

VITOR HUGO FAZOLI DA SILVA

DESENVOLVIMENTO DE FERRAMENTA PARA
ILUMINAÇÃO EM IMAGENS BIDIMENSIONAIS DE
MÉDIA E BAIXA RESOLUÇÃO

Orientador: Prof Dr.Gabriel
Marcelino Alves

SÃO JOÃO DA BOA VISTA

2023

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Objetivos	6
1.1.1	Objetivo Geral	6
1.1.2	Objetivos Específicos	6

2	REVISÃO DA LITERATURA	7
2.1	Modelos de Iluminação	7
2.1.1	Reflexão Ambiente	7
2.1.2	Reflexão Difusa	8
2.1.3	Reflexão Especular	9
2.2	Sombreamento	9
2.2.1	Flat Shading	9
2.2.2	Sombreamento de Gourand	10
2.2.3	Sombreamento de Phong	10
2.3	Ferramentas para iluminação de cenários	12
2.3.1	Vulkan	12
2.3.2	OpenGL	12
2.3.3	Engines	12
2.3.3.1	Unity	13
2.3.3.2	Godot	13
2.4	Trabalhos Correlatos	14
2.4.1	Geração De Mapas De Iluminação Baseado Em Topologia Estimada Para Iluminação De Sprites	14
2.4.2	Analysis and Compilation of Normal Map Generation Techniques for Pixel Art	14
2.4.3	CrossSketch: freeform surface modeling with details	14
2.4.4	Segmentation by grouping junctions	15
3	METODOLOGIA	16
3.1	Levantamento de Requisitos	16
3.2	Estudo: Python e suas bibliotecas	17
3.2.1	PyTorch	17
3.2.2	Scikit-image	18
3.2.3	OpenCV	18
3.2.4	Numpy	18
3.3	Estudo: Inteligência Artificial	18
3.3.1	Segmentação por máscara	19
3.3.2	Segmentação por pontos de clique	19
3.3.3	Segmentação por caixa de delimitação	19
3.3.4	Segmentação por texto	19
3.3.5	Segmentação interativa	19
3.3.6	Segmentação automática	19
3.4	Procedimentos e Técnicas	20
3.5	Avaliar vantagens e limitações das ferramentas abordadas	21
3.6	Modelagem da ferramenta	22
3.7	Justificativa das Tecnologias a serem adotadas	23

3.8	Avaliação Qualitativa	24
3.9	Objetivo da Metodologia	24
3.10	Design da Pesquisa	24
3.11	Amostragem	24
3.12	Procedimentos Analíticos	24
3.13	Critérios de Validação e Confiabilidade	24
3.14	Aspectos Éticos	24
3.15	Cronograma e Recursos	24
3.16	Limitações	24
4	ANÁLISE DOS RESULTADOS PRELIMINARES	25
4.1	Cronograma do Trabalho	25
5	CONCLUSÕES PARCIAIS	26
	REFERÊNCIAS	27

RESUMO

A iluminação é de grande importância para pinturas e esboços digitais nos dias de hoje, pois fornece a percepção de volume ao ambiente. Porém, a iluminação ainda é difícil de replicar em uma imagem digital, pois representa algo muito diferente de um ambiente tridimensional, ainda mais se usarmos uma imagem em baixa resolução. Nesse sentido, O objetivo do trabalho é simular a iluminação em imagens digitais de média e baixa resolução por meio do desenvolvimento de uma ferramenta. Para se alcançar o objetivo, pretende-se que a ferramenta considere informações de luz em um espaço tridimensional para simular a iluminação na imagem bidimensional. Portanto o que se espera deste trabalho é que imagens de média e baixa resolução apresentem boa iluminação.

Palavras-chave: Iluminação. Algoritmo. Ambiente. Bidimensional.

1 INTRODUÇÃO

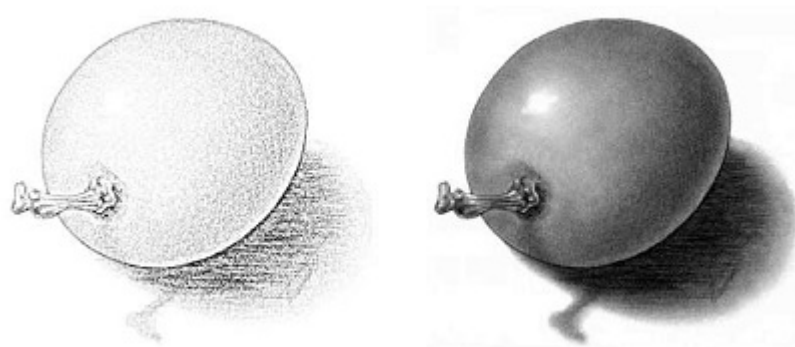
Desde que as pinturas começaram, se formaram vários estilos de arte, ainda que no princípio tudo era voltado a chegar ao realismo das pinturas, por não existir nada que pudesse retratar rostos e momentos melhor do que as artes.

No entanto, isso mudou quando surgiram as câmeras, que poderiam captar tudo de forma praticamente instantânea, com isso era o momento de se pensar que as pinturas iriam sumir, mas ela se renovou e ao invés de buscar o realismo, agora ela estava em busca de trazer novas sensações, como as pinturas abstratas, estilos como cubismo ou surrealismo que usavam de várias curvas e cores para demonstrar expressões (Cezimbra, 2021).

Esses estilos artísticos foram uma revolução no mundo da arte, como o surrealismo por exemplo, que tratava de possibilidades infinitas em suas obras, porque ali tudo era possível e o único limite era imaginação, Nesse período foram apresentadas várias obras que ao olhar não faziam sentido algum, porém era esse o princípio da ideia, pois elas eram criadas para passar uma sensação a quem olhasse, como a de liberdade, entre outros.

Atualmente a iluminação é de extrema importância para as pinturas e desenhos digitais pois é ela quem traz o volume ao ambiente, como pode ser observado na Figura 1.

Figura 1 – Duas imagens demonstrando a diferença da iluminação



(Grafitti, 2019)

Como podemos observar que na figura acima, a imagem da esquerda apresenta apenas um rascunho, sem uso das técnicas de iluminação a portanto a noção de volume e profundidade não fica muito claro, por outro lado a imagem da direita sendo algo mais agradável nas pinturas como é possível notar, além do mais várias composições de cores da luz retratada em uma imagem pode trazer vislumbres fantásticos.

Com a chegada dos primeiros modelos em três dimensões, como cubos, pirâmides e esferas entre outros, foi possível enxergar através de uma tela, um objeto qualquer que

fosse e isso mudou tudo, hoje podemos ver vários exemplos de modelos que está em nossa realidade, como a impressora 3D ou o *Billboard*¹, pois é com ela que foi possível ver a grande beleza dos modelos, é por esse motivo pelo qual atualmente a iluminação é essência para nos causar a sensação de volume, sem pensar nos algoritmos mais robustos como *ray tracing* que segundo Shirley e Morley (2008) retrata, é um algoritmo onde através de uma janela, os raios são direcionados para as imagens. As superfícies são perdidas ou atingidas por cada pixel, que é representado por um raio. Ao atingir uma superfície, o raio se refrata e continua em um novo curso, causando a formação de luzes adicionais nos arredores.

