

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INSTITUTO DE COMPUTAÇÃO

CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS EDUARDO ANDRADE DA SILVA

SIMULAÇÃO DE UM PERFURADOR DE CARTÕES EM REALIDADE VIRTUAL

RIO DE JANEIRO

2023

CARLOS EDUARDO ANDRADE DA SILVA

SIMULAÇÃO DE UM PERFURADOR DE CARTÕES EM REALIDADE VIRTUAL

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Valeria Menezes Bastos

RIO DE JANEIRO

2023

## CIP - Catalogação na Publicação

S586s Silva, Carlos Eduardo Andrade da  
Simulação de um perfurador de cartões em  
realidade virtual / Carlos Eduardo Andrade da  
Silva. -- Rio de Janeiro, 2023.  
82 f.

Orientadora: Valeria Menezes Bastos.  
Coorientador: Realidade virtual .  
Trabalho de conclusão de curso (graduação) -  
Universidade Federal do Rio de Janeiro, Instituto  
de Computação, Bacharel em Ciência da Computação,  
2023.

1. Realidade virtual. 2. Museu. 3. História da  
computação. 4. Simulação. 5. Reconstrução. I. Bastos,  
Valeria Menezes, orient. II. , Realidade virtual,  
coorient. III. Título.

CARLOS EDUARDO ANDRADE DA SILVA

SIMULAÇÃO DE UM PERFURADOR DE CARTÕES EM REALIDADE VIRTUAL

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 03 de Março de 2023.

BANCA EXAMINADORA:

---

Valeria Menezes Bastos, D.Sc. (UFRJ)

---

Paulo Henrique de Aguiar Rodrigues, Ph.D. (UFRJ)

---

Claudia Lage Rebello da Motta, D.Sc. (UFRJ)

---

Nelson Quilula Vasconcelos, M.Sc. (UFRJ)

## RESUMO

Museus podem não ter acesso a todos os recursos necessários para a aquisição e manutenção dos artefatos que gostariam de exibir. Uma solução para este problema pode ser encontrada na adoção da realidade virtual. Este trabalho tem como objetivo desenvolver uma exibição em realidade virtual e examinar sua efetividade em educar e fomentar interesse no artefato simulado e no museu que adotou essa tecnologia. Mais especificamente, o funcionamento de um *keypunch* IBM 029 funcional foi simulado. A qualidade da simulação é medida por meio de um questionário, tanto a presença durante a experiência virtual quanto correlações entre presença e qualidade da exibição são investigadas. A exibição obteve sucesso em aumentar o interesse em ambos, a máquina simulada e o museu, sendo caracterizada pelos participantes na testagem como uma experiência satisfatória e educativa. Uma correlação significativa não foi encontrada entre a presença sentida pelos participantes e sua qualidade relatada. Os resultados indicam que essa tecnologia é digna de maior investimento para museus que desejem enriquecer suas exibições e atrair um público maior.

**Palavras-chave:** realidade virtual; museu; história da computação; simulação; reconstrução; exibição.

## ABSTRACT

Museums might not have access to all the resources necessary for the acquisition and maintenance of the artifacts they would like to display. One solution for this problem can be found in the adoption of virtual reality. This work's objective is to develop an exhibit in virtual reality and examine its effectiveness in educating and stimulating interest on the simulated artifact and the museum that adopted this technology. More specifically, a functional IBM 029 keypunch was simulated. The simulation's quality is measured with a questionnaire, as well as the presence during the virtual experience, and correlations between presence and perceived quality of the exhibit are investigated. The exhibit obtained success in increasing the interest in both the simulated machine and the museum, being characterized by test participants as satisfactory and educational. A significant correlation was found between perceived presence and reported quality. The results indicate this technology is worth increased investment for museums aiming to enrich their exhibitions and attract a broader public.

**Keywords:** virtual reality; museum; computer history; simulation; recreation; exhibition.

## **LISTA DE SIGLAS**

FPS - Frames Per Second, Frames Por Segundo, Quadros Por Segundo

HMD – Head-Mounted Display, visor de realidade virtual, óculos de realidade virtual

3DoF – 3 Degrees of Freedom, 3 Graus de Liberdade: classificação de um sistema de realidade virtual que rastreia a rotação da cabeça do usuário

6DoF – 6 Degrees of Freedom, 6 Graus de Liberdade: classificação de um sistema de realidade virtual que rastreia a rotação da cabeça do usuário e a posição de seu corpo

IPQ – Igroup Presence Questionnaire, questionário utilizado para medir presença em um ambiente virtual

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	OBJETIVO	10
1.2	ESTRUTURA DO TEXTO	11
<b>2</b>	<b>CONHECIMENTOS ESSENCIAIS</b>	<b>12</b>
2.1	REALIDADE VIRTUAL	12
<b>2.1.1</b>	<b>Definição</b>	<b>12</b>
<b>2.1.2</b>	<b>Imersão</b>	<b>14</b>
<b>2.1.3</b>	<b>Ilusões da Realidade Virtual</b>	<b>15</b>
2.2	OBJETOS	17
<b>2.2.1</b>	<b>Cartão Perfurado</b>	<b>17</b>
<b>2.2.2</b>	<b>Perfurador de Cartões IBM 029</b>	<b>18</b>
<b>3</b>	<b>TECNOLOGIAS UTILIZADAS</b>	<b>19</b>
3.1	HMD	19
3.2	CONTROLE	21
3.3	UNITY	22
3.4	BLENDER	24
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>26</b>
4.1	METAS	26
4.2	MODELOS	27
<b>4.2.1</b>	<b>IBM 029</b>	<b>27</b>
<b>4.2.2</b>	<b>Cartão Perfurado</b>	<b>31</b>
<b>4.2.3</b>	<b>Avatar</b>	<b>33</b>
4.3	CENA	34
4.4	PROTOTIPAGEM ITERATIVA	35
<b>4.4.1</b>	<b>Primeiro Protótipo</b>	<b>35</b>
<b>4.4.2</b>	<b>Adições à Cena</b>	<b>37</b>



<b>4.4.3</b>	<b>Movimentação</b>	<b>39</b>
<b>4.4.4</b>	<b>Cartões</b>	<b>39</b>
<b>4.4.5</b>	<b>Perfurações</b>	<b>40</b>
<b>4.4.6</b>	<b>Animação do Cartão</b>	<b>45</b>
<b>4.4.7</b>	<b>Múltiplos Cartões</b>	<b>49</b>
<b>4.4.8</b>	<b>Entrada da Mensagem Perfurada</b>	<b>51</b>
<b>4.4.9</b>	<b>Atualização da Cena</b>	<b>52</b>
<b>4.5</b>	<b>OTIMIZAÇÕES</b>	<b>57</b>
<b>4.5.1</b>	<b>Número de Polígonos</b>	<b>58</b>
<b>4.5.2</b>	<b>Draw Call</b>	<b>58</b>
<b>4.5.3</b>	<b>Iluminação</b>	<b>59</b>
<b>4.6</b>	<b>FLUXO DA EXECUÇÃO</b>	<b>60</b>
<b>4.7</b>	<b>LIMITES DO DESIGN</b>	<b>61</b>
<b>4.8</b>	<b>DESAFIOS</b>	<b>62</b>
<b>5</b>	<b>TESTES</b>	<b>63</b>
<b>5.1</b>	<b>HIPÓTESES</b>	<b>63</b>
<b>5.2</b>	<b>REGISTRO DA POSIÇÃO</b>	<b>63</b>
<b>5.3</b>	<b>QUESTIONÁRIOS</b>	<b>64</b>
<b>5.3.1</b>	<b>Questionário Preliminar</b>	<b>65</b>
<b>5.3.2</b>	<b>Questionário IPQ</b>	<b>65</b>
<b>5.3.3</b>	<b>Questionário da Qualidade</b>	<b>65</b>
<b>5.4</b>	<b>METODOLOGIA</b>	<b>66</b>
<b>5.5</b>	<b>PARTICIPANTES</b>	<b>66</b>
<b>6</b>	<b>RESULTADOS</b>	<b>67</b>
<b>6.1</b>	<b>EXPLORAÇÃO DO AMBIENTE</b>	<b>67</b>
<b>6.2</b>	<b>ANÁLISE QUALITATIVA</b>	<b>68</b>
<b>6.3</b>	<b>ANÁLISE QUANTITATIVA</b>	<b>69</b>
<b>6.3.1</b>	<b>Hipótese 1: Capacidade Como Experiência em Realidade Virtual</b>	<b>69</b>
<b>6.3.2</b>	<b>Hipótese 2: Capacidade Como Exibição de Um Museu</b>	<b>69</b>

<b>6.3.3</b>	<b>Hipótese 3: Interesse em Visitar</b>	<b>70</b>
<b>6.3.4</b>	<b>Hipótese 4: Interesse em Computação</b>	<b>71</b>
<b>6.3.5</b>	<b>Hipótese 5: Relação Entre Presença e Qualidade</b>	<b>72</b>
<b>7</b>	<b>CONCLUSÃO</b>	<b>74</b>
7.1	DISCUSSÃO	74
7.2	TRABALHOS FUTUROS	75
	<b>REFERÊNCIAS</b>	<b>77</b>
	<b>APÊNDICE A - QUESTIONÁRIO PRELIMINAR</b>	<b>81</b>
	<b>APÊNDICE B - QUESTIONÁRIO IPQ</b>	<b>81</b>
	<b>APÊNDICE C - QUESTIONÁRIO DA QUALIDADE</b>	<b>82</b>

## 1 INTRODUÇÃO

Museus são, segundo o Conselho Internacional de Museus, instituições sem fins lucrativos a serviço da sociedade, aberta ao público, que adquire, conserva, pesquisa e exhibe a herança da humanidade e seu ambiente para educar, estudar, e lazer [12]. Museus cumprem papéis indispensáveis no desenvolvimento da sociedade, preservando e exibindo artefatos que, quando devidamente contextualizados, podem contar uma história que seja relevante culturalmente e cientificamente para seu público. Visando cumprir esses papéis, exposições são projetadas de forma a aumentar o interesse do público visitante nos assuntos apresentados e se tornarem experiências memoráveis. Um museu deve ser capaz de apresentar questões de forma que uma pessoa possa simultaneamente aprender e sentir-se entretida.

Visando cumprir sua missão, museus muitas vezes buscam tornar a experiência de visitá-lo mais memorável e educativa, por exemplo acrescentando interatividade em suas exposições e mantendo os artefatos em condições próximas às originais, como seriam encontrados no mundo real em seu contexto histórico. Porém, a aquisição, preservação para gerações futuras, e exibição visando difundir o conhecimento provindo desses objetos culturais são todos trabalhos custosos, não só em tempo e dinheiro, mas também na mão de obra qualificada em adquiri-los e conservá-los.

Tomamos o Computer History Museum (Califórnia, EUA) como exemplo. Nele, as máquinas que tiveram grande impacto na computação e foram ultrapassadas em utilidade encontram um novo uso principalmente para fomentar interesse em computação no público em geral, transmitindo para ele o significado daqueles artefatos no desenvolvimento de sua sociedade. Uma de suas exposições consiste na demonstração da operação de um computador IBM 1401 e seus periféricos (perfurador e leitor de cartão, impressora, unidade de fita), com objetivo de viajar de volta no tempo e experimentar a vista e sons de um centro de computação. O processo de restauração dessa máquina não foi simples, envolvendo a ajuda de um time de voluntários altamente qualificados por anos [13]. Todo esse processo envolve esforço e recursos a um nível que o torna difícil de replicar.

Uma solução para esse problema pode ser encontrada na adoção da realidade virtual. Esse não é um problema novo, e profissionais estão cientes do potencial dessa tecnologia para remediá-lo, mas somente com a recente queda de custos de implementação é que se tornou viável

para a grande maioria das instituições. Com sua aplicação, torna-se possível replicar os artefatos desejados por simulação em um ambiente virtual. Essa adoção permite que o público interaja com a exibição sem correr riscos de prejudicá-la, reduzindo os custos de conservação e expansão das exibições por se tratarem de simulações, além de fornecer uma experiência mais memorável quando comparada a ver um artefato através de uma vitrine.

Além disso, ter uma exibição virtual traz uma série de outros benefícios a uma instituição:

- permite que pessoas que não podem acessar o museu fisicamente, tanto pela distância quanto por alguma deficiência, possam usufruir da experiência;
- permite acesso sem limitações de tempo, caso a experiência esteja disponível online;
- aumenta o interesse do público em visitar o museu (uma pesquisa indica que após visitar o website de um museu, seria mais provável que uma pessoa o visitasse de forma física [14]), incluindo o público que não visitaria museus normalmente.

## 1.1 OBJETIVO

A tese deste trabalho é de que a realidade virtual é uma opção digna de maiores investimentos para museus que desejam enriquecer suas exibições, porém não possuem os recursos, tanto monetários quanto em mão de obra qualificada, necessários para adquirir e realizar manutenção para peças físicas. Para testar esta tese, foi desenvolvido um ambiente virtual no qual o usuário é capaz de ver em ação e interagir com uma das máquinas que impactou o desenvolvimento da computação e já não são mais vistas em funcionamento: o perfurador de cartões IBM 029.

Sua efetividade como uma exibição de um museu e como uma experiência em realidade virtual são avaliados por meio de questionários. Sua qualidade como experiência em realidade virtual é testada pela sua capacidade de imergir o usuário no ambiente artificial e fazê-lo se comportar como faria no mundo real. Sua qualidade como exibição de um museu é testada pela satisfação, aprendizagem, e fomentação de interesse dos usuários.

## 1.2 ESTRUTURA DO TEXTO

O texto está dividido nas seguintes seções: Capítulo 2 (Conhecimentos Essenciais), onde informações sobre a realidade virtual e sobre a máquina que será simulada são apresentadas, permitindo a compreensão do trabalho pelo leitor; Capítulo 3 (Tecnologias Utilizadas), onde as ferramentas utilizadas (hardware e software) são apresentadas, assim como a forma como serão usadas neste trabalho; Capítulo 4 (Desenvolvimento), onde será decorrido todo o processo do desenvolvimento da simulação, desde os estágios do planejamento, até as otimizações; Capítulo 5 (Testes), onde os métodos pelos quais os testes foram realizados são descritos, assim como as hipóteses testadas, como a coleta de dados foi realizada, e uma descrição dos participantes; Capítulo 6 (Resultados), onde os dados coletados são analisados e as hipóteses testadas; Capítulo 7 (Conclusão), onde as conclusões do trabalho são expostas e como este trabalho pode ser complementado por trabalhos futuros.

## 2 CONHECIMENTOS ESSENCIAIS

Esta seção tem como objetivo descrever conceitos que tornarão mais fácil a compreensão do trabalho e de decisões tomadas durante seu desenvolvimento.

### 2.1 REALIDADE VIRTUAL

#### 2.1.1 Definição

O engenheiro de software Fred Brooks define a realidade virtual como a experiência na qual o usuário está efetivamente imerso em um mundo virtual responsivo às suas ações [19]. Uma outra característica central para a definição é o controle que o usuário possui sobre seu ponto de vista, um ponto central no desenvolvimento de mídias em realidade virtual. Aplicações avançadas em realidade virtual elevam ainda mais o nível de controle do usuário sobre o ambiente, tornando-o capaz de reagir a toques, movimentos e outras interações, tendo como objetivo aumentar a sensação de estar realmente no ambiente virtual onde está inserido.

A mídia virtual é destacada de outras formas por sua facilidade em evocar a sensação de que o usuário está presente na mídia, inserido nela. Isso se dá principalmente por três características da realidade virtual [19]: visão estéreo em 3D; controle do ponto de vista; e ter toda a percepção visual controlada pelo aparelho.

- Visão estéreo em 3D:

Dispositivos capazes de exibir imagens em realidade virtual possuem duas telas, uma para cada olho, que reproduzem imagens levemente diferentes emulando a forma como o mundo real é visto por nossos olhos.

- Controle do ponto de vista:

O controle do ponto de vista permite que o usuário decida quais imagens serão reproduzidas em sua HMD (*Head-Mounted Display*, óculos de realidade virtual), geralmente movimentando a própria cabeça para que o aparelho calcule quais as imagens que devem ser reproduzidas no aparelho. Quais imagens serão escolhidas para exibição é uma função, pelo menos, da posição e orientação da cabeça do usuário.

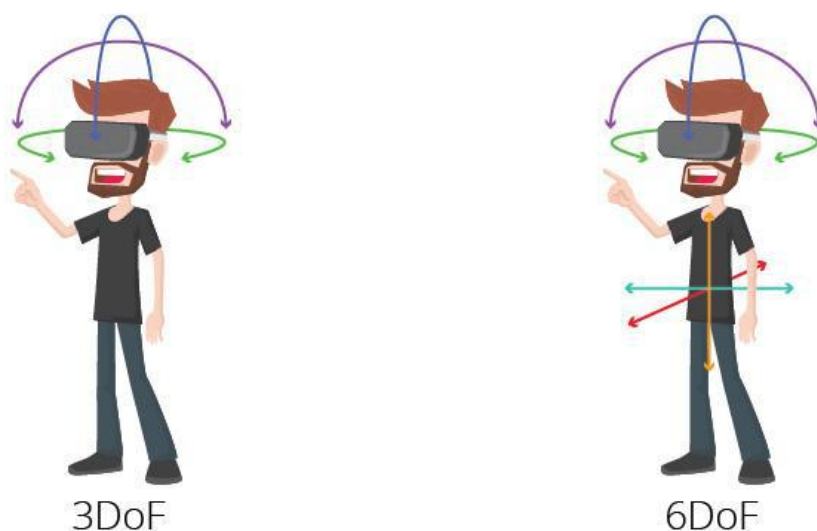
- Quanto mais o campo de visão for coberto por uma tela, aproximando-se do campo de visão possuído no mundo real, maior é a sensação de imersão [19]. Logo, ter o ambiente virtual exibido em todo o campo de visão dá ao usuário uma profunda sensação de estar imerso.

Quando essas três características são implementadas com sucesso, o usuário pode alcançar um estado de presença.

A realidade virtual pode ser classificada pelo grau de liberdade (*degree of freedom*) permitido ao usuário. Caso somente a rotação da cabeça seja rastreada (nos eixos x, y e z), dizemos que o sistema tem 3 graus de liberdade (3DoF, 3 Degrees of Freedom). Caso além da rotação da cabeça, a posição da cabeça do usuário no espaço também seja rastreada (nos eixos x, y e z), dizemos que o sistema tem 6 graus de liberdade (6DoF, 6 Degrees of Freedom), como mostra a Figura 1.

Figura 1: Diferenças do rastreamento entre 3DoF e 6DoF.

Fonte: *3DoF vs 6DoF. Which is better? RedboxVR. 2022*



Sistemas do segundo tipo oferecem uma capacidade de imergir o usuário muito maior do que o primeiro, pois permite a movimentação e a interação similares, como seriam feitas no mundo real. Em sistemas do primeiro tipo a movimentação e a interação são realizadas por botões, joysticks, ou pelo olhar (explicado mais detalhadamente no Capítulo 3 e na Seção 7.2).

