

# Criação de jogos eletrônicos: incentivando a produção

David Tayrone O. Silva<sup>1</sup>, Orlando S. Oliveira<sup>1</sup>

<sup>1</sup>Instituto Federal do Sertão Pernambucano (IFSertão)

dtayrone2001@gmail.com, orlando.silva@ifsertao-pe.edu.br

**Resumo.** Este artigo irá tratar sobre a criação de jogos eletrônicos, com o intuito de gerar interesse de desenvolvimento na área, mostrando ferramentas para a criação de vários espectros que tangem seu desenvolvimento e conceitos básicos necessários para a estruturação do jogo.

**Palavras-Chave:** jogos eletrônicos, desenvolvimento, incentivo e produção.

*Abstract.* This article will deal with the creation of electronic games, in order to generate development interest in the area, showing tools for the creation of various spectra that tangent its development and basic concepts necessary for the structuring of the game.

*Keywords:* electronic games, development, incentive and production.

## 1. Introdução

O mercado de jogos vem crescendo desde os anos 90 e atualmente é um mercado avançado. De acordo com as pesquisas feitas pela NewZoo (2018), já existem aproximadamente 75,7 milhões de jogadores no Brasil. Mas não é apenas o número de jogadores que cresceu nesse tempo, o lucro do mesmo meio no Brasil também, com uma receita a cerca de 1,5 bilhões de dólares. No mundo o número é ainda maior, a receita de jogos é de aproximadamente 137,9 bilhões de dólares, (NEWZOO, 2018). Como mostrada pela PWC(2019, p. 7) o mercado de *video games* já é 60% do capital global de mídia e entretenimento, sendo maior que a junção do que outros tipos de mídias já familiarizados, como cinema, música, teatro.

Mesmo com os números crescentes da indústria de jogos no Brasil e no mundo em geral, o investimento na área de jogos eletrônicos no Brasil é pequeno. Como mostrado no mapa do site de DIAS (2018) Mapa da Indústria de Jogos, a maior concentração dessas empresas estão localizadas no sul e sudeste do país, nas áreas centrais e nordeste, o número decai bastante, para nível de comparação descrito por Dias (2018), no estado de

Pernambuco, existem apenas nove empresas de desenvolvimento de jogos, enquanto na cidade de São Paulo possui aproximadamente 66.

De forma geral, o objetivo do artigo foi desenvolver um jogo, demonstrando meios para sua construção, apresentando possíveis ferramentas para sua criação em todas as áreas que necessitam a produção de um jogo.

O artigo foi dividido em três seções. No referencial teórico será tratado conceitos básicos necessários para entender a produção de um jogo eletrônico, na metodologia iremos tratar a utilização destes conceitos com ferramentas para cada aspecto e no teste de usabilidade faremos um teste com usuários para avaliar a qualidade do *game*.

## **2. Referencial Teórico**

Para definirmos como produzir um jogo é necessário definirmos os princípios de um jogo eletrônico. Segundo Schuytema (2008, p.447, apud Lucchese e Ribeiro, 2009, p.8) um jogo eletrônico é uma atividade lúdica formada por ações e decisões que resultam numa condição final. Tais ações e decisões são limitadas por um conjunto de regras e por um universo, que no contexto dos jogos digitais, são regidos por um programa de computador.

Na produção de um jogo eletrônico, há vários rumos requisitados para uma estruturação adequada e organizada. Dito por Battaiola (2000, p. 83–122, apud Lucchese e Ribeiro, 2009, p.8) afirma que o jogo eletrônico é composto de três partes: enredo, motor e interface interativa. Essas informações são organizadas em um documento chamado *Game Design Document*.

O documento de design do jogo ou Game Design Document (GDD) é o documento que apresenta detalhadamente todas as características de um jogo, como histórias, conceitos, personagens, cenários, mecânicas e sons, por exemplo. Este documento serve de referência para todos os envolvidos entenderem e estarem a par do desenvolvimento do jogo (HIRA et al. 2016, p. 329).

