal cost, and present free software that may be alternatives to paid software.*

***Keywords****: Cost. Development. Digital games. Freeware. Open Source.*

# INTRODUÇÃO

Com a chegada da era digital das coisas e com a chegada do computador, as pessoas começaram a utilizá-lo procurando comunicação, informação, compras, pagamentos e entretenimento. Uma das fontes de entretenimento mais comum de quem usa computador são os jogos digitais, como uma forma de passatempo ou até mesmo de forma profissional como os jogadores que participam de campeonatos de *eSports* por todo o mundo. Consequentemente, o interesse pela parte de desenvolvimento de jogos também vem crescendo, vendo que alguns dos jogos mais jogados atualmente vem de um estúdio de jogos independente.

O desenvolvimento de jogos digitais em geral é um processo muito complexo pois engloba várias áreas como artes e *design* gráfico, programação, sonoplastia, roteirização etc.

*Softwares* como Photoshop, Illustrator, Unity, Unreal, Maya entre outros geralmente utilizados para o desenvolvimento de jogos, exige a compras das suas respectivas licenças que representam uma porcentagem de custo para a equipe durante a fase de desenvolvimento de um jogo. Alguns *softwares* como a *Unity* têm a sua versão gratuita, mas para utilizar esta versão, a empresa não pode ter ganho maior que US 100.000 $ em receita no ano anterior.

O presente trabalho pretende demonstrar que um jogo digital, pode ser desenvolvido usando ferramentas *open source* e ferramentas gratuitas visando o corte de custos durante a fase de desenvolvimento e no produto. Analisar as ferramentas alternativas para os programas pagos e como foi a experiência da transição entre um programa e outro, listando as dificuldades apresentadas e a curva de aprendizado.

É fato que um dos passos mais importantes na etapa de desenvolvimento de um jogo digital é avaliar e projetar os custos do projeto antes de ele ser iniciado, sendo os custos relativo ao número de funcionários que o projeto terá, o local onde a equipe irá se reunir para produzir o jogo, o custo com computadores com os requisitos mínimos para que os desenvolvedores consigam trabalhar sem enfrentar nenhum tipo de problema, a compra de licenças de programas essenciais para que o jogo seja desenvolvido, entre outros custos pormenores.

Examinando os *softwares* geralmente utilizados por equipes de desenvolvimento, criou a curiosidade de estudar e experimentar alternativas gratuitas para esses softwares, a fim de provar que é possível produzir um jogo usando ferramentas *open source* e alternativas gratuitas, e assim, determinar se a experiência acabou se tornando viável com relação quanto ao tempo que levou para que seja dominada essa nova ferramenta e ao custo final do projeto.

A questão que norteia o estudo está relacionada a reflexão sobre alternativas gratuitas aos softwares pagos geralmente usados por desenvolvedores de jogos digitais: É possível produzir um jogo com o mínimo de custo? Quais as alternativas para os programas pagos? Qual a curva de aprendizado desse software? Quais foram as vantagens e desvantagens das alternativas gratuitas em relação aos programas pagos?

Partindo do pressuposto que para o desenvolvimento de um jogo serão necessários softwares de gerenciamento, softwares de imagem, softwares de som e a game engine, cada um desses softwares com seus respectivas licenças e preços, que pode gerar uma porcentagem de custo maior para a equipe, para o caso desenvolvedores iniciantes investir dinheiro no projeto é arriscado, pois o jogo não pode dar lucro após o lançamento, e isso faz que pessoas que têm projetos de jogos não saiam dessa fase. Sim poderia, contudo muitos destes softwares não contam com tantos tutoriais ou manuais na internet ao contrário dos pagos como por exemplo o Adobe Photoshop© que se encontra todo tipo de tutorial, e também o próprio usuário que acaba ignorando a ideia de trocar algo que já está familiarizado para ele ao invés de experimentar um software gratuito semelhante a esse.

# FUNDAMENTAÇÃO TEÓRICA

Neste capítulo pretende-se abordar sobre os movimentos do Software Livre e do *open source* e os processos que ocorrem durante o desenvolvimento de um jogo oferecendo uma contextualização histórica dos tópicos já citados.

