



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

THOMAZ DINIZ PINTO DE MORAIS

**ANIMA: UMA TÉCNICA PARA CRIAR ANIMAÇÕES A PARTIR
DE IMAGENS ESTÁTICAS EM JOGOS**

CAMPINA GRANDE - PB

2021

THOMAZ DINIZ PINTO DE MORAIS

**ANIMA: UMA TÉCNICA PARA CRIAR ANIMAÇÕES A PARTIR
DE IMAGENS ESTÁTICAS EM JOGOS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientador: Professor Dr. Marcelo Alves de Barros

CAMPINA GRANDE - PB

2021



M827A Morais, Thomas Diniz Pinto de.

Anima: uma técnica para criar animações a partir de imagens estáticas em jogos. / Thomas Diniz Pinto de Morais. - 2021.

11 f.

Orientador: Prof. Dr. Marcelo Alves de Barros.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Jogos digitais. 2. Computação gráfica - animações. 3. Animações digitais. 4. Imagens estáticas em jogos. 5. OpenGL Shading Language - GLSL. 6. Application Programming Interface. 7. Game Maker: Studio. 8. Estilo gráfico - jogos digitais. 9. Shader de distorção. 10. Animação de imagens - técnica. I. Barros, Marcelo Alves de. II. Título.

CDU:004.92(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

THOMAZ DINIZ PINTO DE MORAIS

**ANIMA: UMA TÉCNICA PARA CRIAR ANIMAÇÕES A PARTIR
DE IMAGENS ESTÁTICAS EM JOGOS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

**Professor Dr. Marcelo Alves de Barros
Orientador – UASC/CEEI/UFCG**

**Professora Dr. Franklin de Souza Ramalho
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em 25 de maio de 2021.

CAMPINA GRANDE - PB

ABSTRACT

One of the challenges in creating games is to make the graphics of a game as attractive as possible for its players. Thus, to make a product successful, it is necessary to invest time and resources to make the graphics attractive to potential players. A common API (Application Programming Interface) in games is OpenGL [1] which is used to render 2D and 3D graphics. Developers interact with the GPU using OpenGL functions and GLSL[1] (OpenGL Shading Language) commands. In this work, we create and describe a distortion technique to transform still images into animations, focusing on simulating the effect of imperfections in hand-made animations. In addition, we evaluated the technique through a survey using a modified version of a commercial game we are developing and found that 96.7% of respondents prefer images with the technique being applied and 93.3% responded that they feel more confident when drawing with the technique being applied.

Anima: Uma Técnica para Criar Animações a partir de Imagens Estáticas em Jogos

Thomaz Diniz

thomaz.morais@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

Marcelo Alves

mbarros@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

RESUMO

Um dos desafios na criação de jogos digitais é tornar os gráficos de um jogo o mais atrativo possível para os seus jogadores. Desta forma, para fazer um produto de sucesso, é necessário investir tempo e recursos para tornar os gráficos atrativos aos potenciais jogadores. Uma API (Application Programming Interface) comum em jogos é a OpenGL [1] que é utilizada para renderizar gráficos em 2D e 3D. Desenvolvedores interagem com a GPU usando as funções da OpenGL e os comandos da GLSL [1] (OpenGL Shading Language). Neste trabalho, criamos e descrevemos uma técnica de distorção para transformar imagens estáticas em animações, focando em simular o efeito das imperfeições de animações feitas à mão. Além disso, avaliamos a técnica através de uma pesquisa utilizando uma versão modificada de um jogo comercial que estamos desenvolvendo e descobrimos que dos entrevistados 96.7% preferem imagens com a técnica sendo aplicada e 93.3% responderam que se sentem mais confiantes ao desenharem com a técnica sendo aplicada.

Palavras-chave

Jogos digitais, Computação Gráfica, Shaders

ABSTRACT

One of the challenges in creating games is to make the graphics of a game as attractive as possible for its players. Thus, to make a product successful, it is necessary to invest time and resources to make the graphics attractive to potential players. A common API (Application Programming Interface) in games is OpenGL [1] which is used to render 2D and 3D graphics. Developers interact with the GPU using OpenGL functions and GLSL[1] (OpenGL Shading Language) commands. In this work, we create and describe a distortion technique to transform still images into animations, focusing on simulating the effect of imperfections in hand-made animations. In addition, we evaluated the technique through a survey using a modified version of a commercial game we are developing and found that 96.7% of respondents prefer images with the technique being applied and 93.3% responded that they feel more confident when drawing with the technique being applied.

Keywords

Games, Computer Graphics, Shaders

1. INTRODUÇÃO

Jogos eletrônicos fazem parte da cultura popular e são um mercado bastante promissor. Segundo a Newzoo, a tendência é que o mercado de jogos arrecade 174 bilhões de dólares somente em 2020 [4]. Mas, apesar da indústria de jogos ser bastante lucrativa, o custo de desenvolvimento tem se tornado bastante elevado, podendo ultrapassar a casa dos 50 milhões de dólares [2].