A mesma tecnologia que permite o rastreamento em 6DoF para a cabeça pode ser aplicada aos controles, permitindo que suas rotações e posições no espaço sejam conhecidas. Isso permite que um usuário interaja de forma mais realista com o ambiente virtual pois pode mover os controles exatamente como moveria suas mãos. Por exemplo, caso queira pegar um objeto virtual, o usuário move o controle até que a mão de seu avatar virtual toque o objeto, então ao pressionar um botão, o objeto é pego.

A qualidade de uma experiência e a magnitude da sensação de presença que pode ser ocasionada em realidade virtual é ditada por algumas características, entre elas: a qualidade gráfica, o grau de liberdade (3DoF ou 6DoF), a latência do rastreamento, o campo de visão do HMD, entre outras.

Por último, é importante observar que a realidade virtual não se limita somente à visão. Estender o ambiente virtual aos outros sentidos aprofunda a imersão da pessoa, porém além da visão somente a audição pode ser considerada em termos práticos ao desenvolver aplicações em realidade virtual. O tato somente pode ser influenciado por meio de vibrações de um controle, tendo impacto bastante limitado na experiência final.

### 2.1.2 Imersão

Até então, um termo foi utilizado sem uma definição precisa: imersão. Esta seção tem como objetivo defini-lo. Antes de descrevê-lo, é necessário definir um outro conceito: contingências sensorio-motores (*sensorimotor contingencies*). Estas se referem a ações que podem ser realizadas para perceber o mundo ao seu redor, por exemplo mover sua cabeça para mudar seu ponto de vista, ou agachar para poder ver o que está debaixo de uma mesa [1].

A imersão é o conjunto de contingências que são permitidas ao usuário pelo sistema. Esse conceito pode ser utilizado para medir a imersividade de um sistema. Um sistema de realidade virtual que permite ao usuário mover seu corpo inteiro e girar sua cabeça e ter isso refletido no ambiente virtual é mais imersivo do que um sistema que somente reflete a rotação da cabeça.



### 2.1.3 Ilusões da Realidade Virtual

O grau de realidade sentido pelos usuários, e por consequência o quão realistas serão suas respostas a eventos, está atrelado a três ilusões que ocorrem no usuário ao ser imerso e interagir com um ambiente virtual. Essas ilusões são: ilusão de local (Place Illusion), ilusão de plausibilidade (Plausibility Illusion), e ilusão de encarnação (Embodiment Illusion). Essa sensação de realidade, de “estar lá” (no ambiente virtual) recebe o nome de presença, e é uma função dessas três ilusões, que serão explicadas abaixo. Ela também pode ser descrita como a medida na qual um participante responde a eventos no mundo artificial ao invés de eventos no mundo real [3].

Os conceitos de imersão e presença se referem a ideias distintas: a imersão está relacionada ao que um sistema permite ao usuário, a presença é relacionada à resposta do indivíduo ao sistema. Um alto nível de imersão cria uma base a partir da qual a presença possa ocorrer.

A ilusão de local é geralmente descrita como a percepção do ambiente virtual como real. O usuário, ao estar imerso no ambiente virtual, sabe que o ambiente é falso, que na realidade ele está no mundo real, mas ainda sim seu subconsciente o reconhece como verdadeiro. Essa ilusão é uma resposta a um alto nível de imersão em um ambiente virtual, e sua profundidade será limitada pelas contingências sensorio-motores [1]. Por exemplo, se o sistema não rastrear os movimentos do corpo do usuário, ele terá mais dificuldade em acreditar que está no ambiente virtual. Enquanto a correlação entre os movimentos do usuário e mudanças nas imagens refletindo os movimentos for mantida, a ilusão não será quebrada. Relatos dessa ilusão mostraram melhoras quando usuários podiam andar (ou simular andar parados no lugar) para mover-se quando comparado com movimentação por controles [4].

A ilusão de plausibilidade se refere à crença de que o cenário virtual está ocorrendo realmente. Enquanto a ilusão de local se refere ao ambiente, esta se refere aos eventos que ocorrem dentro do ambiente. Sua profundidade é limitada pela capacidade do sistema de produzir eventos que envolvam o usuário e a credibilidade do cenário [1]. Ao serem inseridos em um ambiente virtual, usuários podem interagir com ele. A sensação de naturalidade dada ao usuário pelas reações do ambiente a essas iterações influenciará na profundidade da sensação de presença. Ou seja, o resultado final da experiência dependerá tanto da qualidade da construção

do ambiente quanto das expectativas do usuário. Por exemplo, se o usuário está em um ambiente virtual que simula um bar e decide jogar uma garrafa contra a parede, caso a garrafa não quebre, ou nenhuma das pessoas presentes no bar reajam a este ato, a ilusão de que os eventos são reais será prejudicada. Parâmetros que impactam a profundidade da ilusão são:

- eventos não causados pelo usuário o envolvem (por exemplo, um personagem da cena direciona uma pergunta a ele);
- o mundo responde a suas ações (por exemplo, se o usuário estende a mão para cumprimentar um personagem e ele aperta sua mão);
- credibilidade, isto é, atender as expectativas dos usuários.

Ao alcançar essas duas ilusões, usuários tendem a responder como fariam na vida real [1].

A ilusão de encarnação é única à realidade virtual. A ilusão de encarnação pode ocorrer se um corpo virtual substituir seu corpo real no ambiente virtual. Sua profundidade será ditada pela correspondência entre movimentos do corpo real e corpo virtual. Somente a existência do corpo virtual já é suficiente para provocar uma fraca ilusão, mas para alcançar maiores magnitudes deve-se manter os movimentos do corpo virtual sincronizados aos do corpo real.

Para que essas ilusões sejam preservadas, o ambiente virtual deve ser construído de forma a alcançar as expectativas do usuário de como o mundo deveria responder às suas ações.

Quebras na presença são temporárias. A ilusão de local, sendo perceptual, voltará após certo tempo depois de ser quebrada contanto que o que causou a quebra não ocorra novamente. Por exemplo, se um usuário estiver em um parque virtual e se aproximar bastante de uma árvore, poderá ver os pixels que a formam, quebrando a ilusão de local. Mas como essa ilusão é dependente das contingências sensório-motores (que sempre estão presentes), ela eventualmente retorna. No exemplo, contanto que o usuário se afaste da árvore, a ilusão retornará após certo tempo. A ilusão de plausibilidade, por ser mais cognitiva, não retornará. Caso um usuário tente interagir com um personagem e este não reaja às suas ações, o usuário terá sua ilusão de plausibilidade permanentemente danificada. Como não há motivo para o usuário passar a pensar que o personagem irá começar a reagir de forma realista repentinamente, a ilusão continuará quebrada [2].

## 2.2 OBJETOS

### 2.2.1 Cartão Perfurado

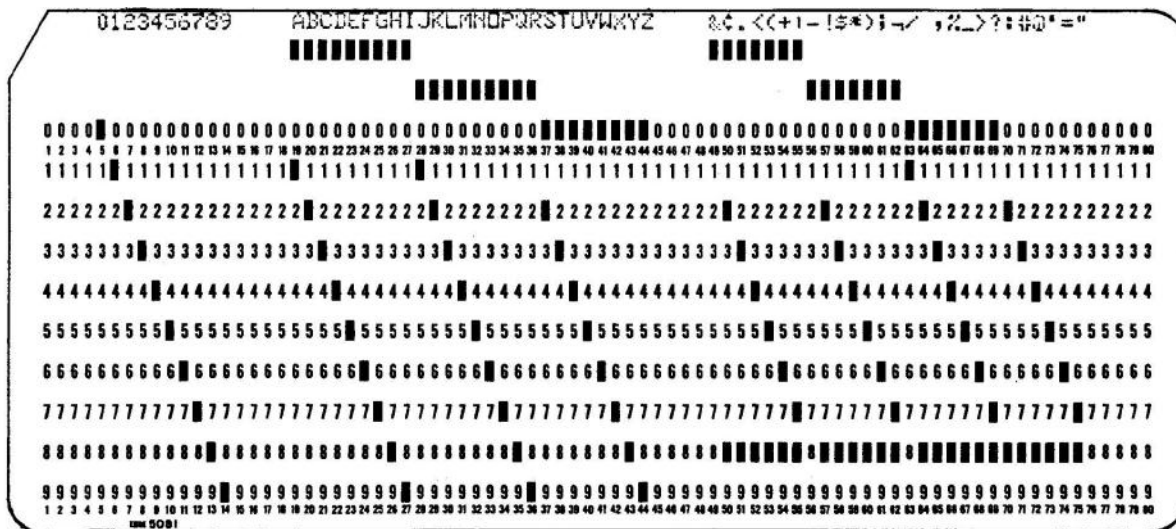
Um cartão perfurado armazena dados pela presença/ausência de perfurações em posições predefinidas. Neste trabalho foi utilizado o cartão do padrão IBM 5081, que possui 80 colunas e 12 linhas. A Figura 2 ilustra todas as perfurações que podem ser feitas pelo perfurador de cartões IBM 029 em uma imagem de um cartão desse padrão. Cada coluna representa um caractere dependendo das linhas nas quais as perfurações foram feitas. Logo, cada cartão pode armazenar 80 caracteres.

Muitos dos primeiros computadores utilizavam cartões perfurados como meio de entrada para códigos e dados. Além de seu uso na área da computação, foram bastante utilizados na indústria de processamento de dados. O IBM 029 era capaz de imprimir texto no topo do cartão para facilitar a leitura por humanos.

Figura 2: Cartão de 80 colunas com a codificação de caracteres da IBM 029.

O caractere representado por cada coluna está impresso acima da coluna.

Fonte: *Reference Manual IBM 29 Card Punch. IBM. 1970*



### 2.2.2 Perfurador de Cartões IBM 029

É uma máquina que perfura cartões em posições dependendo do que fosse digitado pelo usuário. Assim, teve papel central no armazenamento de dados durante o século XX, sendo amplamente empregado na indústria de processamento de dados. Seu uso decresceu à medida que os cartões perfurados foram substituídos como meio de armazenamento por suas inconveniências, entre elas não poderem ser reutilizados e somente serem capazes de armazenar 80 caracteres.

Algumas diferentes versões do IBM 029 (Figura 3) estiveram em produção, cada uma com diferentes funcionalidades, como a habilidade de imprimir o caractere perfurado em uma coluna no topo dela. Este trabalho simula a A22, uma versão com teclado de 64 caracteres com habilidade de impressão.

Figura 3: IBM 029.

*Fonte: IBM 029. DVQ*



### 3 TECNOLOGIAS UTILIZADAS

Para a realização deste trabalho, o auxílio de algumas ferramentas foi preciso. A aplicação foi desenvolvida para que pudesse ser utilizada por smartphones como a HMD. Um Samsung Galaxy A50 foi utilizado para testar a aplicação, porém qualquer smartphone capaz de executar aplicações em realidade virtual poderá executar a aplicação construída. Um controle bluetooth genérico foi utilizado para permitir uma melhor interação do usuário com o ambiente virtual. A aplicação não pode ser utilizada sem algum controle. O ambiente virtual foi criado com auxílio do software Unity, geralmente utilizado no desenvolvimento de jogos, enquanto que a criação de modelos 3D foi feita com auxílio do software Blender, especializado para tratar de modelos.

#### 3.1 HMD

*Head-Mounted Display* (HMD) é um visor fixado à cabeça do usuário com imagens que são atualizadas correspondendo aos seus movimentos, coloquialmente conhecido como “óculos de realidade virtual” [20].

A primeira decisão a ser tomada antes de iniciar o desenvolvimento deste trabalho era o público alvo do produto final. Levando em consideração as propostas de postar o programa online para que a exibição possa ser vista até mesmo por pessoas que não podem acessar o museu, e de fomentar interesse do público em visitar o museu físico, o desenvolvimento deve ter como meta alcançar o maior número de pessoas possível.

O smartphone é o dispositivo compatível com realidade virtual possuído pelo maior número de pessoas, e está projetado para continuar aumentando nos próximos anos [9]. Por essa razão, foi tomada a decisão de desenvolver a simulação de forma que possa ser executada em smartphones, com testes realizados na plataforma Android. Além desse ponto, como as HMD mais avançadas são capazes de executar programas feitos para smartphones sem problemas de performance mas o contrário nem sempre é verdade, o processo de adaptação para distribuição para outros dispositivos seria mais fácil partindo de um produto para smartphones. Nada seria perdido na experiência do usuário pois o HMD mais avançado é mais imersivo do que o smartphone e, portanto, é capaz de rastrear seus movimentos tão bem quanto o smartphone.

Com aparelhos auxiliares de baixo custo e amplamente acessíveis, smartphones podem cumprir o papel de HMDs, porém com performance reduzida. O hardware precisa manter a latência do rastreamento da cabeça do usuário (da rotação e da posição) dentro de um limite para que não cause problemas como náusea. Este limite não é de simples determinação, variando profundamente de pessoa para pessoa, com algumas mal percebendo uma demora de 100ms enquanto outras percebem 4ms [10]. Geralmente é recomendado manter a latência dentro de 20 ms, significando uma performance maior ou igual a 50 frames por segundo, um fato problemático no desenvolvimento de realidade virtual para dispositivos móveis. Um estudo analisando a latência em diversas plataformas de realidade virtual descobriu que a latência da maior parte dos dispositivos móveis podia chegar a 100 ms, e raramente estava abaixo de 50 ms [10]. Logo, otimizações precisam ser feitas para contornar o problema da performance. Para testar este trabalho, um Samsung Galaxy A50 foi escolhido como HMD.

A plataforma Android tem um limite de 500.000 polígonos (dentro do campo de visão do usuário) por frame para manter um framerate consistente e a níveis superiores àquele requisitado para não impactar negativamente a experiência do usuário.

Uma distinção deve ser feita entre a resolução na qual a aplicação é vista e o nível de detalhe do ambiente. A resolução dependerá somente da tela pela qual o ambiente é visto e será constante. É medida em pixels (720p, 1080p), e não é uma preocupação para o desenvolvedor pois mostrar uma imagem já renderizada não é computacionalmente exigente. O nível de detalhe é uma característica do ambiente criado e é o que ditará o quão trabalhoso será renderizar tal ambiente. Esta propriedade pode ser descrita pelo número de polígonos no campo de visão do usuário, levando em consideração que o processo de renderização envolve gerar imagens destes polígonos a partir de dados descritivos de objetos.

Como as HMDs tem duas telas, são precisas duas câmeras, cada uma representando um olho do usuário. Cada câmera renderiza uma imagem distinta, e após a renderização a imagem tem que ser distorcida de uma forma que neutralize a distorção das lentes do HMD (Figura 4). Logo, o hardware tem mais do que o dobro da carga que teria para processar uma cena 3D para um computador normal. Levando em consideração o limite de 500.000 polígonos mencionado acima, os objetos vistos não podem ter mais do que 250.000 polígonos para que possam ser renderizados duas vezes de forma consistente.

Figura 4: Uma imagem pré e pós distorção para exibição em uma HMD.

*Fonte: NVIDIA Explains Pascal's 'Lens Matched Shading' for More Efficient VR Rendering.*

*Lang, Ben. Road to VR. 2016*



### 3.2 CONTROLE

O controle utilizado (Figura 5), conectado por bluetooth ao smartphone, possui um joystick e quatro botões que possibilitam a interação do usuário com o ambiente, por exemplo, permitindo movimentar-se por ele.

Figura 5: Controle utilizado

Fonte: *VR Box Remote. Tech4You*



### 3.3 UNITY

Unity é um software de desenvolvimento de jogos que fornece uma ampla estrutura e funções comuns para acelerar e padronizar seu desenvolvimento [21]. Cada produto criado com o Unity é um conjunto de cenas, cada uma constituída de um conjunto de objetos que interagem uns com os outros. Um exemplo de programação orientada a componentes, cada objeto é um recipiente de componentes, permitindo que funcionalidades sejam adicionadas ou removidas de objetos por meio deles. Se o componente oferece um serviço, uma interface é adotada para que o serviço possa ser utilizado sem precisar entender como ele é implementado.

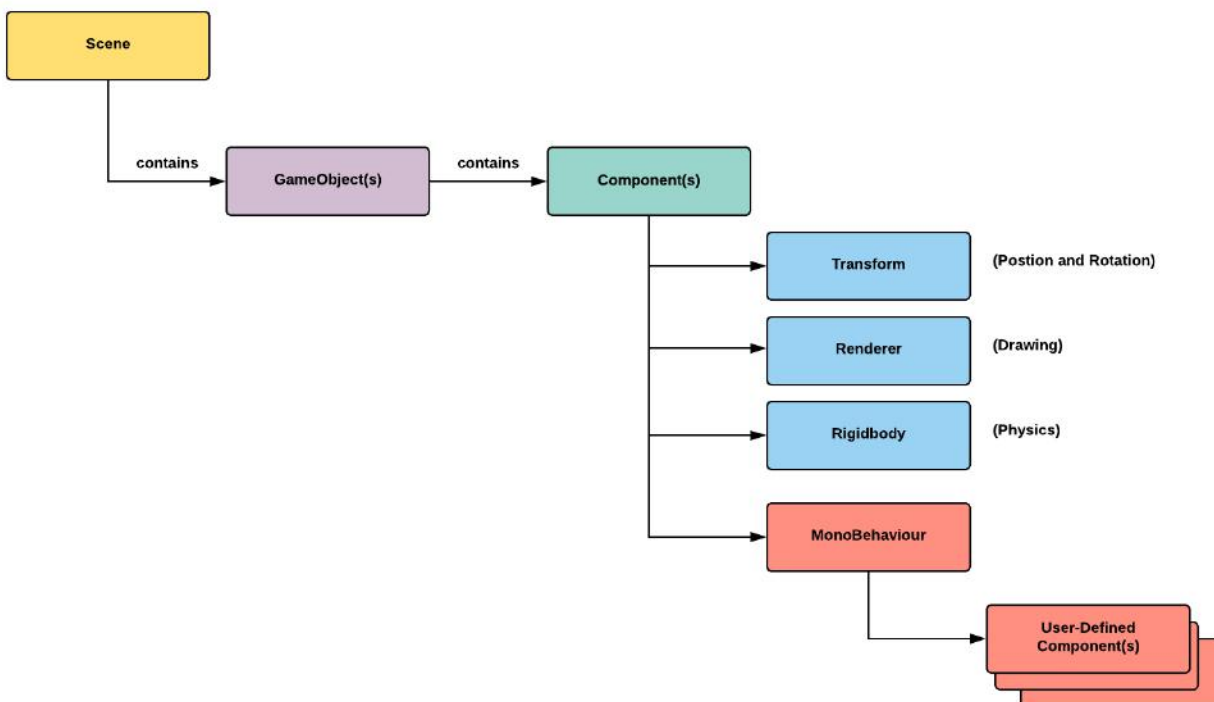
Por exemplo, neste projeto temos dois objetos principais: o keypunch IBM 029 e um cartão perfurado. Como ambos são objetos sólidos, não devem ser capazes de atravessar um ao outro. Adicionando o componente de um script que impeça a malha poligonal do objeto de fazer interseção com a malha de qualquer outro objeto, cumprimos esse requerimento. Dessa forma a produção de aspectos do software como a física do jogo, renderização, entre outras, pode ser



padronizada por abstrações, permitindo que os desenvolvedores foquem sua atenção nos aspectos de mais alto nível do desenvolvimento como o design do software.