## Breve Histórico do movimento Software Livre e do Open Source

Segundo Sabino (2011, p.5) o texto escrito por Bill Gates considerando o compartilhamento de software uma pratica desleal e dessa maneira fazendo a indústria passar a vetar a cópia e a modificação do software gerando o que chamamos de Software de código fechado deu início ao movimento que era contra esse pensamento de Gates e que pretendia trazer de volta a liberdade do software para a cópia, modificação, melhorias e redistribuição.

Em 1983 um pesquisador chamado Richard Stallman enviou uma mensagem a uma comunidade do UNIX informando que ele estava disposto a desenvolver um novo sistema compatível com UNIX chamado GNU, de acordo com Silveira (2004, p.16) A proposta de Stallman era que esse sistema fosse livre para modificações e redistribuições, sem que ele tivesse licenças proprietárias de uso e que rodasse aplicativos desenvolvidos em UNIX. Então foi criado o conceito de Free software, software livre traduzindo para o português, e assim nasceu a Free Software Foundation e as liberdades para que um programa seja Software Livre. É considerado livre um software que contém as seguintes liberdades:

1. A liberdade de usar o *software* para qualquer finalidade;
2. A liberdade de estudar e modificar o programa conforme às suas necessidades, para isso, o acesso ao código-fonte se torna necessário.
3. A liberdade de redistribuir cópias de modo que você possa ajudar ao próximo
4. A liberdade de distribuição de cópias já modificadas de modo a poder aperfeiçoar e dar a conhecer a uma comunidade as melhorias já alteradas ao código fonte. Segundo a *Free Software Foundation*, o acesso ao código fonte é um dos pré-requisitos necessários para tornar credível esta liberdade.
	1. **Processo de desenvolvimento do jogo**

A produção de jogos no Brasil ainda não é forte comparada a outros países como Canadá, Estados Unidos e Japão, mas é nítida o crescimento da indústria nos anos, segundo o 2º censo da indústria brasileira de games realizada pela empresa Homo Ludens o número empresas de desenvolvimento de saltou de 142 para 375.

Sabendo dessa constatação inicia-se a produção de nosso jogo, segundo Clua e Bittencourt (2005, p.16), todos os jogos seguem etapas durante o processo de desenvolvimento, começando pela fase documental que são os seguintes elementos:

* + 1. **Roteiro**

Cada vez mais os jogos estão com roteiros tão bons quantos os dos filmes. Essa é uma parte fundamental para a produção do jogo, trazer uma narrativa que cause diversos sentimentos ao jogador fazendo com que ele queira mais e mais jogar para descobrir o desfecho da história.

* + 1. **Game Design**

Para a elaboração do GDD (*Game Design Document*) o artista conhecido como *game designer* vai agrupar todos os elementos tanto técnicos quanto artísticos nesse documento. Este documento vai servir de referência para as equipes envolvidas no processo de construção de um jogo. Segundo Rabelo (2005, p.36), o *game designer* deve entender o jogo não como uma reprodução da realidade, mas como uma simulação que excede os limites de uma simples imitação. O artista terá de usar sua criatividade para construir as características artísticas e psicológicas dos personagens de forma que o jogador tenha carisma com o personagem e acredite momentaneamente estar vivendo aquela aventura. Ele também define as mecânicas de jogo que serão utilizadas, passando pela criação das regras, objetivos, os riscos e as punições que o jogo terá de apresentar.

* 1. **Produção Gráfica**

A produção gráfica é o momento em que os artistas gráficos da equipe de desenvolvimento esboçam as primeiras ilustrações de personagens e cenário do jogo. Esse processo também é conhecido como arte de conceito (*concept art*), a partir dele é que se tem noção do que deve ser feito digitalmente e portado para o jogo. Em relação a isso Stoneham (2010, p.59) diz que após a conclusão e aprovação do *concept art,* a arte feita será usada como referência de inspiração para o grupo responsável por transformar o que está no papel para o computador.