Porém para os artistas chegarem perto dos algoritmos de iluminação que criam visuais impecáveis em três dimensões, as pinturas e artes foram obrigadas a criar um volume e melhorar a iluminação, pois como a luz funcionava de forma sistemática em três dimensões, as artes puderam abrir espaço em uma variação de cores e paletas diferenciadas que trazem um dinamismo maior.

Por isso atualmente a iluminação se tornou algo tão importante para o mundo, que não importa mas o estilo que é usado, é uma parte essencial. É nesse momento em que chegamos ao cerne do problema, pois com essa importância que a iluminação tem sobre a arte e os avanços tecnológicos cada vez mais ligados a ter um *design* rápido e com eficiência, ferramentas que criam a luz em pinturas, desenhos, objetos, *sprites* ou até logos poderia impactar a arte de maneira eficiente e até mesmo para aumentar a comunidade, fazendo com que várias pessoas iniciantes que não conseguem bons resultados possam aparecer no mundo do *design*.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo do trabalho é simular a iluminação em imagens digitais de média e baixa resolução por meio do desenvolvimento de uma ferramenta.

1.1.2 Objetivos Específicos

- Encontrar algoritmos que têm o propósito de iluminar cenas bidimensionais;
- Criar um protótipo e desenvolver a ferramenta para iluminação;
- Realizar testes na ferramenta Relight que será analisado para comparação dos resultados.

¹ apresentado em Tóquio, uma televisão imensa que pode captar imagens como se fossem 3D, criando um aspecto de profundidade pela sua curvatura

2 REVISÃO DA LITERATURA

Este capítulo será organizado de forma a se iniciar com as formas mais primitivas e os modelos de iluminação do mais simples para o mais complexo, logo após mostrar as técnicas de sombreamento e por fim mostrar as ferramentas e trabalhos correlatos ao proposto neste

2.1 Modelos de Iluminação

Segundo Azevedo, Conci e Vasconcelos (2022) o tratamento da luz é um passo essencial para visualizações realistas em ambientes tridimensionais, porém mais do que isso, os processos para chegar a tal são cada vez mais complexos conforme avançam os sistemas de hardware.

Se pensarmos em como funciona a luz no mundo real, é somente através dela que podemos enxergar qualquer coisa, pois mesmo a noite ainda recebemos a luz do sol refletida pela lua, ou seja, tudo o que vemos é somente porque os raios de luz atingem a nossa visão (Manssour; Cohen, 2006).

A princípio, a iluminação de forma geral necessita de uma fonte, algo que possa emitir luz, como por exemplo o sol as estrelas, uma vela ou até mesmo um fósforo. tais fontes criam uma aura ao redor que é conhecida como radiação eletromagnética, e podem emitir entre 380 a 750 nanômetros. Azevedo, Conci e Vasconcelos (2022)

Porém apenas com fontes de luz a cena não se torna interessante, por isso se iniciou a iluminação de forma a ser um ponto tão longe no espaço que ilumina todos os ambientes, e para esse tipo de iluminação surgiram vários modelos.

2.1.1 Reflexão Ambiente

A luz ambiente ou reflexão ambiente é um modelo simplificado para entendermos a luz, ela utiliza padrões da realidade de forma a possibilitar a visualização de cores através da cor do objeto.

Ela "representa um nível constante de luz que define a silhueta dos objetos. Esse nível constante é a simplificação do modelo de iluminação global proposto por Goral et al. (1984) que considera múltiplas reflexões da luz nos diversos objetos que compõe a cena"(Scalco, 2012, p. 150)

De forma a esclarecer graficamente a reflexão ambiente é possível observar o modelo

sendo renderizado¹ através da seguinte Figura 2.

Figura 2 – Renderização usando apenas a componente ambiente.



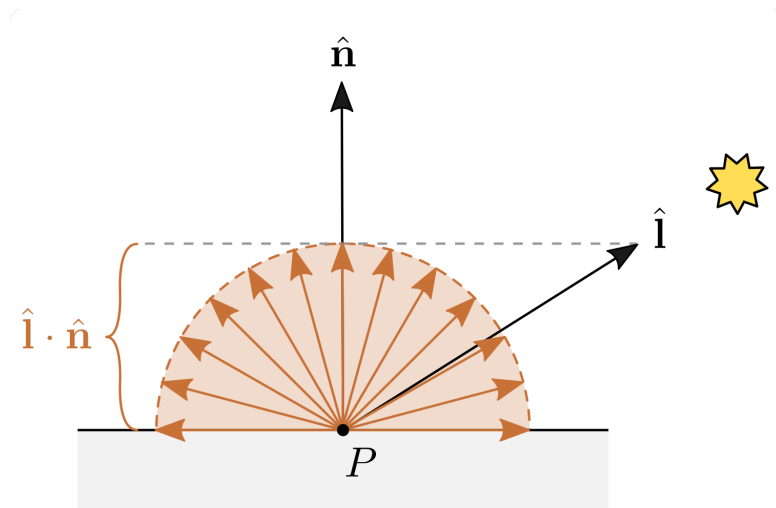
Fonte: (Batagelo, 2021)

2.1.2 Reflexão Difusa

A reflexão difusa não deixa resultado sombras no ambiente, isso acontece porque a luz difusa provém de diferentes posições criando um volume no objeto sem sombras a sua volta, essa ideia foi feita a partir do Alcalde (2012) e um estudo de luzes em superfícies rugosas.

Matematicamente, falando podemos utilizar dos estudos de (Marschner; Shirley, 2015) Quando uma superfície lambertiana² é iluminada, a luz que incide sobre o ponto P é refletida uniformemente em todas as direções. Isso é ilustrado na figura 3 a seguir.

Figura 3 – Reflexão difusa ideal.



Fonte: (Batagelo, 2021)

¹ É o processo pelo qual se obtém imagens digitais resultantes de modelos tridimensionais

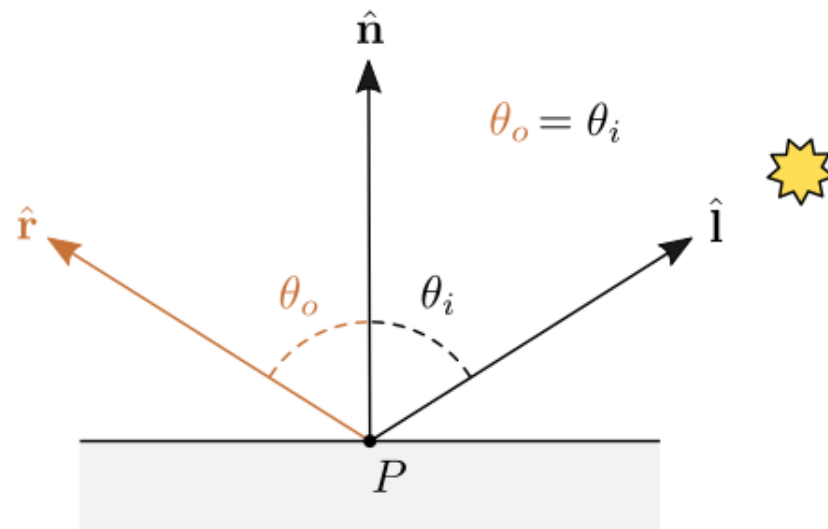
² De forma controlada e uniforme, a superfície lambertiana reflete uniformemente a luz, independentemente da direção em que a luz é refletida.

