

# CENTRO PAULA SOUZA

---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Tecnologia em Jogos Digitais**

**Larissa Arruda Leite da Silva**

**USO DE SOFTWARE LIVRE E DE CÓDIGO ABERTO**  
**NO DESENVOLVIMENTO DE JOGOS**

**Americana, SP**  
**2014**

# CENTRO PAULA SOUZA

---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Tecnologia em Jogos Digitais**

**Larissa Arruda Leite da Silva**

## **USO DE SOFTWARE LIVRE E DE CÓDIGO ABERTO NO DESENVOLVIMENTO DE JOGOS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso de Tecnologia em Jogos Digitais, sob a orientação do Prof. Me. Gabriel de Souza Fedel

Área de concentração: Software Livre

**Americana, SP**  
**2014**

S581u	<p>Silva, Larissa Arruda Leite da Uso de <i>software</i> livre e de código aberto no desenvolvimento de jogos. / Larissa Arruda Leite da Silva. – Americana: 2014. 47f.</p> <p>Monografia (Graduação em Tecnologia em Jogos Digitais). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza. Orientador: Prof.Me. Gabriel de Souza Fedel</p> <p>1.Jogos digitais I. Fedel, Gabriel de Souza II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.</p> <p style="text-align: right;">CDU: 681.6</p>
-------	--

SILVA, Larissa Arruda Leite Da.

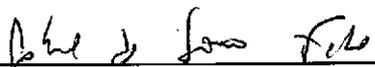
## USO DE SOFTWARE LIVRE NA EVOLUÇÃO DO DESENVOLVIMENTO DE JOGOS

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso de Tecnologia em Jogos Digitais, sob a orientação do Prof. Me. Gabriel de Souza Fedel

Área de concentração: Software Livre

Americana, 01 de dezembro de 2014.

### **Banca Examinadora:**

  
\_\_\_\_\_  
Gabriel de Souza Fedel (Presidente)  
Mestre  
FATEC Americana

  
\_\_\_\_\_  
Maria Cristina Aranda (Membro)  
Doutora  
FATEC Americana

  
\_\_\_\_\_  
Bruno Darros Lorençon (Membro)  
Graduado  
FATEC Americana

## **AGRADECIMENTOS**

Agradeço aos maiores amigos que tenho e aos maiores mestres que já conheci: meus irmãos e meus pais, por acreditarem na minha palavra, por serem meus confidentes em todas as horas e pela cautela dedicada a mim em cada passo que trilhei.

Agradeço também pela oportunidade honrosa que é fazer parte de uma comunidade científica extremamente talentosa como a FATEC de Americana e por ter sido recebida calorosamente durante o curso pelos professores mais dedicados que poderia ter desejado como meus guias e, especialmente, ao meu orientador por toda a ajuda concedida.

Também registro aqui toda a minha gratidão ao orientador, prof. me. Gabriel Fedel de Souza, uma vez que sem sua ajuda e todo o conhecimento que carrega consigo este trabalho não seria possível.

E, incluindo todas as razões já citadas, agradeço ao Senhor pela graça concedida, pelas forças enviadas nos momentos que não existem motivações e por trazer à tona nas horas conflitantes as verdadeiras razões pelas quais devemos acreditar nos sonhos que habitam nossos corações.

## DEDICATÓRIA

Às minhas amigas Taynara e Roberta e suas incríveis famílias, que me ajudaram  
imensamente na conclusão deste trabalho.

Ao meu pai.

## RESUMO

A construção de programas a partir de colaboração e código aberto pode parecer estranho para muitos programadores, mas é uma realidade que já trouxe projetos interessantes como o Linux, que é utilizado por grandes instituições como a NASA. No campo dos jogos digitais essa ideia ainda é um pouco distante, mas já existem iniciativas que apontam nessa direção. Esse trabalho faz um estudo das ferramentas livres e de código aberto disponíveis para a produção de jogos digitais e traz um levantamento de jogos e ferramentas de produção de conteúdo igualmente distribuídas sob licenças livres e de código aberto. Dentro do cenário de produção de software livre e aberto no mercado brasileiro, existe um projeto do governo federal que busca investir no talento nacional e trazer à tona soluções inovadoras tanto para softwares comerciais quanto para jogos e entretenimento e foi realizado um estudo de caso para compreender a situação atual desta área no Brasil.

**Palavras Chave:** software; livre; jogos.

## ABSTRACT

*Creating programs through people's collaboration and open source might sound odd to many programmers, but it's a reality that already brought us many interesting projects such as Linux, which is used by reputed institutions like NASA. On digital games programming this idea might sound distant, however there are purposes growing up towards it. This monograph studies the software, digital game, engine and game production stages concepts and a few multimedia creation tools and games distributed under free and open source licenses are related here. About Brazilian's free software and open source scenario, there's a project made by the federal government that looks for bring to the surface brand new solutions for commercial software and for games and entertainment and a survey were conducted in order to understand the recent situation of free and open source softwares in Brazil.*

**Keywords:** *free; software; games.*

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	12
2. Conceitos Básicos .....	14
2.1. <i>Software</i> .....	14
2.2. <i>Jogo Digital</i> .....	14
2.3. <i>Produção de um Jogo Digital</i> .....	15
2.4. <i>Engine</i> .....	16
3. Software Livre e de Código Aberto.....	17
3.1. <i>Contexto Histórico</i> .....	18
3.2. <i>Free Software Foundation</i> .....	19
3.3. <i>Open Source Initiative</i> .....	21
3.4. <i>Liberdade Tecnológica no Mundo Empresarial</i> .....	23
3.5. <i>Vantagens e Benefícios Comerciais</i> .....	25
3.6. <i>Desvantagens e Problemas Comerciais</i> .....	26
4. A liberdade nos Jogos Digitais .....	28
4.1. <i>Criação de Imagem Livres e Abertas</i> .....	28
4.2. <i>Produção de Áudio</i> .....	30
4.3. <i>Engines Livres</i> .....	31
4.4. <i>Títulos de Jogos Livres e de Código Aberto</i> .....	33
4.5. <i>Impulso governamental para Jogos Livres e Abertos no Brasil</i> .....	36
5. Estudo de Caso .....	38
5.1. <i>Metodologia de Pesquisa</i> .....	38
5.2. <i>Resultados Obtidos</i> .....	38
5.3. <i>Análise dos dados</i> .....	41
6. Conclusão.....	43
<b>7. REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	45
<b>8. APÊNDICE</b> .....	47

## LISTA DE FIGURAS E DE TABELAS

Figura 4.1 – Modelagem 3D no Blender .....	28
Figura 4.2 – Screenshot Gimp.....	29
Figura 4.3 – Screenshot Audacity.....	29
Figura 4.4 – Engine Panda 3D.....	31
Figura 4.5 – Mapa do Counter Strike do Rio de Janeiro.....	32
Figura 4.6 – Screenshot do jogo Tremulous.....	33
Figura 4.7 – Screenshot do jogo Assault Cube.....	34
Figura 5.1 – Usuários por plataforma.....	39
Figura 5.2 – Importância dos Softwares Livres e Abertos para Produção de Jogos Digitais.....	40
Figura 5.3 - Entrevistados que já produziram jogos de código aberto .....	41

## 1. INTRODUÇÃO

O setor de entretenimento voltado para jogos digitais não possui muitos incentivos governamentais, sejam estes através de atos significantes de valorização desta cultura ou incentivos fiscais (CAVICHIOILLI et. al., 2014). Poder contar com ferramentas de criação de conteúdos livres, ferramentas que não possuem restrições de uso e modificação, pode favorecer os desenvolvedores e programadores, que por sua vez são os responsáveis por aplicar as técnicas de desenvolvimento de em jogos, gerando estímulo para o crescimento desse setor em âmbito nacional.

Caso um desenvolvedor de software de mente mais curiosa pretenda não depender somente das escolas e dos (poucos) cursos de desenvolvimento de jogos, uma solução para esta situação pode estar nas ferramentas de criação e nos jogos distribuídos sob as licenças de software livre. Dentre outras características, os códigos-fontes destes softwares livres são disponibilizados para que o usuário faça o uso que desejar do produto. Dessa forma, os desenvolvedores são então livres para estudarem os códigos-fonte dos softwares declarados livres e compartilhar o conhecimento adquirido, aplicar novas técnicas estudadas em software de autoria de terceiros ou em seus próprios produtos e, de maneira geral, fomentar e expandir de forma benéfica os estudos dentro da programação de softwares.

Um bom exemplo da representatividade que o software livre pode trazer à programação de jogos é o Blender, uma ferramenta de animação 3D livre multiplataforma onde é possível modelar, animar e simular movimentos de personagens para a criação de jogos (BLENDER, 2013). Além das funções de modelagem, este software também conta com um ambiente de desenvolvimento de jogos, o que o torna um dos softwares mais completos do mercado e mais vantajoso por não taxar preços pela sua licença, diferentemente das taxas de licenciamento de seus concorrentes diretos – os ambientes 3DS Max e Maia.

Considerando estes conceitos, o presente trabalho busca compreender a aplicação do software livre dentro da indústria de jogos, através da análise de casos bem sucedidos de softwares, jogos, ferramentas e *frameworks* (do inglês, plataformas de trabalho) que abriram o código de seus produtos ou colocaram eles

sob licenças livres e por que as empresas e desenvolvedores adotaram essa filosofia.

A dissertação se organiza da seguinte forma:

O **Capítulo 2** inicia os estudos ao apresentar brevemente os conceitos básicos e essenciais sob os quais a dissertação se apoia: o que é um software, o que é um jogo digital, produção de um jogo digital e o que são *engines* de desenvolvimento para jogos.

No **Capítulo 3** são apresentados o contexto histórico e surgimento do software livre e de um movimento paralelo de código-fonte aberto chamado Open Source Initiative (OSI). Serão discutidos os benefícios de adotar a política de software livre no cenário atual da tecnologia e como a obsolescência pode ser combatida através das liberdades propostas pelo software livre. As propostas de corporações que usam Linux e softwares livres serão apresentadas, como a Google, IBM, RedHat e Netscape e o que motivou a escolha por estes sistemas operacionais e demais aplicações. Ao fim do capítulo são apontadas algumas das desvantagens comerciais e técnicas que o software livre pode apresentar em relação ao software proprietário e como isso pode vir a desfavorecer o produto.