* 1. **Motores de jogos**

Motores de jogos, em inglês *game engines* são usadas para simplificar o desenvolvimento de jogos, uma *engine* pode contar outras *engines* dentro dela como por exemplo: *engine* de som, *engine* de física e *engine* gráfica. Também conta com suporte a animação, sons, inteligência artificial e pode fazer a gerência de arquivos, de memória, linhas de códigos entre outras funções. A respeito de *engine* Clua e Bittencourt (2005, p.25) dizem que ela pode se encarregar de várias tarefas como renderização, entrada de dados, processamento de baixo nível, permitir que os desenvolvedores possam criar vários jogos usando uma *engine*, reaproveitamento de código e possibilita integração de maneira mais pratica entre o código e a modelagem 3D.

* 1. **Sonoplastia**

Em jogos o som é parte importante da experiência do jogador, ela aproxima o usuário e contribui para que a sensação de imersão no jogo seja completa, de acordo com Murray (2016, p.232) imersão está relacionada a abstração humana do fascinante mundo imaginário presente em diversas mídias, transformando em uma exclusiva sensação de prazer.

 Lemos (2009, p.112) ainda se referindo a sons em jogos diz que um conjunto de sons presente em um jogo também pode ser chamado de paisagem sonora, ela pode variar entre uma cena e outra sendo composta pelos sons *on-screen* quando a fonte sonora é visualizada na tela e sons *off-screen* quando não é perceptível a fonte sonora na tela.

Um jogo que consegue atingir o nível satisfatório em paisagem sonora contribui positivamente para a imersão do jogador, para o acontecimento disso Shum(2008, p.7) o artista de som terá que elaborar estratégias concepção e produção de paisagem sonora exigindo muito esforço do mesmo.

* 1. **Ferramentas para produção de Jogos.**

O presente tópico apresenta a história e o dados técnicos de softwares relacionados com o desenvolvimento de jogos.

* + 1. ***Adobe Photoshop***

O Adobe Photoshop é um *software* desenvolvido pela Adobe Systems caracterizado por ser um dos mais populares *softwares* de edição e criação de imagens do mundo, altamente usado pelos profissionais de design e fotografia. A sua primeira versão foi lançada em 1990, e a sua última versão estável lançada é a CC 2018.

O software é composto por uma grande quantidade de recursos, que são usados para a melhorar a qualidade das imagens, extração de detalhes, pinturas digitais, entre outras funções. O Photoshop com o conceito de camadas, onde é possível editar a imagem por partes.

* + 1. ***Krita***

O Krita é uma ferramenta para criação de ilustrações, arte conceitual, animações, histórias em quadrinhos, suportando vários modelos de cores e pintura HDR, está disponível para as plataformas Linux, Windows e macOS© sob a licença de software GNU GPLv3 de software livre e código aberto. É um dos softwares gratuitos e open source em alternativa a programas como o Adobe Photoshop© e Adobe Illustrator©. Basicamente, o Krita oferece comandos e funcionalidades semelhantes aos softwares citados anteriormente e acabou tornando uma opção viável já que o tempo para adaptar ao programa seria relativamente curto. (KRITA, 2018).

Como é um *free software* e *open source*, conta com uma grande quantidade de plugins disponíveis produzidos pela comunidade criada em torno do Krita, com isso conta também com suporte a diversos formatos de arquivos de imagem.

* + 1. ***Godot***

Godot é uma game *engine* de jogos 2D e 3D de código aberto e gratuito desenvolvido pela Godot *Engine*, os jogos produzidos usando essa *engine* podem ser exportados para computadores Microsoft Windows, Mac OS e Linux e para dispositivos moveis do seguintes Sistemas Operacionais Android e IOS, e *Web* (HTML5), a game *engine* ainda não consegue portar os para consoles. Godot possui uma boa comunidade de apoio, que soluciona problemas da plataforma, e esclarece as dúvidas de desenvolvedores iniciantes desta *engine*. O site da documentação do Godot oferece mais informações sobre o motor de jogo. Atualmente está na versão 3.0.6, lançada em julho de 2018. (GODOT, 2018).

A linguagem utilizada para o desenvolvimento de jogos dentro do Godot é o GDscript, uma linguagem de alto nível, semelhante ao Python, o objetivo proposto pelos criadores é que a linguagem seja otimizada e firmemente integrada ao motor de jogo, deixando ser mais flexível para a criação e integração do conteúdo. A partir da versão 3.0 do Godot foi inserida o *Visual Scripting* que visa atrair iniciantes no desenvolvimento de jogos que ainda não têm experiência com linguagens de programação. E também a partir dessa versão o Godot começou a adotar o C# para ser utilizado para a programação, mas para isso deve ser instalado um SDK e é recomendável instalar um editor externo ou uma IDE.