Um obstáculo que eleva o custo de desenvolvimento é o imenso esforço necessário para criar arte que seja atrativa aos jogadores. Alguns jogos simulam arte feita à mão para estilização de sua arte. Exemplos dessa estilização são os jogos Hollow Knight [6], Cuphead [7], e Drawn to Death [8]. O ciclo de desenvolvimento de tais jogos pode chegar a até mesmo 5 anos e grande parte do esforço é voltado justamente para a arte [3].

Neste contexto, existem programas, chamados de shaders, capazes de realizar uma série de funções especializadas para alterar imagens e recursos visuais [1]. Shaders são utilizados para reduzir o esforço manual na criação de arte. Atividades como iluminação, efeitos visuais e pós-processamento de imagens são comumente feitas de maneira automática pelo computador utilizando shaders.

Este projeto descreve a criação do Anima, um shader que simula as imperfeições de animações feitas à mão a partir de imagens estáticas. Estas imperfeições são o resultado de quando um desenhista tenta desenhar múltiplas vezes o mesmo desenho para uma animação e as imperfeições de tentar desenhar o mesmo frame acabam sendo visíveis ao olho humano. O que parece um defeito, a priori, pode ser visto como uma escolha artística visual. Estas imperfeições podem ser observadas em alguns desenhos animados, em filmes e até mesmo em jogos (e.g. Gertie The Dinosaur [9], propagandas da Red Bull [10], Drawn to Death [8], Waking Life [11], skribbl.io [12], OMORI [13], Later Alligator [14], Tangle Tower [15], Wuppo [16], Ittle Dew [17]).

O shader Anima está sendo desenvolvido para *Drawgaem* [18], um jogo comercial que estamos produzindo onde os jogadores também desenham dentro do jogo. O jogo está em fase beta e com lançamento previsto para agosto de 2021. O Anima poderá tanto ser utilizado para melhorar as artes visuais do jogo como para melhorar os desenhos que os nossos jogadores desenharão durante uma partida do jogo. Nosso objetivo é que a nossa técnica possa ser utilizada para reduzir o esforço e custo de desenvolvimento de arte de certos tipos de jogos e animações. Além disso, esse método pode tornar os jogos mais atrativos. Para avaliarmos nossa técnica, fizemos uma pesquisa com um total de 30 jogadores.

Avaliamos a técnica fazendo um roteiro para os jogadores em uma versão modificada do jogo e um formulário de pesquisa.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção descrevemos técnicas e aplicações utilizadas neste projeto.

2.1 OPENGL, GLSL E SHADERS

OpenGL é uma API utilizada pela indústria para criação de aplicações gráficas. Desta forma, sua função é a de renderizar imagens, ou seja transformar dados que vem da aplicação em algo visível pela tela do computador, videogame ou dispositivo móvel. Aplicações que necessitam de processamento gráficos (e.g. simuladores, jogos digitais e visualizadores de dados) utilizam OpenGL ostensivamente para renderizar seus gráficos, imagens e efeitos visuais. [5]

Para isto, existe a GLSL uma linguagem de domínio específico onde o desenvolvedor é capaz de ter controle sobre a GPU para fazer pequenos programas gráficos que podem gerar, alterar e renderizar imagens. Estes programas são chamados de Shaders [1]. Comumente um shader é composto de vertex shaders (shader que manipula os vértices de um determinado modelo ou imagem) e fragment shader (shader que manipula a imagem pixel a pixel). No contexto deste trabalho, estaremos trabalhando com fragment shader para fazer o efeito de distorção ondular.

2.2 GAME MAKER: STUDIO 2(GMS2)

Para implementar nossa técnica, utilizamos GMS2 que é uma ferramenta de desenvolvimento de jogos da Yoyogames. Para o desenvolvimento da lógica dos jogos ela utiliza uma linguagem própria chamada de game maker language (gml). Nela é possível criar jogos em 2D ou 3D, mas o foco principal desta ferramenta é no desenvolvimento de jogos em 2D. Neste trabalho, focamos nas ferramentas de desenvolvimento de shaders. Dentro do GMS2 há um módulo de shaders que também utiliza a DSL apropriada para desenvolvimento de shaders (GLSL).

2.3 ESTILO GRÁFICO EM JOGOS DIGITAIS

Existem diversos métodos de colocar gráficos na tela de jogos que afetam a maneira como o conteúdo dos gráficos dos jogos digitais são mostrados na tela. Alguns jogos podem ser em 2 dimensões, utilizando imagens e desenhos feitos em softwares, outros podem ser em 3 dimensões, utilizando técnicas como cel shading [21] que trazem uma estilização um pouco mais cartunesca, enquanto outros tentam fazer uma estilização mais realística. Estes diferentes estilos podem ser conhecidos como estilização gráfica ou estilo gráfico. [20]

O estilo gráfico em jogos digitais é uma parte importante no desenvolvimento de jogos. Segundo Keo, as publishers afirmam que os gráficos de jogos são a parte mais importante no marketing dos jogos uma vez que os gráficos são de maneira literal a primeira impressão que o jogador terá com o jogo. Por tanto, é essencial que os desenvolvedores escolham um estilo gráfico que melhor se encaixe com a proposta do jogo [20]. No contexto deste trabalho estamos lidando com jogos que querem simular especificamente o estilo gráfico de animações e artes feitas a mão.