2.1.3 Reflexão Especular

"a iluminação especular identifica os realces especulares brilhantes que ocorrem quando a luz atinge uma superfície de objeto e reflete de volta em direção à câmera. A iluminação especular é mais intensa do que a luz difusa e incide mais rapidamente na superfície do objeto." dito por (Steven White, John Parente , Michael Satran , Alexander Koren, 2022)

É possível dizer que a partir de uma superfície perfeitamente especular é semelhante a um espelho ideal. Se a luz atinge o ponto P, ela é refletida apenas na direção reflexa \hat{r} do vetor \hat{l} em torno de \hat{n} . Isso é ilustrado na Figura 4.

Figura 4 – Reflexão especular ideal.



Fonte: (Batagelo, 2021)

2.2 Sombreamento

De acordo com Batagelo (2021) "Sombreamento ou tonalização (do inglês *shading*) é o processo de modificar a intensidade das cores de uma imagem através de tons claros e escuros, de modo a produzir a percepção de volume e profundidade de um objeto tridimensional."

2.2.1 Flat Shading

O sombreamento plano está em sua eficiência. A frequência de cálculo do modelo de iluminação é diretamente proporcional ao número de faces, que geralmente é menor que o número de *pixels*, resultando em menos avaliações.

Segundo o Marschner e Shirley (2015) O sombreamento plano geralmente expõe as arestas irregulares de uma malha poligonal, resultando em uma aparência fragmentada.

Isso pode ser um efeito atraente se o objeto em questão for um poliedro. No entanto, se a malha pretende aproximar uma superfície lisa, o resultado desejado só pode ser alcançado dividindo a malha em faces menores. Neste exemplo, uma malha triangular atua como uma imitação de uma esfera. Ao aumentar o número de subdivisões, é fácil ver que os intervalos de luz entre as faces são equilibrados suavemente.

Atualmente, o uso de sombreado plano na GPU³ não melhora a eficiência da renderização. Isso ocorre porque o pipeline gráfico atual é otimizado para lidar com atributos de vértice em vez de atributos de triângulo.

2.2.2 Sombreamento de Gourand

A suavização das transições de luz entre as faces dá ao sombreado Gouraud uma vantagem sobre a técnica de sombreado plano, pois melhora significativamente o aspecto visual de um objeto, superando visualmente sua contraparte com igual número de faces. Embora digno de nota, o efeito visual apelidado de bandas de Mach⁴ geralmente faz com que as bordas dos rostos pareçam mais pronunciadas do que a área interna. Isso é importante para que as imagens possam ser retratadas como objetos realistas em desenhos, esse efeito resulta do contraste exagerado criado nas bordas, que o sistema visual humano detecta facilmente com base na obra de Gourand (1971)

Onde faltam vértices de malha, pode haver um efeito de brilho cintilante, causado pela perda de brilho especular. Outro problema que pode ser enfrentado com o sombreado Gouraud é essa ocorrência. Para aliviar isso, pode-se aumentar o número de refinamentos de malha, embora isso possa aumentar o tempo de processamento. Se as subdivisões forem baixas, há uma chance de encontrar esse problema como mostrado por Batagelo (2021)

Calcular o vetor normal do vértice é uma etapa crucial na utilização do sombreado de Gouraud, pois ele pode ser obtido tomando-se a média dos vetores normais da face próxima.

2.2.3 Sombreamento de Phong

Segundo (Gomes,) No método de sombreado de Phong, a partir das normais aos vértices, realizada do mesmo modo que o sombreado de Gourand retratado abaixo, é calculada a normal a cada quadrícula através da interpolação das normais.isto é então usada no modelo de iluminação de Phong para calcular a intensidade da energia luminosa reflectida. Batagelo (2021) Explica que o sombreado de Phong é que exige mais dos sombreados, e consiste em avaliar a equação do modelo de iluminação para cada

³ *Graphics Processing Unit*, ou unidade de processamento de gráficos

⁴ O físico Ernst Mach se deparou com esse fenômeno na década de 1860, que pode ser atribuído a um contraste elevado das bordas em tons de cinza.

fragmento. O resultado é muito superior ao sombreamento de Gouraud para um mesmo número de faces. Marschner e Shirley (2015) retrata que para o modelo de Phong precisamos calcular um vetor \hat{n} para cada fragmento. Isso geralmente ocorre da interpolação linear das coordenadas (x,y,z) dos vetores normais de vértice, igual aos componentes RGB das cores dos vértices são interpoladas para criar um degradê de cores. Porém, neste caso, o vetor de coordenadas interpoladas é normalizado novamente no *fragment shader*⁵ para ser utilizado como vetor \hat{n} da equação.

No sombreamento de Phong, as bandas de Mach são praticamente imperceptíveis. Além disso, o brilho especular é mantido independentemente do refinamento da malha.

Para garantir que o decodificador tenha acesso a informações geométricas críticas, as codificações posicionais são adicionadas à incorporação da imagem sempre que participam de uma camada de atenção. Além disso, os tokens de prompt originais completos (incluindo suas codificações posicionais) são re-adicionados aos tokens atualizados sempre que participam de uma camada de atenção. Isso permite uma forte dependência tanto da localização geométrica quanto do tipo do token de prompt.

Após a execução do decodificador, ampliamos a incorporação da imagem atualizada em $4\times$ usando duas camadas convolucionais transpostas (resultando em uma escala $4\times$ menor em relação à imagem de entrada). Em seguida, os tokens realizam uma última atenção para a incorporação da imagem, e passamos a incorporação do token de saída atualizada para uma pequena MLP de três camadas que gera um vetor correspondente à dimensão de canal da incorporação da imagem ampliada. Finalmente, prevemos uma máscara através de um produto ponto a ponto espacial entre a incorporação da imagem ampliada e a saída da MLP.

O Transformer utiliza uma dimensão de incorporação de 256. Os blocos MLP do Transformer têm uma grande dimensão interna de 2048, mas a MLP é aplicada apenas aos tokens de prompt, que são relativamente poucos (raramente ultrapassando 20). No entanto, em camadas de cross-atenção, onde temos uma incorporação de imagem de 64×64 , reduzimos a dimensão do canal das queries, keys, e values em $2\times$ para 128, visando eficiência computacional. Todas as camadas de atenção utilizam 8 cabeças.

As convoluções transpostas usadas para ampliar a incorporação da imagem de saída são 2×2 , com stride 2 e dimensões de canal de saída de 64 e 32, e utilizam ativações GELU. Elas são separadas por normalização em camadas.

⁵ através da openGL

2.3 Ferramentas para iluminação de cenários

Para a iluminação ser feita em ambiente gráfico foram criadas diversas ferramentas, dentre elas Vulkan e OpenGL, que trabalham em cima de toda a forma como a luz e os materiais funcionam no ambiente, proporcionando configurações avançadas de baixo nível para que exista controle total do ambiente sendo desenvolvido.