Adiante, no **Capítulo 4**, serão apresentados jogos de licenças livres que se popularizaram, bem como ferramentas de criação e de desenvolvimento que ganharam espaço na indústria e o mais recente apoio do Ministério das Comunicações para o desenvolvimento de aplicativos em software livre, que busca expandir a inclusão digital pelo Brasil e premiar talentos nacionais.

No **Capítulo 6** encontra-se a análise de dados do estudo de caso realizado com profissionais da área de desenvolvimento sobre seus conhecimentos a respeito das ferramentas livres e de código abertos.

E por fim, no **Capítulo 7**, após os estudos das informações e situações aqui apresentadas, são divulgados os resultados esperados e os resultados inesperados encontrados durante o desenvolvimento desta dissertação, relatados na conclusão do estudo abordado.

## 2. CONCEITOS BÁSICOS

Antes de abordar o problema desta dissertação, estão aqui descritos alguns conceitos básicos e essenciais para a compreensão do estudo.

### 2.1. SOFTWARE

Para a ciência da computação, software é uma instrução da qual um computador é capaz de interpretar e processar. Um software possui uma sequência de instruções, chamadas de comandos, projetadas para manipular dados e informações para diversos fins (FERNANDES, 2002).

Um software é construído através de uma **linguagem de programação**. O objetivo de uma linguagem de programação é tornar compreensíveis para entendimento humano as instruções dadas a um computador. Uma instrução compreensível é chamada de **código-fonte**. O **compilador** é o software que traduz um código-fonte para o código compreensível pela máquina: o **código binário**. Um compilador também é responsável por verificar erros de sintaxe no decorrer dos códigos-fonte e aperfeiçoar um código binário para melhorar sua execução. O código binário é então referenciado em rotinas do computador, conectado em bibliotecas já presentes na máquina, e se torna um **código executável** (ALEXANDRE; GUTIERREZ, 2002).

### 2.2. JOGO DIGITAL

Um jogo digital é conceituado como uma tarefa realizada por microcomputadores, destinada primordialmente ao entretenimento do usuário. Tal tarefa tende a ser projetada para imersão de quem a executa através de enredos mesclados com mundos e situações fantasiosas. Como consequência, as respostas dos jogadores costumam apresentar emoções referentes às situações experimentadas e alguns destes jogos aproveitam-se da capacidade de imersão e aprendizado proporcionado durante o jogo e adquirem cunho educativo. Estes jogos possuem propostas de comportamento dedicadas ao entretenimento sugerido e para que isto não seja alterado, os produtores os mantêm como sistemas fechados (SILVA et. al., 2009).

### 2.3. PRODUÇÃO DE UM JOGO DIGITAL

A produção de um jogo, assim como a produção de um software, exige documentos e etapas a serem seguidas. Inicialmente, o projeto é iniciado através da documentação. Nesta documentação, encontrando o documento chamado de **bíblia do projeto** (*design bible*), em que estão descritas detalhadamente todas as características do jogo e funciona como um manual de instruções para os envolvidos no projeto. Na bíblia do projeto encontra-se o **documento do projeto** (*design document*), o **gameplay** com a jogabilidade (funcionalidade do jogo) e a **interface gráfica** (LIMA & BATISTA, 2009).

No documento do projeto encontra-se o roteiro, o conceito, o nome, a apresentação resumida, o público-alvo do jogo, o estilo, a história, as especificações técnicas e os dispositivos de entrada e saída. Este documento é essencial e requer total atenção, pois é a partir do conceitual descrito nele que os investidores vão decidir se o jogo é atrativo ou não. Já o *gameplay* descreve as regras, guiando o programador na etapa de desenvolvimento dos módulos ou fases do jogo. Na etapa de interface gráfica é descrita os modos de entrada de dados do jogador, configuração, instruções, modo de apresentação do jogo, entre outras funções diretamente relacionadas com a apresentação do produto ao jogador. (LIMA & BATISTA, 2009).

A produção dos jogos, pode ser dividida em três outras etapas: etapa de produção de áudio, produção de imagens e desenvolvimento através de uma *engine*. Na **produção de áudio**, existem músicos que se dedicam a criar sons que contextualizam adequadamente a história do jogo. Os áudios são grandes fatores para definir o grau de imersão, ou seja, interação do usuário com o jogo e para a criação destes arquivos de som, são usados softwares de criação de áudio como o SoundForge e Audacity (LIMA; BATISTA, 2009).

Na **produção de imagens**, existem ainda outras duas etapas: produção de imagens 2D e 3D. As imagens 2D (duas dimensões, baseadas em um plano cartesiano formado por coordenadas entre os eixos x e y, respectivamente eixos horizontal e vertical) costumam ser produzidas por softwares de manipulação de imagem, como o Adobe Photoshop ou o GIMP. As imagens 3D (três dimensões,

baseadas em um plano cartesiano de três eixos x, y e z, sendo respectivamente os eixos horizontal, vertical e de profundidade) são criadas por outros softwares que atuam exclusivamente na criação de imagens 3D, como o Blender e o 3D Studio Max. Esses softwares realizam também o que é chamado de **renderização**, processo computacional responsável por calcular os eixos de uma imagem 3D e apresentar a imagem na tela do computador (LIMA; BATISTA, 2009).

Após a conclusão de todas as etapas descritas acima, o jogo passa para a **etapa de desenvolvimento**. Para tanto, são utilizadas ferramentas de produção de jogos, conhecidas como *engines*, que auxiliam durante todo o processo de programação do jogo.

#### 2.4. ENGINE

No início da produção de jogos, os programadores contavam com poucos recursos acessíveis devido a seus altos preços de hardware das máquinas e dos softwares que utilizavam para a criação. A empresa id Software mudou essa realidade em 1993 quando lançou o jogo *Doom* e o kit de desenvolvimento correspondente, o id Tech 1, que trouxe formas de geração de gráficos 3D para microprocessadores e, como consequência, diminuiu os custos de produção de jogos para os desenvolvedores que, até então, só encontravam softwares caros e computadores pouco acessíveis. Com o id Tech era possível trabalhar com mais recursos em computadores menos potentes. Ao ambiente de desenvolvimento da id Tech é atribuído a evolução tecnológica dos jogos contemporâneos, e esses ambientes passaram a ser denominados de *games engines* (LEWIS; JACOBSON, 2002).

Para Lima e Batista (2009), “*game engine*, ou motor de jogo, pode ser definido como um conjunto de funcionalidades reutilizáveis que auxiliam o desenvolvedor a implementar as tarefas mais comuns de um jogo”. Estas ferramentas trazem muitas interfaces gráficas e diminuem a quantidade de código que o programador precisa escrever, aperfeiçoando o tempo necessário para criar etapas do jogo.

### 3. SOFTWARE LIVRE E DE CÓDIGO ABERTO

O modelo atual mais popular de produção de softwares é baseado na exclusividade dos direitos autorais da empresa que os distribuem. Os softwares pertencentes a esse modelo são chamados de softwares proprietários (ou não-livres). Isso significa que a redistribuição ou modificação de um software é feita exclusivamente pela produtora que detém os direitos exclusivos de todo o conteúdo do produto e em alguns casos, as produtoras buscam em métodos para ocultar o código. (GUESSER, 2005).

Por mais bem estruturado e planejado que um produto seja não seria correto descartar a hipótese de que, ao longo do uso de um determinado software, o usuário venha a notar outras necessidades para aquela aplicação. Em muitos programas e aplicativos, os usuários acabam sendo obrigados a trabalhar subordinados ao comportamento do software, ao invés de se favorecerem de um software que permite adequações de uso específico de um usuário diferenciado (GUESSER, 2005).

As questões a serem discutidas são: até que ponto as empresas estão de fato preocupadas em trazer o melhor para seus clientes e usuários? Como se desligar do monopólio de grandes corporações, que moldam como as pessoas devem usar seus produtos, e não como os aparelhos devem funcionar para as pessoas?

Buscando responder a essas perguntas, o **software livre** propõe um modelo de produção e distribuição de produtos nos quais o usuário tem total controle de seu desempenho, código-fonte, redistribuição e modificações (FSF, 1991). Não somente os usuários são favorecidos com o surgimento de novas versões e atualizações, mas o produto também pode ser beneficiado, tendo em vista que independentemente do sucesso ou fracasso do desenvolvedor que criou o produto, ele vai continuar acessível aos interessados. Já no software proprietário não há esta opção: quando uma empresa decide encerrar suas atividades, todos seus produtos se tornam obsoletos e descontinuados.

Para compreender a proposta de liberdade de um software, será analisado contexto histórico do surgimento da proposta, os projetos e fundações que foram

criados para apoiar e oficializar as licenças de uso livre e os benefícios e problemas destes softwares em relação ao software proprietário.

### 3.1. CONTEXTO HISTÓRICO

No início década de 1970, época da programação de cartões perfurados, todo código produzido era compartilhado. Todo código-fonte que existia passava por alterações, às vezes obrigatórias, para se adaptar melhor ao hardware do computador responsável por compilar os blocos de código. Softwares poderiam demonstrar comportamentos diferentes de máquina para máquina e alterações nos códigos-fonte poderiam - ou deveriam - ser aplicadas para melhor execução. No final da mesma década, surgiram os softwares portáteis, que se adequavam em diferentes plataformas e sistemas operacionais e, aproveitando da oportunidade, as empresas e produtoras de programas passaram a não distribuir mais esses códigos com a intenção de comercializá-los, atribuindo aos softwares valores equiparáveis aos valores dos hardwares dos computadores (CAMPOS FILHO, 2009).

Durante este período, onde os códigos-fonte eram distribuídos livremente, Richard Stallman, funcionário do laboratório de Inteligência Artificial (AI Lab) do Massachusetts Institute of Technology (MIT), era acostumado a consertar e acrescentar aos softwares das máquinas do laboratório pequenas necessidades que acabavam sendo descobertas no dia-a-dia, como um comando enviado pelo computador às impressoras que as faziam checar periodicamente se haviam ou não documentos na fila de impressão. Quando as empresas tornaram-se proprietárias dos softwares, Stallman, frustrado porque agora já não conseguia mais adequar os erros do software da impressora doada ao AI Lab pela Xerox Corporation, deixou o instituto em 1984. No ano seguinte da sua saída do MIT, o programador publicou o manifesto que deu origem ao Projeto GNU (WILLIAMS, 2009).