3. ARQUITETURA DA SOLUÇÃO

Nesta seção explicamos o desenvolvimento da técnica para simular o efeito de animações feitas à mão. Ela está dividida nas etapas por ordem de implementação. A técnica é composta por duas etapas distintas: a etapa de distorção de imagem e a etapa de criação da animação.

Na seção 3.1 discutimos sobre o shader de distorção ondular, que é responsável por fazer a etapa de distorção da imagem. Demonstramos a implementação do shader, juntamente com alguns exemplos práticos do shader em ação, bem como problemas que podem vir a aparecer se utilizarmos esta técnica em algumas situações. Em seguida, na seção 3.2 demonstramos sobre como fazer para transformar a imagem distorcida em animação através da ativação, desativação do shader e variação dos parâmetros utilizados.

3.1 SHADER DE DISTORÇÃO

Como queríamos desenvolver uma técnica para distorcer sutilmente imagens, nossa ideia então foi aplicar um shader de distorção ondular para gerar imagens levemente distorcidas com a intenção de simular o efeito de imperfeições de desenhos feitos à mão. Escolhemos então em desenvolver um shader de distorção ondular, pela facilidade e praticidade de desenvolvimento.

```
1 varying vec2 v_vtexcoord;
2 varying vec4 v_vColour;
3 uniform float amount;
4 uniform float distortion;
5
6 void main(){
7     vec2 uv = v_vtexcoord;
8     uv.x = uv.x+cos(uv.y*amount)/distortion;
9     uv.y = uv.y+sin(uv.x*amount)/distortion;
10    gl_FragColor = cv_vColour*texture2D(gm_BaseTexture,uv);
11 }
```

Quadro 1 - Fragment Shader de distorção ondular

O código para executar esta distorção pode ser visto no Quadro 1. Esta é uma versão modificada de um shader distorção ondular de imagem. Na linha 8 o shader recebe a textura que queremos desenhar e no eixo x deslocamos o pixel utilizando uma função cossenoidal recebendo como parâmetros a posição y e que aquele pixel se encontra multiplicado por um parâmetro recebido pelo shader chamado de "amount" que é responsável por definir quantas ondas teremos naquela textura; O valor da função cossenoidal então é dividido pelo parâmetro "distortion" que vai ser responsável por definir o quão alta ou baixa será a onda de distorção. O mesmo processo é então feito na linha 9 só que desta vez utilizando uma função senoidal e o eixo x para definir o eixo y. Ou seja, ele causa um efeito de distorção ondular tanto vertical quanto horizontal.

Para exemplificar de maneira mais prática, definimos parâmetros que consideramos bons para a utilização do shader. Gostaríamos que o shader fosse utilizado somente para fazer uma leve distorção, então definimos um "amount" igual a 51.2 e um "distortion" = 643.2. Criamos uma imagem de telha para facilitar a visualização do que acontece quando aplicamos o shader em uma imagem e o resultado pode ser visto na diferença entre a figura 1 (sem o shader) e na figura 2 (com o shader).

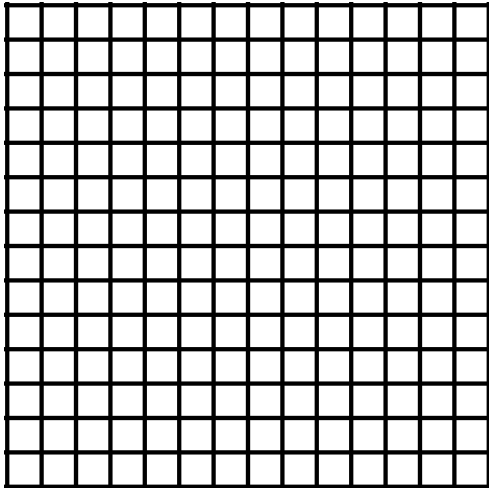


Figura 1 - Exemplo de imagem de telha sem o shader

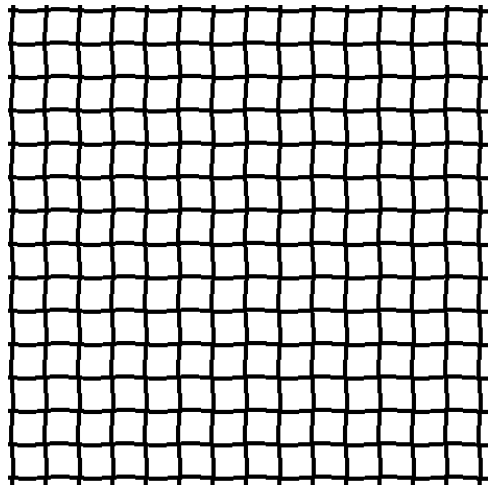


Figura 2 - Exemplo de imagem de telha com o Shader

Comparando a figura 1 com a figura 2, linhas que antes eram retas agora estão sendo distorcidas de maneira ondular (devido a aplicação da função de distorção cossenoidal no eixo x e senoidal no eixo y).