A engine usa o conceito de *scenes* e *nodes*, cada node tem uma funcionalidade definida por um *script*, esse *script* é usado para controlar como o *node* funciona e como ele interage com outros *nodes*.

* + 1. ***Unity***

Unity é uma *engine* de jogos 2D e 3D de código fechado criado pela Unity Technologies, os jogos criados por esse motor gráfico conseguem ser exportados para dispositivos movéis como Android e IOS, para computadores Microsoft Windows, Mac OS, Linux, e para consoles Playstation 3, Playstation 4, Xbox 360, Xbox One, Nintendo 3DS e Nintendo Switch, além de jogos para Web. A *engine* conta com um recurso bem interessante que é a loja da Unity, onde-se pode realizar o *download* de texturas, modelos 3D, sons, entre outros elementos de forma paga ou gratuita. Jogos como *Cuphead* (2017) e *Life is Strange: Before the Storm* (2017) foram produzidos com o motor de jogo da *Unity*, cada vez mais o motor está sendo usado para desenvolver jogos, segundo Rodrigues; Lopes; Carvalho (2010, p.4) entre 1000 jogos feitos para dispositivos móveis, 34% são feitos na *Unity*.

Lemos (2009, p.125) aponta que a *Unity* mudou o paradigma de desenvolvimento de jogos, pois ela é versátil apesar de ser uma ferramenta paga, e o custo benefício dela é maior, se tornando uma vantagem para os produtores *indies.*

Segundo Rodrigues; Lopes; Carvalho (2017, p.3) as linguagens utilizadas dentro da *Unity* são o C# e o Javascript e a empresa oferece suporte a quem quer aprender a desenvolver jogos na *Unity*, ela disponibiliza em seu site uma documentação completa sobre o motor do jogo e apresentando vários tutoriais aos iniciantes em *Unity*.

# Procedimentos metodológicos

Essa seção aborda como foi implementada a pesquisa, explicitando como foi classificada e as técnicas usadas para coleta de dados que foram utilizadas para obtenção do objetivo do trabalho.

Para Marconi e Lakatos (2003, p.155) a pesquisa é entendida como “um procedimento formal, com método de pensamento reflexivo, que requer um tratamento científico e se constitui no caminho para conhecer a realidade ou descobrir verdade parciais”.

Quanto à natureza da pesquisa ela é classificada como aplicada. A pesquisa aplicada de acordo com Barros e Lehfeld (2000, p. 78) tem como motivação a necessidade de produzir conhecimento para aplicação de seus resultados, com o objetivo de “contribuir para fins práticas, visando à solução mais ou menos imediata do problema encontrado na realidade”.

Este trabalho foi feito a partir de uma pesquisa qualitativa, segundo Neves (1996, p.01) a pesquisa qualitativa não busca listar e mensurar eventos. Ela compreende um conjunto de diferentes técnicas interpretativas que visam descrever e decifrar os componentes de um sistema complexo de significados.

Quanto aos objetivos ela é classificada como exploratória segundo Gil (2008, p. 27) o objetivo desta pesquisa é proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou construir hipóteses. Ao final desse trabalho será analisada a experiência adquirida com esses programas gratuitos listando as vantagens e desvantagens deles e se com isso o custo final do produto será reduzido.

Quanto aos procedimentos técnicos utilizados nesta pesquisa serão utilizados a pesquisa bibliográfica. Segundo GIL (2008, p.50) “A principal vantagem da pesquisa bibliográfica reside no fato de permitir ao investigador a cobertura de uma gama de fenômenos muito mais ampla do que aquela que poderia pesquisar diretamente”. Para esta pesquisa o foco foram artigos e livros sobre o movimento *open source*, sobre desenvolvimentos de jogos digitais e os jogos *indies*.