Stallman chamou a atenção de outros programadores que estiveram dispostos a ir contra o cenário comercial de desenvolvimento de software e contribuir com o seu projeto: um sistema operacional completamente livre, visto que naquele cenário não havia sistemas operacionais livres. A esse projeto baseado em Unix deu-se o nome de GNU - acrônimo recursivo de *GNU's Not Unix* (GNU Não é Unix, tradução nossa), já que, diferentemente do Unix, o GNU traria liberdade de

modificação aos usuários. O nome também faz referência ao animal mascote do projeto, que em inglês é chamado de *gnu*. Trata-se de um antílope selvagem e indomável das savanas africanas, característica de liberdade do animal que o aproxima do projeto de mesmo nome (GUESSER, 2005).

### 3.2. *FREE SOFTWARE FOUNDATION*

Após surgimento do Projeto GNU, foi criada, também por Stallman, a organização sem fins lucrativos *Free Software Foundation* (FSF), que amplia o conceito de liberdade do projeto GNU a outros produtos. No Brasil, chamada de Fundação do Software Livre, a FSF defende a liberdade de total escolha do usuário sobre um software que utiliza, conferindo controle completo em seu uso, modificação, distribuição e aperfeiçoamento para adequá-lo da maneira mais conveniente aos usuários (CAMPOS, 2006). Isso significa que a todo o momento os softwares estão evoluindo, se adaptando e sendo corrigidos pelos próprios usuários que, ao observarem uma necessidade, desenvolvem soluções e aplicam-nas nos produtos sem depender de permissões do autor responsável pelo programa. Ou então, caso não saibam programar, tem a liberdade de buscar um desenvolvedor ou empresa pra fazer a modificação desejada. Essa fundação promove, até os dias atuais, os projetos de toda a comunidade favorável ao seu manifesto.

A FSF conta com quatro liberdades essenciais. Um programa só é considerado software livre se possuiu as seguintes liberdades:

"A liberdade de executar o programa, para qualquer propósito (liberdade zero).

A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades (liberdade um). Para tanto, acesso ao código-fonte é um pré-requisito.

A liberdade de redistribuir cópias de modo que você possa ajudar ao próximo (liberdade dois).

A liberdade de distribuir cópias de suas versões modificadas a outros (liberdade três). Desta forma, você pode dar a toda comunidade a chance de beneficiar de suas mudanças. Para tanto, acesso ao código-fonte é um pré-requisito" (FSF, 1991).

Essas definições quanto ao uso do produto foram unidas e estão incluídas na licença *copyleft* General Public License, a GPL, que regulamenta formalmente o uso do software livre (AUGUSTO, 2003). A licença *copyleft*, ao contrário da licença *copyright*, autoriza a qualquer pessoa o direito de usar, aperfeiçoar e distribuir um software.

Diante das propostas da FSF sobre o que é software livre, é comum ocorrer confusões quanto à classificação do programa, especialmente no sentido da palavra livre. A interpretação da expressão *software livre* costuma confundir na conotação atribuída à palavra *livre*, que neste caso, define a liberdade de manipulação do software a favor do usuário e não *livre* quanto à gratuidade do software. O desenvolvedor deve ser livre para praticar as alterações cabíveis quando lhe for conveniente, mas isso não significa que, antes ou após as alterações feitas, o produto deva ser distribuído de graça (FSF, 1991).

O conceito de liberdade conforme a explicação própria dos organizadores do FSF está ligado unicamente à liberdade de expressão, ou seja, não é incorreto a prática de cobrança de taxas e preços pelo produto modificado. Cada desenvolvedor deverá avaliar qual a sua proposta inicial com o produto e caso comercializá-lo seja interessante e a licença sob a qual se encontra permita tal comércio, não existem empecilhos para que o faça. A FSF deixa claro em suas definições posteriores que esta é sim uma prática legal e apoia os desenvolvedores que assim fazem.

“Uma vez que software livre não é uma questão de preço, um preço baixo não é mais livre ou mais próximo de ser livre. Então, se você está redistribuindo cópias de software livre, pode também cobrar uma taxa e *ganhar dinheiro*. Redistribuir software livre é uma atividade boa e válida; se você a efetua, pode também lucrar com a mesma.

Software livre é um projeto comunitário e quem depende do software livre deve procurar maneiras de contribuir para a construção da comunidade. Para um distribuidor, o modo de fazer isso é dando uma parcela de seu lucro para a Fundação pelo Software Livre ou para algum outro projeto de desenvolvimento de software livre.

Financiando o desenvolvimento, você pode promover o mundo do software livre".<sup>1</sup>

Quem demonstrar interesse pode sentir-se livre quanto à cobrança de taxas pelo seu trabalho, vender o software que modificou e adaptou, entretanto, devem ser mantidas as mesmas licenças das quais o software tinha quando foi adquirido e não podem existir restrições quanto ao uso e modificações do mesmo produto (FSF, 1991). De forma resumida, as licenças GPL devem ser integralmente aplicadas ao produto, caso contrário o software passa a ser não-livre.

Outra implicação da licença GPL é o código-fonte do produto aberto. Sem o código aberto, não é possível modificação, então o software é classificado proprietário. Infringir as regras do software proprietário ao abrir o código, fazer alterações e até mesmo divulgar esse produto pode ser considerado crime de pirataria em alguns países (GUESSER, 2005).

Além do movimento FSF, existe paralelo a este outra organização chamada *Open Source Initiative* (OSI) que também defende a liberdade de escolha do usuário. Ao deparar-se com conceitos de software livre e código aberto é possível confundir-se a respeito de suas definições, uma vez que a prática do código-fonte aberto é aplicada em ambos os conceitos. Muitas pessoas partem do princípio de que os contextos soam similares, que talvez abordem os mesmos conceitos e por fim acabam inferindo que se tratem unicamente de uma mesma proposta.

Entretanto, são dois movimentos com propostas diferentes, e é de extrema importância para o desenvolvimento desta dissertação que as diferenças entre eles sejam pontuadas.

### 3.3. OPEN SOURCE INITIATIVE

Alguns anos após o início do movimento FSF, alguns de seus colaboradores decidiram abandonar por não concordarem integralmente com as propostas de Stallman. Alguns destes colaboradores relatam que o movimento de software livre tem uma filosofia restrita demais, e entre as restrições inclui-se que um software deixava de ser GPL caso estivesse incluído em seu código algum trecho de um

---

<sup>1</sup> Disponível em <<https://www.gnu.org/philosophy/selling.html>>. Acesso em 12 de setembro de 2014.

software não-GPL. Ainda segundo a OSI, a FSF se ocupa muito com questões filosóficas, como o quão livre um usuário é, e pouca atenção é de fato conferida ao software em si (OSI, 1998).

A OSI nasceu em 1998, através da iniciativa sem fins lucrativos do antigo colaborador da FSF Bruce Perens, com o propósito de “construir pontes entre os diferentes integrantes da comunidade código aberto” (OSI, 1998). Por se tratar de ideal derivado do software livre, o código aberto possui algumas similaridades com as liberdades propostas pela FSF. Os softwares submetidos a essa iniciativa podem ser usados, distribuídos, modificados e compartilhados livremente.

A OSI busca ser mais flexível do que a FSF. Foi criada, assim como na ideologia precursora, uma definição que por sua vez é composta por dez itens que descrevem os quesitos que um software deve atender para ser considerado código aberto:

"1. Ter distribuição livre. 2. Disponibilizar acesso ao código-fonte. 3. Permissões para criação de trabalhos derivados. 4. Integridade do autor do código-fonte. 5. Não discriminação contra pessoas ou grupos. 6. Não discriminação contra áreas de atuação. 7. Distribuição da licença. 8. Licença não específica a um produto. 9. Licença não restritiva a outros programas. 10. Licença neutra em relação à tecnologia" (OSI, 1998).

Os objetivos conceituais vêm de acordo nas duas vertentes libertárias. Na prática, porém, a principal diferença entre a OSI e a FSF é que para a OSI, os softwares de código aberto podem ser trabalhados em qualquer plataforma. O foco é o software e como o programador ou o usuário podem se beneficiar do produto, isto em qualquer sistema operacional livre ou proprietário, exatamente como a décima regra do acordo citado descreve: “Licença neutra em relação à tecnologia”. Já na FSF, o sistema operacional livre deve usar exclusivamente outros softwares livres, excluindo todas as barreiras que um software proprietário é capaz de impor na utilização de seus recursos. Segundo Elias e Mattos (2007), a OSI permite licenças de código aberto exclusivas para um determinado produto, cada uma delas feita para defender as interesses comerciais de seus investidores. Um exemplo é a

Mozilla Public License (MPL), que defende que um software deverá portar a mesma MPL caso seja alterado ou distribuído, mantendo-se um produto Mozilla.

Uma das licenças mais comuns entre os programas de código-fonte aberto é a licença elaborada pelo Instituto de Tecnologia de Massachusetts (MIT, da sigla em inglês para *Massachusetts Institute of Technology*). Dentre as licenças de código aberto, esta é a menor delas e sua única restrição obriga a redistribuição do produto com a mesma licença <sup>2</sup>.

Ambos os ideais são positivos e podem contribuir, cada qual à sua forma, para a evolução tecnológica da sociedade e assim, serão considerados ambas as vertentes durante a presente pesquisa e serão analisados softwares, ferramentas e jogos de ambas as licenças.

#### 3.4. LIBERDADE TECNOLÓGICA NO MUNDO EMPRESARIAL

Com a evolução da internet, os movimentos libertários de softwares livres se espalharam pelo mundo, fortalecendo o poder da comunidade da tecnologia da informação através do conhecimento ao alcance de todos. Na época da ascensão, um jovem universitário finlandês chamado Linus Torvalds desenvolveu o Linux, sistema operacional baseado em Unix, e publicou seu produto na internet sob a licença GPL do projeto GNU. Em pouco tempo, o projeto Linux criou raízes e passou a ser chamado de GNU/Linux. Não distantes da perspectiva de observação do crescimento de uma nova solução tecnológica, grandes empresas do ramo demonstraram interesse neste modo colaborativo de criação que, quanto mais colaboradores possuírem, maior será a qualidade de um software desenvolvido. (ELIAS; MATTOS, 2007).