2.3.1 Vulkan

Segundo o (Khronos Group, 2016), a Vulkan⁶ é uma API ⁷ de gráficos de baixo nível, criada pela Khronos Group, com o objetivo de fornecer aos desenvolvedores um acesso mais direto ao hardware do computador para criar aplicativos gráficos de alto desempenho em várias plataformas e dispositivos. Ela pode ser usada para desenvolver aplicativos para diversos casos de uso. Embora as aplicações em Vulkan possam escolher usar um subconjunto das funcionalidades descritas abaixo, a API foi projetada para que um desenvolvedor possa usar todas elas em uma única API.multiplataforma.

2.3.2 OpenGL

OpenGL⁸ é uma biblioteca de renderização (Khronos Group, 2016). No entanto, o que o OpenGL não faz é reter informações sobre um "objeto". Tudo o que o OpenGL vê é uma esfera de polígonos e um conjunto de estados com os quais renderizá-los. não se lembra que foi desenhado uma linha em um local determinado e uma esfera em outro.

Por causa disso, a maneira geral de usar o OpenGL é desenhar tudo o que você precisa desenhar e, em seguida, mostrar essa imagem com um comando de troca de *buffer* dependente da plataforma. Se você precisar atualizar a imagem, desenhe tudo novamente, mesmo que precise atualizar apenas parte dela. Se você quiser animar objetos se movendo na tela, precisa de um *loop* que constantemente limpe e redesenhe a tela.

2.3.3 Engines

Está sessão tem o intuito de apresentar as *engines* do mercado atual, afinal elas tem sua própria forma de criar iluminação e alterar de diversas maneiras para atender a um estilo visual único com *shaders* ou outros efeitos.

De acordo com (Felipe, 2017) Em sua essência, a *engine* é um software vital que compila todos os arquivos e bibliotecas essenciais dos quais um jogo depende. Essa ferramenta serve como base para o desenvolvimento de jogos, permitindo que os desenvolvedores criem habilmente sua visão. No entanto, a criação de jogos requer muito mais do que

⁶ disponível: <https://www.vulkan.org/>

⁷ Application Programming Interface

⁸ disponível: <https://www.opengl.org/>

apenas um mecanismo. Os desenvolvedores também devem fazer uso de editores de imagem, software de áudio e vídeo, modeladores 3D e software vetorial, dependendo da situação. Uma vez criados, esses elementos ganham vida dentro do motor, onde recebem animação, física, efeitos sonoros e outros recursos cruciais.

2.3.3.1 Unity

A marca registrada do *Unity* reside em seu estilo de programação, utilizando de vários objetos conectando vários pontos em um mesmo código. Distinto e estrutura de projeto que possui uma simplicidade inigualável. O *Unity* permite que os desenvolvedores utilizem as alternativas disponíveis e abre vários caminhos para os criadores se concentrarem em seus conhecimentos, principalmente na conduta dos personagens, de ambas as classes segundo (Henrique, 2014)

Embora o *Unity* tenha um objetivo de desenvolvimento bem definido, seus recursos se estendem além disso para diversos tipos de projeto. Gráficos altamente realistas são a combinação perfeita para jogos de aventura como RPGs ⁹, TPSs ¹⁰ e FPSs ¹¹ ao usar o *Unity*. A capacidade de incorporar elementos feitos por outros desenvolvedores de jogos aos nossos é um dos maiores pontos fortes do *Unity*. Essa funcionalidade é tremendamente vantajosa para indivíduos sem habilidades gráficas extensas, como modelagem 3D ou ilustração.

2.3.3.2 Godot

De acordo com Juan Linietsky, Ariel Manzur and the Godot community (2014), na Godot você tem liberdade criativa para personalizar o código-fonte do mecanismo de jogo, totalmente adaptado às suas necessidades. O software é flexível no sentido de que pode facilmente gerar jogos 2D e 3D, com uma abundância de recursos como gráficos, som, física e animação. Além disso, sua compatibilidade entre plataformas se estende a *Windows*, *macOS*, *Linux*, *Android*, *iOS* e *web*.

Comparado a outros mecanismos de jogos de código aberto, o Godot se destaca por sua interface amigável e processo de aprendizado simples. Com documentação abrangente, os aspirantes a criadores de jogos podem mergulhar direto no processo de criação do jogo. Os jogadores também podem fazer sua programação em várias linguagens diferentes, incluindo CSharp e GDScript. Esse recurso torna mais fácil para os desenvolvedores selecionarem o idioma de sua preferência.

⁹ Role Playing Game ou Jogo de interpretação de personagem

¹⁰ Third Person Shooter ou Tiro em terceira pessoa

¹¹ First Person Shooter ou Tiro em primeira pessoa

2.4 Trabalhos Correlatos

Nesta seção são apresentados os trabalhos correlatos ao proposto neste trabalho

2.4.1 Geração De Mapas De Iluminação Baseado Em Topologia Estimada Para Iluminação De Sprites

Este trabalho de autoria Santana (2017) trata-se de uma pesquisa prática em torno do mapeamento de uma imagem, e criação de uma topologia, no âmbito das pixel artes atualmente esse processo de iluminação é realizada de duas formas, com modelos em 3D ou de forma manual, mas o que realmente é tratado desse trabalho é de como esse tipo de tecnologia pode facilitar a vida do profissional.

Em seu desenvolvimento é retratado métodos já criados para tentar resolver o problema, um deles é o *Cross Shade* proposto por (Shao et al., 2012) que cria em um desenho uma especie de cruz para que em cada plano do desenho possa ser identificado a sua normal de forma como a cruz está, pegando sua curvatura para identificar um objeto em três dimensões

Dentre os métodos é utilizado para os resultados uma criação de mapa topológico a partir da ferramenta *sprite illuminator* e algumas customizações por cima do que foi gerado para que fosse possível uma bom resultado em pixel artes bem pequenas.

2.4.2 Analysis and Compilation of Normal Map Generation Techniques for Pixel Art

Esse trabalho do autor Moreira Flávio Coutinho (2013) tem como influencia a luz que interage com cada pixel do material. Existem vários métodos para gerar mapas normais em jogos 3D, mas aplicá-los na arte pixel pode não produzir resultados precisos por causa das particularidades do estilo. Este trabalho reúne diferentes métodos de geração de mapas normais e estuda sua aplicação na arte pixel, contribuindo para uma análise qualitativa do comportamento desses métodos em diferentes estudos de caso e diminuindo a escassez de material existente sobre essas técnicas.

2.4.3 CrossSketch: freeform surface modeling with details

Este artigo do autor Andre e Nakajima (2007) apresenta uma nova técnica para modelar uma superfície tridimensional livre, incluindo sua forma global e pequenos detalhes, usando uma interface de desenho a partir de um único ponto de vista. Em sistemas de modelagem anteriores que utilizavam esboços como entrada, a forma era reconstruída a partir da silhueta e o usuário tinha pouco controle sobre as partes internas do resultado. É gerado uma grade de linhas co-planas a partir de um pequeno número de traços

desenhados pelo usuário, foi estimado o vetor normal onde ele é restrito e formamos a superfície propagando essa informação para toda a grade. Como resultado, traços menores atuam localmente para adicionar detalhes, enquanto traços maiores modificam toda a superfície. Esse trabalho oferece uma nova abordagem para o problema de modelagem a partir de esboços e tem a intenção de ser parte de um sistema de modelagem mais complexo.