Também será utilizada a pesquisa documental. Conforme GIL (2008) a pesquisa documental é semelhante a bibliográfica, contudo a fonte dos materiais da pesquisa documental vale de documentos que não receberam tratamento analítico. Aqui analisaremos relatórios de usuários do software *open source* de imagem Krita, apresentando a experiência que eles tiveram com o programa, quais as vantagens e desvantagens encontradas por eles.

O levantamento de Campo também irá fazer parte de nosso estudo. A esse respeito GIL declara:

As pesquisas deste tipo se caracterizam pela interrogação direta das pessoas cujo comportamento se deseja conhecer. Basicamente, procede-se à solicitação de informações a um grupo significativo de pessoas acerca do problema estudado para em seguida, mediante análise quantitativa, obter as conclusões correspondentes dos dados coletados. (2008, p.55).

Para o levantamento de campo será elaborado um questionário de forma de perguntas abertas que serão respondidas por empresas desenvolvedoras de jogos. Para Marconi e Lakatos (2003, p.201) “Questionário é um instrumento de coleta de dados, constituído por uma série ordenada de perguntas, que devem ser respondidas por escrito e sem a presença do entrevistador”.

# DESENVOLVIMENTO

O Krita é um software *open source*, ele foi utilizado para a criação dos cenários dos jogos desenvolvidos neste trabalho, personagens e objetos de colisão. Para a criação dos personagens e dos inimigos foi utilizada a técnica do *Pixel Art* que remete ao visual gráficos dos primeiros jogos desenvolvidos e a escolha da utilização desta arte é por trazer o sentimento de nostalgia dos memoráveis jogos das primeiras gerações, ela é um tipo de arte digital que através da manipulação por pixel permite a produção de uma matriz diversificada de imagens artísticas.

**Figura 1 - Interface do Krita**



Fonte: elaborado pelos autores (2018).

Com o Krita foram criados:

*Tilesets:*

**Figura 2 - Tilesets feito usando o Krita**



Fonte: elaborado pelos autores (2018).

Personagem Principal:

**Figura 3 - Pehtalla**



Fonte: elaborado pelos autores (2018).

Inimigos:

**Figura 4 - *Sprite Sheet Blue Spider***



Fonte: elaborado pelos autores (2018).

**Figura 5: *Sprite Sheet Canon Sentinel***



Fonte: elaborado pelos autores (2018).

A engine escolhida para produzir o jogo foi a Unity, que por mais que não seja um programa *open source*, ela é grátis para empresas com uma receita abaixo de $100.000 dólares e oferece suporte para desenvolver jogos para consoles.

# RESULTADOS E DISCUSSÃO

Durante a fase de desenvolvimento do jogo foram observados os softwares open *source* e *freewares* com a finalidade de descobrir quais fossem mais intuitivos ao uso com recursos semelhantes à dos softwares pagos e que o aprendizado nesta ferramenta se desse de forma ágil, para que não fosse perdido muito tempo durante a fase de adaptação ao novo programa.

O software Krita atendeu bem esses requisitos, mostrou ser uma ferramenta que pode ser uma boa alternativa ao Photoshop ou Illustrator, ou outros softwares de produção gráfica que necessitam de licenças.

# CONSIDERAÇÕES FINAIS

Após a finalização do projeto foi constatado a aplicação de ferramentas *open source* no desenvolvimento de jogos, a utilização destes *softwares* trouxe vantagens como a economia de gastos com licenças, fazendo com que o jogo fosse desenvolvido com o mínimo de custo. E as ferramentas utilizadas se mostraram grande alternativa aos softwares pagos, pois apresentaram recursos semelhantes, iguais ou até superiores a dos programas pagos mais utilizados em desenvolvimento de jogos. O uso de ferramentas *open source* podem ser uma alternativa para empresas de jogos com orçamento mais enxuto.

**REFERÊNCIAS**

BARROS, A. J. P.; LEHFELD, N. A. S. Fundamentos de metodologia: um guia para a iniciação científica. 2. Ed. São Paulo, 2000.

BESSA, Aline; SOUZA, Caio T.; BEZERRA, Carlos E.; MONTEIRO, Ivan; BANDEIRA, Humberto; SOUZA, Rodrigo. O Desenvolvimento de um Motor Multiplataforma para Jogos 3D. São Leopoldo,2007.