Figura 6: Composição de uma cena Unity.

Fonte: *Unity's Architecture. Packtpub. 2019*



A Figura 6 apresenta as principais funções de componentes: quais transformações aplicar aos objetos, de que forma eles devem ser renderizados e como a física deve ser aplicada ao objeto. Em mesmo nível de importância está a classe `MonoBehaviour`. Isso se dá pois é a classe da qual todos os scripts (que devem ser escritos em C#) em Unity derivam. Ela fornece a *framework* que permite anexar scripts a objetos.

Unity foi escolhido por duas razões: sua base de usuários e o plugin XR Interaction Toolkit. Sua base de usuários é a maior entre os ambientes de desenvolvimento para realidade virtual, e portanto a quantidade de recursos disponíveis para auxiliar o desenvolvimento é a maior. O plugin contém uma série de componentes que dão suporte a funções vitais para o desenvolvimento de aplicações em realidade virtual. Para este trabalho, os principais componentes foram um gerenciador de interações e o *Cross Platform Input Manager*. O primeiro

permite executar interações entre objetos, fornecendo interatividade ao ambiente criado. O segundo fornece um mapeamento único para input de diferentes controles, tornando possível escrever código que capte entradas independentemente do tipo de controle utilizado, facilitando o desenvolvimento de software para diferentes plataformas. Utilizando o *Cross Platform Input Manager*, é possível referenciar os botões no controle da Figura 5 por meio de *handles* comuns a qualquer dispositivo de entrada (inclusive teclados). Logo, a execução da simulação em outro hardware não será impactada por diferenças nos controles. A Tabela 1 relaciona a *handle* ao botão do controle e à tecla de um teclado comum. A Figura 7 apresenta um trecho do código escrito para este projeto, onde uma instância de um cartão é criado caso a entrada “Fire1” seja reconhecida.

Tabela 1: Relação entre teclas e *handles* do *CrossPlatformInputManager* utilizadas no projeto.

Handle	Controle	Teclado
Jump	D	Espaço
Fire1	C	Ctrl
Fire2	A	Alt
Fire3	B	Cmd

Figura 7: Exemplo de uso de uma *handle* em código.

Ao reconhecer a entrada *Fire1*, um novo cartão é criado e instanciado no ambiente.

```
if(CrossPlatformInputManager.GetButtonDown("Fire1") && ordinal_number < num_max_cards)
{
    card_instance = Instantiate(card_model, spawnPosition.position, spawnPosition.rotation);
    card_instance.GetComponent<cardPerfurator>().card_ordinal_number = ordinal_number;
    ordinal_number++;
}
```

### 3.4 BLENDER

Unity, por ter como objetivo principal ser um *game engine*, não apresenta funcionalidades de modelagem. Isto é, embora permita aplicar algumas transformações básicas (translação,

rotação, e escalonamento), qualquer edição mais detalhada do modelo terá de ser feita em outro software.

Como será visto em uma seção posterior, foi necessário modificar a geometria de certos modelos e criar outros, ações impossíveis de serem realizadas no Unity. Blender, por outro lado, oferece ferramentas justamente para esses cenários [22]. Neste trabalho ele foi utilizado para criar o modelo do cartão perfurado e para modificar o modelo do IBM 029.

## 4 DESENVOLVIMENTO

Nessa seção será descrito todo o processo do desenvolvimento da exibição em realidade virtual, desde o planejamento, até os detalhes das otimizações.

### 4.1 METAS

O objetivo do desenvolvimento foi produzir uma experiência na qual um usuário pudesse utilizar um perfurador de cartões. Para tal, uma simulação de um perfurador de cartões deveria ser desenvolvida, além de um ambiente virtual que retratasse um ambiente onde o perfurador seria visto naturalmente. Assim, buscou-se criar uma “sala de computação”, uma sala onde diversas máquinas importantes na área da computação seriam vistas, uma delas o próprio perfurador. Para aumentar a sensação do usuário de que o ambiente poderia ser real, seria importante que ele pudesse navegar por este ambiente. Nenhuma das outras máquinas no ambiente seriam simuladas (fora do escopo deste trabalho), logo a capacidade de interação com elas não precisaria ser programada. Além desses requerimentos, seria preciso que o usuário pudesse interagir com o perfurador, para perfurar uma mensagem de sua escolha, e interagir com os cartões perfurados, para que pudesse analisá-los.

O design precisava ser receptivo e adaptável ao feedback de usuários. O objetivo do desenvolvimento foi chegar a um ambiente que alcançasse as expectativas dos usuários e, portanto, sucessivos testes e adaptações foram necessárias para ajustar o software. Não é conhecido de antemão como os usuários irão interagir com o ambiente virtual e não foi possível preparar o programa para contornar todas as formas que um usuário pode prejudicar sua presença. Logo, seu feedback pode ser necessário mesmo em estágios tardios do desenvolvimento. Por essas razões, os protótipos foram rotineiramente testados com dois indivíduos, além do próprio desenvolvedor.

Por meio da simulação é desejado também testar de que formas a sensação de presença pode impactar a experiência. Por exemplo, embora tenham sido realizadas em um ambiente específico de ensino e não em um ambiente virtual de um museu, pesquisas passadas [17] correlacionam a imersão com a qualidade da aprendizagem em simulações em realidade virtual. Para que esses testes sejam realizados, é preciso que a simulação alcance alguns requisitos

mínimos de design para que possa fornecer uma sensação de presença significativa. Porém, deve também ser definido de forma suficientemente vaga para permitir adaptações durante o desenvolvimento. Foi decidido, então, que o produto final deveria ter, minimamente, as seguintes características:

- Permitir ao usuário se movimentar pelo ambiente criado;
- Permitir ao usuário perfurar um cartão com o IBM 029;
- Permitir ao usuário manusear um cartão perfurado;
- Apresentar um ambiente realista;
- Apresentar um avatar como substituto virtual do corpo do usuário.

Tudo o que não estiver incluso (ou detalhado) nessa lista não foi decidido antecipadamente. Por exemplo, não foi definida a forma como a movimentação do usuário seria implementada neste estágio do planejamento.

Para alcançar uma versão definitiva que atingisse esses requisitos, sucessivos protótipos tiveram que ser desenvolvidos, que foram julgados pela sensação de presença que eles eram capazes de fornecer. Embora essa seja uma tarefa subjetiva, alguns dos componentes que influenciam a sensação podem ser medidos quantitativamente, o FPS (frames por segundo) alcançado no ambiente, por exemplo. Otimizações visando a melhoria dessas métricas foram implementadas em diversos estágios do desenvolvimento e serão detalhadas em seções posteriores.

Assim, as primeiras preocupações eram conseguir os modelos necessários. Obtidos os modelos, cenas rudimentares poderiam ser construídas para que a prototipagem pudesse ser iniciada.

## 4.2 MODELOS

### 4.2.1 IBM 029

O modelo do IBM 029 pôde ser adquirido na internet (Figura 8). Esse modelo não foi criado visando ser utilizado em uma plataforma com pequeno poder gráfico, evidente pelo seu

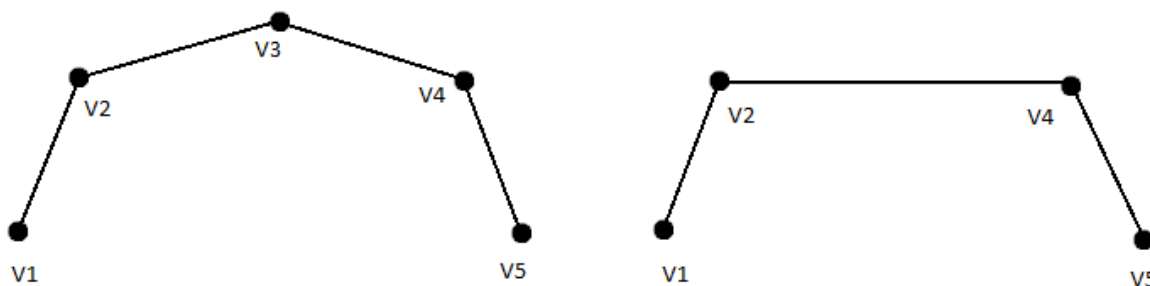
grande número de vértices (205.623), arestas (393.463) e triângulos (352.342), lembrando que quanto maior o número de polígonos, mais lenta será a renderização do objeto. Diretrizes comuns para desenvolvimento da realidade virtual para smartphones aconselham manter o número de triângulos limitado a 300.000, algumas até têm como limite 100.000 triângulos. Uma performance constante não poderia ser mantida se este modelo fosse utilizado: assim que ele entrasse no campo de visão, o FPS médio despencaria, somente para voltar a um bom patamar assim que ele saísse do campo de visão, dando uma experiência inconsistente e impedindo qualquer sensação de presença de se instalar.

Figura 8: Modelo original do IBM 029 [5].



Para contornar isso, utilizamos um algoritmo modificador de modelos chamado *decimate*, que permite remover vértices e arestas desnecessárias, reduzindo o número de faces. Seu uso pode modificar o formato do modelo, então foi utilizado até que as distorções se tornassem aparentes. O algoritmo funciona unindo em uma face, faces próximas que tenham ângulos entre elas menores do que o ângulo dado como parâmetro. Uma ilustração simples do efeito deste algoritmo é mostrado na Figura 9.

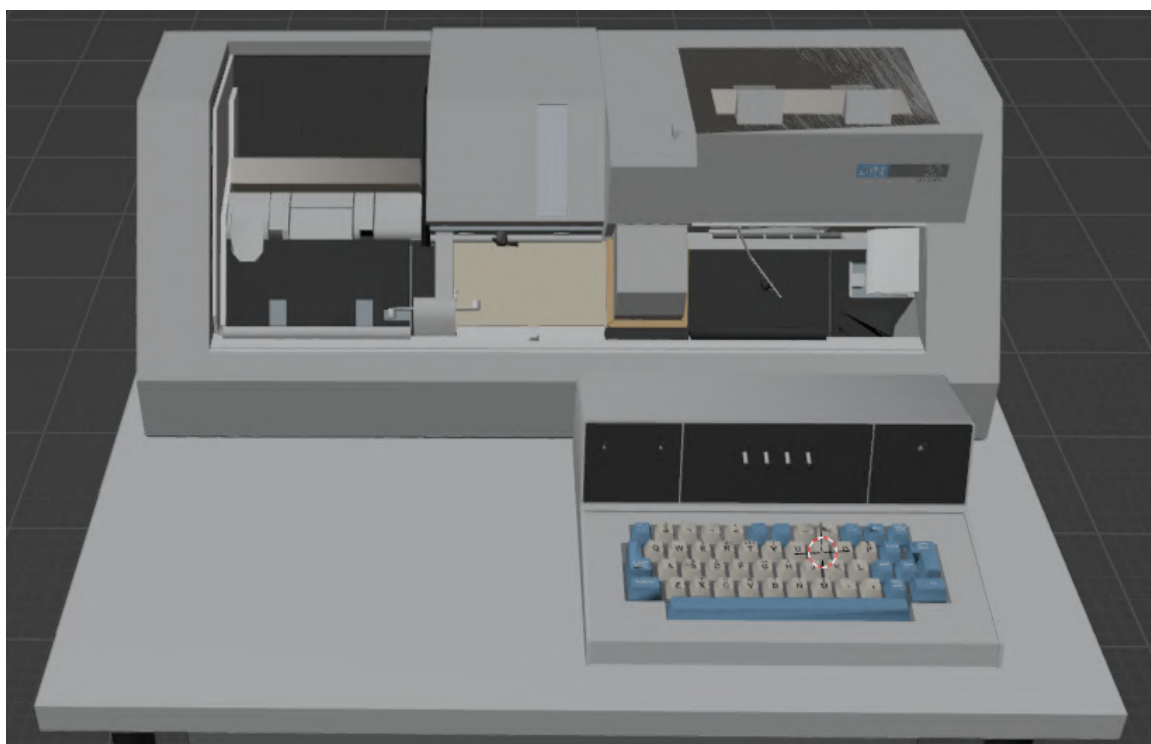
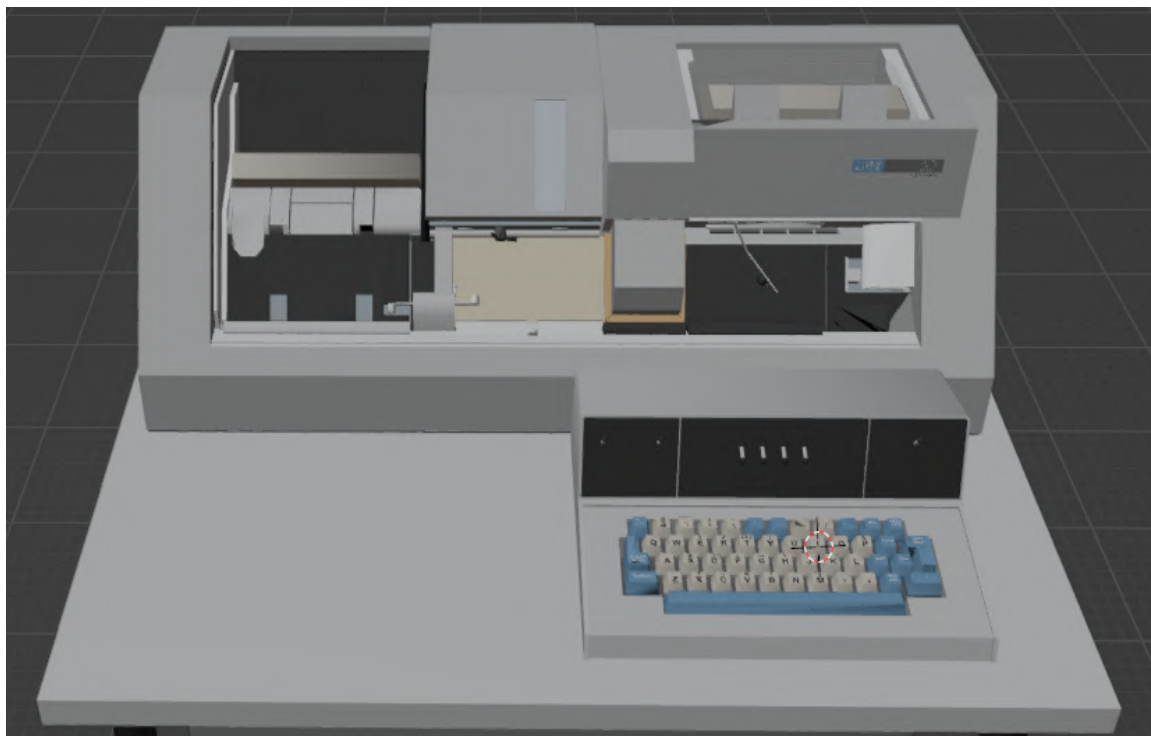
Figura 9: Simplificação de um modelo pelo algoritmo *decimate*. A imagem à esquerda mostra parte do modelo antes da simplificação, a imagem à direita mostra o resultado.



Experimentando com diversos ângulos, o valor de  $33^\circ$  foi alcançado como ponto ótimo entre diminuição do número de triângulos e deformação do modelo. O resultado da simplificação está na Figura 10a. Não há grandes diferenças entre os modelos. Considerando também que a maioria dos possíveis visitantes ao museu de computação não conhecem um 029 detalhadamente, e com isso não poderão notar as pequenas discrepâncias. Logo, a aplicação do algoritmo não impactará negativamente a simulação. Mesmo que o modelo não seja visualmente muito diferente do original, uma grande simplificação pôde ser alcançada: o modelo é agora composto de 90.729 vértices, 131.106 arestas e 127.626 triângulos. O novo modelo possui quase um terço do número de triângulos do modelo original, e agora está dentro do limite para o número máximo de triângulos nas diretrizes seguidas.

Uma simplificação maior por este caminho produziu impactos negativos. Aplicando o algoritmo com um ângulo de  $34^\circ$ , o resultado final possuía claros erros em sua malha, tais como o tapamento do *card feed*, a área onde os cartões ainda não perfurados são inseridos na máquina, por uma nova face (canto superior direito na Figura 10b).

Figuras 10a e 10b: Modelo modificado com ângulo de  $33^\circ$  (10a) e com  $34^\circ$  (10b). Um erro pode ser visto com o tapamento do *card feed* na Figura 10b (canto superior direito).



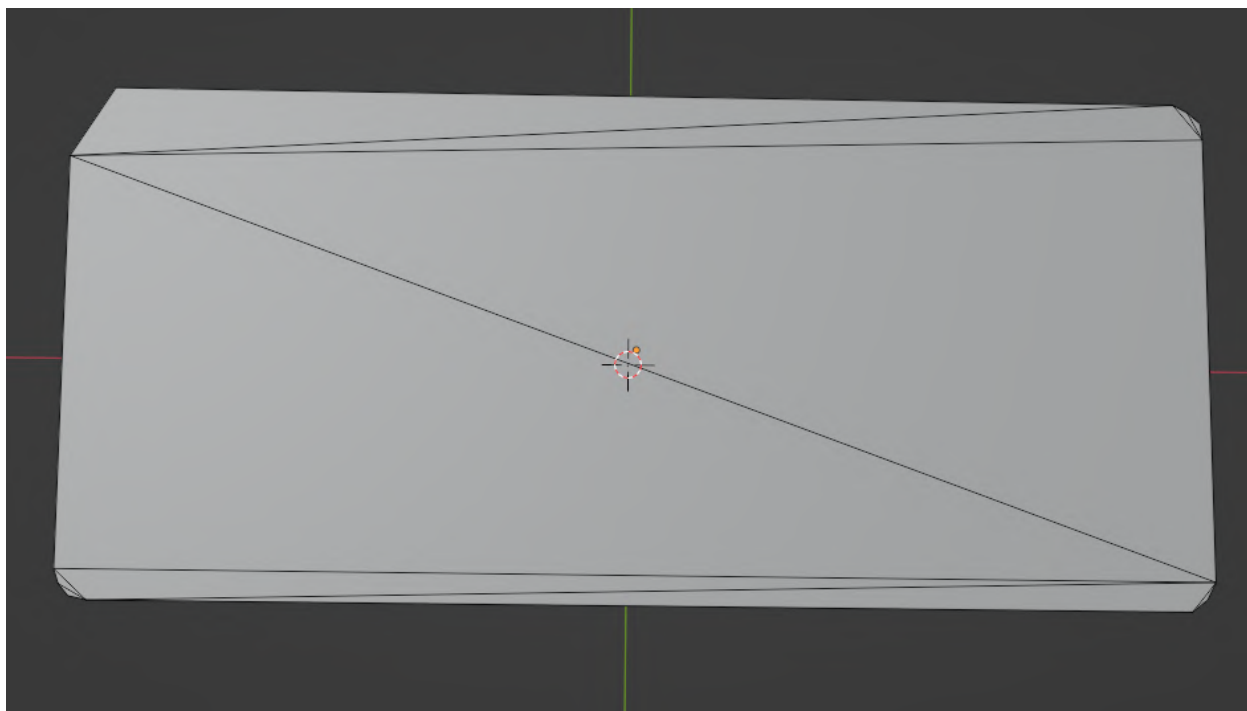


O propósito desta simplificação inicial do modelo, obter um modelo com qualidade suficiente para permitir o início do desenvolvimento, foi atingido. Posteriormente, em estágios mais avançados do desenvolvimento, seria julgado se outras simplificações seriam necessárias, caso o modelo causasse quedas de performance. Porém, como será visto, modificações adicionais ao modelo não foram necessárias.