Testando com jogadores, percebemos que algo que poderia causar estranheza com os jogadores é o fato de que pode ficar claro que o desenho está sendo distorcido de maneira ondular, ou seja, parte do nosso desafio para buscar valores que consideramos ideais era para que a imagem fosse distorcida, mas não ficasse óbvio o padrão de distorção aos jogadores. Por isso tentamos deixar o efeito o mais sutil possível. Vale ressaltar que este padrão de distorção não é facilmente perceptível se o desenho não for completamente alinhado com o eixo x ou com o eixo y. Na figura 3 (sem o shader), temos o desenho de uma paisagem com sol, grama, nuvens, flores e um personagem. Já na figura 4 (com o shader), este efeito ondular só é claramente perceptível na linha de horizonte da grama, no restante da imagem, acaba parecendo uma distorção aleatória. Isto ocorre exatamente porque a linha de horizonte está alinhada com o eixo x da imagem.

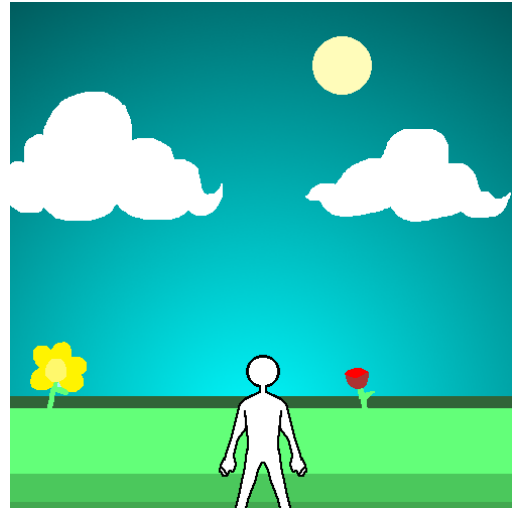


Figura 3 - Exemplo de desenho sem o Shader

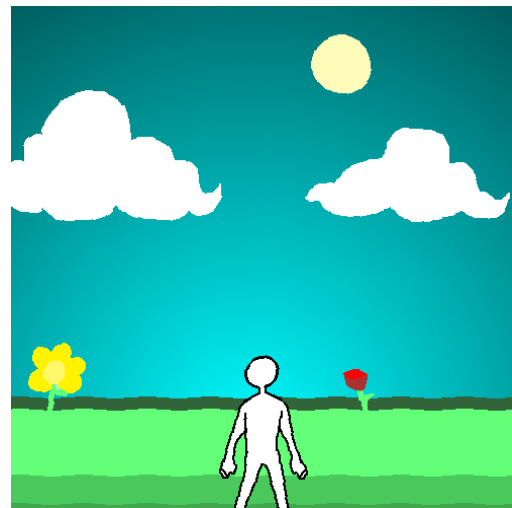


Figura 4 - Exemplo de desenho com o shader

3.2 A TÉCNICA DE ANIMAÇÃO

Podemos a partir do shader derivar uma técnica de animação de imagens estáticas, a técnica consiste em ativar e desativar o shader em intervalos regulares. Sendo assim, a partir de testes empíricos onde variamos o tempo de alternância até encontrarmos um valor de alternância que consideramos bom, definimos que a cada 0.2 segundo o efeito alternaria entre desativado e ativado. O que notamos ao fazer isto é que antes tínhamos uma imagem, mas agora temos uma animação com duas imagens (a imagem com o efeito ativado e com o efeito desativado). Percebemos que a alternância com o efeito ondular é muito constante e, por isso, decidimos variar os valores que havíamos definido. Definimos valores aleatórios de "amount" para pertencer a valores entre $[51-10, 51+10]$ e "distortion" entre $[643-100, 643+200]$. O código referente a definição dos parâmetros, bem como momentos de ativação e desativação do shader, podem ser observados no Quadro 2.


```

1 function dg_shader_start() {
2   var time = current_time/200;
3   var effect = (round(time) mod 2);
4   if (effect && !global.dg_shader_activated){
5     global.dg_shader_activated=1;
6     global.amount = 51 + random_range(-10,10);
7     global.distortion = 643 + random_range(-100,200);
8   }
9   if (!effect && global.dg_shader_activated)
10    global.dg_shader_activated=0;
11 }
12 if (effect && global.shader) {
13   shader_set(shd_wave);
14   var amount = shader_get_uniform(shd_wave,"amount");
15   var distortion = shader_get_uniform(shd_wave,"distortion");
16   shader_set_uniform_f(amount,global.amount);
17   shader_set_uniform_f(distortion,global.distortion);
18 }
19 }

```

Quadro 2 - Código da função responsável pela alternância de parâmetros e ativação e desativação do shader.