Interface interativa é conhecido como HUD (*Head up display*), que se traduziria para interface a frente da cabeça. Esse sistema será o meio de visualização de alertas, barra de vida, habilidades, mapas e qualquer outra informação considerada útil. É a interface vista pelo jogador. Como explicado pelo Fagerholt e Lorentzon(2009, p. 1, tradução nossa),

HUD é qualquer exibição transparente que apresente dados sem a necessidade do usuário desviar o olhar do seu ponto visual habitual<sup>1</sup>.

O motor é chamado de *Game Engine*, segundo Gregory(2009) Game Engines ou motores de jogos são middlewares responsáveis por simplificar e abstrair elementos do desenvolvimento de aplicações gráficas ao prover um conjunto de ferramentas integradas, entre as quais: motores gráficos, de física e de conexão e rede (network, linguagens de script, softwares para o controle de som, gerenciadores de arquivos, etc) (apud MACEDO, 2015, p. 182).

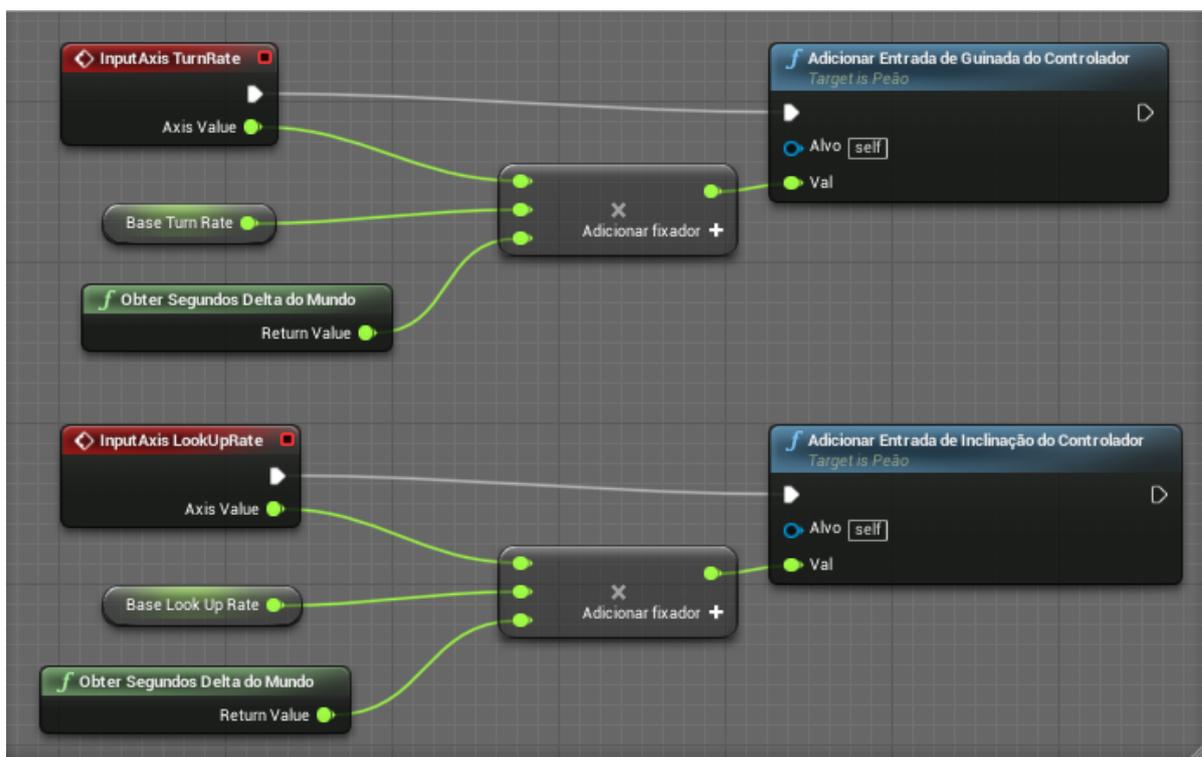
Assim, a Game Engine é uma ferramenta na qual se pode gerir todas as necessidades de um jogo eletrônico, caracterizados por: gráficos, sonorização, física, animação, programação.