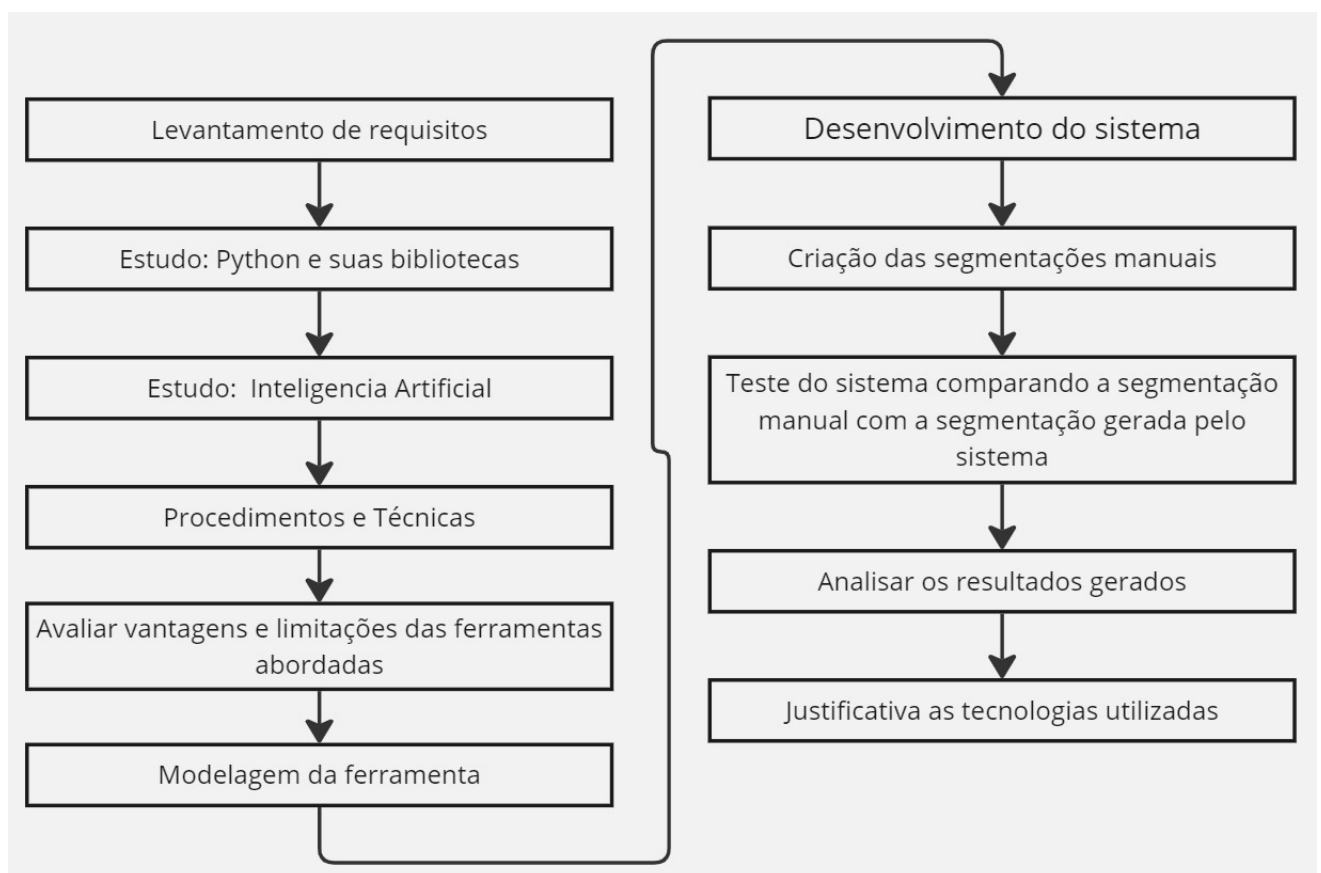
2.4.4 Segmentation by grouping junctions

Escrito por Ishikawa e Geiger (1998) este trabalho nos apresenta uma abordagem inovadora para a segmentação de imagens em tons de cinza, focando em junções, que são pontos críticos onde as bordas se encontram. Este método enfatiza a importância dessas junções na definição de regiões significativas dentro de uma imagem, contrastando com métodos tradicionais que frequentemente dependem apenas de informações de intensidade ou gradiente.

3 METODOLOGIA

Descrever metodologia, materiais e métodos utilizados no estudo, bem como os procedimentos experimentais realizados, nesta etapa será descrito vários assuntos sobre os passos a serem realizados iniciando com o levantamento dos dados, para uma análise e estudo posteriormente analisando todos os processos que são realizados, afim de iniciar o desenvolvimento.

Figura 5 – Passos feito na plataforma miro



Fonte: Feito pelo autor

3.1 Levantamento de Requisitos

Nessa sessão será abordado a pesquisa qualitativa feita a partir da ferramenta *Relight* que usa de um algoritmo para identificação de camadas em uma imagem, assim, quando a imagem é colocada na ferramenta ela faz um mapeamento das áreas altas e baixas para criar as camadas, por esse motivo as camadas que são denominadas como mais alta recebe mais luz do que as mais baixas.

Porém a ferramenta tem uma funcionalidade a mais onde é possível definir em que nível está a luz, fazendo com que a iluminação do objeto possa vir das camadas mais a baixo para as camadas acima, esta ferramenta com a luz funciona como um pincel de adicionar em softwares de pintura, pois ela recebe a luz já existente na foto e complementa por cima com a luz própria, podendo ser de qualquer cor.

A seguir será mostrado na tabela, onde terá a avaliação qualitativa referente a vários tipos de imagens testadas a partir da ferramenta, que para efeitos deste trabalho serão utilizados os valores de alta resolução como acima de 720 pixels, média resolução entre 720 e 256 pixels e baixa resolução abaixo de 256 pixels, retratado na tabela:

Avaliação qualitativa da ferramenta

Imagem	Alta Resolução	Média Resolução	Baixa Resolução
Objetos	Funciona	Funciona	Restrição
Humanos	Funciona	Funciona	Restrição
Paisagens	Funciona	Problema	Problema

Nessa tabela, foi adotado três categorias para descrever o funcionamento: Funciona, Restrição e Problema. A primeira categoria é utilizada para descrever que a imagem utiliza se comportou de forma correta, Restrição é a categorização para as imagens que funcionam mas em alguns casos ou áreas da imagem geram problemas, já a categoria Problema indica os casos onde a luz não reconhecia as formas de maneira correta.

3.2 Estudo: Python e suas bibliotecas

Nesta seção, será abordado sobre as diferentes bibliotecas estudadas para que a análise pudesse ser feita com o melhor aproveitamento da linguagem. será mostrado também alguns tópicos referente ao estudo do python como linguagem em geral.

Para iniciar o estudo das bibliotecas antes, deve se entender melhor sobre os ambientes do python como o ambiente virtual, além de estudar como é o funcionamento das IDEs com o ambiente python dá mesma forma.

3.2.1 PyTorch

Para o uso da SegmentAnything o pyTorch é uma das dependencias cruciais, já que ele aborda todo o processamento usado por placas de video e sistemas integrados com maior facilidade. O estudo dessa biblioteca se faz necessário apenas para a instalação de seus softwares dependentes como o uso do NVIDIA CUDA entre outros sistemas terceiros para o funcionamento do Segment Anything.