CLUA, E. W. G.; BITTENCOURT, J. R. Desenvolvimento de Jogos 3D: concepção, design e programação. São Leopoldo, 2005. p. 1313-1357. Disponível em: <http://www2.ic.uff.br>. Acesso em: 03 jun. 2018.

GIL, Antônio Carlos, Métodos e Técnicas de Pesquisa Social. São Paulo, 2008. Disponível em: <https://ayanrafael.files.wordpress.com/2011/08/gil-a-c-mc3a9todos-e-tc3a9cnicas-de-pesquisa-social.pdf>. Acesso em: 27 mai. 2018.

G. Disponível em: <http://docs.godotengine.org/en/3.0/> Acesso em: 18 set. 2018.

LAKATOS, Eva Maria; MARCONI, Maria de Andrade. Fundamentos de Metodologia Científica. 5.ed. São Paulo, 2003. Disponível em: <https://docente.ifrn.edu.br/olivianeta/disciplinas/copy\_of\_historia-i/historia-ii/china-e-india>. Acesso em: 27 mai. 2018.

NEVES, José Luiz. Pesquisa Qualitativa – Característica, Usos e Possibilidades. São Paulo, 1996. Disponível em: < http://ucbweb.castelobranco.br/webcaf/arquivos/15482/2195/artigo\_sobre\_pesquisa\_qualitativa.pdf>

RABELO, Cláudio, Game Design In: AZEVEDO (org.) Desenvolvimento de Jogos 3D e Aplicações em Realidade Virtual. Rio de Janeiro, Elsevier, 2005.

SABINO, Vanessa. Um Estudo Sistemático de licenças de software livre. São Paulo, 2011.

SHUM, Lawrence Rocha. A Construção de Paisagens Sonoras em Games a partir dos Modelos Estruturais de Samsel e Wimberley. Belo Horizonte, 2008. Disponível em <http://www.sbgames.org/papers/sbgames08/ad/papers/p03.pdf>.

SILVA, Felipe Rodrigues da; LOPES, Vinícius Luiz; CARVALHO, Marcos Alberto de. Desenvolvimento de Jogos na Plataforma Unity. Alfenas, 2016.

SILVEIRA, Sergio Amadeu. Software livre – A luta pela liberdade do conhecimento. São Paulo, 2004.

SILVEIRA, Sergio Amadeu. Software livre, cultura hacker e ecossistema da colaboração. São Paulo, 2009.

STALLMAN, Richard. About the GNU Project – Initial Annoucement. Disponível em: <http://www.gnu.org/gnu/initial-announcement.html> Acesso em: 21 jun 2018.

STALLMAN, Richard. Linux and the GNU Project. Disponível em: <http://www.gnu.org/gnu/linux-and-gnu.html> Acesso em: 21 jun 2018.

STALLMAN, Richard. Linux, GNU and Freedom. Disponível em: <http://www.gnu.org/philosophy/linux-gnu-freedom.html> Acesso em: 21 jun 2018.

STONEHAM, Bill. How to Create Fantasy Art for Video Games: A Complete Guide to Creating Concepts, Characters, and Worlds. Barron’s Educational Series. Londres, 2010.

**O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)**

1. - Aluno de graduação do Curso de Tecnologia em Jogos Digitais, na FATEC Carapicuíba – *e-mail*: danilo.santos61@fatec.sp.gov.br. [↑](#footnote-ref-1)
2. - Aluno de graduação do Curso de Tecnologia em Jogos Digitais, na FATEC Carapicuíba – *e-mail*: joaovitor\_kbo@hotmail.com. [↑](#footnote-ref-2)
3. - Aluno de graduação do Curso de Tecnologia em Jogos Digitais, na FATEC Carapicuíba – *e-mail*: lukeyoshioka@gmail.com. [↑](#footnote-ref-3)
4. - Aluno de graduação do Curso de Tecnologia em Jogos Digitais, na FATEC Carapicuíba – *e-mail*: rafael.tomyo.yamaguti@gmail.com.

5 - Professor Mestre do Curso de Tecnologia em Jogos Digitais, na FATEC Carapicuíba– *e-mail*: antonio.carvalho@fatec.sp.gov.br. [↑](#footnote-ref-4)