Existem várias Game Engines gratuitas no mercado, sendo algumas delas a Unity, que é gratuito, mas possui uma delimitação e Godot, que é *open source*, com vários *plugins* da comunidade. A Game Engine escolhida para a produção foi o Unreal Engine 4 (UE4), uma game engine gratuita criado pela Epic Games<sup>2</sup>. Esta é uma das mais utilizadas no desenvolvimento de jogo no mercado, sendo usado pelas maiores empresas do mercado atual, como Ubisoft, EA Games, NetherRealm Studio, Sega. O Unreal Engine é um conjunto completo de ferramentas de criação projetadas para atender ambiciosas visões artísticas, sendo flexível o suficiente para garantir o sucesso de equipes de todos os tamanhos, (Epic Games, 2019). Em sua programação, há 3 linguagens de programação possíveis para a programação do jogo: C++, Blueprint Scripting e Python (sendo preciso ativar um plugin para essa função). Segundo Valcasara(p. 2, 2015, Tradução nossa) Blueprint é um sistema de *scripting* visual de alto nível que fornece intuitividade, interface baseada em nós que pode ser usado para criar qualquer tipo de evento de *script* no Unreal Editor. A ferramenta fornecida pode ser usada por level designers, artistas, e qualquer não programador, para criação rápida e *gameplay* sem precisar escrever uma linha de código, como é ilustrado na Figura 1.

---

<sup>1</sup> “...HUD, is any transparent display that presents data without requiring the user to look away from his or her usual viewpoint.”

<sup>2</sup>Epic Games. Disponível em: <https://www.epicgames.com/store/en-US/>



**Figura 1 - Programação Blueprint baseada em nós.**

**Fonte: Captura de tela do próprio autor.**

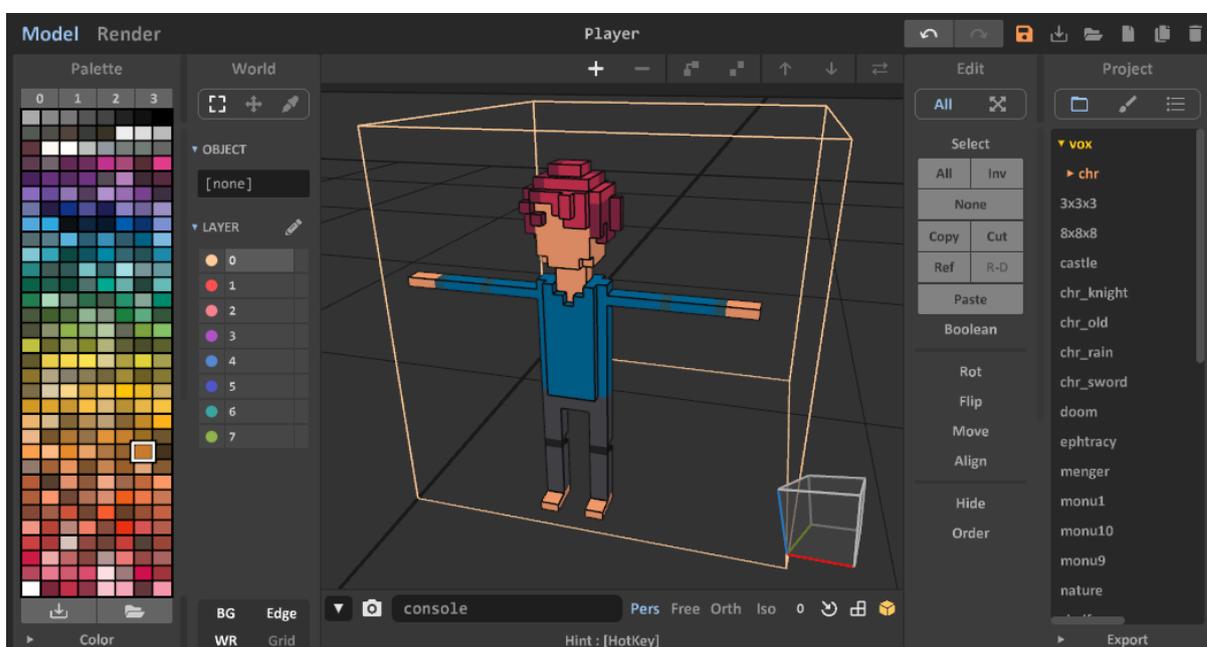
Com a linguagem Blueprint um desenvolvedor que está iniciando conseguirá com mais entender e programar mais facilmente as funções do jogo.