No Quadro 2, temos a função `dg_shader_start()` que é responsável por definir que o shader está ativo para desenhar na tela. Na linha 2 temos a definição da variável `time` que é definida como `current_time` (variável global do GMS2 responsável por pegar o tempo atual em milissegundos) dividida por 200. Na linha 3 definimos, a flag `effect` que é responsável por definir quando o shader será aplicado. Ele é aplicado toda vez que a variável `time` arredondada for divisível por 2. Isso fará a alternância acontecer a cada 0.2 segundos. Definimos então para cada momento de ativação nas linhas 6 e 7 os valores dos parâmetros `amount` e `distortion`. Na linha 13, o shader é ativado (`shd_wave` é uma referência para o shader descrito no quadro 1). Em seguida, passamos os parâmetros `amount` e `distortion` para o shader. Desta forma, em tudo que for desenhado a partir da chamada desta função será aplicado o shader de distorção ondular, se o código de alternância permitir.

```

1 function dg_shader_stop() {
2   if (shader_current() != -1)
3     shader_reset();
4 }

```

Quadro 3 - Função responsável por desativar o efeito de distorção do shader

Ao finalizar a chamada, temos a função de desativação do shader, para que outros elementos do jogo possam ser desenhados sem a utilização deste efeito.

O resultado da técnica, exemplo de códigos, imagens e vídeos estão publicados em um repositório no github¹ em código aberto.

¹ <https://github.com/ThomazDiniz/AnimaShader>

4. AVALIAÇÃO

Com o intuito de testar se a técnica melhora as artes visuais do jogo e se ela pode trazer personalidade aos desenhos dos jogadores de *Drawgaem*, decidimos fazer uma pesquisa de avaliação tanto para saber se o shader era aprovado de maneira geral como para saber se os jogadores se sentem mais confiantes desenhando com a nossa técnica.

Nesta seção temos a subseção questões de pesquisa, onde definimos o que queremos responder com a avaliação feita; Composição das amostras, onde definimos qual é a população amostral da nossa pesquisa, formulário de avaliação onde mostramos as perguntas feitas e resultados e discussões, onde demonstramos os resultados da avaliação e discutimos sobre eles.

Nossa avaliação foi feita utilizando uma pesquisa no modelo quali-quantitativa por meio de uma entrevista estruturada. Fizemos uma entrevista com questões abertas e fechadas tentando descobrir se os jogadores gostaram da técnica aplicada e o quanto eles gostaram. Para isto, utilizamos a escala de likert para avaliar o quanto os jogadores preferem nossa técnica e o quanto os jogadores se sentiram mais confiantes utilizando a nossa técnica [19].

4.1 QUESTÕES DE PESQUISA

Nossa pesquisa sobre aprovação da técnica está sendo feita para avaliar se os nossos jogadores acham que a técnica melhora os gráficos do nosso jogo. A técnica encontra-se em fase de estudo e neste relatório estamos fazendo um estudo preliminar sobre ela. Por isso, nossa avaliação visou responder às seguintes questões de pesquisa (QP):

- QP1: Comparando imagens aplicando a técnica e sem aplicar a técnica, os jogadores acreditam que a técnica melhora a arte visual do jogo?
- QP2: Os jogadores se sentem mais confiantes desenhando com esta técnica aplicada em seus desenhos?

Além disso, gostaríamos de coletar feedback e ideias para melhorar a nossa técnica. Então pretendemos, também, dedicar parte desta avaliação para coleta de possíveis falhas e melhoramentos da nossa técnica.

4.2 COMPOSIÇÃO DA AMOSTRA

Para compor a amostragem desta avaliação, fizemos um levantamento de todos os jogadores que estão testando o jogo. Atualmente temos um total de 127 jogadores. Definimos como população da nossa pesquisa 30 usuários de *Drawgaem* escolhidos aleatoriamente para ter uma amostra representativa e não enviesada da população de jogadores atuais.

4.3 PROCEDIMENTO DA PESQUISA

Para responder às questões de pesquisa, utilizamos pesquisa explicativa com registro, análise e interpretação de fenômenos que, segundo Gil [22], valem-se quase exclusivamente de método experimental. Utilizamos como variáveis de pesquisa a preferência do usuário, a qualidade percebida das imagens criadas e a confiança declarada pelo usuário sobre os seus desenhos. Utilizamos de um método experimental, utilizando-se de observação a partir de experimentos controlados com mensuração de variáveis e coleta de opiniões bem como análise estatísticas dos

dados. Além disso, fizemos uma análise quantitativa para as questões objetivas dos formulários de pesquisa e análise qualitativa das questões abertas.

Portanto, desenvolvemos um roteiro onde compartilhamos com os usuários uma versão modificada de Drawgaem. Nesta versão, temos uma conversa inicial com o jogador explicando a técnica enquanto mostramos imagens lado a lado com e sem o efeito aplicado, em seguida pedimos para o usuário desenhar algo que ele queira (e.g. uma árvore, uma pessoa, um objeto). Este primeiro desenho é demonstrado sem a aplicação da técnica. Ao finalizar o desenho, pedimos para que o usuário refaça o desenho, mas desta vez ele será mostrado na tela com a técnica que desenvolvemos.

Ao finalizar o roteiro, pedimos para que os usuários preencham um formulário de avaliação da técnica empregada. Utilizamos o Google Forms (serviço de formulários do google) por ser gratuito e pela facilidade de construir, distribuir e analisar os formulários. Fizemos 5 questionamentos relacionados às imagens demonstradas na aplicação. Todo o processo está resumido no fluxograma da figura 4.