3.2.2 Scikit-image

O Scikit será extremamente necessário para a análise dos recursos e resultados obtidos, pois é com ele que conseguimos facilmente gerar as estatísticas como o MSE por exemplo. Não apenas isso mas o estudo dessa ferramenta deve vir analógica ao estudo dos métodos de estimativa comparando duas imagens.

3.2.3 OpenCV

O OpenCV (`cv2`) será empregado para lidar com tarefas relacionadas ao processamento de imagens, como leitura, conversão e redimensionamento. Ele será utilizado para ler as imagens a partir do disco (`cv2.imread`) e convertê-las entre diferentes espaços de cores para garantir a consistência de dados durante o processamento. Por exemplo, a função `cv2.cvtColor` será usada para converter as imagens do formato BGR para RGB, alinhando-as ao formato de cores esperado pelo modelo de segmentação. Além disso, o OpenCV será usado para redimensionar as imagens (`cv2.resize`) a fim de assegurar que elas possuam o mesmo tamanho antes de realizar comparações e cálculos de correlação. Um dos métodos-chave será o cálculo da Correlação Cruzada Normalizada (NCC), que será realizado por meio da função `cv2.matchTemplate`. Esse método permitirá medir a similaridade entre a imagem segmentada e a imagem esperada, fornecendo uma métrica quantitativa para avaliar a precisão do processo de segmentação.

3.2.4 Numpy

No projeto, o `numpy` será fundamental para a manipulação eficiente de arrays multidimensionais, que representarão as imagens processadas. O `numpy` será utilizado para criar e gerenciar esses arrays, facilitando a realização de operações matemáticas e comparações pixel a pixel nas imagens. Por exemplo, a função `np.zeros` será utilizada para inicializar uma matriz de zeros que servirá como base para a imagem segmentada final na função de geração de máscara. Além disso, o `numpy` permitirá a execução de operações eficientes, como a comparação entre arrays de imagens para verificar a igualdade de pixels com a função `np.array_equal`. As imagens serão normalizadas convertendo os valores de pixel para um formato de ponto flutuante, possibilitando uma análise mais precisa ao dividir os valores por 255.0.

3.3 Estudo: Inteligência Artificial

Nesse estudo será demonstrado algumas das partes que serão necessárias para a criação da análise posteriormente, inicialmente foi preciso uma serie de estudos sobre todas as possibilidades relacionadas as possíveis segmentações realizadas, a seguir cada tipo de segmentação será melhor explicada.

3.3.1 Segmentação por máscara

A segmentação por máscara utiliza um modelo para identificar e isolar regiões específicas dentro de uma imagem. O modelo gera uma máscara binária onde a região segmentada é representada por 1 e o fundo por 0. Este método é eficaz para separar objetos ou áreas de interesse em imagens, permitindo uma análise mais detalhada dessas regiões específicas.

3.3.2 Segmentação por pontos de clique

Neste método, o usuário seleciona pontos dentro da área que deseja segmentar. É utilizado esses pontos como referências para definir os limites da região de interesse. A segmentação é então ajustada com base na localização desses pontos, oferecendo um controle mais preciso sobre as regiões segmentadas, especialmente em imagens complexas.

3.3.3 Segmentação por caixa de delimitação

Aqui, o usuário fornece uma caixa delimitadora ao redor do objeto ou área de interesse. O modelo então realiza a segmentação dentro dessa caixa. Esse método é útil para rapidamente identificar e segmentar objetos que podem ser facilmente contidos dentro de uma área retangular, facilitando a definição da região de interesse.

3.3.4 Segmentação por texto

A segmentação por texto permite ao usuário descrever a região ou objeto desejado em termos textuais. O modelo utiliza essa descrição para identificar e segmentar a área correspondente na imagem. Este método é útil quando a descrição do objeto é mais clara em palavras do que em detalhes visuais, oferecendo uma maneira eficiente de segmentar com base em descrições contextuais.

3.3.5 Segmentação interativa

Este método combina várias abordagens interativas, como pontos de clique e caixas de delimitação, permitindo que o usuário refine a segmentação com base no feedback contínuo. A segmentação inicial pode ser ajustada e melhorada conforme o usuário fornece mais informações ou faz ajustes na área segmentada, resultando em uma segmentação mais precisa e adaptada às necessidades específicas.

3.3.6 Segmentação automática

Na segmentação automática, é aplicado algoritmos de segmentação sem intervenção do usuário. O modelo usa padrões aprendidos para identificar e segmentar automaticamente

os objetos de interesse na imagem. Esse método é ideal para processar grandes volumes de dados ou quando a segmentação precisa ser realizada de forma rápida e eficiente em cenários bem definidos.

3.4 Procedimentos e Técnicas

Para o desenvolvimento desta pesquisa, foram realizados testes preliminares explorando toda a capacidade do modelo Segment Anything (SAM), com o intuito de validar a eficácia dos processos envolvidos na análise de segmentação de imagens. Durante essa fase inicial, foi constatado que o tempo de processamento se mostrou elevado, mesmo em um sistema com hardware moderno. Como resultado, optou-se por não utilizar a tecnologia NVIDIA CUDA, uma vez que o uso intensivo de processamento pela GPU não seria imprescindível para o escopo do projeto.

Com essa decisão tomada, iniciou-se uma nova bateria de testes envolvendo diferentes tipos de imagens, abrangendo diversas categorias visuais, a fim de avaliar o potencial de segmentação do SAM em variados contextos. Os resultados foram, em sua maioria, satisfatórios, com o modelo demonstrando desempenho adequado na maioria dos casos analisados.

Posteriormente, iniciou-se o desenvolvimento de imagens específicas, criadas pelo autor em estilo pixel art, para validar a performance do modelo em situações de segmentação por cor. Essa etapa exigiu ajustes no código, especialmente no que diz respeito à manipulação das camadas e formatos das imagens, além da tipagem adequada em Python. Esse processo envolveu pesquisas extensivas e múltiplas tentativas até a obtenção de um resultado satisfatório na geração de grupos de cores a partir das imagens.

Com o sistema de segmentação de imagens em funcionamento, iniciou-se a etapa de criação manual das camadas de segmentação. O autor definiu como deveriam ser os resultados esperados da segmentação proposta pela IA, segmentando um conjunto de 50 imagens em estilo pixel art com base em cores específicas para realizar as análises. A partir desse ponto, foi possível aplicar métodos existentes para a comparação entre duas imagens.

Inicialmente, foi desenvolvido um método próprio, denominado pelo autor como "Método de Espalhamento". Esse método parte do canto da imagem e realiza uma varredura pixel a pixel ao redor de cada ponto analisado, comparando diretamente com o resultado esperado. Caso a cor do pixel na imagem gerada seja diferente da cor prevista no resultado esperado, era gerada uma pontuação negativa. Se as cores coincidisse, o método prosseguia, continuando a busca por outras colorações na imagem base.

Sempre que vemos uma cor clara e logo depois uma escura, isso significa que há um relevo muito grande ali, por exemplo na Figura 6

Figura 6 – História em quadrinhos, Batman



Fonte: (Bazela, 2022)

Nesse quadrinho do Batman, podemos ver que em seu ombro está muito claro e logo acima onde sua capa está vem uma cor muito escura, isso nos mostra que a profundidade da capa é grande, ao ponto de não chegar nenhuma luz até ela, é claro que precisamos levar em consideração que nas HQ's em geral os contrastes são muito maiores, porque traz esse volume nos trajes.