### 3. Metodologia

A criação de um jogo sem a organização de ideias, intuítos, ferramentas e tudo que o compõe pode deixá-lo caótico e mal feito. Para compor um jogo de forma não caótica, é extremamente necessário a criação do registro *Game Design Document*, utilizado a sigla GDD. Nele foi colocado tudo que havia de ideias, ferramentas e modos de execução, além de elementos, de forma explicada, das características do jogo, como inimigos, interfaces, personagem do jogador e todas as outras. Com sua conclusão, a produção começa com a *game engine*. Na criação do projeto no UE4, o modelo base usado é o “Side Scrolling Game”. Um jogo de plataforma, também chamado de *Side Scrolling Game*, é um jogo, com apenas os vetores X e Y, a qual o jogador sairá de um ponto A a um ponto B, apenas se movimentando nesses dois eixos, podendo ser com estilo 2D ou 3D. Em um modelo 3D, o eixo Z terá, geralmente, uma função de profundidade, sendo conhecido popularmente por jogo 2.5D. De acordo com Sajina e Orehovacki (2018, p. 1826) side-scrolling é usado em

jogos onde a câmera é colocada em um ângulo de visão lateral. Esses tipos de jogos é usualmente em 2D com personagens se movendo da esquerda para a direita da tela.

O jogo, no qual foi denominado Realixel, será no estilo plataforma com visual 3D. Os modelos 3D usados foram feitos em um software chamado MagicaVoxel, um programa de modelação que utiliza o estilo de voxel, o software é gratuito e *open source*, exemplificado na Figura 2, e possibilita a exportação do modelo em Obj, formato de arquivo utilizado para salvar malhas poligonais 3D para ser utilizado por outro software.