#### 4.2.2 Cartão Perfurado

O modelo de um cartão perfurado não pôde ser adquirido, então teve que ser criado do zero. Primeiramente, uma imagem de um cartão perfurado genérico foi importada no software *blender* e vértices foram posicionados em torno da imagem e conectados, de forma a criar uma malha poligonal 2D (Figura 11).

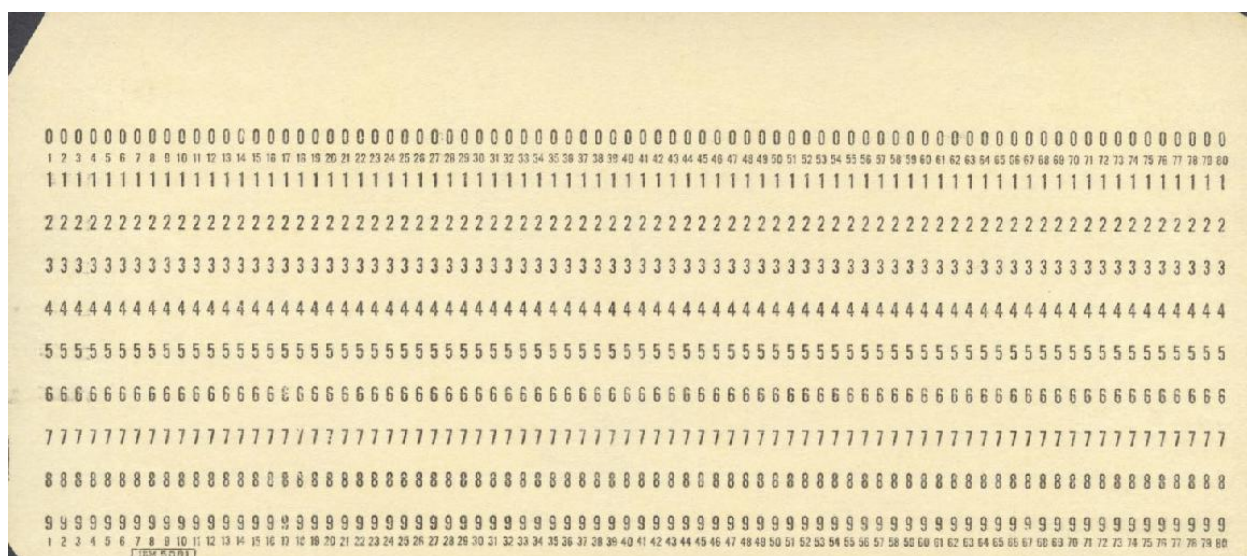
Figura 11: Malha poligonal inicial do cartão.



A imagem importada foi mapeada como textura ao modelo. O verso do cartão foi criado com a mesma malha, porém texturizada somente com uma cor sólida. O modelo resultante pode ser visto na Figura 12.

Porém, o modelo do cartão deve ser modificado durante a execução do programa para que as perfurações possam ser feitas (a única outra opção seria ter disponíveis modelos distintos para cada possível combinação de perfurações, e trocar os modelos à medida que elas forem realizadas, desperdiçando o espaço e o tempo que seria necessário para criar todos esses modelos). Por essa razão, a malha do modelo foi alterada de tal forma que essas modificações fossem mais fáceis de serem realizadas. Isso foi feito definindo faces em sua malha para cada posição no cartão que poderia ser perfurada, permitindo que as posições das perfurações possam ser referenciadas exatamente. A definição das faces foi realizada por meio da criação de novas arestas na malha.

Figura 12: Cartão texturizado.

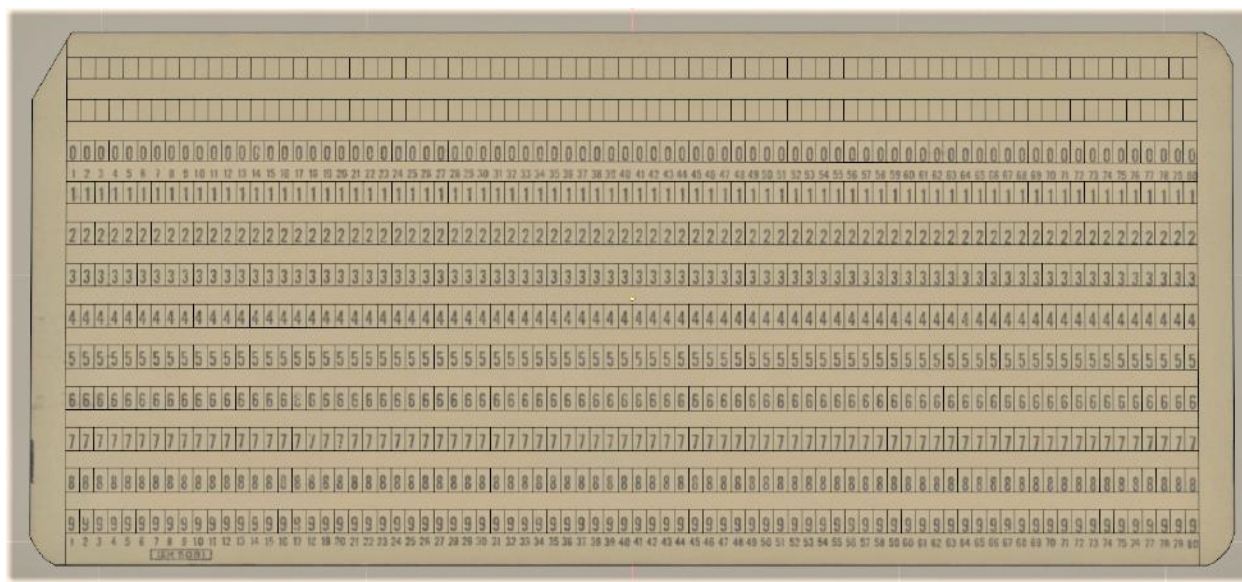


Cada coluna de um cartão pode ser perfurada em 12 posições diferentes, 10 delas são numeradas e 2 estão na parte superior do cartão, como indicado na Figura 13. Cada cartão também possui uma área em cima de cada coluna para manter a estrutura (essa área foi utilizada para imprimir em caracteres o que foi perfurado), e uma área abaixo de cada coluna pelo mesmo motivo. Essas faces criadas podem ser referenciadas por código, permitindo que elas sejam alteradas, ou mais especificamente para este projeto, removidas para simular a perfuração. Cada face que pode ser perfurada (os quadriláteros na Figura 13) é formada por dois triângulos, logo ao ser perfurada a mudança está sendo aplicada a esses triângulos. Como uma perfuração é feita foi desenvolvido posteriormente, e é explicada em mais detalhes na Seção 4.4.5.

Esse processo de criação de faces teve como consequência um aumento no nível de detalhes de cada cartão. No total, cada cartão possui 3938 vértices, 5888 arestas, e 7822 triângulos. Conseqüentemente, a performance da simulação pode ser impactada caso muitos deles estejam no campo de visão do usuário. Um limite para o número de cartões que podem estar no ambiente simultaneamente foi investigado em um estágio mais avançado do desenvolvimento, detalhado na Seção 4.4.7.

Em um cartão real as perfurações são feitas de tal forma que duas perfurações em colunas adjacentes em uma mesma linha deixariam uma área intacta entre elas. Porém, a divisão de faces da malha para acomodar essa área aumentaria demais seu número de triângulos, portanto não foi implementada. Esse modelo mais autêntico poderia levar o número de triângulos calculados por frame fora dos limites recomendados para a plataforma.

Figura 13: Cartão com malha visível para mostrar a posição das faces  
(A malha não é visível durante a execução do programa).



#### 4.2.3 Avatar

O modelo do avatar também pôde ser adquirido online (Figura 14). Nenhuma modificação foi necessária. O modelo raramente se encontra no campo de visão do usuário, somente entrando nele caso o usuário escolha olhar para baixo. A câmera tem posição fixada

entre os olhos do modelo. Ambos a câmera e o avatar fazem parte de um outro objeto que tem como componentes scripts que permitem sua movimentação. Dessa forma, o movimento de ambos, o modelo e a câmera são os mesmos, garantindo que o avatar sempre estará onde o usuário esperaria encontrá-lo. Foram dados ao avatar animações para quando o usuário estiver parado (na qual o avatar assume uma pose mais natural) ou se movendo, agregando ao realismo da cena. Somente um avatar masculino foi utilizado neste trabalho.

Figura 14: Avatar do usuário.



### 4.3 CENA

Em realidade virtual é importante que todo o ambiente seja realista pois, ao contrário de filmes, não se pode controlar para onde o usuário está olhando. No set de um filme nada que está

atrás da câmera será visto, logo sua aparência não importa. Em realidade virtual o ponto de vista pode ser movido para qualquer posição, logo toda a cena precisa estar pronta para ser vista. O máximo que pode ser feito é orientar o olhar dos usuários para áreas específicas no ambiente.

Esta orientação foi utilizada como ferramenta para que a simulação se conformasse com o limite de 300.000 triângulos em uma frame e as cenas foram criadas de forma que objetos densos em polígonos não precisassem ser renderizados simultaneamente, por exemplo posicionando-os em lados opostos de uma sala. Assim, para que um objeto entre no campo de visão do usuário, outro teria que sair dele.

Inicialmente, grande foco não foi dado ao design da cena: durante as primeiras iterações do protótipo, o IBM 029 simplesmente existia no meio do nada (Figura 15). Maior esforço só pôde ser dado a esta área após a implementação das funcionalidades principais; seu progresso é detalhado em diversos pontos da Seção 4.4.

## 4.4 PROTOTIPAGEM ITERATIVA

### 4.4.1 Primeiro Protótipo

O primeiro protótipo teve como principal objetivo estimar as restrições que seriam impostas à simulação pela plataforma para a qual seria desenvolvida. Isso dará base para o que é possível ou prático e minimizará futuros custos de redesign.

Para isso, nenhuma cena foi construída. O IBM 029 simplesmente foi posto no centro do mundo virtual. Sendo o único objeto a ser renderizado, seu impacto poderia ser medido mais precisamente. Medir algumas métricas eram de interesse durante a utilização desse protótipo:

- O realismo do objeto, buscando falhas em seu modelo que podem ter passado despercebidas durante o processo de simplificação;
- O FPS médio durante a cena, contrastando valores obtidos quando próximo ou longe do objeto, e também quando o objeto estiver dentro ou fora do campo de visão.

A medição da segunda métrica necessitou a escrita de um script que calculasse o valor do FPS em um dado instante e o pusesse na tela durante a execução da simulação. O valor é atualizado a cada 10 frames e é exibido em um objeto canvas que foi atrelado à câmera, seguindo os

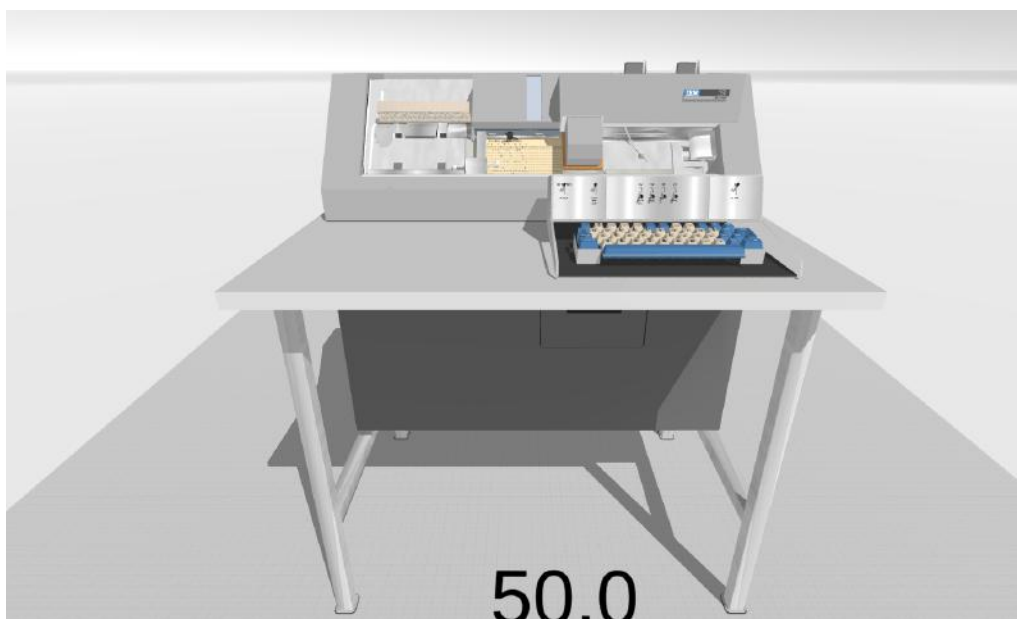
movimentos da câmera e sobrepondo qualquer outro objeto de forma a sempre estar visível (Figura 15).

Primeiramente foi utilizado o modelo original do IBM 029. Após a realização de testes com este modelo, ele foi substituído pelo modelo simplificado. Os testes consistiram somente da rotação do ponto de vista do usuário, simulando as ações que um usuário faria, para verificar o impacto que o IBM 029 traria ao FPS ao entrar no campo de visão.

Unity atualiza que partes de um objeto devem ser renderizadas baseando-se nos limites do campo de visão do usuário, para evitar chamadas desnecessárias a funções de renderização. À medida que objetos (ou partes deles) entram no campo de visão, um overhead que potencialmente pode impactar o FPS é causado.

Neste ponto do desenvolvimento, a movimentação do usuário pelo ambiente ainda não havia sido implementada.

Figura 15: Primeiro protótipo, demonstrando a exibição do FPS.



Resultados com o modelo original: com o objeto fora do campo de visão, o FPS se manteve constante em 60; com o objeto parcialmente no campo de visão, o FPS variou entre 54 e 60, geralmente em torno de 54; com o objeto inteiramente no campo de visão, o FPS variou entre 46 e 54, geralmente em torno de 50.

Resultados com o modelo simplificado: com o objeto fora do campo de visão, o FPS se manteve constante em 60; com o objeto parcialmente no campo de visão, o FPS variou entre 54 e 60, geralmente em torno de 58; com o objeto inteiramente no campo de visão, o FPS variou entre 54 e 60, geralmente em torno de 58. Essas velocidades para a atualização não são ruins e não provocam a quebra de presença.

Com o modelo simplificado a velocidade não só foi melhor mas também foi mais consistente, apresentando menor flutuação entre os valores máximos e mínimos. Foi feita a hipótese de que essa tendência deve se manter em futuros protótipos: o FPS atingido com ambos os modelos irá diminuir à medida que a simulação ficar mais complexa, porém melhores resultados serão atingidos com o modelo simplificado.

Nenhum dos modelos apresentou qualquer falha em sua geometria e nenhuma diferença era aparente na qualidade deles. Pelos resultados, pode-se ver que o protótipo pode servir como base para futuros desenvolvimentos. Porém, duas limitações tornaram-se aparentes: não foi possível examiná-los detalhadamente pois não era possível navegar pelo ambiente e aproximar-se dos objetos; o senso de presença foi dificultado pela falta de um ambiente à volta do usuário.

#### 4.4.2 Adições à Cena

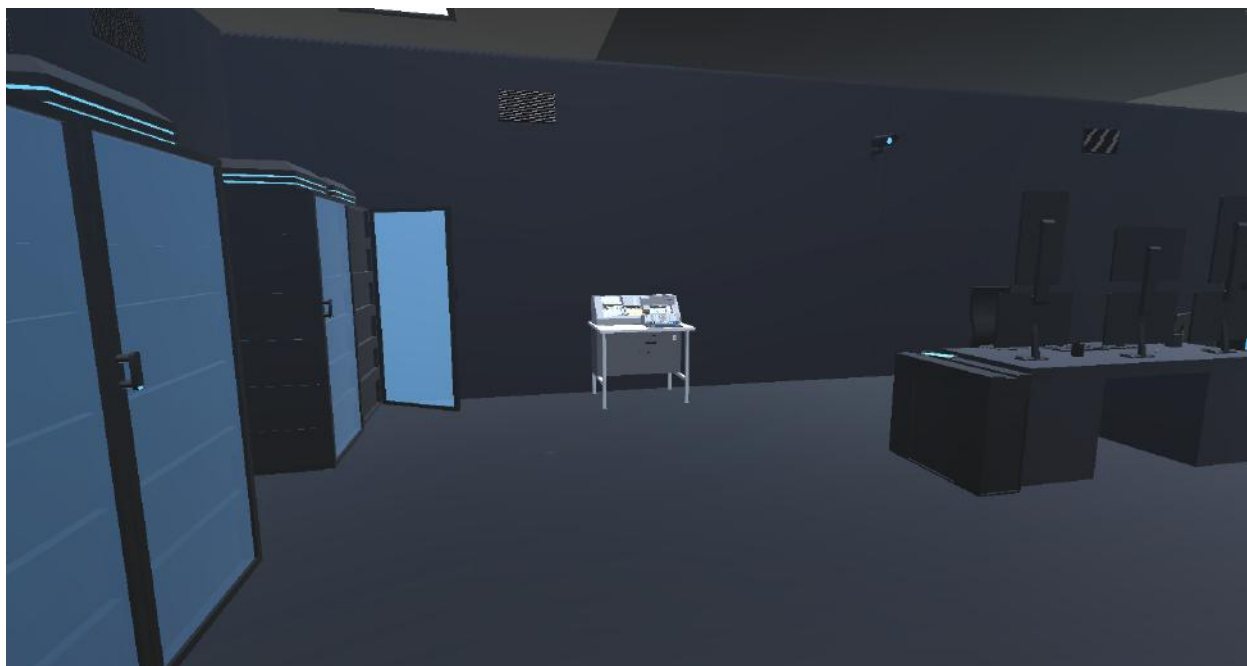
A partir do protótipo inicial, foram incluídas funcionalidades e objetos para a cena iterativamente. Primeiramente foram adicionados outros objetos formando um ambiente no qual inserir o IBM 029, para que seu impacto em uma cena mais realista pudesse ser julgado. Esses objetos serviriam apenas como substitutos até um estágio mais avançado do projeto, quando seriam trocados por modelos de máquinas tão antigas quanto o IBM 029 para aprofundar o senso de realismo da cena. Como mostra a Figura 16, esses objetos consistem de computadores e servidores modernos com poucos polígonos. Os mesmos testes foram realizados com a nova configuração.