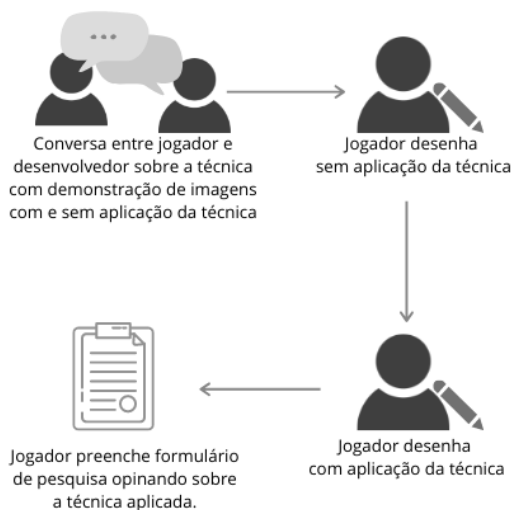


Figura 4- Fluxograma do roteiro do procedimento de pesquisa

Para responder a Questão de pesquisa 1, fizemos dois questionamentos aos jogadores:

1. Você prefere as figuras demonstradas com a técnica aplicada ou sem a técnica aplicada?
2. Você acredita que as imagens com aplicação da técnica são muito piores, piores, indiferentes, melhores ou muito melhores que imagens sem a técnica aplicada?

Enquanto a primeira pergunta foca em saber se o jogador prefere com ou sem a aplicação da nossa técnica, a segunda pergunta é para entendermos o quanto os jogadores gostam da nossa técnica.

Para responder a questão de pesquisa 2, fizemos o seguinte questionamento aos usuários:

1. O quanto você se sentiu mais confiante com seus desenhos com esta técnica aplicada?

Por fim pedimos um feedback e ideias em uma caixa de texto opcional e sem limites de caracteres.

4.4 Resultados e Discussões

Em geral tivemos resultados positivos com a avaliação dos nossos jogadores.

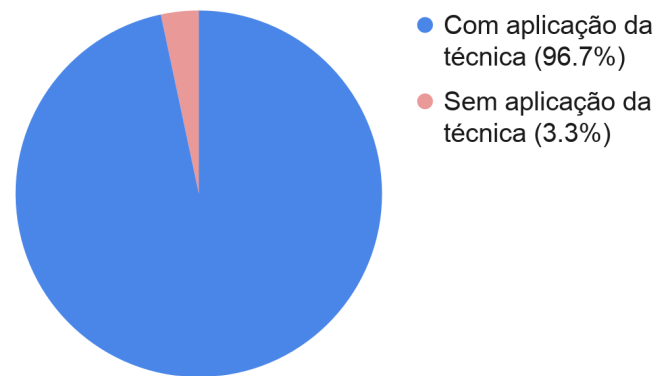


Figura 6 - Você prefere as figuras demonstradas com a técnica aplicada ou sem a técnica aplicada?

Como é possível ver na figura 6, avaliamos se os jogadores preferem as figuras com a técnica sendo aplicada ou sem a técnica ser aplicada. A maioria dos jogadores preferiram as figuras com a utilização da técnica. Isso indica para a gente que esta técnica pode ser utilizada para estilizar o jogo e trazer um diferencial artístico em relação a outros jogos que tentam simular a mesma estilização gráfica.

Avaliamos também o quanto os jogadores preferem imagens com a técnica utilizada. O resultado desta avaliação pode ser visto na figura 7, onde a maioria dos usuários acharam muito positiva a utilização da técnica, mas ainda assim tivemos um jogador que achou que a técnica piorou muito e outro que achou indiferente.

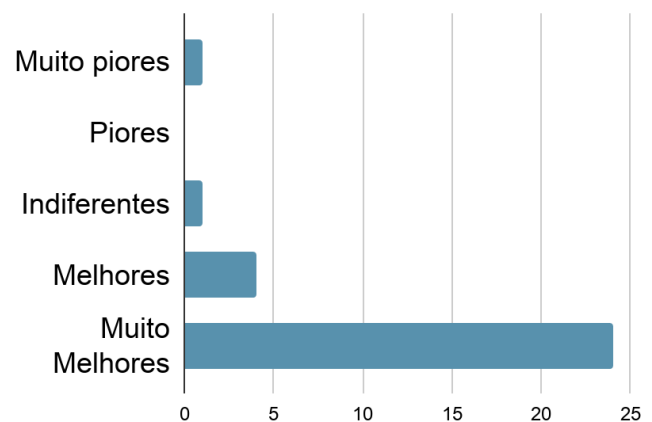


Figura 7 - Escala de Likert da questão: "Você acredita que as imagens com aplicação da técnica são muito piores, piores, indiferentes, melhores ou muito melhores que imagens sem a técnica aplicada?"

Desta forma podemos responder a primeira questão de pesquisa, onde tivemos somente 1 dos 30 jogadores afirmando que a técnica não é positiva e 2 dos 30 jogadores afirmando que as imagens

com a técnica são muito piores ou indiferentes das imagens sem a técnica. Em outras palavras, 96.7% dos jogadores entrevistados preferem as imagens com a técnica aplicada e 93.3% dos entrevistados acreditam que a técnica melhora os gráficos do jogo.