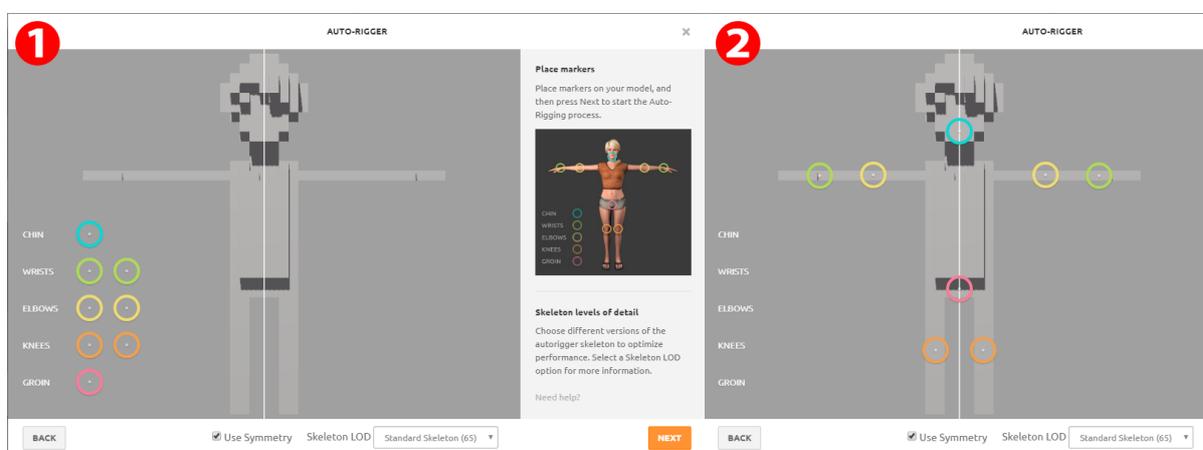


**Figura 2 – Criação do personagem em voxel no MagicaVoxel.  
Fonte: Captura de tela do próprio autor.**

Seguindo ainda a criação dos personagens, o site Mixamo da Adobe foi utilizado para o *auto-rigging* dos modelos exportados do MagicaVoxel. Rigging, ou animação esquelética são ossos interligados, que são usados para animar malhas 3D. Como descrito por Hasse et al(2019), *rigging* é o processo pelo qual um objeto em animação 3D (em alguns casos em 2D digital também) passa para ser manipulado e executar os movimentos coordenados pelo animador, e é algo fundamental para quase todos os processos gráficos em terceira dimensão. O Mixamo fornece a possibilidade de ser colocada uma malha 3D e, por meio de círculos disponibilizados ao lado esquerdo, posicionando cada parte nas áreas

corretas (queixo, cotovelos, pulsos, virilha e joelhos), como mostrado na Figura 3, o mesmo automatiza a criação de *rigging* e disponibiliza animações já feitas para o uso.

Na Figura 3, o primeiro ponto mostra a tela inicial do auto-rigging, como vem posicionado os círculos de rig, já no segundo ponto é um exemplo de como utilizar os círculos de rig. O arquivo que é fornecido estará em formato FBX, que é suportado pelo UE4. A própria Epic Games oferece materiais para desenvolvimento, como modelos de cenário e efeitos visuais.



**Figura 3 – Auto-rigger do site Mixamo.  
Fonte: Captura de tela do próprio autor.**

Na criação de artes 2D, utilizadas para os botões, barra de vida, HUD e ícones, foi empregado o Asesprite, software pago para design em pixel, utilizado para desenhos e animação, podendo ser substituído por um software gratuito como o Gimp.

O desenvolvimento foi todo feito no UE4, onde podemos fazer o gerenciamento de personagens, HUD's, programação e *level design*. *Level design* é a área de cunho artístico onde é feito a criação do nível, como bem nos assegura Fagerholt e Lorentzon (2009, p. 2, tradução nossa), um level designer constrói a geometria do jogo, construindo o mundo real a qual o jogador percorre e explora<sup>3</sup>. A programação foi toda feita em Blueprint Scripting, por ser uma linguagem de fácil aprendizado e completa o suficiente para a criação de um

<sup>3</sup> “A level designer builds the geometry of the game, by constructing the actual world that the player moves through and explores.”

jogo completo. A criação do mapa, foi baseado em estilos visuais medievais, como castelos de pedra e masmorras.

Para a análise da qualidade do jogo, foram formuladas perguntas nas quais pontuam partes importantes na construção do jogo, tais perguntas foram:

- Você ficou satisfeito com o jogo;
- O jogo foi divertido;
- Qualidade da mecânica;
- Qualidade da dificuldade;
- Qualidade dos gráficos;
- Qualidade dos menus;

Todas as perguntas são respondidas no formato de formulário utilizando a escala Likert. Como explicado por Aquiar, Correia e Campos (2011 ,p. 2) escalas Likert são uma das escalas de autorrelato mais difundidas, consistindo em uma série de perguntas formuladas sobre o pesquisado, onde os respondentes escolhem uma dentre várias opções, normalmente cinco, sendo elas nesta ocasião como: Muito bom, bom, razoável, ruim e muito ruim. Tal questionário terá, portanto, cinco respostas para cada pergunta, sendo enumeradas de 1 a 5. O critério utilizado para avaliar se as expectativas foram alcançadas será, se a resposta for de 3 acima. Os usuários que testaram possuem um perfil de jogadores (*gamers*), pessoas que já convivem na área de jogos como consumidores, tendo uma base de noção sobre as áreas avaliadas.