Partindo para o ramo da *Unity* será preciso um aprendizado todo relacionado a iluminação dentro da ferramenta, uso dos objetos em cena e manipulação deles, pois para o desenvolvimento será necessário a criação de uma malha que forme a imagem, um sistema de camadas para que possa estabelecer profundidade, um sistema de alteração e manipulação da posição, cor e luminosidade do ponto de luz. E por fim um estudo básico de toda interface da *Unity*.

Outro ponto importante no estudo, será a linguagem de programação **C#** que é usada como alicerce para qualquer código que precise ser estruturado lá dentro, desde instanciar objetos dentro da cena até modificar configurações de câmera como movimento, posicionamento e ângulo até para recebimento dos arquivos

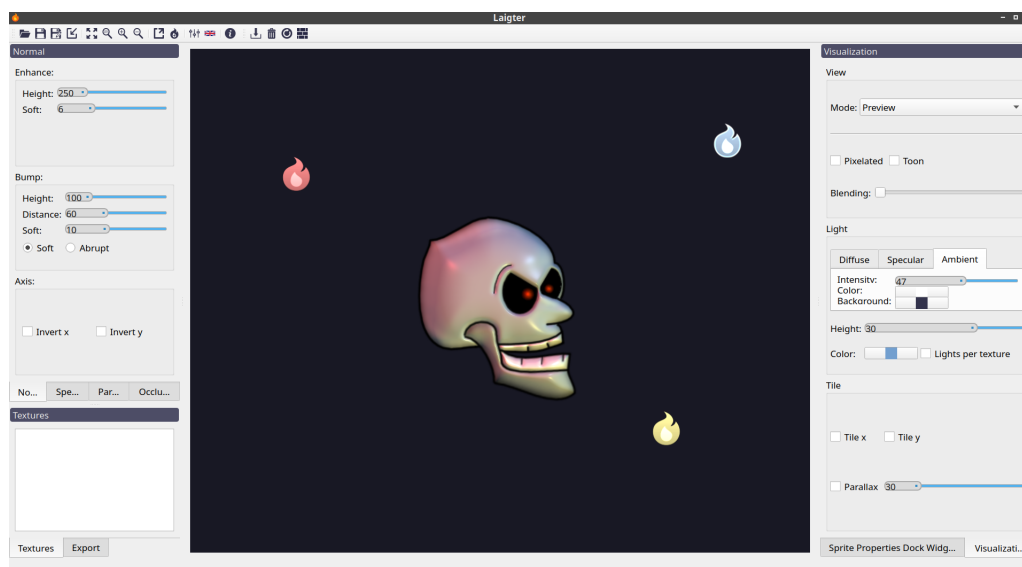
3.5 Avaliar vantagens e limitações das ferramentas abordadas

O *Relight* é uma ferramenta muito boa para processos de iluminação tanto para cenários como personagens, com uma boa modificação como profundidade, posição, cor e luminosidade, dando para o usuário a liberdade de dar personalidade as suas imagens.

Mas ainda existe várias limitações e elas que serão retratadas nessa seção, como foi visto na análise a cima na tabela é possível notar que quanto menor a imagem é, menos funcional ela passa a ser, por exemplo imagens em baixa resolução ou pixeladas começam a ter muitos problemas pois a luz não consegue distinguir os objetos na cena e muito menos paisagens pois com tantas cores perto uma das outras transforma a cena em uma desordem visual.

Além da ferramenta abordada acima existe também algumas outras opções no mercado para este tipo de processo, como o *laigter* mas que ainda se envolve em uma limitação, que é o mapa de normais em apenas uma camada, o *laigter* atual na imagem fornecida pelo usuário como retratado na 7.

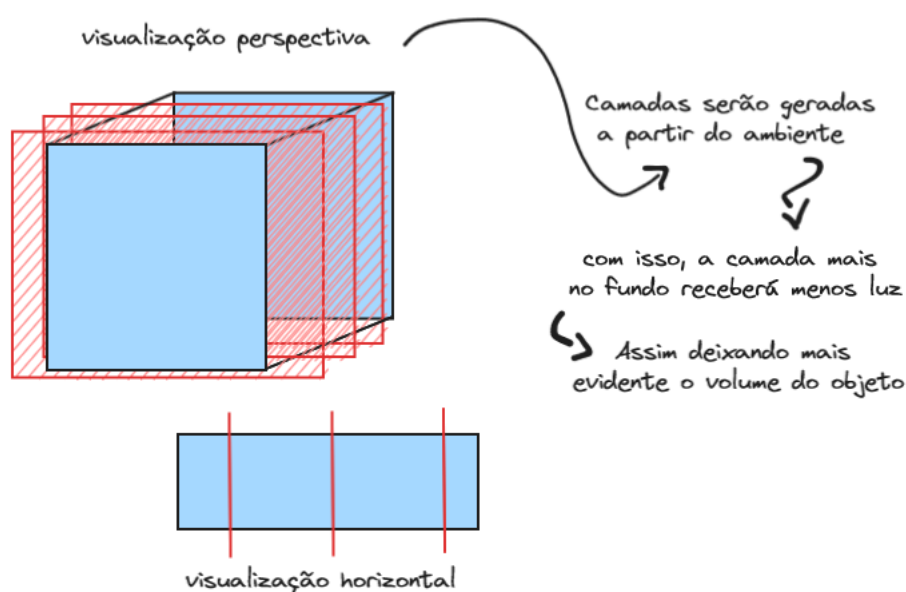
Figura 7 – Imagens retiradas do github da ferramenta



3.6 Modelagem da ferramenta

Uma malha deverá ser criada por cima de toda a imagem e definir um inteiro para a profundidade, cada *pixel* da imagem deve receber uma profundidade como pode se observar na Figura 8

Figura 8 – Rascunho elaborado pelo autor



Assim ele vai numerar toda a malha de *pixels* na tela com números que podem ter uma variação maior dependendo do tamanho da imagem (no caso de exemplo)

Será calculado de 1 a 9, mas com imagens imensas em alta resolução podemos pensar em usar de 1 a 100 ou até mais

Podemos pensar que dessa forma será possível utilizar essa profundidade para produzir uma luz pelas laterais ou pela frente e até atrás de elementos, sem perder sua funcionalidade até mesmo em imagens pequenas

Além disso, cada um desses valores definidos em cada *pixel* da imagem, vão servir de referencia para criação de vários objetos, ele levará em consideração a distância e a variação de cor que foi atingida

Como por exemplo, se existir um objeto a frente da camada 1 até a 10 e um objeto atrás que está na camada 15, essa distancia de 5 camadas irá fazer o objeto dá frente se separar com o de trás transformando a cena com dois objetos ao invés de um

3.7 Justificativa das Tecnologias a serem adotadas

Para a criação da ferramenta de análise será utilizado o ambiente de desenvolvimento Visual Studio Code utilizando Python, foi optado este ambiente por sua praticidade com os diversos formatos de arquivos que serão utilizados ao decorrer do projeto. O python também servirá como unica linguagem de programação para a ferramenta pela facilidade com a criação e uso das tecnologias de inteligencia Artificial que irá adiantar muitos dos processos necessários para a criação da análise

3.8 Avaliação Qualitativa

3.9 Objetivo da Metodologia

3.10 Design da Pesquisa

3.11 Amostragem

3.12 Procedimentos Analíticos

3.13 Critérios de Validação e Confiabilidade

3.14 Aspectos Éticos

3.15 Cronograma e Recursos

3.16 Limitações

[illegible]

5 CONCLUSÕES PARCIAIS

Foi descrita a modelagem da ferramenta própria, que envolve a criação de uma malha por cima da imagem para definir a profundidade e a variação de cor, permitindo a manipulação da luz. Além disso, foi citado as tecnologias adotadas, como o Visual Studio para programação em *C#* e a Unity como a game *engine* escolhida.