Resultados com o modelo original: com o objeto fora do campo de visão, o FPS variou de 30 a 60 FPS, com média de 38; com o objeto parcialmente no campo de visão, o FPS variou entre 30 e 38, com média de 32; com o objeto inteiramente no campo de visão, o FPS variou entre 24 e 29, com média de 25.

Resultados com o modelo simplificado: com o objeto fora do campo de visão, o FPS variou de 30 a 60 FPS, com média de 46; com o objeto parcialmente no campo de visão, o FPS variou entre 35 e 40, com média de 37; com o objeto inteiramente no campo de visão, o FPS variou entre 31 e 35, com média de 33.

A presença de conjuntos de modelos reduz a capacidade de manter um alto FPS na cena. Porém, mesmo no pior caso utilizando o modelo simplificado, o FPS não fica abaixo de 30, um nível comum para aplicações em realidade virtual. Logo, o desenvolvimento pôde continuar partindo deste ponto. Utilizando o modelo original, o FPS não pode alcançar esse nível. Tendo confirmado o benefício para a performance pela simplificação do modelo, e não podendo observar grande diferença de qualidade entre o modelo original e o simplificado, o original deixa de ser usado pelo resto do projeto.

Figura 16: IBM 029, ao centro, em uma sala com computadores modernos.



Os novos objetos não possuem o mesmo nível de realismo do IBM 029. Somado ao contraste entre ele e as máquinas modernas, a instauração de uma sensação de presença demora a ocorrer. Mas como ainda não foi possível analisar os modelos de perto, a navegação pela cena foi a próxima funcionalidade a ser implementada.



#### 4.4.3 Movimentação

Existem duas formas principais de implementar a movimentação na realidade virtual: movimentação contínua e teleportação. A teleportação consiste em escolher um dentre vários pontos predefinidos para se posicionar, movimentando-se entre eles instantaneamente, sem passar por pontos intermediários. Desorientação ao se transportar para uma nova posição é comum com esse modo. A movimentação contínua permite que o usuário se mova de forma fluida, utilizando os inputs de ambos o *joystick* do controle e a rotação de sua cabeça, para decidir a direção e velocidade da locomoção. A movimentação contínua foi escolhida por se assemelhar mais a forma como nos movemos no mundo real.

Podendo mover a posição da câmera em tempo real, finalmente torna-se possível examinar os modelos. Mesmo quando analisados de forma minuciosa, o IBM 029 simplificado parece realista. Importaneamente, tão realista quanto o modelo original, implicando que o algoritmo de simplificação de sua geometria pôde trazer seus benefícios sem trazer quase nenhum impacto negativo à qualidade visual do modelo.

Mantendo o realismo do ambiente, para poder utilizar a máquina deve-se primeiro sentar à frente dela. Para tal, foi adicionado ao script que controla a movimentação do usuário um gatilho (ligado a um botão do controle) que, se acionado, fixa a posição (mas não a rotação) do usuário em frente à máquina e remove sua capacidade de se mover (simulando uma pessoa se sentando à frente da máquina).

#### 4.4.4 Cartões

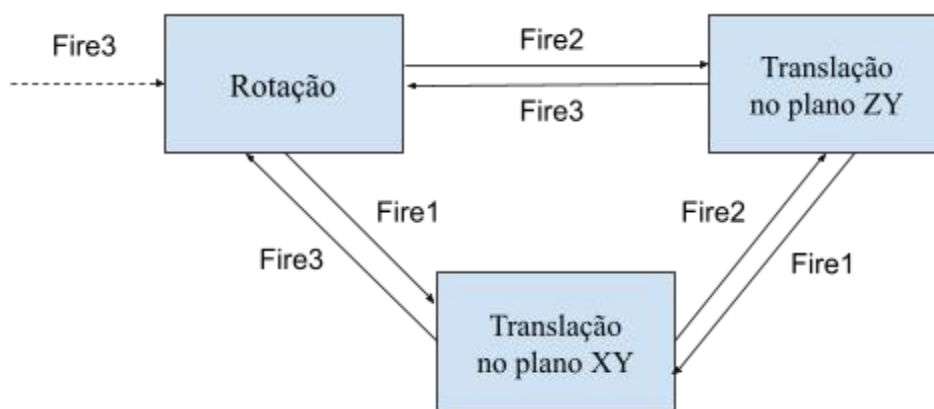
Foi decidido neste ponto incluir os cartões no ambiente. Inicialmente um único cartão foi utilizado pois somente era desejado julgar seu realismo e quão bem se encaixava na cena. Futuramente, vários cartões foram utilizados simultaneamente para que a performance da simulação em um cenário realista pudesse finalmente ser conhecida.

A aparência do cartão é realista, e sua qualidade é de nível similar ao IBM 029. Como era de se esperar, a presença do modelo do cartão não impactou a performance de forma significativa. Porém, ainda não era possível movê-lo durante a simulação, uma capacidade necessária para poder examiná-lo de forma detalhada após a perfuração. Foi decidido que após a

perfuração do cartão (ainda não implementada), o cartão deveria ser movido com o *joystick*, o que seria possível já que o movimento do avatar foi impedido após sentar-se à frente da máquina.

Após o cartão ser perfurado pelo IBM 029, *Fire3* deve ser pressionado para permitir sua movimentação. O movimento permitido por default ao entrar nesse estado é a rotação. Utilizando o *joystick* pode-se girar o cartão para cima/baixo ou para direita/esquerda, permitindo examinar ambos os lados dele. Caso *Fire2* seja pressionado, torna-se possível mover o cartão no plano ZY, ou seja, mover o cartão para cima/baixo, ou movê-lo para mais perto/longe do usuário (tomando como referência o usuário virado para o IBM 029). Caso *Fire3* seja pressionado, torna-se possível mover o cartão no plano XY, ou seja, mover o cartão para cima/baixo, ou movê-lo para direita/esquerda.

Figura 17: Diagrama de estado do esquema de movimentação do cartão



Alguns testes revelaram que esse esquema de movimentação é um pouco complicado por ser uma forma de movimentar um objeto diferente da que estão acostumados no mundo real, com os usuários demorando um pouco para se acostumar a ela.

#### 4.4.5 Perfurações

A movimentação do cartão agora permite que a qualidade de perfurações possam finalmente ser examinadas. Portanto, a perfuração foi a próxima funcionalidade a ser implementada.

A divisão de faces do modelo foi útil para a perfuração pois permitiu referenciar exatamente as posições a serem perfuradas. Cada face que representa uma posição que pode ser perfurada na malha do cartão (os quadriláteros na Figura 13) é formada por dois triângulos.

Todos os triângulos na malha do cartão estão armazenados em um vetor. Este vetor é composto por sequências de vértices, onde cada sequência de três vértices representa um triângulo na malha. Por exemplo, em uma simples malha com apenas quatro vértices e dois triângulos (Figura 18), este vetor poderia ser (em pseudocódigo): *malha.triangulos* = [(1, 2, 4), (2, 3, 4)].

Foi decidido implementar a perfuração modificando os valores presentes nesse vetor. Utilizando o exemplo anterior, caso a sequência de vértices (1, 2, 4) seja modificada para 0, 0, e 0, o triângulo seria efetivamente removido da malha (Figura 19) e o vetor resultante seria: *malha.triangulos* = [(2, 3, 4)]. Ao renderizar o objeto, a aplicação não encontrará dados indicando que existe um triângulo formado pela sequência de vértices (1, 2, 4), logo não iria renderizá-lo. Aplicando esta técnica a ambos os triângulos que formam uma face que pode ser perfurada na malha do triângulo, obtemos uma perfuração no cartão.

Figura 18: Malha simples de dois triângulos.

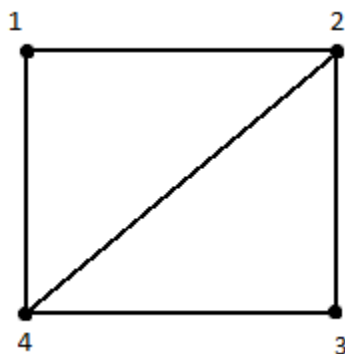
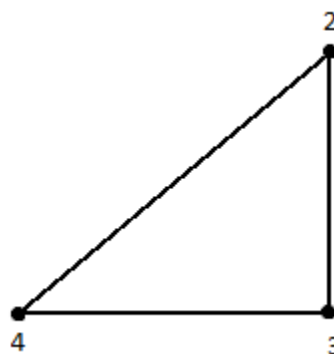


Figura 19: Malha com um triângulo removido.



Porém, para que a face escolhida seja a correta a ser removida, é preciso referenciá-la corretamente. Foi necessário saber quais eram os índices dos vértices que formavam os triângulos que deveriam ser removidos da malha. Por tentativa e erro, alguns triângulos puderam ser removidos da coluna #1 e alguns dados puderam ser coletados. Estes constam na Tabela 2, lembrando que cada posição que pode ser perfurada no cartão é formada por dois triângulos.

Pode-se ver que há certa regularidade na distribuição desses índices: os índices em cada linha diferem da linha de cima por 480. Isso foi explorado para descobrir os índices pelos quais é possível referenciar todos os triângulos na coluna. A mesma coleta de dados foi realizada para a coluna #2: os índices da segunda coluna diferem dos da primeira por 6, como pode ser visto na coluna 3 da Tabela 3.

Tabela 2: Relação entre linha e vértices na coluna #1.

Número da linha da perfuração	Vértices do primeiro triângulo em uma face	Vértices do segundo triângulo em uma face
0	4908, 4909, 4910	4911, 4912, 4913
1	4428, 4429, 4430	4431, 4432, 4433
2	3948, 3949, 3950	3951, 3952, 3953
3	3468, 3469, 3470	3471, 3472, 3473

Tabela 3: Relação entre linha e vértices na coluna #2.

Número da linha da perfuração	Vértices do primeiro triângulo em uma face	Vértices do segundo triângulo em uma face
0	4914, 4915, 4916	4917, 4918, 4919
1	4434, 4435, 4436	4437, 4438, 4439
2	3954, 3955, 3956	3957, 3958, 3959
3	3474, 3475, 3476	3477, 3478, 3479

Temos os seguintes resultados: a diferença entre os vértices de linhas adjacentes na mesma coluna é de 480, e a diferença entre os vértices de colunas adjacentes na mesma linha é de 6. Com estes resultados e os índices para a primeira posição na coluna 1 e a última na coluna 80, torna-se possível escolher o alvo de uma perfuração. Um exemplo de cartão perfurado pode ser visto nas Figuras 20 e 21, enquanto que um diagrama do fluxo tomado pela função de perfuração pode ser visto na Figura 22. As posições das perfurações seguem a Figura 2, com a exceção de dois caracteres que constam no teclado de um IBM 029, porém não estão na tabela ASCII e portanto não puderam ser implementados: ¢ (centavo) e ¬ (negação).

Para que o usuário pudesse confirmar que as perfurações realizadas correspondem à string desejada, o caractere correspondente às perfurações em uma coluna é impresso em seu topo. Isso foi implementado por meio de um objeto canvas que foi entrelaçado ao cartão, o sobrepondo, e possui texto que é modificado durante a execução.

O método selecionado para realizar as perfurações não foi a primeira escolha. Originalmente, foi planejado modificar o vetor normal dos triângulos para que aparentasse como uma remoção. A renderização é realizada em somente um lado de uma superfície, que depende da orientação da superfície, que em torno depende do vetor normal a ela. Logo, invertendo o vetor de forma que ele aponte para a direção oposta, o lado da superfície que previamente era visível se tornava invisível. Em contrapartida, a renderização ainda ocorreria, porém no lado oposto da superfície. Isso pareceu produzir um resultado semelhante às perfurações na Figura 20, porém os triângulos da malha da frente do cartão que tornavam-se invisíveis apareciam em seu verso, inviabilizando a solução de inverter as normais. Por exemplo, caso o número 1 fosse perfurado na coluna 1, a perfuração aparentaria ter ocorrido com sucesso olhando a frente do cartão, porém o número 1 passaria a ser visível no verso do cartão.

Figura 20: Exemplo de cartão perfurado com impressão.

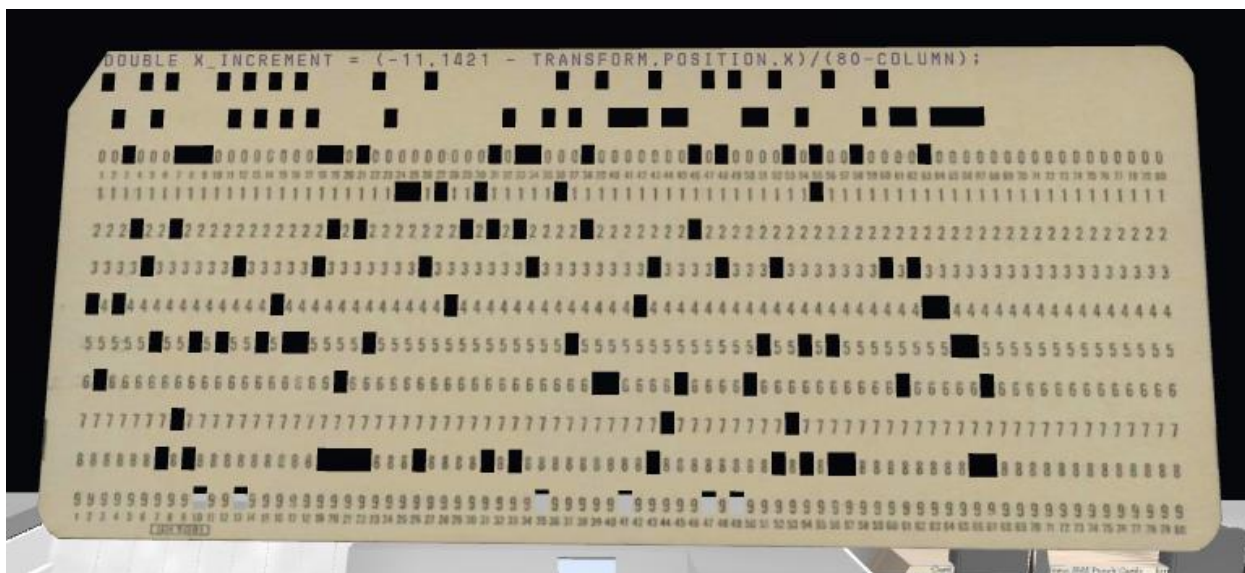


Figura 21: Verso do cartão perfurado (mesma perfuração da Figura 20).

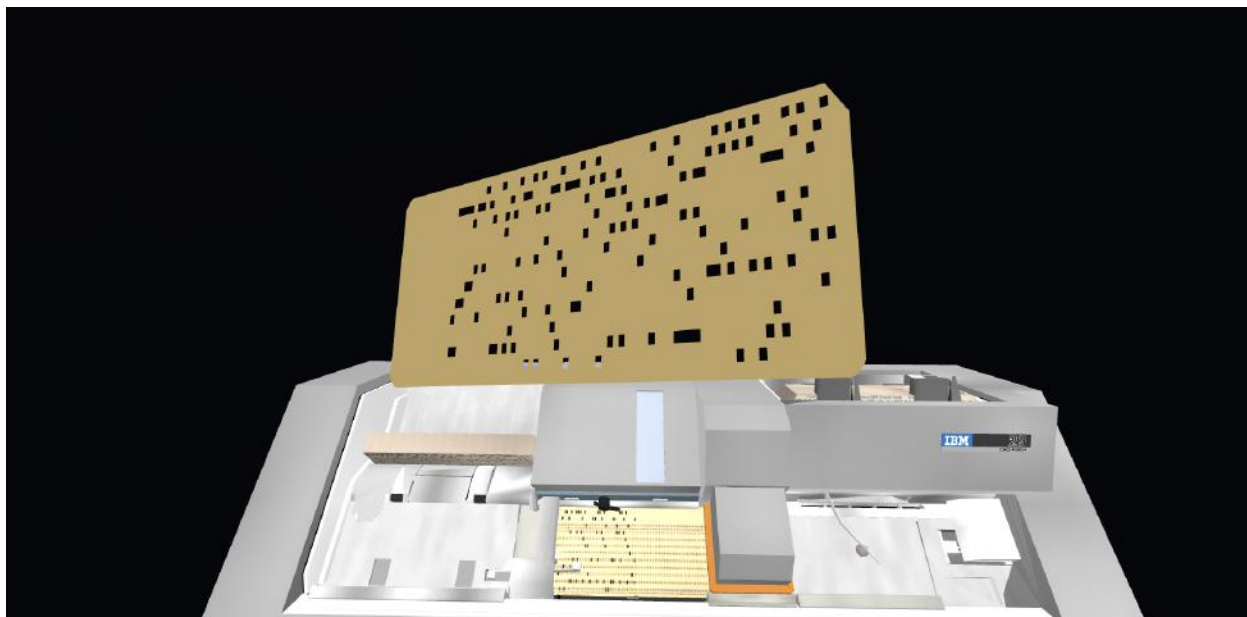
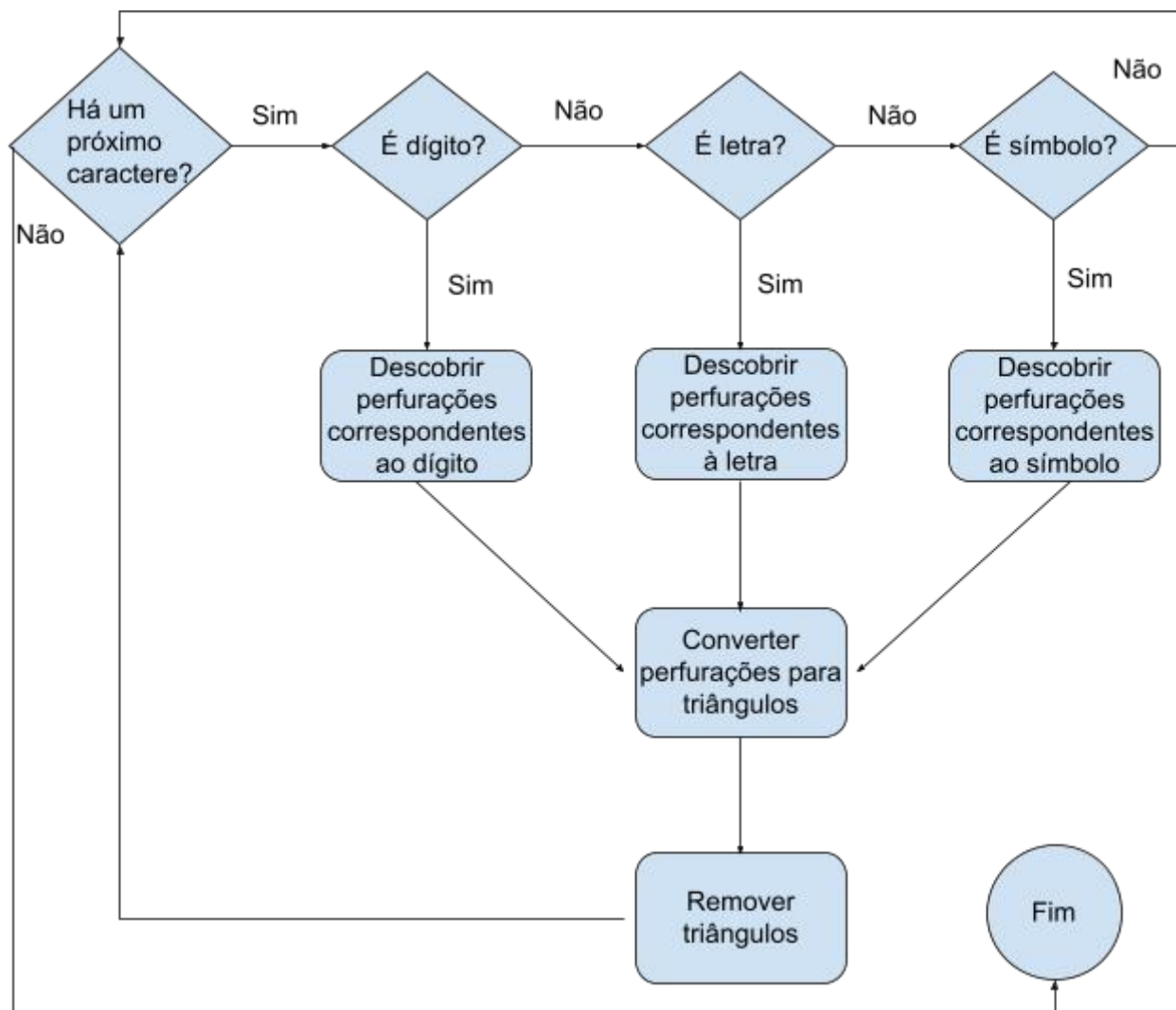


Figura 22: Fluxo tomado pela aplicação ao perfurar uma coluna.