QP1: Comparando imagens com e sem a técnica sendo aplicada, os jogadores acreditam que a técnica melhora a arte visual do jogo? Dos entrevistados, 96.7% dos jogadores preferem as imagens com a técnica sendo aplicada e que 93.3% dos entrevistados acreditam que a técnica melhora os gráficos do jogo.

Fizemos também uma análise sobre o quanto os jogadores gostam dos desenhos que eles próprios fizeram com a técnica aplicada em seus desenhos. Acreditamos que como a nossa técnica poderia dar uma estilização artística para imagens, talvez os jogadores se sintam mais confiantes em desenhar com a técnica sendo aplicada, portanto questionamos também os jogadores sobre o seu nível de confiança ao desenhar com a técnica aplicada.

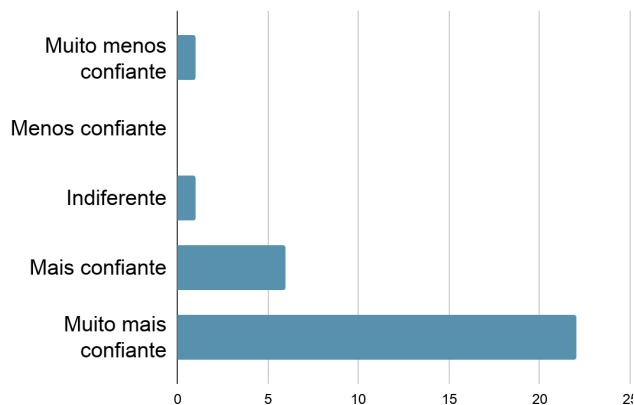


Figura 8 - Escala de Likert da questão: “O quanto você se sentiu confiante desenhando com esta técnica sendo aplicada em relação a desenhar sem a técnica ser aplicada?”

Na figura 8 é possível observar a avaliação que os jogadores têm sobre o quão mais confiantes eles se sentem utilizando esta técnica. Em geral, os jogadores se sentiram muito mais confiantes, tendo 93.3% dos jogadores se sentindo mais confiantes desenhando com a técnica sendo aplicada em seus desenhos. Somente um jogador se sentiu muito menos confiante e outro jogador se sentiu indiferente desenhando com a técnica aplicada.

QP2: Os jogadores se sentem mais confiantes desenhando com esta técnica aplicada em seus desenhos? Dos entrevistados, 93.3% dos jogadores se sentiram mais confiantes com seus desenhos quando a técnica estava sendo aplicada.

Obtivemos diversos feedbacks, em sua grande maioria foram bastante positivos, mas alguns usuários deram sugestões e melhoramentos que consideramos relevantes.

Destacamos dois feedbacks que são relativos aos jogadores que não gostaram da técnica. Sendo o jogador 1 aquele que classificou a técnica como muito pior e que a técnica o deixava muito menos confiante com o seu desenho e o jogador 2 aquele que classificou a técnica como indiferente.

Jogador 1: “Não acho que esta técnica adicione muito pro jogo. Quando desenhei não conseguia acompanhar meu desenho da maneira correta. As linhas entortavam no meio do meu desenho e isto me deixou confuso. Recomendo que se for utilizar esta técnica no jogo, coloque uma opção para desativá-la nas opções do jogo.”

Jogador 2: “Achei o efeito da técnica muito forte. Acho que vocês poderiam colocar uma opção para controlar a intensidade do efeito.”

Apesar de negativas, ambos os comentários são bem relevantes. Gostaríamos que a técnica agradasse a toda a nossa base de jogadores, mas a possibilidade de adicionar uma opção para regular a intensidade do efeito ou, até mesmo, desativá-lo por completo foram sugestões muito boas para agradarmos o público que não gostou da técnica.

Além disto, dos jogadores que gostaram da técnica e avaliaram que a técnica os deixaram muito mais confiantes com seus desenhos e que a técnica deixa os gráficos do jogo muito melhores, destacamos um comentário de um jogador:

Jogador 3: “Gostei bastante. Uma ideia interessante para vocês é tentar aplicar outros shaders junto com esse. Existe um shader (Acho que chama sketch) faz parecer com rabiscos. Deve ficar legal.”

Este comentário foi uma sugestão de um desenvolvedor de jogos que nos deu a ideia bastante relevante de tentar melhorar a técnica utilizando-a em conjunto com outros efeitos que possam melhorar a estilização gráfica do nosso jogo.

5. CONCLUSÕES

Neste artigo descrevemos o desenvolvimento de uma técnica para, a partir de uma imagem estática, simular os efeitos de imperfeições de animações feitas à mão utilizando um shader de distorção e alternância de parâmetros passados por este shader. Portanto, acreditamos que esta técnica possa ser utilizada exatamente para melhorar ou facilitar a adoção de um estilo gráfico que tente simular animações e desenhos feitos a mão.

Avaliamos o shader em um contexto real utilizando um jogo que estamos desenvolvendo (*Drawgaem*) e descobrimos que 93.3% dos entrevistados acharam que a técnica melhora os gráficos do jogo. E que 93.3% dos jogadores entrevistados se sentiram mais confiantes com seus desenhos ao desenharem com a técnica sendo aplicada. Apesar disso, este trabalho trata de um estudo preliminar e deverá ser aprofundado por meio de uma avaliação mais detalhada e com uma quantidade maior de pessoas para avaliar a técnica.