#### **4. Avaliação de resultados de pesquisa**

Com o jogo pronto no UE4, podemos compila-lo. Para compilar um projeto final no UE4, necessita configurar o estado de construção, para isso, nas “configurações de compilação” alteramos o modo *Development* para *Shipping*, assim a versão finalizada não terá comandos do console e estatísticas de desenvolvimento. Na Unreal Engine, temos a possibilidade de compilar para Windows, Linux, HTML5 e Android. As figuras 4 e 5 ilustra o visual final do jogo.



Figura 4 – Menu inicial do jogo.  
Fonte: Captura de tela do próprio autor.

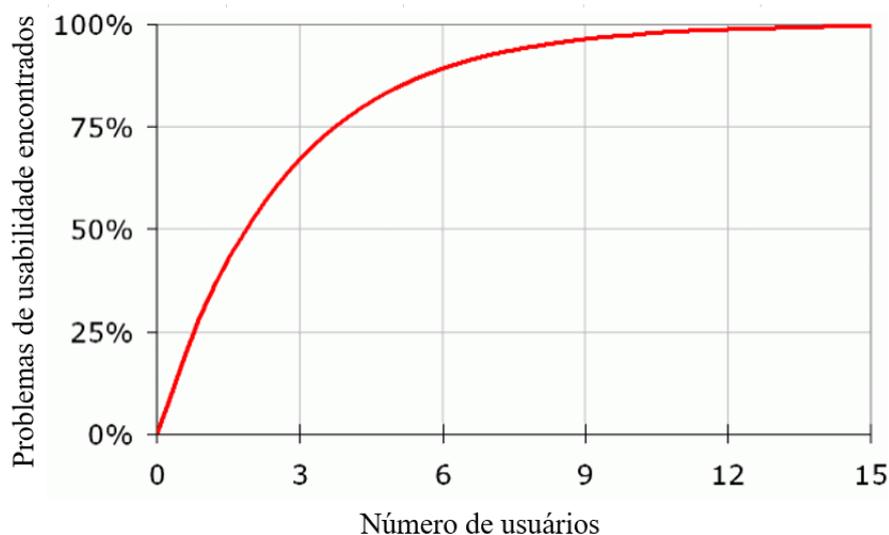
A compilação feita foi para Windows 32 bits e publicada em uma plataforma para exposição de jogos indies, chamada de Itch.io, como dito pela própria Itcho Corp (2019, tradução nossa), a Itch.io foi feito para dar ao desenvolvedor de jogos uma loja na qual eles controlam como o seu conteúdo é vendido. Na versão final, o menu de *pause* possui um botão no qual carrega uma página web no HUD, onde os usuários que testaram poderão preencher os questionários, que serão utilizados para esta avaliação de resultados.



**Figura 5 – Mapa 1 com personagem e monstro.**  
**Fonte: Captura de tela do próprio autor.**

#### **4.1. Teste de Usabilidade**

O teste foi realizado com seis pessoas, que em um teste de usabilidade, já é o suficiente para definir acertos e erros para a nossa proposta de jogo. “Para identificar os problemas de usabilidade mais importantes de um projeto, testar 5 usuários geralmente é suficiente. Ao invés de executar um estudo grande e caro, é um melhor uso dos recursos para executar muitos testes pequenos e revisar o design entre cada um para que você possa corrigir as falhas de usabilidade ao identificá-las”, (NIELSEN, 2016). Demonstrado por Nielsen, 5 usuários já conseguem encontrar aproximadamente 75% dos problemas em um projeto, visualizável no gráfico 1. As respostas dadas pelos usuários são demonstradas na Tabela 1.