#### 4.4.6 Animação do Cartão

A perfuração do cartão deve ser animada para a simulação. Para garantir o realismo das animações, os movimentos do cartão em um vídeo demonstrando a operação do keypunch foram copiados. As animações foram divididas em três partes (Figura 23): fornecimento do cartão do *card hopper* (armazena cartões prontos para perfuração) até a *punching station* (área de perfuração), a perfuração em si, e a passagem do cartão já perfurado para a *reading station* e então empilhamento do cartão no *stacker* (armazena cartões já perfurados). As animações são realizadas por meio de transformações que são aplicadas aos objetos.

Figura 23: Partes do IBM 029 marcadas em vermelho.

*Card Hopper - 1; Punching Station - 2; Reading Station - 3; Stacker - 4*

*Fonte: IBM 029. DVQ*

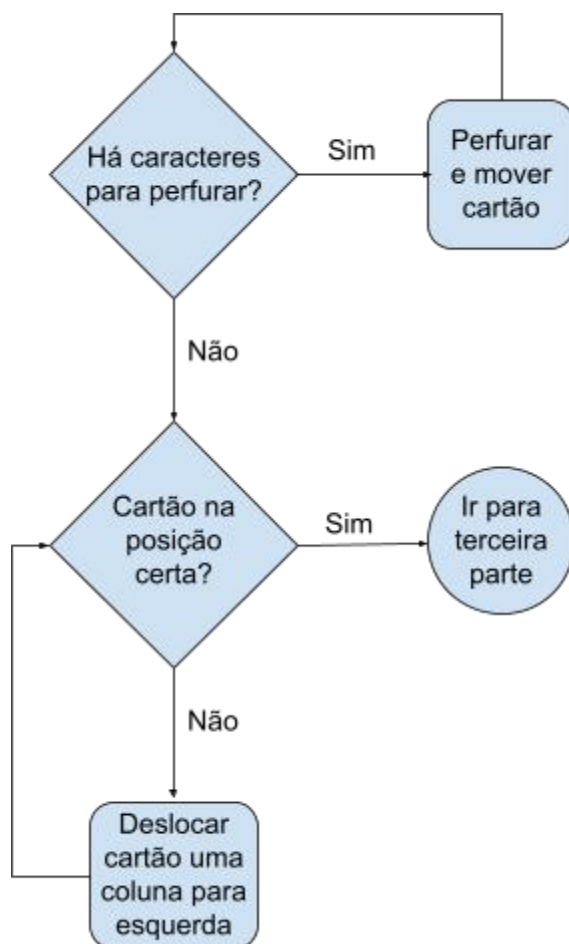


Durante a primeira parte, o cartão é deslocado do *card hopper* diretamente para baixo até alcançar a altura certa, e então é deslocado para a esquerda até alcançar o início da *punching station*. A segunda parte é realizada puramente por código para sincronizar o movimento do cartão com as perfurações, que devem ser realizadas por código. A cada perfuração, o cartão é movido o correspondente à largura de uma coluna para a esquerda. Após a perfuração terminar, o cartão é deslocado para a *reading station* e então empilhado no *stacker*. Sons retirados do vídeo demonstrando a operação da máquina foram adicionados às animações, replicando os sons emitidos pela máquina em cada uma das partes. Durante a etapa da perfuração, cada perfuração individual chama uma função para tocar um clipe de áudio de uma perfuração.

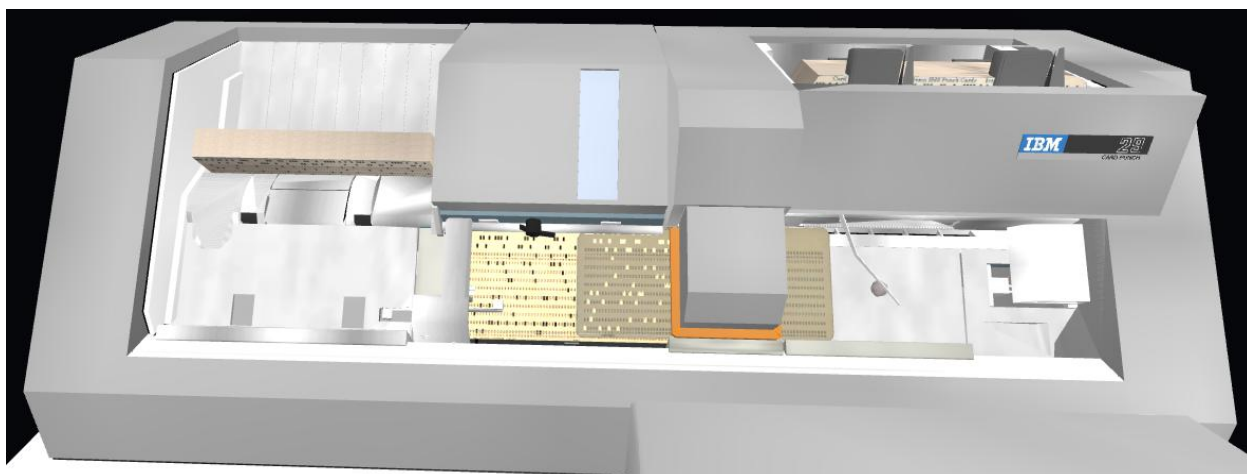
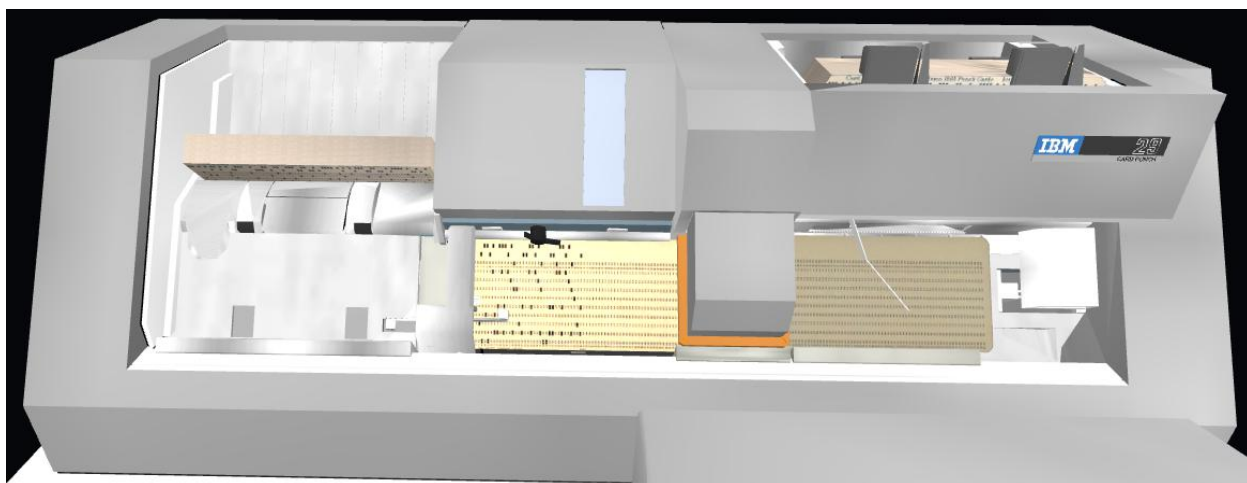


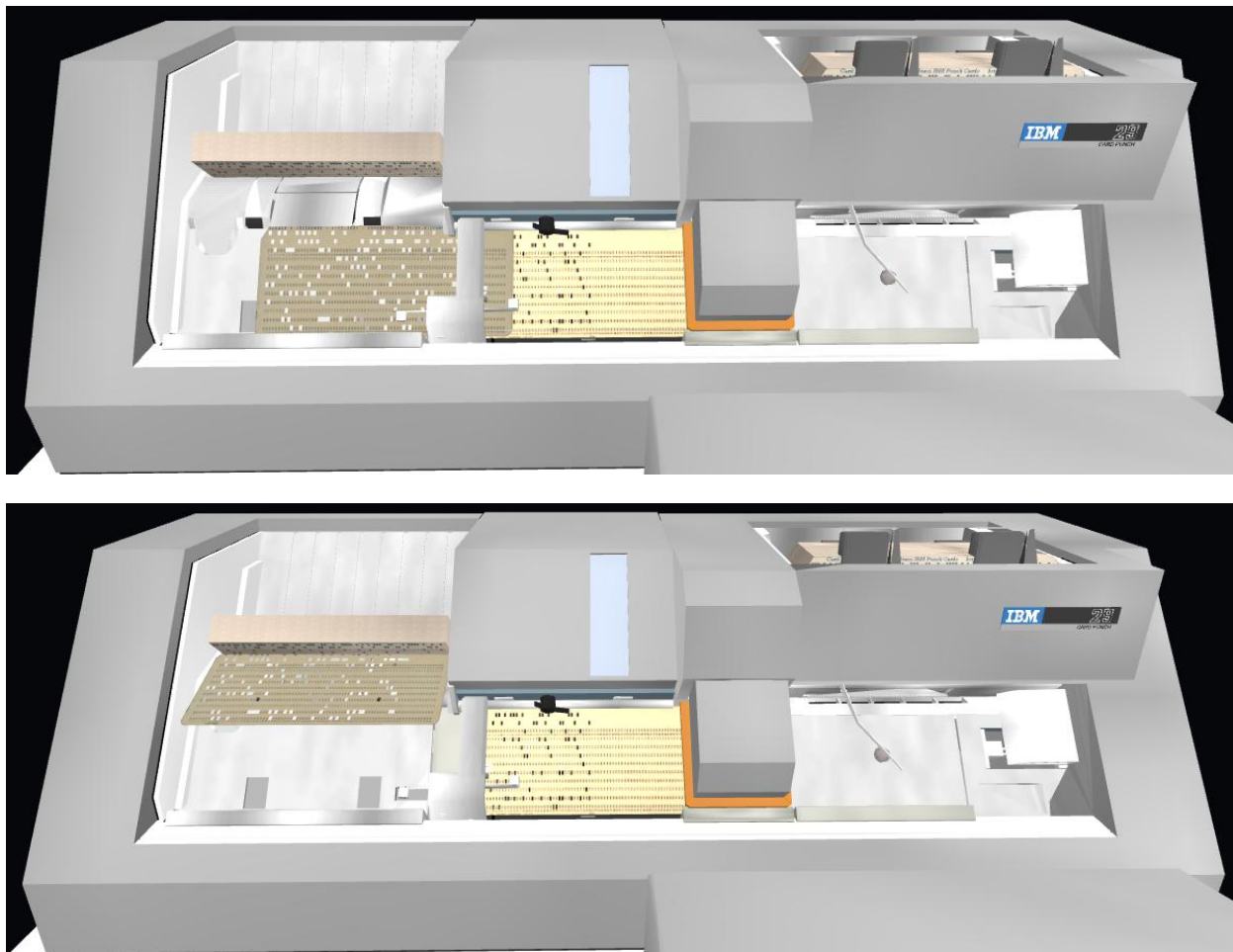
Para que a última animação possa ser ativada o cartão deve estar na posição na qual ela é iniciada, isto é, a posição na qual o cartão se encontra após serem realizadas 80 perfurações nele. Caso a string perfurada não tenha 80 caracteres, após a perfuração o cartão é movido até a posição correta.

Figura 24: Segunda parte da animação da perfuração.



Figuras 25a, 25b, 25c, 25d, 25e: Animação do cartão em diferentes pontos.





#### 4.4.7 Múltiplos Cartões

Neste ponto a simulação permitia que o usuário perfurasse um único cartão. As alterações seguintes tiveram como objetivo medir o impacto que múltiplos cartões trariam à simulação.

Primeiramente, um script foi criado para permitir a criação de cartões pelo input do usuário, impedindo que um número pequeno demais ou grande demais de cartões perturbasse a experiência. Cada cartão criado possui os mesmos componentes, como script controlando a perfuração e as animações, pois é instanciado a partir de um único modelo que os contém. Cada cartão é criado na posição do *card hopper*, na posição inicial da primeira animação, assim permitindo que ela seja executada.

Para facilitar o manuseio de múltiplos cartões, um novo objeto foi criado: uma pilha de cartões. Inicialmente um objeto vazio, é preenchido por cartões à medida que suas perfurações terminam.

Figura 26: Close up da pilha com 10 cartões.



Com múltiplos cartões, foi adicionado ao esquema de movimentação apresentado na Figura 17 a função de trocar o cartão atualmente sendo movido pelo próximo na pilha de cartões, o que pode ser feito pressionando *Jump*. O cartão que estava sendo movido antes da troca é enviado para o fundo da pilha, permitindo assim que o usuário circule por todos os cartões perfurados.

A performance pôde então ser testada com diferentes números de cartões. Até a criação do quarto cartão pouco impacto pode ser observado, com o framerate podendo diminuir até 28, porém mantendo a média de 30 FPS. Com 5 cartões a performance começa a ser prejudicada, com nova média de 28 FPS, e ponto baixo de 25. Cada novo cartão criado impacta a performance. Com 9 cartões perfurados no campo de visão, o framerate eventualmente caia até 22 FPS, embora geralmente mantinha em torno de 24 FPS.

Com múltiplos cartões a performance sofre a ponto de causar desconforto nos usuários. O FPS passa repentinamente de valores superiores a 30 para menos de 25, causando

dessincronização entre movimentos da cabeça do usuário e a atualização das imagens no HMD, o que pode eventualmente quebrar a sensação de presença.

Portanto, alguma otimização é necessária. Mas simplificar o modelo do cartão dificultaria a realização das perfurações além de causar a reescrita do script que controla as perfurações. Simplificar o modelo do keypunch também não seria benéfico pois diminuiria demais sua qualidade gráfica. A rota tomada para melhorar a performance está descrita na Seção 4.5.

#### 4.4.8 Entrada da Mensagem Perfurada

Idealmente a entrada da mensagem que é perfurada seria feita dentro da simulação utilizando o teclado do IBM 029. Porém o controle impede qualquer método de entrada por dentro de ser eficaz: um cursor (ou contorno escolhendo uma tecla) teria que ser movimentado por cada tecla uma por uma até alcançar a correta utilizando o *joystick*, um processo demorado e potencialmente frustrante considerando que deve ser repetido potencialmente 80 vezes. Essa frustração poderia prejudicar a sensação de presença, logo este método foi evitado.

Durante a prototipagem a entrada foi feita dentro do código, como uma string ou vetor de strings. Esse método requer que o usuário tenha conhecimento prévio de como modificar o código da aplicação caso queira dar entrada a uma mensagem sua, uma suposição que não pode ser feita para o público em geral. Consequentemente, a entrada da mensagem deveria ser feita por alguém do museu, limitando a interação do usuário com o sistema e prejudicando a experiência daqueles que viessem a encontrar a experiência online. Por essas falhas, esse método foi modificado para uma versão definitiva em estágios mais avançados do desenvolvimento.

A versão definitiva envolve digitar a mensagem em um arquivo de texto antes da simulação começar, o que pode ser feito por qualquer pessoa. Essa forma foi escolhida pois permite que a interação do usuário afete o resultado final e envolve sua digitação, mas sem exigir processos complicados que poderiam subtrair a qualidade da experiência.

Um claro problema com o método escolhido é que a entrada não é feita de dentro do ambiente virtual, dando menos plausibilidade ao cenário. Um método pelo qual a entrada da mensagem pode ser realizada de dentro da simulação envolve a detecção do olhar do usuário. Essa técnica e esse método são detalhados na Seção 7.2 (Trabalhos Futuros). Como mencionado na seção, o método tem potencial para oferecer uma melhor imersão e melhor interatividade, e

por tal deve ser investigado em um trabalho futuro. Não foi implementado neste trabalho por somente ter sido idealizado após a realização dos testes.

Vale também notar que alguns caracteres comuns, tais como ‘{’ e ‘}’ não constam no teclado do IBM 029, e erros ocorrem ao tentar perfurá-los.

#### 4.4.9 Atualização da Cena

Com as funções principais da implementação prontas, as máquinas modernas que até então serviam como substitutos temporários puderam finalmente ser substituídas. Os objetos escolhidos para povoar a cena foram:

- IBM 2401: unidade de fita magnética (Figura 28);
- IBM 1401: unidade de processamento central (Figura 27);
- IBM 3178: estação de display (Figura 30);
- IBM 3350: armazenamento de acesso direto (Figura 29).

A Figura 31 apresenta todos esses modelos: 2401 ao fundo, 1401 à sua esquerda, 3178 na parte inferior direita, e 3350 no centro da imagem. Separado à direita está o 029, que foi isolado para reduzir a carga na GPU de renderizar vários objetos simultaneamente. A cena atualizada aumenta a sensação de presença e adiciona valor à cena, que agora consiste de uma experiência mais ampla, também permitindo ao usuário examinar essas máquinas de perto. A cena não foi preenchida com mais objetos para evitar causar maiores problemas de desempenho.

Figura 27: Unidade de processamento central 1401



Figura 28: Unidade de fita magnética 2401





Figura 29: Armazenamento de acesso direto 3350



Figura 30: Estação de display 3178



Figura 31: Visão da versão final da cena.