Ao longo dos anos, muitas empresas e instituições globais se tornaram adeptas do voluntarismo global presente no software livre e código aberto. Uma dessas empresas é a IBM, detentora de inúmeras patentes, pioneira no mercado de produção de hardware e software, que além de utilizar certas distribuições Linux nos microcomputadores internos e em muitos de seus servidores, desenvolve um projeto para código aberto chamado Open Source and Standards, focado em desenvolver

---

<sup>2</sup> Disponível em <<http://opensource.org/licenses/MIT>>. Acesso em 3 novembro de 2014.

um planeta mais inteligente e aprimorar como bilhões de pessoas trabalham e vivem<sup>3</sup>. A empresa define que Linux é a melhor solução corporativa, uma vez que estão “empenhados fornecer soluções para ajudar a resolver negócios reais necessidades de liderança, a um custo de propriedade menor”<sup>4</sup>.

A década inicial da expansão do software livre, em meados de 1990, não só fortaleceu o desenvolvimento tecnológico de empresas já estruturadas no mercado competitivo como também influenciou a exploração das distribuições Linux, dando origem à criação de novas empresas, como a Red Hat, que trabalha para o desenvolvimento da distribuição Linux de mesmo nome. Outros centros de pesquisa como universidades e entidades não governamentais tem se dedicado a outros projetos, como as distribuições Debian, Slackware, e muitas outras (ELIAS; MATTOS, 2007).

Foi com a abertura do código-fonte em 1998 de seu navegador que a NetScape conseguiu se recuperar de uma provável falência. A Microsoft vinha ganhando cada vez mais espaço nos microcomputadores com o seu navegador próprio, o navegador Internet Explorer, e a companhia estava perdendo usuários. A NetScape, então, liberou o código-fonte do software para que mais pessoas pudessem contribuir e trazer inovações para o navegador, e conseqüentemente, deu-se origem ao projeto sem fins lucrativos chamado Projeto Mozilla (ELIAS; MATTOS, 2007). Ao tentar recuperar espaço nos microcomputadores, a empresa se recuperou de maneira inovadora e deu origem a um dos navegadores mais populares no mundo – o Mozilla Firefox – e que atualmente é disponível para várias plataformas de sistemas operacionais, desde computadores a dispositivos móveis.

Existem incontáveis outros projetos ao redor do mundo geridos com softwares livres e códigos abertos que buscam trazer melhoras para a sociedade com novas soluções que estejam ao alcance de todos. A empresa Google conta com projetos focados em esforços para a valorização do código aberto, como a *Google Summer of Code*, que busca recrutar globalmente estudantes dispostos a contribuir com a

---

<sup>3</sup> Disponível em <<http://www-03.ibm.com/linux/ossstds/>> , tradução nossa. Acesso em 27 outubro de 2014.

<sup>4</sup> Disponível em <<http://www-03.ibm.com/linux/>>, tradução nossa. Acesso em 27 outubro de 2014.

empresa<sup>5</sup>. Estes esforços incluem o desenvolvimento do sistema operacional para dispositivos móveis Android, o projeto Chromium que desenvolve o navegador Google Chrome e o sistema operacional Chrome OS, entre outros<sup>6</sup>. De outro lado, a Administração Nacional do Espaço Aéreo dos Estados Unidos, mais conhecida pela sigla em inglês NASA, disponibiliza os códigos de softwares desenvolvidos para simulação de satélites, análise de missões e explorações de voo e até uma ferramenta que analisa fissuras de superfícies<sup>7</sup>, tudo isso com a intenção de trazer o conhecimento a todos.

### 3.5. VANTAGENS E BENEFÍCIOS COMERCIAIS

Uma das grandes vantagens do software livre e código aberto é que a grande maioria dos produtos associados a essa liberdade podem ser adquiridos gratuitamente. Ainda que a prática de comercialização não seja banida pelos termos oficiais da FSF, grande parte das organizações que desenvolvem software livre não busca fins lucrativos com os produtos.

Outro ponto favorável da prática da liberdade de um software é o incentivo à criatividade. Para que mais descobertas e inovações sejam desenvolvidas, é necessário existir mais fontes de estudo e pesquisas. Quanto mais códigos-fonte puderem ser compartilhados e contribuírem para pesquisa, mais a ciência da computação poderá avançar, uma vez que para o avanço da computação o código-fonte se faz essencial para o estudo e reavaliações de métodos empregados (VERAS, 2006).

Quanto maior o número de colaboradores em uma pesquisa, maiores as chances de obter sucesso em um novo projeto, que ao contar com vários programadores envolvidos tem a chance de experimentar novos conceitos e ambientes de testes para aprimoramentos que um programador não seria capaz de realizar sozinho, minimizando os esforços individuais (AUGUSTO, 2003). Este padrão de contribuição de conhecimento para a evolução de um projeto foi

---

<sup>5</sup> Disponível em <<https://developers.google.com/open-source/soc/>>. Acesso em 21 de setembro de 2014.

<sup>6</sup> Disponível em <<https://developers.google.com/open-source/projects/>>. Acesso em 21 de setembro de 2014.

<sup>7</sup> Disponível em <<http://code.nasa.gov/>>. Acesso em 3 outubro de 2014.

classificado por Raymond (1998) como um “bazar”, estruturado pela liberação de novas versões a cada semana, que conta com a divulgação de *feedbacks* fornecidos, e dessa maneira, as versões aprimoradas se mantêm disponíveis na internet para os usuários, que podem escolher quais produtos lhes são mais interessantes ou apropriados.

Para as escolas, universidades e outras instituições de ensino, o software livre pode ser ainda mais vantajoso, pois além do custo muito menor se comparado a um software proprietário, é um ótimo aliado do aprendizado dos estudantes e por ser tão poderoso quanto o software proprietário, pode ser usado na estrutura de informática das instituições públicas, de servidores aos microcomputadores. Outro ponto é que as universidades são grandes pontos de pesquisas inovadoras e ao incentivar o uso de sistemas operacionais livres, induzem os estudantes a criarem novas soluções para este mercado (VERAS, 2006).

### 3.6. DESVANTAGENS E PROBLEMAS COMERCIAIS

Para os usuários não familiarizados com soluções de informática, trabalhar com software livre pode ser um problema, pois geralmente sua instalação é mais trabalhosa e muito pouco intuitiva se comparada à instalação dos softwares proprietários. Por valorizar a escolha de liberdade dos usuários, estes sistemas operacionais tendem a fornecerem mais opções de personalização de ambiente e para aqueles que não dominam certo conhecimento sobre o funcionamento de um computador, esta tarefa pode se tornar complicada (PETRUCELLI et. al., 2011).

Outros problemas dos softwares livres e de código aberto estão relacionados com a dificuldade de identificação do produtor de um código devido à grande quantidade de colaboradores voluntários que um projeto pode adquirir. Para Michlmayr; Hunt e Probert (2005) é impossível se apoiar completamente na presença dos atuais participantes e identificar alguém que não esteja contribuindo ou negando desempenhar devidamente as tarefas que lhes cabem. Dessa forma, a qualidade do projeto é afetada.

Os projetos mais atrativos de software livre e código aberto também enfrentam problemas como a dificuldade de identificação do criador de um código

devido à rotatividade de colaboradores de um projeto, o que interfere diretamente na qualidade de um produto. Já outros projetos são abandonados pela comunidade devido a um aspecto de seleção e competição, onde projetos mais inovadores atraem mais voluntários para si, enquanto outros acabam sendo deixados para trás. O fracasso de projetos de software livre e código aberto acaba sendo atribuído à falta de uma boa prática de gerência de projetos (MICHLMAYR; HUNT; PROBERT, 2005).

## 4. A LIBERDADE NOS JOGOS DIGITAIS

Ainda que o modelo livre e de código aberto de produção de jogos não seja tão popular quanto é para a produção de sistemas operacionais e outros softwares, observa-se através das ferramentas de produção de jogos que existe um mercado em crescimento, buscando fortalecer a indústria do entretenimento de jogos através dos princípios do voluntariado global.

A seguir, serão apresentados *engines* de produção de jogos e outras ferramentas que vieram para fortalecer a produção de jogos. Entretanto, serão apresentados produtos com licenças livres e código aberto, com o propósito de ressaltar o modelo colaborativo de criação de conteúdo e que se fazem úteis à criação de um novo jogo. Através da página da FSF pode-se encontrar uma listagem de alguns dos softwares livres mais populares. Este grupo que utiliza os softwares mais populares é chamado de *Free Software Gang*<sup>8</sup> e abrange diferentes categorias de softwares, como clientes de *e-mail*, linguagens de programação, ferramentas gráficas e sistemas operacionais. Dentre os produtos disponíveis, é possível encontrar vários dos quais são utilizados nas diferentes etapas de produção de um jogo digital.

### 4.1. CRIAÇÃO DE IMAGEM LIVRES E ABERTAS

Um destes projetos da *Free Software Gang* é o **Blender**, um software multiplataforma para modelagem e animação 3D, muito popular entre os designers iniciantes de jogos. Além de criar modelos 3D, também é possível criar animações e inclusive animações interativas para jogos eletrônicos. Usuários avançados do Blender e que possuem mais familiaridade com Python (linguagem em que o Blender é desenvolvido) desenvolvem scripts com novas funções e que passam a fazer parte do programa a partir do lançamento de versões mais novas<sup>9</sup>. A versão 2.72b mais recente do software foi lançada em 22 de outubro de 2014, que por sinal já é estável. A cada versão de atualização que é lançada, o Blender se mantém mais completa e funcional, sempre se beneficiando do conhecimento de seus usuários que ajudam a aprimorar a experiência de design e modelagem (**figura 4.1**).

---

<sup>8</sup> Disponível em <<https://www.fsf.org/working-together/gang>>. Acesso em 3 de outubro de 2014.