Para avaliar a eficácia da ferramenta, foi planejado realizar uma avaliação qualitativa dos resultados obtidos, incluindo a análise da alteração de imagens, a viabilidade da ferramenta e a eficácia dos cálculos realizados. Para fim é esperado que a ferramenta esteja concluída mesmo que de forma manual, para que ocasionalmente possa ter um progresso posterior em outros trabalhos.

REFERÊNCIAS

- ALCALDE, A. K. G. Interacción de luz difusa con superficies rugosas. 2012. 8
- ANDRE, S. S. A.; NAKAJIMA, M. Crosssketch: freeform surface modeling with details. p. 52, 2007. Disponível em: <https://dl.acm.org/doi/abs/10.1145/1384429.1384442?casa_token=ynVUdINw8eEAAAAA:mu8ZoSKq_D8QfR4RwMT4RHnyB6rCD1ScV4kjGh33Ie6rt-kafs7Uked4g0i_zABu9NkJc7RStqhfig>. 14
- AZEVEDO, E.; CONCI, A.; VASCONCELOS, C. **Computação Gráfica Teoria e Prática: Geração de imagens**. 1. ed. [S.l.: s.n.], 2022. v. 1. 7
- BATAGELO, B. M. H. **Reflexão Especular**. 2021. <<https://www.brunodorta.com.br/cg/lighting.html>>. Accessed: 2023-4-24. 8, 9, 10
- BAZELA, C. **Batman: 3 quadrinhos que influenciaram o novo filme**. 2022. <<https://www.boletimnerd.com.br/batman-3-quadrinhos-que-influenciaram-o-novo-filme/>>. Accessed: 2023-5-20. 21
- CEZIMBRA, D. J. **História da Arte e Design**. 1. ed. [S.l.]: Contentus, 2021. 142 p. 5
- FELIPE, L. **O que é uma engine? Qual sua importância no desenvolvimento de games?** 2017. Accessed: 2023-4-30. 12
- GOMES, M. R. Computação gráfica. In: **Iluminação e Sombreamento**. [S.l.: s.n.]. 10
- GORAL, C. M. et al. Modeling the interaction of light between diffuse surfaces. **SIGGRAPH Comput. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 18, n. 3, p. 213–222, jan 1984. ISSN 0097-8930. Disponível em: <<https://doi.org/10.1145/964965.808601>>. 7
- GOURAND, H. **Computer Display of Curved Surfaces**. Tese (Doutorado) — University of Utah, 1971. 10
- GRAFITTI. **Aprenda a criar efeitos de luz e sombra para as suas ilustrações**. 2019. Disponível em: <<https://blog.grafittiartes.com.br/aprenda-criar-efeitos-de-luz-e-sombra-para-as-suas-ilustracoes>>. Acesso em: 27 de mar. de 2023. 5
- Henrique. **Unity 3D: Introdução ao desenvolvimento de games**. 2014. <<https://www.devmedia.com.br/unity-3d-introducao-ao-desenvolvimento-de-games/30653>>. Accessed: 2023-4-27. 13
- ISHIKAWA, H.; GEIGER, D. **Segmentation by grouping junctions**. [S.l.]: IEEE Computer Society, 1998. 125-131 p. 15
- Juan Linietsky, Ariel Manzur and the Godot community. **Introdução ao editor do Godot**. 2014. <https://docs.godotengine.org/pt_BR/stable/getting_started/introduction/introduction_to_godot.html>. Accessed: 2023-4-27. 13

Khronos Group. **Vulkan Guide**. 2016. <<https://www.vulkan.org/>>. Accessed: 2023-4-30. 12

MANSSOUR, I. H.; COHEN, M. Introdução à computação gráfica. **RITA**, v. 13, n. 2, p. 43–68, 2006. 7

MARSCHNER, S.; SHIRLEY, P. **Fundamentals of Computer Graphics**. [S.l.]: CRC Press, 2015. v. 1. Drawing on an impressive roster of experts in the field, Fundamentals of Computer Graphics, Fourth Edition offers an ideal resource for computer course curricula as well as a user-friendly personal or professional reference. Focusing on geometric intuition, the book gives the necessary information for understanding how images get onto the screen by using the complementary approaches of ray tracing and rasterization. It covers topics common to an introductory course, such as sampling theory, texture mapping, spatial data structure, and splines. It also includes a number of contributed chapters from authors known for their expertise and clear way of explaining concepts. Highlights of the Fourth Edition Include: Updated coverage of existing topics Major updates and improvements to several chapters, including texture mapping, graphics hardware, signal processing, and data structures A text now printed entirely in four-color to enhance illustrative figures of concepts The fourth edition of Fundamentals of Computer Graphics continues to provide an outstanding and comprehensive introduction to basic computer graphic technology and theory. It retains an informal and intuitive style while improving precision, consistency, and completeness of material, allowing aspiring and experienced graphics programmers to better understand and apply foundational principles to the development of efficient code in creating film, game, or web designs. ISBN 978-14-8222-941-7. 8, 9, 11

MOREIRA FLÁVIO COUTINHO, L. C. R. D. Analysis and compilation of normal map generation techniques for pixel art. 2013. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/9961116>>. 14

SANTANA, F. F. G. A. C. D. Geração de mapas de iluminação baseado em topologia estimada para iluminação de sprites. 2017. Disponível em: <<https://app.uff.br/riuff/handle/1/5683>>. 14

SCALCO, R. Uso do geogebra 3d como estúdio de tonalização para análise das componentes dos coeficientes de reflexão do modelo de iluminação local. **Revista do Instituto GeoGebra Internacional de São Paulo**, v. 1, n. 1, p. XVI – XXVII, mar. 2012. Disponível em: <<https://revistas.pucsp.br/index.php/IGISP/article/view/8198>>. 7

SHAO, C. et al. Crossshade: Shading concept sketches using cross-section curves. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 31, n. 4, jul 2012. ISSN 0730-0301. Disponível em: <<https://doi.org/10.1145/2185520.2185541>>. 14

SHIRLEY, P.; MORLEY, R. **Realistic Ray Tracing, Second Edition**. Taylor & Francis, 2008. ISBN 9781568814612. Disponível em: <<https://books.google.com.br/books?id=knpN6mnhJ8QC>>. 6

Steven White, John Parente , Michael Satran , Alexander Koren. **Microsoft Learn: Iluminação Especular**. 2022. <<https://learn.microsoft.com/pt-br/windows/uwp/graphics-concepts/specular-lighting>>. Accessed: 2023-4-24. 9