Além disso, apesar de termos feito um trabalho interessante para poder simular o efeito de animações feitas à mão, acreditamos que ainda existe muito trabalho a ser feito nessa área, como por

exemplo buscar por outras técnicas de distorção de imagens para aplicação da técnica (e.g displacement shader). É possível que esta técnica misturada com outros shaders (e.g. sketch shader, toon shader, outline shader) possam trazer resultados ainda melhores para as artes, desenhos de jogos e animações.

REFERÊNCIAS

- [1] KESSENICH, John; SELLERS, Graham; SHREINER, Dave. OpenGL programming guide: the official guide to learning OpenGL, Version 4.5 with SPIR-V. Addison-Wesley Professional, 2016.
- [2] Koster, R., 2018. The Cost Of Games. [online] VentureBeat. Disponível em: <<https://venturebeat.com/2018/01/23/the-cost-of-games/>> [Acessado em 24 de março de 2021].
- [3] Polygon. 2017. Cuphead Artists Illustrate The Game'S Five-Year Dev Cycle. [online] Disponível em: <<https://www.polygon.com/features/2017/12/7/16745144/cuphead-artists-illustrate-five-year-development-cycle>> [Acessado em 24 de março de 2021]
- [4] Wijman, T., 2020. Global Game Revenues Up An Extra \$15 Billion This Year As Engagement Skyrockets | Newzoo. [online] Newzoo. Disponível em: <<https://newzoo.com/insights/articles/game-engagement-during-covid-pandemic-adds-15-billion-to-global-games-market-revenue-forecast/>> [Acessado em 24 de março de 2021]
- [5] Rost, R. J., Licea-Kane, B., Ginsburg, D., Kessenich, J., Lichtenbelt, B., Malan, H., & Weiblen, M. (2009). OpenGL shading language. Pearson Education.
- [6] Hollowknight.com. 2021. Hollow Knight – An atmospheric adventure through a surreal, bug-infested world. [online] Disponível em: <<https://hollowknight.com/>> [Acessado em 24 de março de 2021]
- [7] Cupheadgame.com. 2021. Cuphead: Don't Deal With The Devil | Available on Xbox One - Windows 10 - Nintendo Switch – PlayStation 4 - Steam - GOG - Mac. [online] Disponível em: <<http://cupheadgame.com/>> [Acessado em 24 de março de 2021]
- [8] IGN. 2021. Drawn to Death Review - IGN. [online] Disponível em: <<https://www.ign.com/articles/2017/04/08/drawn-to-death-review>> [Acessado em 24 de março de 2021]
- [9] IMDb. 2021. Gertie the Dinosaur (1914) - IMDb. [online] Disponível em: <<https://www.imdb.com/title/tt0004008/>> [Acessado em 24 de março de 2021]
- [10] Red Bull Cartoon World. 2021. O MUNDO DOS CARTOONS RED BULL. [online] Disponível em: <<https://www.redbull.com/br-pt/cartoons/>> [Acessado em 24 de março de 2021]
- [11] IMDb. 2021. Waking Life (2001) - IMDb. [online] Disponível em: <<https://www.imdb.com/title/tt0243017/>> [Acessado em 24 de março de 2021]
- [12] skribbl - Free Multiplayer Drawing & Guessing Game. 2021. skribbl - Free Multiplayer Drawing & Guessing Game. [online] Disponível em: <<https://skribbl.io/>> [Acessado em 24 de março de 2021]
- [13] OMORI on Steam. 2021. [online] Disponível em: <<https://store.steampowered.com/app/1150690/OMORI/>> [Acessado em 24 de março de 2021]
- [14] Later Alligator. 2021. Later Alligator. [online] Disponível em: <<https://www.lateralligatorgame.com/>> [Accessed 25 March 2021].
- [15] Tangle Tower on Steam. 2021. [online] Disponível em: <https://store.steampowered.com/app/359510/Tangle_Tower/> [Acessado em 24 de março de 2021]
- [16] Wuppogame.com. 2021. Wuppo the Game - Official website. [online] Disponível em: <<https://wuppogame.com/>> [Acessado em 24 de março de 2021]
- [17] Ittle Dew.com. 2021. Ittle Dew 2. [online] Disponível em: <<https://ittledew.com/>> [Acessado em 24 de março de 2021]
- [18] Drawgaem on Steam. 2021. [online] Disponível em: <<https://store.steampowered.com/app/923000/Drawgaem/>> [Acessado em 24 de março de 2021]
- [19] Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert Scale: Explored and Explained. Current Journal of Applied Science and Technology, 7(4), 396-403. <https://doi.org/10.9734/BJAST/2015/14975>
- [20] Keo, Mary. "Graphical Style in Video Games." (2017).
- [21] Luque, R. Reyes. "The cel shading technique." (2012).
- [22] Gil A. C. "Como elaborar projetos de pesquisa" 4a edição editora atlas. (1987)