<sup>9</sup> Disponível em <<http://www.blender.org/about/>>. Acesso em 3 de outubro de 2014



Figura 4.1 - Modelagem 3D no Blender<sup>10</sup>

Para aqueles que preferem imagens 2D, o **GIMP** (acrônimo de *GNU Image Manipulation Program*, Programa de Manipulação de Imagem GNU, tradução nossa) é usado para manipulação, retoques e criação de imagens. Sua versatilidade permite ser usado como um simples programa para pintar, dar retoques em imagens profissionais, conversor de imagens, entre outros<sup>11</sup>. Assim como o Blender, o GIMP (**figura 4.2**) é multiplataforma e seu desenvolvimento é feito através de versões lançadas, que são melhorias de bugs encontrados pelos usuários. A versão estável mais recente lançada é de setembro de 2014, e o GIMP não inclui preços de uso e também é integrante da *Free Software Gang*.

Outra opção para manipulação de imagem é o **Inkscape**, que trabalha com gráficos vetoriais para imagem, semelhante aos concorrentes não-livres como Corel Draw e Adobe Ilustrador. Sua versão estável mais recente é de julho de 2014. O que o torna diferente de seus concorrentes é ser livre de custos e quaisquer taxas para uso e todos os formatos podem ser exportados para formatos familiares aos navegadores de internet ou para lojas gráficas, que costumam trabalhar com o tipo de imagem vetorial.<sup>12</sup>

<sup>10</sup> Disponível em <<http://www.rickety.us/wp-content/uploads/2008/11/blender.jpg>>, acesso em 7 de novembro de 2014.

<sup>11</sup> Disponível em <<http://www.gimp.org/about/introduction.html>>, acesso em 8 de novembro de 2014

<sup>12</sup> Disponível em <<https://inkscape.org/pt/acerca-de/visao-geral/>>, acesso em 8 de novembro de 2014.

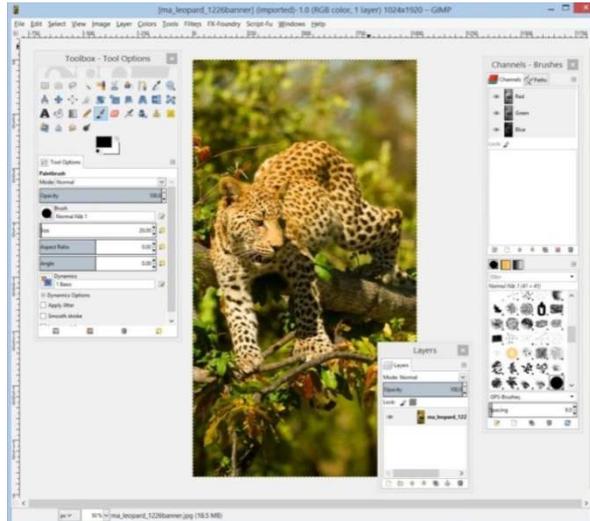


Figura 4.2 - Screenshot GIMP<sup>13</sup>

#### 4.2. PRODUÇÃO DE ÁUDIO

Para criação de áudio, encontra-se o **Audacity** (figura 4.3), software para criação e edição de sons. Sua versão estável e mais recente é de setembro de 2014. O Audacity é popular entre aqueles que buscam aprender a cortar, copiar, mixar e inserir inúmeros efeitos sonoros. É multiplataforma e também é distribuído sob a licença GPL<sup>14</sup>.

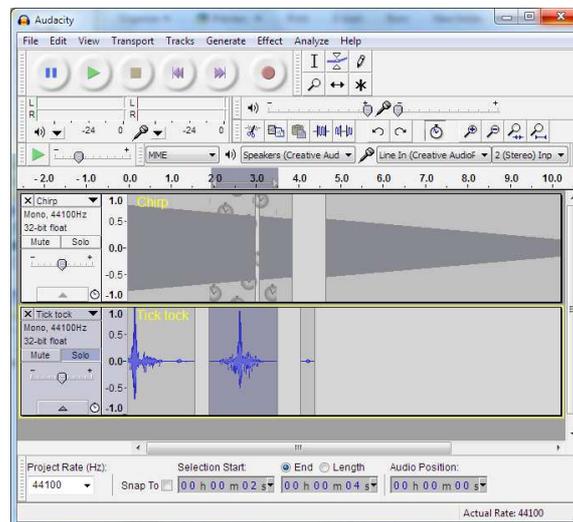


Figura 4.3 - Screenshot Audacity<sup>15</sup>

<sup>13</sup> Disponível em <<http://www.extremetech.com/wp-content/uploads/2013/10/Gimp-UI-screenshot.jpg>>. Acesso em 5 de novembro de 2014.

<sup>14</sup> Disponível em <<http://audacity.sourceforge.net/about/>>. Acesso em 4 de novembro de 2014.

<sup>15</sup> Disponível em <<http://audacity.sourceforge.net/about/images/audacity-windows.png>>. Acesso em 5 de novembro de 2014.

### 4.3. ENGINES LIVRES

Quanto às plataformas de produção livres, cita-se o **Pygame**, uma plataforma de desenvolvimento na linguagem Python, que também é uma linguagem livre, voltada para jogos. O Pygame é desenvolvido primordialmente para aqueles que buscam facilidade no aprendizado da programação de jogos. Sua versão estável e mais recente é de agosto de 2009. É multiplataforma, compatível a grande maioria dos sistemas operacionais disponíveis no mercado, desde as distribuições Linux, Windows, Mac OS até sistemas operacionais pouco populares, como FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, e QNX. Ainda, pode ser utilizado somente para processamento de imagem, coleta de entradas de dados pelo usuário, coletado através dos chamados *joysticks* (controles eletrônicos geralmente dedicados para jogos digitais) ou somente executar os sons do jogo. O Pygame é completamente livre, gratuito, e distribuído sob a licença GPL<sup>16</sup>.

A plataforma **Rubygame** foi produzida para os interessados em produzir jogos utilizando a linguagem de programação livre Ruby. Rubygame é flexível, fácil de aprender e muito poderosa. Ainda que seja desenvolvida para produção de jogos, pode ser utilizada em diferentes soluções e aplicações interativas. Sua versão mais atual é datada de abril de 2010 e também é distribuída sob as diretrizes da GLP<sup>17</sup>.

Entre as plataformas de desenvolvimento de código aberto podemos encontrar **Panda 3D (figura 4.4)**, gratuita para quaisquer intenções de uso, inclusive uso comercial. A *engine* Panda 3D trabalha com Python e C++. É compatível com os sistemas operacionais Windows e Linux<sup>18</sup>. Sua versão mais recente é de 2013. A biblioteca 3D utilizada é a *OpenGL* (Open Graphic Library), um conjunto de funções gráficas que se comunica diretamente com a placa de vídeo para gerar e processar as imagens (BORGES, 2006). Panda 3D é distribuída sob uma versão da licença de código aberto, que é datada de 2008 e que, entre outras regras, exclui o nome da produtora, a Universidade de Carnegie Mellon, como co-autora de qualquer produto

---

<sup>16</sup> Disponível em <<http://www.pygame.org/wiki/about#Facts%20About%20Pygame>>. Acesso em 12 de novembro de 2014.

<sup>17</sup> Disponível em <<http://rubygame.org/>>. Acesso em 10 de novembro de 2014.

<sup>18</sup> Disponível em <<https://www.panda3d.org/>>. Acesso em 10 de novembro de 2014.

feito quando há intenção de se favorecer inadequadamente em nome da universidade ou de seus colaboradores.<sup>19</sup>



Figura 4.4 – Engine Panda 3D<sup>20</sup>

Sob a licença de distribuição de código aberto do MIT, existem duas outras *engines*. A primeira, a **Phaser**, tem seu foco para jogos em HTML5 com *scripts* em *TypeScript* e *JavaScript* e pode ser exportado para jogos em navegador, *desktop* ou *mobile* e é totalmente gratuito<sup>21</sup>. A segunda *engine* é a **LÖVE**, igualmente gratuita e código-fonte aberto, podendo ser usada para fins comerciais sem restrições e é suportada para os sistemas operacionais Windows, Mac OS e Ubuntu<sup>22</sup>. A programação é baseada em Lua<sup>23</sup>, linguagem de programação licenciada pelos termos da licença de código aberto do MIT e inteiramente brasileira, desenvolvida e mantida pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ).

A **OGRE** (*Object-Oriented Graphics Rendering Engine*, Motor para Renderização de Gráficos Orientados a Objeto, tradução nossa) é outra solução distribuída pela licença do MIT e, diferentemente das ferramentas descritas

<sup>19</sup> Disponível em <<https://www.panda3d.org/license.php>>. Acesso em 2 de novembro de 2014

<sup>20</sup> Disponível em <<https://www.panda3d.org/randomscreens/ssg-code3d/code3D05.jpg>>. Acesso em 2 de novembro de 2014.

<sup>21</sup> Disponível em <<http://html5gameengine.com/details/25/phaser>>. Acesso em 1 de novembro de 2014.

<sup>22</sup> Disponível em <<http://love2d.org/>>. Acesso em 1 de novembro de 2014.

<sup>23</sup> Disponível em <<http://www.lua.org/license.html>>. Acesso em 29 de outubro de 2014.

anteriormente, não se trata de uma *engine*. A OGRE é uma biblioteca com funções para gráficos 3D em C++, feita para tornar a programação com ambientes 3D mais intuitiva. Suas soluções não se aplicam exclusivamente a jogos digitais, podendo ser utilizadas para quaisquer simulações de ambientes 3D.

#### 4.4. TÍTULOS DE JOGOS LIVRES E DE CÓDIGO ABERTO

Assim como foi para os primeiros softwares livres e de código aberto, a disseminação dos jogos de código aberto se faz através da internet. Ao disponibilizar um jogo livremente, pode-se obter mais tempo de vida na mão de seus usuários. Depois de muito jogar, o usuário tende a ficar entediado, uma vez que já se aproveitou de praticamente quase todas as maneiras de jogo. Quando é aberta aos usuários a oportunidade de expandir o cenário atual do jogo, aumenta-se as chances de mantê-lo atraente ao público (MIZUTANI, 2014).