**Gráfico 1 – Crescimento de problemas encontrados por usuário.**

**Fonte: Nielsen Norman Group<sup>4</sup>.**

<sup>4</sup> Nielsen Norman Group. **Why You Only Need to Test with 5 Users.** Disponível em: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>

	Você ficou satisfeito com o jogo?	Achou o jogo divertido?	Mecânica	Dificuldade	Gráficos	Menus
Usuario 1	3	2	2	3	5	2
Usuario 2	5	5	3	3	3	3
Usuario 3	3	3	4	4	5	4
Usuario 4	3	4	2	4	5	1
Usuario 5	4	3	3	4	5	3
Usuario 6	3	3	2	3	5	2
Acima de 3	6	5	3	6	6	3
Porcentagem	100%	83%	50%	100%	100%	50%

**Tabela 1 – teste de usabilidade.**

**Fonte: Próprio autor.**

Na formulação da satisfação geral do jogo, utilizando o critério de que respostas acima de 3 serão contabilizadas como “satisfeito”, possuímos uma satisfação de 100% dos usuários, com uma moda de resultado 3, mostrando que, por mais que possua falhas, ainda é cativante em seus aspectos.

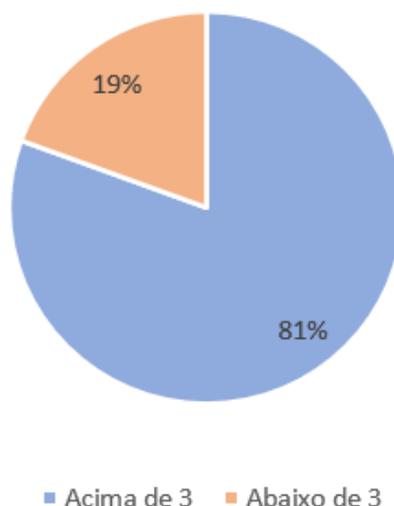
O critério diversão seguiu a mesma linha da satisfação, porém desta vez, apenas 83% dos usuários achou cativante.

A mecânica já decai bastante de nota comparando com as anteriores, agradando somente 50%. Provavelmente tenha sido um dos fatores que contribuiu para a diminuição de sua cativação.

Tanto a dificuldade quanto o gráfico atingiram 100%, mas a maior notoriedade que temos, é em relação aos gráficos, que possui uma moda de 5, mostrando que é o grande ponto de qualidade do jogo. Já os menus, possuem uma nota bem abaixo da maioria, atingindo a mesma que a nota em “mecânica”, de 50%.

Analisando todas as notas contabilizadas, temos no total 36 respostas. De 36, 29 delas são acima de 3, ficando apenas 7 abaixo disso, como demonstrado no gráfico 2. Isso mostra que, no mínimo o jogo atinge a nota razoável.

## Porcentagem de notas razoáveis



**Gráfico 2 – Porcentagem de notas razoáveis.**

**Fonte: Próprio autor.**

Observando agora, apenas os dados acima de 3, visualizamos que, o mais representante é o próprio 3, que possui uma nota maior que a soma das notas 5 e 4, visível na tabela 2.

	Número de respostas	Porcentagem total
Total de respostas	36	100%
Acima de 3	29	81%
Abaixo de 3	7	19%
nota 5	nota 4	nota 3
7	7	15

**Tabela 2 – número de notas contabilizadas.**

**Fonte: Próprio autor.**

Isto é, o jogo possui pontos bons, mas em sua maioria são pontos específicos, e como já visualizado nos exame anterior, os gráficos são o grande poder do jogo, que mais cativou os usuários, muito auxiliado e impulsionado pela *game engine* utilizada, a Unreal

Engine 4, que entrega uma gama grande de detalhes para aprimoramento gráfico, como texturas, efeitos de luz, ambientação e materiais, que tornam o jogo com uma estética robusta.