#### 4.5 OTIMIZAÇÕES

Após todas as funcionalidades centrais serem adicionadas, o desenvolvimento passou a ser focado na otimização do que havia sido criado. A otimização pode ser alcançada por dois caminhos: diminuindo a carga posta na CPU, ou diminuindo a carga posta na GPU. As otimizações realizadas relacionadas com o primeiro caminho consistiram em reduzir o número de objetos que precisam ser renderizados, reduzir o número de vezes que cada objeto precisa ser renderizado, ou reduzir o número de chamadas às funções renderizadoras. As relacionadas ao segundo caminho consistiram em otimizar a geometria dos modelos utilizados.

Como mencionado anteriormente, uma das métricas mais importantes para a qualidade de uma experiência em realidade virtual é o número de frames gerados por segundo. Diversas técnicas de otimização, descritas a seguir, foram testadas visando a melhoria dessa métrica, por vezes piorando outras métricas, como a qualidade visual, em contrapeso.

#### 4.5.1 Número de polígonos

Funções envolvidas no processo de renderização são executadas para cada polígono que está visível. Cada luz que ilumina um polígono tem seu efeito sobre aquele polígono calculado por funções de renderização. Assim sendo, o número de operações necessárias para completar cada frame é da ordem do produto entre o número de polígonos e o número de luzes. Uma otimização relacionada que pôde ser efetuada foi a simplificação de geometria já detalhada na Seção 4.2.1.

#### 4.5.2 Draw Call

Para renderizar cada frame, tudo o que será visível na frame deve ser individualmente renderizado, ou mais coloquialmente, desenhado (*drawn*). Sempre que uma função para ‘desenhar’ uma malha é chamada (*called*), temos uma *draw call*. O número dessas chamadas depende, majoritariamente, do número de objetos e do número de materiais desses objetos. Um material é um conjunto de informações sobre a aparência de uma superfície, por exemplo, ditando como a iluminação deve interagir com ela.

Estas *draw calls* devem ser processadas pela CPU antes de serem enviadas à GPU. Dependendo do poder computacional da CPU, um gargalo pode ser formado, diminuindo o FPS. Assim, quanto menor for o número de *draw calls*, menor será o impacto na performance.

No projeto isso foi alcançado pelo uso de *static batching* (agrupamento estático). Esse método combina malhas que não são movidas e têm os mesmos materiais em uma malha visando simplificar o processo de *draw calls*. Como estas malhas nunca se movem e são renderizadas de forma similar (devido aos mesmos materiais), podem ser tratadas como uma grande malha, permitindo diminuir o número de *draw calls*.

Levando isso em consideração, um baixo número de materiais foi utilizado por grande parte dos objetos para permitir maior otimização por este método. Por exemplo, todos os objetos que compõem as paredes compartilham o mesmo material, assim como os objetos que compõem o teto e o chão compartilham do mesmo material. Todos os objetos, com exceção dos cartões perfurados, da câmera controlada pelo usuário, e do avatar são marcadas como estáticas, indicando que *static batching* deve ser utilizada neles, caso seja viável.

Utilizando esta técnica, o único impacto está na forma como os cálculos são feitos, não em seus resultados finais. Ou seja, não há diferença gráfica causada por esta técnica. Um impacto positivo no desempenho pôde ser observado. Até a criação do quarto cartão o *framerate* manteve-se constantemente acima de 30 FPS, com média de 34, quando antes da adoção dessa técnica com quatro cartões a média era de 30 FPS. Além disso, o desempenho só foi prejudicado a ponto de levar a média abaixo de 30 FPS quando nove ou mais cartões são criados, quando a média assume 28 FPS. Antes da utilização dessa técnica, com nove cartões a média era de 24 FPS.

Um ponto importante de ser notado é o pulo no FPS quando os objetos com geometria mais complexa entram no campo de visão, que continua a ser alarmante e prejudica a sensação de presença.

#### 4.5.3 Iluminação

Por default, Unity utiliza iluminação em tempo real para iluminar suas cenas. Essa técnica pode ser útil quando a cena é composta de objetos móveis, onde cada frame deve ser renderizada para refletir a mudança na interação da luz com o objeto, representando a movimentação. Porém, a cena foi construída de forma a minimizar a movimentação dos objetos e diminuir o número de objetos com os quais o usuário pode interagir. Consequentemente, boa parte do poder computacional poderia ser desperdiçado renderizando as mesmas imagens a cada nova frame. Isso foi contornado com o uso de *lightmaps* pré-calculadas, estruturas que armazenam o resultado da incidência de luz sobre a superfície de um objeto. Elas podem ser utilizadas para gerar texturas que podem ser aplicadas na superfície de objetos, fornecendo um substituto à iluminação em tempo real e eliminando o overhead de calculá-la durante a execução. Essa técnica só pode ser utilizada em objetos estáticos, pois as texturas não podem ser modificadas durante a execução. Logo, essas estruturas são utilizadas para todos os objetos, com exceção dos cartões perfurados e do avatar. A Figura 31 demonstra o ambiente iluminado por esta técnica. Cada luz incidente sobre um objeto causa novos cálculos em cada polígono [16]. A substituição de luz em tempo real para luz pré-calculada diminui o número de polígonos que precisam ser considerados ao realizar os cálculos de iluminação.

Pela adoção dessa iluminação, a simulação pode se manter no intervalo [50, 60] FPS, com resultado final dependente do número de cartões no campo de visão:

- Sem a criação de nenhum cartão: 60 FPS constantes;
- Com até 5 cartões, geralmente 55 FPS pode ser mantido;
- Com até 10 cartões, valores não ficam abaixo de 50 FPS, mantendo média de 52.

Mas quem sabe o mais importante, a mudança abrupta no *framerate* que ocorria sempre que o *keypunch* passava para o campo de visão desapareceu completamente.

Pela aplicação deste método a experiência agora obedece os limites para a latência para aplicações em realidade virtual primeiramente mencionados na Seção 3.1.

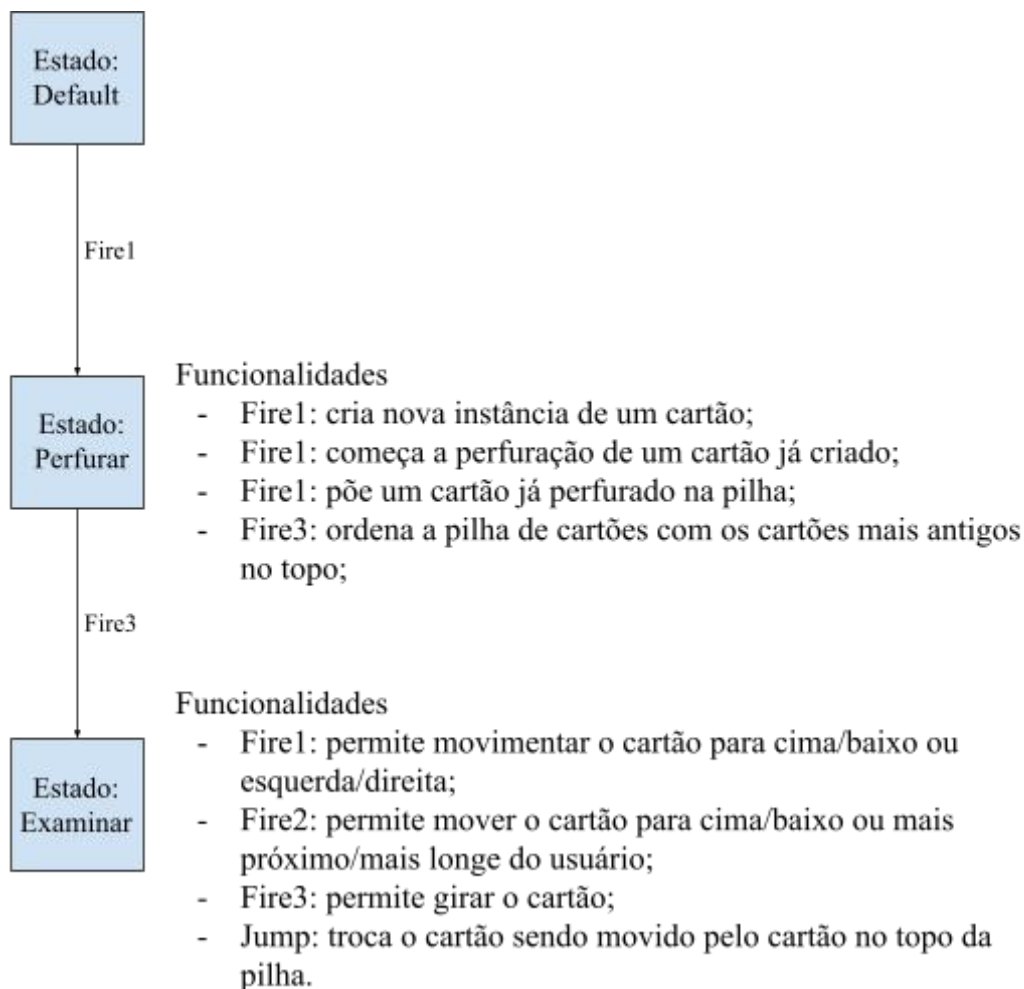
#### 4.6 FLUXO DA EXECUÇÃO

O produto final consiste em uma simulação onde o usuário é inserido em uma sala onde pode examinar diversas máquinas que fizeram parte da história da computação, uma das quais pode ser utilizada. A Figura 32 apresenta o fluxo seguido pela simulação, apresentando funções que podem ser executadas pelo usuário. O fluxo foi dividido em diferentes estados:

- Default: é permitido ao usuário navegar pelo ambiente;
- Perfurações: é permitido ao usuário criar e perfurar cartões;
- Examinar: é permitido ao usuário mover cartões para poder examiná-los.

O número de botões é limitado, portanto, a quantidade de maneiras pelas quais o usuário pode interagir com o ambiente também será. Isso provocou o design baseado em estados apresentado no diagrama, onde cada estado fornece aos botões novas funções.

Figura 32: Diagrama do fluxo da simulação com funções específicas a cada estado.



#### 4.7 LIMITES DO DESIGN

Para falar sobre as limitações, é importante ter em mente as concessões (*affordances*) desses artefatos. Elas são as possíveis ações relacionadas a objetos que são reconhecidas prontamente por um ator ao identificar um objeto [11]. Por exemplo, ao ver uma porta, uma pessoa imediatamente sabe que pode abri-la ou fecha-la. Certamente usuários terão a expectativa de que essas ações são possíveis, então caso não sejam no ambiente virtual sua imersão será prejudicada. No caso deste trabalho, o IBM 029 possui alguns interruptores próximos ao teclado, visíveis aos usuários, e testes posteriores confirmaram que alguns usuários esperavam poder utilizá-los. Idealmente todos seriam funcionais, porém trazer todas as funcionalidades do 029, até as raramente utilizadas, estaria fora do escopo deste trabalho.

Implementar uma versão virtual da máquina que funcionasse como uma versão física poderia não ser útil, visto que para utilizar todas as funcionalidades da máquina seria preciso treinamento por parte do usuário, uma ideia incompatível com uma visita a um museu. Uma exibição interativa deve ser simples o suficiente para ser utilizada pela maior parte do público visitante.

Entre algumas diferenças entre a versão física e a virtual do perfurador, podemos listar:

- A simplificação do processo de perfuração: um operador teria que pressionar a tecla “FEED” (alimentar) para que um cartão fosse pego do *card hopper* para ser perfurado, digitar um caractere de cada vez, e então apertar a tecla “REL” (liberar, do inglês *release*) para passar o cartão para a *reading station*; na versão virtual, após apertar um botão para “alimentar” um cartão para a máquina, a perfuração ocorre automaticamente e será liberada para a *reading station* após sua perfuração, tudo com um botão;
- Funcionalidades mais obscuras, tais como a automatização de sua operação por meio de *program cards*, não foram programadas.

#### 4.8 DESAFIOS

O maior entre os problemas foi a performance do software, que é limitada pela escolha da plataforma mobile onde o poder de processamento é naturalmente mais baixo. Isto está diretamente ligado à náusea que pode ocorrer durante seu uso. A náusea é causada pelo conflito entre informações captadas pelos nossos sistemas visuais e vestibulares. O sistema vestibular nos dá o senso de balanço e orientação espacial, e nos permite coordenar movimento com balanço. Caso a HMD não seja capaz de atualizar as imagens mostradas ao usuário em uma velocidade similar a de seu movimento, um conflito de informações fornecidas por esses dois sistemas ocorrerá. Este foi um problema prevalente durante todo o desenvolvimento, ao qual algumas pessoas são naturalmente mais suscetíveis. O problema desapareceu após as otimizações serem efetuadas.



## 5 TESTES

Esta seção descreve de que forma os dados foram julgados para que se possa compreender se o software criado alcançou suas metas como experiência em realidade virtual de imergir o usuário, e suas metas como exibição de um museu de satisfazer e educar o usuário.

### 5.1 HIPÓTESES

Os testes realizados forneceram uma base de dados a partir da qual as seguintes hipóteses puderam ser julgadas:

- Primeira hipótese: a experiência é capaz de imergir usuários em seu ambiente artificial de forma que estes ajam como agiriam no mundo real;
- Segunda hipótese: a experiência cumpre os papéis de uma exibição de um museu de fornecer lazer e educação;
- Terceira hipótese: a adição de uma experiência em realidade virtual a um museu aumenta a probabilidade de que ele seja visitado no mundo real;
- Quarta hipótese: a adição de uma experiência em realidade virtual a um museu aumenta o interesse do público no tema do museu;
- Quinta hipótese: a qualidade da experiência relatada pelos usuários está diretamente relacionada à sensação de presença alcançada durante a experiência.

### 5.2 REGISTRO DA POSIÇÃO

Para analisar a efetividade da cena criada, um registro da posição do usuário a cada instante é mantido. Torna-se possível dizer o quanto a cena foi explorada e os objetos analisados.

Na Figura 33, a proporção do tempo passado em uma posição é indicado por cores mais quentes: mais tempo foi passado na parte central em baixo, indicado pelo ponto amarelo. A Figura 34 mostra como a cena seria vista de cima. As posições dos objetos na cena podem ser discernidas na imagem pois foram contornadas ao serem examinadas pelo usuário. No canto superior direito estão o IBM 1401 e os IBM 2401, no centro estão os IBM 3350, no lado esquerdo está o IBM 3178, e na parte de baixo se encontra o perfurador IBM 029.

Figura 33: Mapa de calor da posição em uma simulação.

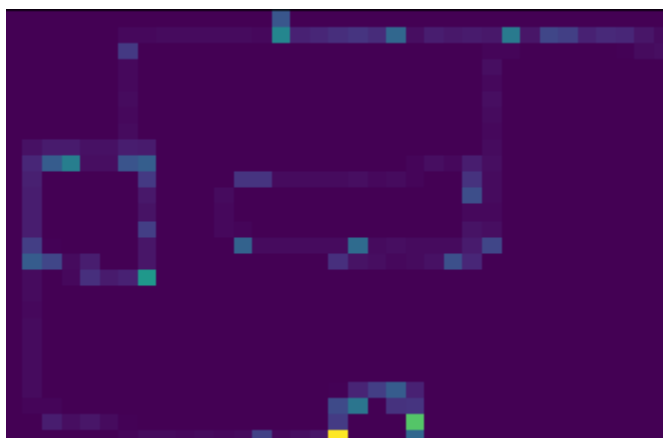
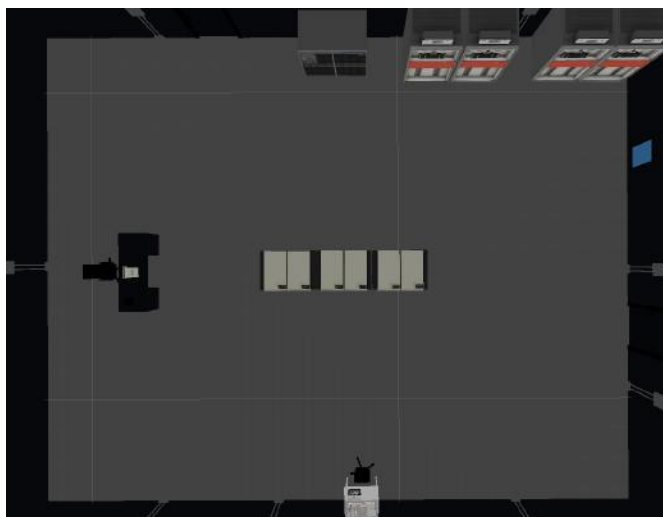


Figura 34: Vista da cena de cima



### 5.3 QUESTIONÁRIOS

Dados quantitativos foram coletados para poder compreender o efeito da experiência nos respondentes. São utilizados três questionários para a coleta de dados, sendo que dois foram criados pelo autor do trabalho baseado em questionários utilizados na literatura da área.

### 5.3.1 Questionário Preliminar

O primeiro que é respondido é um dos criados pelo autor e foi denominado “preliminar” pois deve ser respondido antes da simulação. Ele tem como objetivo principal medir o nível de interesse em uma possível visita a um museu e interesse na computação em geral, servindo como uma base para comparar os níveis de interesse após experienciar a simulação. Pode ser encontrado no **Apêndice A**.

### 5.3.2 Questionário IPQ

Um padrão para medir a sensação de presença é indispensável para qualquer pesquisa sobre realidade virtual. Sem uma forma confiável e reproduzível de realizar essas medidas, comparações entre resultados de diferentes pesquisas se tornam complexas. Este trabalho utiliza a forma mais comum de medição na literatura da área: questionários pós-imersão padronizados consistindo de itens em uma escala Likert. Estes questionários têm sido o padrão na pesquisa da área por anos, e foi demonstrado serem efetivos na medição de presença [15]. O questionário escolhido para ser utilizado foi o IPQ (*Igroup Presence Questionnaire*), que pode ser encontrado no **Apêndice B**.

### 5.3.3 Questionário da Qualidade

Além de medir a sensação da presença, também foi tido como necessário medir a qualidade da experiência como uma exibição para um museu. Para tal, um questionário foi criado para confirmar que a simulação atinge os objetivos de um museu. Seus itens foram divididos em seções, cada uma focando em diferentes quesitos que podem tornar a experiência uma boa adição para um museu: a relação subjetiva do visitante com a exibição; o realismo da exibição; e a aprendizagem pela experiência. Assim, o questionário pode confirmar se um participante gostou da experiência, o que achou dos controles e movimentação pelo ambiente, se a experiência foi educativa, assim como medir o interesse após a experiência, como mencionado anteriormente. O questionário completo se encontra no **Apêndice C**.

Este questionário foi baseado na escala MES (*Museum Experience Scale*), que tem como objetivo medir a experiência de um visitante a um museu. O desenvolvimento da escala está descrito em [6], e [7, 8] apresentam sua utilização em estudos relacionados a museus.