Um jogo exemplo de que é possível manter um jogo atualizado e longe da obsolescência com a colaboração de seus usuários é o *Counter Strike*, jogo de tiro em primeira pessoa lançado no início dos anos 2000 pela empresa Valve. A empresa também lançou para o público uma *engine* para produção de conteúdo para seus jogos, a *Valve Hammer Editor* VHE.

Mesmo com quase quinze anos do lançamento de *Counter Strike*, o jogo ainda é mantido por muitos fãs de jogos *online* (já que um dos modos de jogo é *online* e permite a formação de dois times inimigos: os terroristas e contra terroristas), e tem como atrações mapas e modos incomuns<sup>24</sup>. Entre os mapas, podem ser encontrados na internet mapas similares à cidade de São Paulo e do Rio de Janeiro (**figura 4.5**). Nem o jogo e nem a *engine* são livres ou de código aberto, mas sua representatividade no modo colaborativo de criação de conteúdo se faz necessário para defender o sucesso de um projeto gerido por todos que participam da comunidade.

---

<sup>24</sup> Disponível em < <http://goo.gl/BY5jem>>. Acesso em novembro de 2014.



Figura 4.5 - Mapa do Counter Strike do Rio de Janeiro<sup>25</sup>

Um jogo com enredo similar ao *Counter Strike* e que entretanto é livremente distribuído através da licença GPL é o **Tremulous** (figura 4.6), jogo multiplataforma em que os jogadores devem escolher para jogar uma entre as duas raças de personagens: alienígenas ou humanos. A última versão lançada do jogo lançada pela sua produtora, a Dark Legion Development, é de 2009, mas encontram-se versões de correções de *bugs* datadas de 2011 e outras adaptações desenvolvidas por usuários na internet<sup>26</sup>.



Figura 4.6 - Screenshot do jogo Tremulous<sup>27</sup>

<sup>25</sup> Disponível em <<http://goo.gl/7TMmQK>>. Acesso em 27 de outubro de 2014.

<sup>26</sup> Disponível em <<http://tremulous.net/about/>>. Acesso em 4 novembro de 2014.

<sup>27</sup> Disponível em <<http://tremulous.net/media/files/promo03.jpg>>. Acesso 1 de novembro de 2014.

O jogo **Lincity**, nome resultante de um trocadilho intencional com *SimCity*, ambos jogos de simulação para administração de uma cidade. O jogador é responsável por manter a harmonia da cidade alimentando todos os cidadãos, construindo casas e edifícios, gerando emprego e ainda mantendo a cidade sustentável, através do aproveitamento de materiais recicláveis e fontes de energia renováveis. Sua última versão estável tem mais de dez anos – data de agosto de 2004 e é distribuído pela licença GPL<sup>28</sup>. Ainda que Lincity seja disponível para os sistemas operacionais ultrapassados Windows, Mac e Linux, se destaca por trazer uma experiência de jogo igualmente satisfatória que *SimCity*. Para aqueles que por ventura se interessam em aprender, renovar e recriar, o projeto se encontra disponível para *download* até os dias de hoje.

Outro representante da categoria de jogos que simula situações de tiroteio e de código aberto é **Assault Cube** (figura 4.7), desenvolvida através da *Cube Engine*, que pode ser jogado *online* em modo de cooperação ou guerra entre os jogadores. Seu diferencial está no aproveitamento da banda larga do computador e por ser, enquanto aplicativo, de tamanho consideravelmente menor do que outros jogos do mercado – ocupa cerca de 40MB no computador enquanto outros jogos costumam passar os 3GB de tamanho. Sua versão mais recente é de novembro de 2013 e o jogo é distribuído por uma licença de código aberto própria, que garante a liberdade de modificação que cada usuário possui<sup>29</sup>.



Figura 4.7 - Screenshot do jogo Assault Cube

<sup>28</sup><http://lincity.sourceforge.net/>, acesso em novembro de 2014.

<sup>29</sup><http://assault.cubers.net/>, acesso em novembro de 2014.

#### 4.5. *IMPULSO GOVERNAMENTAL PARA JOGOS LIVRES E ABERTOS NO BRASIL*

Devido à grande quantidade de jovens com acesso fácil à internet e computadores, o Brasil é um dos maiores mercados consumidores de jogos do mundo. O país vem chamando a atenção das grandes produtoras pelo potencial de consumo, pela rápida disseminação dos jogos entre diferentes faixas etárias e pela iniciação tecnológica precoce da população brasileira. Por outro lado, encontram grandes barreiras para investir mais no Brasil devido à falta de profissionais qualificados e as altas taxas de tributação. Muitos dos desenvolvedores de jogos brasileiros, confrontados com a falta de reconhecimento da área no país, preferem deixar o Brasil e se dedicar a suas carreiras em países com mais incentivos e aceitação da cultura de produção de jogos (CAVICHIOILLI et. al., 2014).

Imagine-se quantas não foram as boas oportunidades que programadores de jogos perderam devido as barreiras impostas pelas taxas e tributos federais. Um país com um público que, segundo a publicação do Portal R7, é o quarto maior consumidor de jogos do mundo<sup>30</sup> deveria ter a mesma classificação quanto ao número de profissionais da área e ao número de cursos disponíveis em universidades em todo o território.

Como apresentado anteriormente, os movimentos libertários de defesa de um software foram responsáveis por uma grande movimentação e revolução da maneira em que os softwares estavam sendo produzidos, gerando softwares com maior qualidade e tornando-os favoritos por grandes empresas de tecnologia da informação, ao usá-los como ferramentas principais devido a sua confiabilidade. Partindo da ideia o modo colaborativo de criação determina que o conhecimento seja de todos, projetar o modelo de produção de softwares livres e de códigos abertos para os jogos digitais poderia impulsionar o setor brasileiro ao atrair mais pessoas interessadas em se tornarem profissionais da área de entretenimento digital.

Diante das respostas econômicas que o setor de jogos trouxe ao país, o Ministério das Comunicações lançou em agosto de 2014 o concurso INOVApps que

---

<sup>30</sup><http://noticias.r7.com/educacao/fotos/brasil-ja-e-o-quarto-maior-mercado-consumidor-de-jogos-no-mundo-salarios-podem-chegar-a-r-10000-11092013>

buscava “premiar 25 aplicativos de utilidade pública e 25 jogos sérios<sup>31</sup> para dispositivos móveis e smart TVs<sup>32</sup>”. O objetivo deste concurso foi apoiar o desenvolvimento de conteúdo para dispositivos digitais, estimular o crescimento da economia do setor, fortalecer a produção de aplicativos e jogos nacionais e difundir o modelo de negócio baseado em software livre e código aberto no território brasileiro (MINISTÉRIO DAS COMUNICAÇÕES, 2014).

As inscrições foram aceitas até 19 de setembro e a temática do concurso se limitou aos seguintes temas: Direitos e Defesa do consumidor; educação / Ensino; Saúde; Mobilidade Urbana; Segurança Pública; Turismo e Grandes Eventos; Gestão colaborativa de utilidade pública; Tratamento de indicadores de políticas públicas; Difusão de campanhas de utilidade pública. Somadas, as inscrições para aplicativos e jogos sérios totalizaram 865 inscrições de projetos variados, de acordo com o edital preliminar de resultados do concurso. Ao final de todas as etapas de julgamento, os selecionados serão premiados com a quantia de oitenta mil reais para cada um dos vinte e cinco projetos de jogos sérios, e cem mil reais para cada um dos projetos de aplicativos, um prêmio significativo para aqueles que contribuem para o crescimento tecnológico do país.

Para Veras (2006), outro ponto extremamente favorável do software livre no Brasil e que deve ser mais explorado pelos órgãos governamentais é seu custo baixo. O país conta com inúmeras universidades prontas para receberem investimentos tecnológicos e, entretanto, ficam presas burocraticamente entre as licitações de compra de licenças de software proprietário. As universidades e outras instituições de ensino são os precursores de grandes ideias novas e ao incentivar tal proposta, não somente abrem-se margens para a criação de projetos igualmente promissores ao projeto Lua como também diminuem drasticamente os custos de manutenção e prorrogação de licença de produtos proprietários (VERAS, 2006). Software livre e de código aberto não é somente um incentivo ao desenvolvimento tecnológico. É também uma solução financeira para todos os tipos de negócios e instituições.

---

<sup>31</sup>Por jogos sérios, entende-se que são os jogos com objetivos didáticos nos temas propostos.

<sup>32</sup><http://www.mc.gov.br/inovapps>

## 5. ESTUDO DE CASO

Para somar as análises de ferramentas, jogos e iniciativas envolvendo software livre e de código aberto com jogos digitais, apresenta-se um estudo de caso: uma pesquisa com profissionais e estudantes envolvidos com programação de jogos.

O objetivo da pesquisa efetivamente é avaliar, através das respostas obtidas, a representatividade de tais ferramentas livres e abertas no cotidiano do desenvolvedor de jogo, o quão importante as ferramentas livres são para quem já desenvolve jogos, e descobrir se estes mesmos profissionais entendem a proposta de liberdade deste modelo de negócio.

### 5.1. METODOLOGIA DE PESQUISA

A metodologia da pesquisa foi moldada em formato de questionário e disponibilizada através dos formulários do Google Docs<sup>33</sup>. O questionário gira em torno das três questões principais, sendo elas respectivamente: quais ferramentas e plataformas de desenvolvimento, livres ou de código aberto, são utilizadas na produção de jogos digitais, classificação de importância de softwares livres ou de código aberto na produção de jogos digitais e se o entrevistado já desenvolveu algum jogo com o código aberto.

O questionário foi divulgado através da rede social Facebook, através de listas de e-mails e contatos profissionais. O formulário utilizado na coleta de dados se encontra na íntegra no **Apêndice** deste trabalho.