## **5. Considerações finais**

A produção do jogo atingiu as expectativas, utilizando em sua grande maioria, ferramentas gratuitas que entregam o necessário para o desenvolvimento do jogo. As avaliações feitas sobre o jogo, mostra que em geral, o jogo tem potencial de entretenimento, porém, possui aspectos que precisam ser reparados para torna-lo mais qualificado. Um estudo aprofundado de *design* de menus tornaria mais intuitivo e agradável ao jogador. De todo modo, os resultados da avaliação, expressa que com essas tecnologias, conseguimos desenvolver um jogo eletrônico. Ao final, demonstramos ferramentas que podem ser usadas para a criação de *games*, de forma que possa gerar um game interessante para o público.

Esse artigo deixa em aberto para análise futura, possíveis ferramentas para a produção de um jogo, de forma 100% gratuita, com *softwares* livres. Outro ponto também, é a avaliação mais precisa da indústria de vídeo games no Brasil, qual o âmbito que elas atingem, sendo apenas local ou globalizado.

## 7. Referências

AGUIAR, Bernardo; CORREIA, Walter; CAMPOS, Fábio. Uso da Escala Likert na Análise de Jogos. **Anais do X Simpósio Brasileiro de Games e Entretenimento Digital, 07-09 de novembro de 2011 Salvador**, 2011.

DE MACEDO, Daniel Valente; RODRIGUES, Maria Andréia Formico; SERPA, Yvens Rebouças. Desenvolvimento de Aplicações Gráficas Interativas com a Unreal Engine 4. **Revista de Informática Teórica e Aplicada**, v. 22, n. 2, p. 181-202, 2015.

DIAS, Raphael. **Mapa da Indústria de Jogos**. [S. l.], 2019. Disponível em: <http://mapadaindustriadejogos.com.br/>. Acesso em: 3 jun. 2019.

Epic Games. **What is Unreal Engine 4**. United States, [2019]. Disponível em: <<https://www.unrealengine.com/en-US/>>. Acessado em: 01 maio 2019.

FAGERHOLT, Erik; LORENTZON, Magnus. Beyond the HUD-user interfaces for increased player immersion in FPS games. 2009.

HASSE, Rodrigo Marcondes et al. Criação de um Modelo-Base de Rigging 3D com Movimento Muscular Realista para Animações. 2019.

HIRA, Willian Kenji et al. Criação de um modelo conceitual para Documentação de Game Design. **XV Simpósio Brasileiro de Jogos e Entretenimento Digital**, p. 329-336, 2016.

ITCH CORP. Frequently Asked Questions. San Francisco, 2019. Disponível em: <https://itch.io/docs/general/faq#why-was-itchio-made>. Acesso em: 19 ago 2019.

LUCCHESI, Fabiano; RIBEIRO, Bruno. **Conceituação de jogos digitais**. Sao Paulo, 2009.

NACKE, Lennart. In: NACKE, Lennart. **Game Industry Sales Data**. [S. l.], 18 jan. 2009. Disponível em: <http://www.acagamic.com/research/stats/game-industry-sales-data/>. Acesso em: 3 ago. 2019.

NEWZOO. **Brazil Games Market 2018**. [S. l.], 6 jul. 2018. Disponível em: <https://newzoo.com/insights/infographics/brazil-games-market-2018/>. Acesso em: 16 maio 2019.

NIELSEN, Jakob. Nielsen norman group. **Internet**: <https://www.nngroup.com>, v. 24, 2016.

PWC. Getting Personal: Putting the me in entertainment and media. **Perspectives from the Global Entertainment & media Outlook 2019 - 2023**. Estados Unidos, 2019. Disponível em: <https://www.pwc.com/gx/en/industries/tmt/media/outlook.html>. Acesso em: 14 ago. 2019.

SAJINA, Robert; OREHOVACKI, Tihomir. User experience evaluation of 2D side-scrolling game developed using Overlap2D game editor and LibGDX game engine. In: **2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)**. IEEE, 2018. p. 1580-1585.

VALCASARA, Nicola. **Unreal Engine Game Development Blueprints**. Packt Publishing Ltd, 2015.