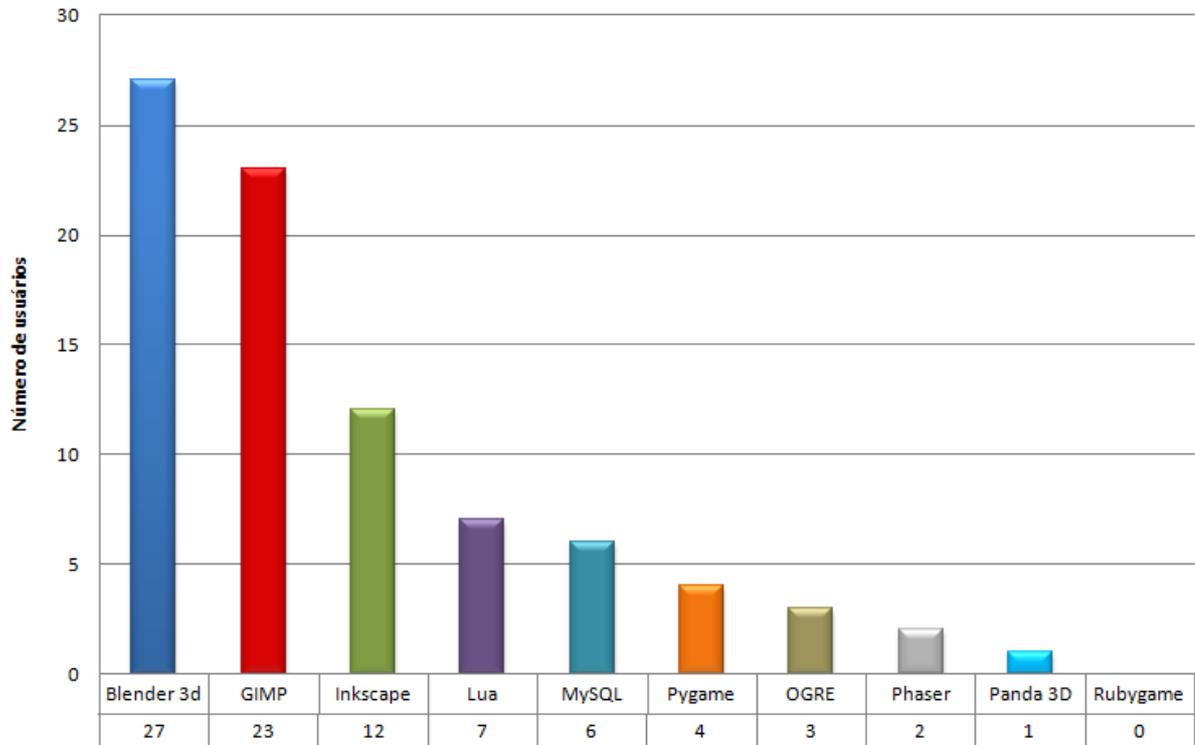
### 5.2. RESULTADOS OBTIDOS

A soma de 35 respostas submetidas na pesquisa foram quantificadas e moldadas em gráficos para a compreensão e análise dos resultados obtidos. Conforme a **Figura 5.1**, nota-se o gráfico quantitativo das respostas referentes à pergunta que avalia quantos softwares livres e abertos o entrevistado conhece.

---

<sup>33</sup> Disponível em <<http://goo.gl/forms/YOitTjoxdz>>. Acesso em 29 de outubro de 2014.

**Quais das seguintes ferramentas e plataformas de desenvolvimento, livres ou de código aberto, você utiliza ou já utilizou na produção de jogos digitais?**

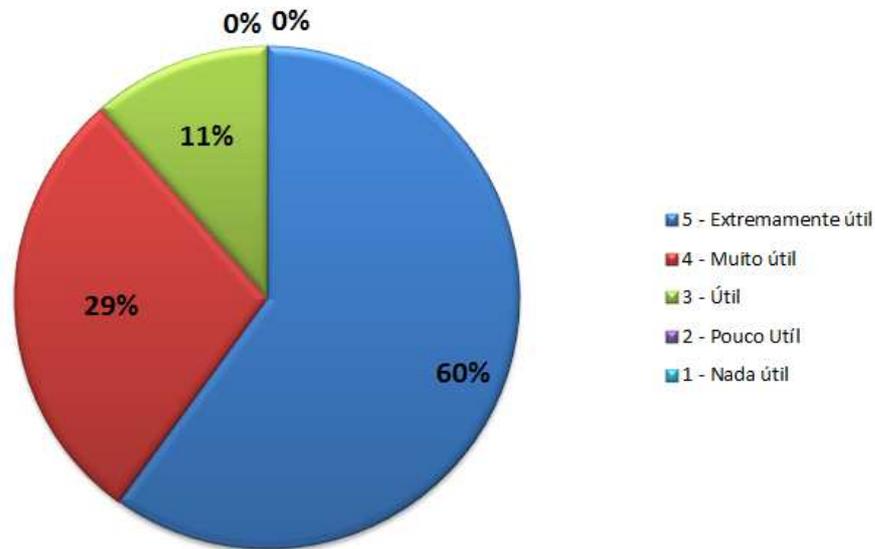


**Figura 5.1 – Usuários por plataforma. Fonte: próprio autor.**

Os softwares mais populares foram o Blender, o GIMP e o Inkscape. Do total de entrevistados, 27 declaram-se usuários do Blender em seus projetos. Outros 23 do mesmo total de entrevistados declararam que utilizam o GIMP e 12 pessoas utilizam o Inkscape. Os outros softwares, como Lua, MySQL, Pygame, OGRE, Phaser e Panda 3D contabilizaram respectivamente 7, 6, 4, 2 e 1 usuários e nenhum dos entrevistados utiliza ou já utilizou a *engine* Rubygame.

Também foi questionado aos usuários o quão relevante eles considerariam que estes modelos de software são para o atual cenário de desenvolvimento tecnológico nacional e global.

**Como você classifica a importância de softwares livres ou de código aberto na produção de jogos digitais?**



**Figura 5.2 – Importância dos Softwares Livres e Abertos para Produção de Jogos Digitais.**

Fonte: próprio autor

Numa escala numérica de um a cinco, os usuários classificaram a importância dos softwares livres e abertos na produção de jogos digitais, sendo cinco a importância mais alta. Do total de respostas, 60% classificaram que os softwares livres e abertos está na escala “5” - extremamente úteis para a produção de jogos digitais (**figura 5.2**). A parcela corresponde a 29% classificou software livre e aberto como “4” - muito úteis e outra parcela de 11% classificou como “3” – úteis. Não houve respostas que classificaram os softwares livres e abertos como pouco úteis ou nada úteis.

A última questão da pesquisa perguntava se o profissional já desenvolveu algum jogo de código aberto. Dos entrevistados, 46% afirmaram que já contribuíram com softwares de código aberto e/ou livres. Outros 31% afirmaram que não fizeram um jogo de código aberto porque nunca avaliaram a hipótese, 8% tem medo de plágio/roubo, 3% tinham restrições específicas quanto às plataformas que utiliza e 6% não distribuiu nenhum jogo com o código aberto porque não finalizaram um projeto (**figura 5.3**).

### Você já fez algum jogo com o código aberto?

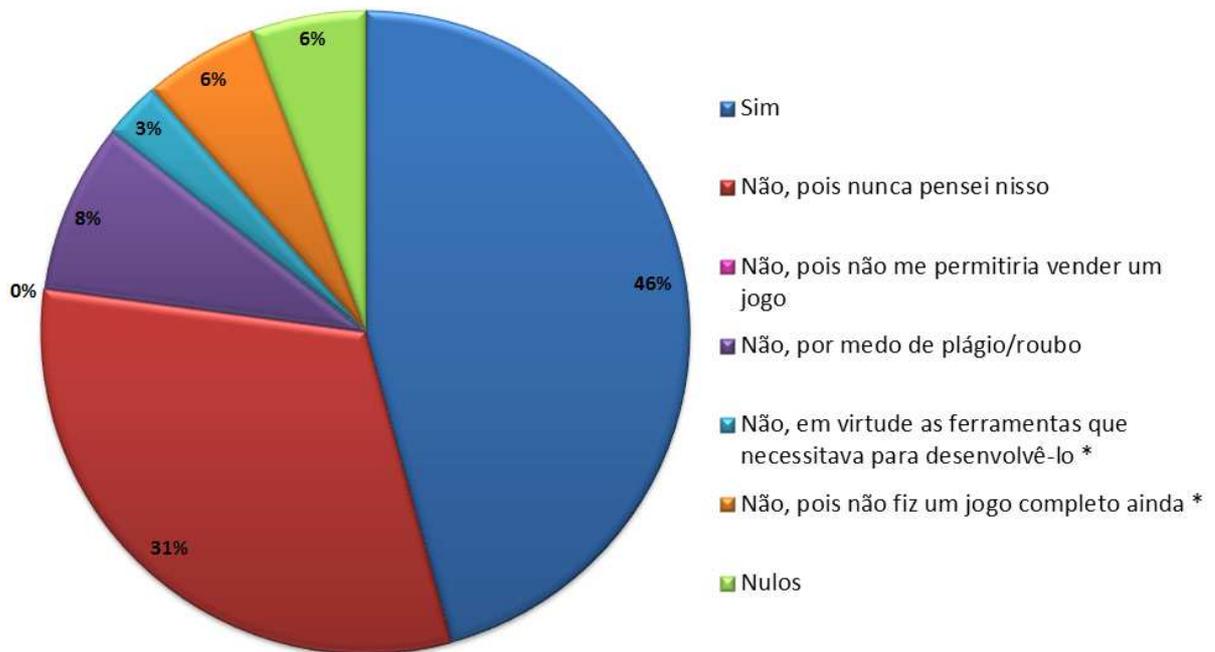


Figura 5.3 - Entrevistados que já produziram jogos de código aberto - Fonte: próprio autor

As legendas do gráfico destacadas com o asterisco indicam respostas submetidas no campo “outros motivos”, que ficou aberto a quaisquer outras razões não citadas pelas alternativas anteriores. Votos nulos contabilizaram 6% e se restringem às respostas recebidas em “outros motivos” que não puderam ser classificadas por se esquivarem do assunto proposto. Nenhuma pessoa declarou não ter produzido um jogo de código aberto por empecilhos comerciais.

#### 5.3. ANÁLISE DOS DADOS

A tendência observada foi que quanto mais versões e atualizações recentes um produto tem, maior a chance de possuir maior quantidade usuários. As últimas versões do Blender, GIMP e Inkscape são respectivamente de outubro, setembro e julho de 2014. Já as versões recentes do Panda 3D, Rubygame e Pygame são de 2013, 2010 e 2009, respectivamente. Este fato está ligado com as razões apresentadas por Michlmayr, Hunt e Probert (2005), que relatam a competitividade é muito alta num mercado democrático como o de softwares livres e abertos que permitem que muitas pessoas representem, ao mesmo tempo, papel de desenvolvedores, usuárias e colaboradoras. Quanto mais tempo a gerência de um projeto for capaz de mantê-lo inovador e atrativo, maiores as chances de manter um

usuário interessado no produto. Considerando que um usuário pode se tornar um colaborador para o projeto, o gerenciamento de projetos é extremamente importante para a vida útil de um software livre e/ou de código aberto.

Um campo aberto para digitação chamado de “outros” foi incluído na pergunta II. “Como você classifica a importância de softwares livres ou de código aberto na produção de jogos digitais?” caso o desenvolvedor entrevistado notasse a necessidade de inserir outro projeto não citado pelo enunciado. O Audacity foi um dos poucos citados que é realmente livre. Considerando outros produtos que foram citados, observa-se confusões na definição do conceito de software livre e aberto, pois muitos softwares que são apenas gratuitos, e que não permitem alterações e modificações em seus códigos-fonte e nem abragem a liberdade total de uso de um usuário foram citados. A liberdade de usuário não é confundida como liberdade de uso e modificação, mas sim liberdade de taxas. Não está completamente claro e disseminado entre os profissionais da área da ciência da computação qual o verdadeiro propósito do software livre e de código aberto. O receio que ainda existe em muitas pessoas de abandonar o modelo proprietário pode ser consequência dessa confusão.

A pesquisa realizada não possuiu um embasamento estatístico. As análises feitas são apenas indícios de uma pequena quantidade de profissionais da área da ciência da computação. Seria necessário em um trabalho futuro abordar um número maior de pessoas para coleta de dados e uma melhor análise estatística para resultados satisfatoriamente confiáveis.

## 6. CONCLUSÃO

Através de todos os aspectos aqui estudados, foi possível compreender qual a real proposta dos movimentos que defendem a liberdade de um software através do controle total que um usuário possui em sua execução e distribuição. Para os precursores destes movimentos, o mais importante é o direito de escolha que todos deveriam exercer nas aplicações que os ajudam, e não deixar que as empresas, que além de deter todo o conhecimento para si, deixem que as máquinas determinem como deve-se agir. Cada solução foi feita a seu modo, e através da junção de mundos e realidades que a internet trouxe ao decorrer das décadas, a um custo de propriedade consideravelmente menor, surgiram soluções novas e igualmente moldáveis para cada usuário que se interessasse pelo modelo de construção digital independente de empresas e seus direitos exclusivos.

Observando os dados obtidos na pesquisa realizada durante esse trabalho, pode-se ter uma pequena visão de que o jogo livre e aberto representa atualmente para o setor de produção de jogos. Poucas pessoas sabem que é possível fazer um jogo e deixar seu código-fonte disponível. O setor de desenvolvimento de jogos ainda não percebeu o potencial de sucesso que pode ser obtido através da abertura de código. Faltam casos de jogos que fizeram tanto sucesso quanto o Linux fez. Todavia, projetos como o INOVApps que buscam popularizar a produção de softwares livres e de código aberto são os principais incentivadores para que a tecnologia avance nesse caminho. Outro caminho que pode levar à evolução tecnológica do setor de software livre e de código aberto é incentivar programadores, analistas, técnicos e tecnólogos desde o início de seu desenvolvimento profissional nas universidades e escolas de formação a tornarem seus códigos livres e abertos, para que mais pessoas possam fazer uso benéfico do conhecimento.

Este é um assunto pouco abordado entre os acadêmicos da ciência da computação. Encontra-se pouca quantidade de pesquisas e análises realizadas sobre software livre e de código aberto para jogos digitais. Para que este trabalho seja um grande colaborador para os interessados na área, inclui-se em seus objetivos futuros avaliar o impacto gerado aos programadores que abrem seus códigos fontes em sites de gerenciamento e armazenamento de código, como o

*GitHub*, em que é possível deixar projetos pessoais disponíveis. Algumas empresas buscam exclusivamente candidatos que já contam com portfólios disponíveis na internet e usam isto como artifício para avaliar o grau de conhecimento de um possível novo colaborador para a empresa. Outro objetivo futuro é avaliar com o público que já conheceram algum dos jogos livres e abertos que existem e convidá-los para testar os jogos dessa categoria e classificá-los quanto ao divertimento proporcionado, facilidade de compreensão, qualidade e desempenho do jogo.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

ALEXANDRE, Patrícia Vieira Machado. GUTIERREZ, Regina Maria Vinhais. **Complexo Eletrônico: Introdução ao Software**. Departamento da Indústria Eletrônica do BNDES, setembro de 2004.

AUGUSTO, Maurício Pires. **Um estudo sobre as motivações e orientações de usuários e programadores brasileiros de software livre**. Universidade Federal do Rio de Janeiro, 2002.

BORGES, Thiago Trezza. **Utilização de opengl e linguagem orientada a objetos para desenvolvimento de um pacote gráfico para análise de redes**. Universidade Federal de Juiz de Fora, Minas Gerais, agosto de 2006.

CAMPOS, Augusto. **O que é software livre**. BR-Linux. Florianópolis, março de 2006. Acesso em janeiro de 2014.

ELIAS, Paulo César. MATTOS, Fernando Augusto M. Informação e software livre no capitalismo contemporâneo. Revista Digital de Biblioteconomia e Ciência da Informação, Campinas, v.5, n. 1, p 55-76. Jul/dez. 2007.

FERNANDES, Jorge Henrique Cabral. **O que é um programa (software)?** Universidade de Brasília, 2002.

FILHO, Ademir Constantino. **Software livre**. Faculdade Metropolitana de Blumenau – FAMEBLU. Florianópolis, junho de 2009.

FSF, Free Software Foundation. **O que é software livre?**. Estados Unidos, 1991.

GUESSER, Adalto Herculano. **Software livre e controvérsias tecnocientíficas: uma análise sociotécnica no Brasil e em Portugal**. Universidade Federal de Santa Catarina. Florianópolis, 2005.

JACOBSON, Jeffrey. LEWIS, Michael. **Game engines in scientific research. communications of the ACM**. Janeiro de 2002. Vol. 45, No. 1.

KON, Fabio. LAGO, Nelso. MEIRELLES, Paulo. SABINO, Vanessa. **Software livre e propriedade intelectual: aspectos jurídicos, licenças e modelos de negócio**. Universidade de São Paulo. JAI/SBC, 2011.

COMUNICAÇÕES, Ministério das. **Regulamento de concurso de seleção de projetos de aplicativos e jogos sérios de interesse público**. Edital Nº 11/2014/SEI-MC, 2014.

MIZUTANI, Wilson Kazuo. **Jogos digitais e software livre**. São Paulo, Jornal Estadão de São Paulo, 14 de fevereiro de 2014.

OSI, Open Source Initiative. **The open source definition**. Estados Unidos, 1998.

PETRUCELLI, Ana Clara Faleiro; et. al. **Vantagens e desvantagens do uso do software livre no mundo acadêmico e profissional.** 2011.

PROBERT, David. HUNT, Francis. Michlmayr, Martin. **Quality practices and problems in free software projects.** Universidade de Cambridge, Cambridge, Reino Unido, 2005.

RAYMOND, Eric S. **The cathedral and the bazaar.** Estados Unidos, 1998.

SERRA, Glades Debei. **Software livre – um bem pertencente à humanidade.** CDCC-USP. São Carlos, maio de 2008. Acesso em abril de 2014.

SILVA, Maycon Prado Rocha. COSTA, Paula Dornhofer Paro. PRAMPERO, Paulo Sérgio. FIGUEIREDO, Vera Aparecida. **Jogos digitais: definições, classificações e avaliação.** Universidade Estadual de Campinas, 2009.

STALLMAN, Richard. **O que é software livre?** Estados Unidos. Acesso em 12 de maio de 2014.

TORVALDS, L. Diamond, D. **Just for fun: the story of an accidental revolutionary.** Harper Collins Publishers, 2002.

VAZ, Thaís Emília Rodrigues. OLIVEIRA, Eliane Vendramini de. CAVICHIOILLI Adriane. AFFONSO, Elaine Parra. **Jogos digitais: estudo sobre mercado de trabalho e perfil do profissional.** Revista RETC – Edição 14ª, abril de 2014. Acesso em 6 de setembro de 2014.

VERAS, Jaclason Machado. **Software livre: um fator indispensável para a estruturação educacional das universidades.** 2006.

WILLIAMS, Sam. **Free as in Freedom: Richard Stallman's Crusade for Free Software.** Califórnia Estados Unidos, 2002. Acesso em 19 de setembro de 2014.

## 8. APÊNDICE

# Software Livre e Código Aberto na evolução do desenvolvimento de Jogos Digitais

Estudo de caso para o trabalho de conclusão de curso

\*Obrigatório

Idade \*

Cidade \*

Área de atuação profissional \*

Quais das seguintes ferramentas e plataformas de desenvolvimento, livres ou de código aberto, você utiliza ou já utilizou na produção de jogos digitais? \*

Selecione quantos achar conveniente, desde ferramentas de criação e modelagem à gerenciamento de dados.

- Blender 3D
- Inkscape
- GIMP
- Pygame
- Rubygame
- Panda 3D
- Phaser
- OGRE
- Lua
- MySQL
- Outro:

**Como você classifica a importância de softwares livres ou de código aberto na produção de jogos digitais? \***

Considere de 1 a 5 a escala de relevância, sendo 5 a mais importante.

- 5 - Extremamente útil
- 4 - Muito útil
- 3 - Útil
- 2 - Pouco útil
- 1 - Nada útil

**Você já fez algum jogo com o código aberto? \***

- Sim
- Não, pois nunca pensei nisso
- Não, pois não me permitiria vender um jogo
- Não, por medo de plágio/roubo
- Outro:

**Email para contato posterior**

Caso você se interesse, podemos enviar a conclusão da nossa pesquisa para você.