#### 5.4 METODOLOGIA

Cada participante foi testado separadamente. Em nenhum momento foi explicado quais eram os objetivos dos testes, como forma de minimizar o enviesamento dos dados.

O processo de testes era iniciado com o participante respondendo ao questionário preliminar. Então, uma breve explicação era dada sobre o que veriam ao pôr a HMD e como poderiam interagir com o ambiente e se movimentar nele. Após digitar a mensagem que gostaria de ver perfurada, o participante seria imerso no ambiente virtual, onde poderia explorar o ambiente e perfurar seus cartões até completar a simulação. Após o término da simulação o questionário da presença era preenchido e então o da experiência.

Não foram limitados por um limite de tempo, permitindo que pudessem se familiarizar com os controles. A partir do momento no qual a simulação era iniciada, os participantes somente poderiam receber explicações sobre os controles, caso pedissem, caso contrário não seriam incomodados por nada fora do ambiente virtual.

#### 5.5 PARTICIPANTES

O número de participantes foi 27, sendo 17 homens e 10 mulheres, com idades variando entre 19 e 61 anos, porém não distribuídas uniformemente: 19 dos participantes estavam no intervalo de 19 a 30 anos. São de diversas origens, com diferentes formações e carreiras: estudantes de engenharia, de psicologia, professores de ensino fundamental, comerciantes sem formação superior. Assim, embora a amostragem seja de pequeno tamanho, pela sua diversidade, compõe um retrato do público geral. A maior parte deles relatou ser confortável com novas tecnologias, porém foi observado que alguns, mais idosos, mal estavam confortáveis com seus celulares. Somente 4 deles relataram qualquer experiência prévia com realidade virtual.

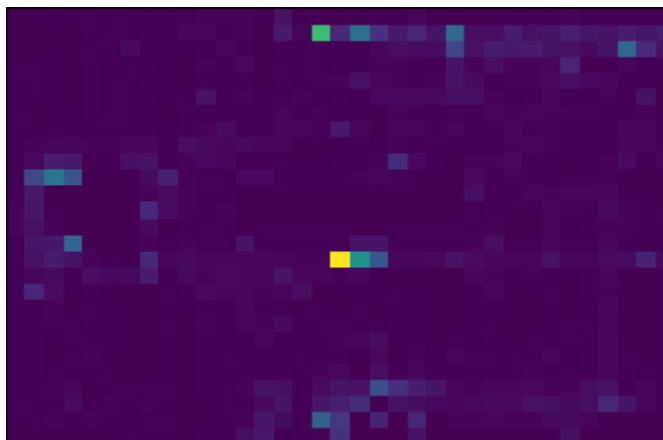
## 6 RESULTADOS

Esta seção analisa os dados coletados, buscando construir algum conhecimento a partir deles.

### 6.1 EXPLORAÇÃO DO AMBIENTE

Analisar o registro da posição de todos os usuários sobrepostos em uma única imagem nos auxilia a determinar seu comportamento na simulação. A área amarela no centro da imagem indica a posição onde mais tempo foi passado. Esta é a posição inicial, onde o usuário começa a simulação. Assim que a simulação começava, os participantes, principalmente aqueles sem experiência com realidade virtual, simplesmente olhavam à sua volta por certo tempo, se acostumando à imersão no ambiente virtual e testando o rastreamento da rotação de sua cabeça. As outras áreas claras confirmam que a presença de outros objetos pela cena os incentivou a explorá-la, estando localizadas próximas a estes objetos. Muitos participantes pausavam em frente a algumas das máquinas, como o IBM 1401 (centro da parte de cima, indicado pelo ponto verde), para analisar todos os botões e escritas em sua frente.

Figura 35: Registro das posições de todos os participantes sobrepostos.



## 6.2 ANÁLISE QUALITATIVA

Alguns pontos foram percebidos observando a experiência dos participantes e conversando com eles sobre ela. Aqueles que foram julgados serem mais importantes foram discutidos a seguir.

Os participantes relataram interesse nas máquinas antigas, porém aqueles que tentaram interagir com eles ficaram decepcionados ao descobrir que eram meramente decorativos. Isso está ligado ao conceito de *affordances* mencionado anteriormente: usuários em uma experiência que envolve a operação de uma máquina têm a expectativa de que, ao ver outras máquinas, estas também sejam funcionais.

Alguns participantes, notavelmente os mais idosos, apresentaram problemas com os controles, relatando dificuldade em mover-se pelo *joystick* por falta de familiaridade. Um indivíduo chegou a remover o HMD para poder ver o controle enquanto já estava na experiência.

A confusão ao mover os cartões perfurados foi mais comum, sendo apresentada por mais da metade dos respondentes e sendo relatada como o maior ponto negativo da experiência. Os participantes não puderam compreender como a movimentação dos cartões ocorreria até tentar movê-los, então inicialmente muitos apresentaram confusão sobre como realizar os movimentos que imaginavam. Uma forma de interação mais intuitiva poderia ser alcançada pelo uso dos controles mencionados na Seção 2.1.1. Com essa forma, pegar e movimentar os cartões poderia ser feito como no mundo real, assim como a digitação.

Ajuda não foi oferecida durante a simulação para testar simular o caso onde alguém sem acesso físico ao museu encontrou a experiência online, visto que estes não poderiam ter apoio nessa questão. Eventualmente, todos aprenderam a mover os cartões como desejavam.

Um outro ponto negativo comum foi não poder utilizar o teclado do IMB para poder digitar sua mensagem durante a experiência. Uma possível solução para este problema foi discutida na Seção 7.2 (Trabalhos Futuros).

Por último, todos os participantes acharam a experiência única, e aparentavam apreciá-la mais por isso, e todos relataram ter aprendido algo novo (nem aqueles com interesse na computação conheciam os cartões perfurados).

## 6.3 ANÁLISE QUANTITATIVA

### 6.3.1 Hipótese 1: Capacidade Como Experiência em Realidade Virtual

Pode ser observado, pela Tabela 4, que uma boa presença foi alcançada pelos participantes. Porém, questões como a interação com o ambiente não ser muito intuitiva dificultaram a instalação da ilusão de plausibilidade, enfraquecendo a sensação de presença. As Tabelas 5 e 6 demonstram a baixa naturalidade e intuitividade da interação com o ambiente virtual.

Tabela 4: Dados da presença dos participantes.

Mínimo	Máximo	Média	Desvio padrão
3.3	6.2	4.804	0.788

Tabela 5: Dados para o item “A interação com o ambiente era natural”.

Mínimo	Máximo	Média	Desvio padrão
2	6	3.407	1.185

Tabela 6: Dados para o item “Eu pude produzir os resultados desejados intuitivamente”.

Mínimo	Máximo	Média	Desvio padrão
1	5	3.296	1.103

Os dados confirmam que os participantes não interagiram, durante a experiência, da forma como o fariam no mundo real. Assim, embora um bom nível de presença tenha sido alcançado, a primeira hipótese não pôde ser confirmada.

### 6.3.2 Hipótese 2: Capacidade Como Exibição de Um Museu

Pode ser observado pelas Tabelas 7 e 8 que as características de uma exibição são alcançadas pela simulação criada. Portanto, a segunda hipótese é confirmada.

Tabela 7: Dados para o item “Eu gostei da experiência”.

Mínimo	Máximo	Média	Desvio padrão
3	7	5.630	1.150

Tabela 8: Dados para o item “Minha experiência enriqueceu meu conhecimento”.

Mínimo	Máximo	Média	Desvio padrão
3	7	5.259	1.022

### 6.3.3 Hipótese 3: Interesse em Visitar

Para medir o interesse dos participantes na computação (tema do museu) e em visitar um museu que tratasse desse tema, dois itens específicos constavam no questionário preliminar, e dois outros no questionário da qualidade da experiência. Os dois primeiros são:

- “Tenho interesse na computação”
- “Tenho interesse em visitar um museu sobre computação”

As duas no questionário da qualidade são:

- “Após realizar a experiência, eu estou mais interessado em aprender sobre seu tópico”
- “Após realizar a experiência, eu estou mais inclinado a visitar um museu que a ofereça”

Estas duas últimas têm como objetivo medir a divergência no interesse com relação às respostas dadas para as duas primeiras perguntas. Assim, uma resposta de “7” para o item “tenho interesse em visitar um museu sobre computação”, seguida por uma resposta “4” para o item “após realizar a experiência, eu estou mais inclinado em visitar um museu que a ofereça”, não significa que o participante em questão ficou menos interessado, mas que seu interesse não diminuiu nem aumentou.

Tabela 9: Dados do interesse de participantes antes da experiência.

	Mínimo	Máximo	Média	Desvio padrão
Computação	1	7	3.778	1.783
Visitar	1	7	3.889	1.826



Tabela 10: Dados da mudança no interesse de participantes após a experiência.

	Mínimo	Máximo	Média	Desvio padrão
Computação	2	7	4.704	1.171
Visitar	2	7	5.370	1.305

Para testar a primeira hipótese, a hipótese nula  $H_0$  e a hipótese alternativa  $H_1$  são construídas:

$H_0$ : Não há aumento no interesse na visita após realizar a experiência.

$H_1$ : Há aumento no interesse na visita após realizar a experiência.

Qualquer resposta no intervalo  $[1, 4)$  para o item “Após realizar a experiência, eu estou mais inclinado a visitar um museu que a ofereça” indica que houve diminuição no interesse na visita, enquanto que 4 indica que não houve mudança, e qualquer valor no intervalo  $(4, 7]$  indica aumento no interesse. Assim, as hipóteses são reescritas como:

$$H_0: \mu \leq 4 \quad H_1: \mu > 4$$

Como nível de significância, é definido  $\alpha = 0.05$ . A média observada é de 5.370, com desvio padrão de 1.305. Com esses parâmetros, um teste t é realizado com 26 graus de liberdade, retornando um valor de 5.454 com valor-p, a probabilidade de encontrar esse valor caso a hipótese nula seja verdade, menor do que 0.001. Logo, a hipótese nula é rejeitada e, por consequência, é confirmado que a experiência aumenta o interesse em visitar o museu.

#### 6.3.4 Hipótese 4: Interesse em Computação

Um teste similar é feito para confirmar a segunda hipótese. A hipótese nula  $H_0$  e a alternativa  $H_1$  são:

$H_0$ : Não há aumento no interesse pela computação após realizar a experiência.

$H_1$ : Há aumento no interesse pela computação após realizar a experiência.

Qualquer resposta no intervalo  $[1, 4)$  para o item “Após realizar a experiência, eu estou mais interessado em aprender sobre seu tópico” indica que houve diminuição no interesse na visita, enquanto que 4 indica que não houve mudança, e qualquer valor no intervalo  $(4, 7]$  indica aumento no interesse. Assim, as hipóteses são reescritas como:

$$H_0: \mu \leq 4 \quad H_1: \mu > 4$$

Como nível de significância, é definido  $\alpha = 0.05$ , a média observada é de 4.704, com desvio padrão de 1.171. Com esses parâmetros, um teste t é realizado com 26 graus de liberdade, retornando um valor de 3.125 com valor-p = 0.002. Logo, a hipótese nula é rejeitada e, por consequência, é confirmado que a experiência aumenta o interesse pela computação.

### 6.3.5 Hipótese 5: Relação Entre Presença e Qualidade

A terceira hipótese é testada por meio da correlação de Pearson. Primeiro, a presença é analisada. Pela forma do questionário IPQ, seu resultado é dado em uma escala de 1 a 7, onde 1 indica que o respondente não sentiu presença, e 7 indica uma profunda sensação de presença. Os dados da presença relatada podem ser encontrados na Tabela 4.

O nível de presença foi similar ao alcançado em outras pesquisas nas quais experiências realistas foram implementadas [18], utilizando o mesmo questionário (IPQ). Esses resultados indicam que um bom nível de presença foi alcançado pela maior parte dos participantes. Pode ser dito pelos dados que o projeto foi bem sucedido em sua meta de ser uma experiência em realidade virtual que possa imergir usuários.

Tabela 11: Dados da qualidade da experiência dos participantes.

Mínimo	Máximo	Média	Desvio padrão
3.3	6.4	4.996	0.147

O valor da correlação entre a presença e a qualidade da experiência é de  $R = 0.5394$ , valor- $p = 0.003$ , o que indica uma correlação significativa entre as variáveis, confirmando a terceira hipótese.

Porém, o questionário da qualidade da experiência consiste de vários itens, tais como “eu gostei da experiência”, ou “foi fácil me movimentar pelo ambiente virtual”, permitindo analisar qual aspecto da experiência tinha mais relação com a presença. Analisando a correlação do item “Eu gostei da experiência” com a presença, é encontrado  $R = 0.039$ , valor- $p = 0.847$ , o que não implica uma correlação significativa. Este não era o resultado esperado, e contradiz resultados em diversas outras pesquisas que encontraram uma correlação positiva entre satisfação e presença. Uma correlação também foi encontrada com o item “Foi fácil me movimentar pelo ambiente virtual”, com  $R = 0.351$ , valor- $p = 0.072$ . Isto implica que uma maior facilidade na movimentação dentro do ambiente permite uma maior presença.

## 7 CONCLUSÃO

### 7.1 DISCUSSÃO

Este trabalho demonstra, por meio de um estudo de caso, a viabilidade da realidade virtual como um meio pelo qual museus possam enriquecer suas exposições sem grandes gastos. O projeto obteve sucesso em evidenciar que a replicação de artefatos de forma virtual fornece uma experiência de lazer e aprendizagem, permitindo ainda que o público interaja com a exposição, demonstrando que um museu pode ter valor agregado a suas exposições por meio da realidade virtual. Sucesso também foi alcançado em evidenciar o aumento do interesse em visitar um museu que ofereça esse tipo de experiência. Além disso, a menos que um museu adquira uma máquina que gostaria de exibir por doação, e seja mantida por trabalho voluntário, a utilização da realidade virtual apresenta um método de menor custo.

Pela demonstração, a aplicação de realidade virtual em smartphones é um rumo válido, mas idealmente equipamentos dedicados que permitam 6 DoF devem ser utilizados, ponto evidente pelas dificuldades relatadas com a navegação e controles durante a experiência.

Os testes foram realizados em um local remoto a um museu, simulando um cenário onde alguém sem acesso físico a ele encontrou a experiência online. Assim, os resultados não puderam ser impactados por uma experiência de visita a um museu que poderia influenciar suas impressões positivamente. Assim, é evidenciado que sua disponibilização online seria benéfica às pessoas que não possam visitar o museu fisicamente e aumentaria o interesse em uma visita ao museu para aquelas que possam visitá-lo.

Dificuldades com os controles foram os problemas mais relatados. Isso provavelmente foi causado pela falta de naturalidade dos controles: não há muita similaridade entre ações feitas fisicamente e resultados no ambiente virtual, por exemplo mover objetos por meio de um *joystick*. Uma outra razão contribuinte foi a falta de familiaridade com o tipo de controle. A melhor forma de resolver estes problemas é adquirir equipamento feito especificamente para realidade virtual. Algum tutorial poderia beneficiar aqueles que apresentaram dificuldade com os controles. Porém, isso estenderia o tempo de duração de cada experiência. Neste caso, um balanço deve ser encontrado entre nível de detalhe e duração do tutorial. Além disso, um tutorial

também beneficiaria quem vier a encontrar essa experiência online, tendo em vista que não poderá ter nada explicado sobre os controles por alguém.

Alguns participantes estavam sobre a impressão de que a organização das máquinas refletia a maneira como elas seriam organizadas no mundo real na época na qual eram utilizadas. Este não é o caso, e futuros trabalhos devem estar atentos ao fato de que a organização do ambiente deve, além de ser realista, mostrar objetos em seu contexto histórico, para evitar que usuários aprendam algo errôneo.

Por último, é notável a baixa correlação entre a presença e a satisfação. Embora a satisfação relatada tenha sido boa, como mostra a Tabela 7, era esperada uma relação mais forte assim como a relatada em outras pesquisas [3]. Uma resposta para seu baixo valor não foi encontrada.

## 7.2 TRABALHOS FUTUROS

Como mencionado diversas vezes, modificar a forma como se dá a interação com o ambiente virtual, permitindo uma interação mais natural, é vital para progredir nesta área. As respostas às perguntas “Foi fácil me movimentar pelo ambiente natural”, “A interação com o ambiente era natural”, e “Eu pude produzir os resultados desejados intuitivamente”, obtiveram médias de 4.67, 3.41, e 3.29, respectivamente. Porém, hardwares específicos para realidade virtual não são possuídos por grande parte da população, limitando o alcance de qualquer futuro trabalho que implemente essa opção. Uma opção mais viável para futuros trabalhos seria tentar implementar um esquema de controle comum em realidade virtual para smartphones: detecção do olhar. Ele permite ao usuário interagir com um objeto pondo a posição central do seu campo de visão sobre ele. Com sua adoção seria possível digitar uma mensagem no IBM 029 enquanto dentro do ambiente virtual, olhando para a tecla por um período predeterminado e longo o suficiente para garantir a intenção do usuário em “apertar” a tecla para a qual está olhando. Essa forma de digitação não seria tão rápida quanto a digitação implementada neste trabalho, mas seria mais natural por ser realizada durante a experiência e mais rápido e menos frustrante do que utilizar um joystick para indicar uma tecla. Esse método forneceria um meio termo entre o método implementado e o método de seleção da tecla por joystick. Um outro benefício desse esquema é que ele permite a navegação pelo ambiente virtual sem a necessidade de um controle.

No caso, a navegação é realizada pondo o ponto central do campo de visão na posição predeterminada para a qual se quer teleportar por um certo período de tempo. Isso eliminaria a necessidade de um controle bluetooth, abrangendo o alcance da experiência, porém uma forma de movimentar os cartões perfurados somente com o olhar provavelmente seria de difícil implementação, e não foi encontrada pelo autor deste trabalho. Com a adoção desse método, espera-se observar maiores qualidades e presenças relatadas.

Para complementar a experiência criada, adicionar outras máquinas que seriam vistas juntas ao IBM 029, seria ideal para compreender melhor o papel da computação em seu contexto histórico.

Além do dito acima, seria importante modificar os questionários para incluir perguntas que permitam adaptar a experiência às expectativas dos participantes, tais como “O que você mudaria na experiência?”.

Um outro caminho a ser tomado por futuros trabalhos seria a replicação do museu de computação da UFRJ em um ambiente virtual, permitindo que pessoas sem acesso físico a ele ainda possam desfrutar da experiência de visitá-lo. Porém, deveria ser investigado se essa experiência aumentaria ou diminuiria a intenção em visitá-lo.

## REFERÊNCIAS

- [1] SLATER, Mel. Place Illusion and Plausibility Can Lead to Realistic Behaviour in Immersive Virtual Environments. **Philosophical Transactions of the Royal Society B: Biological Sciences**, vol. 364, 2009. Disponível em: <https://doi.org/10.1098/rstb.2009.0138>. Acesso em: 26/09/2022
- [2] SLATER, Mel. “Break of Presence.” Introduction to Virtual Reality, **Coursera**. University of London. Interview. Disponível em: <https://www.coursera.org/learn/introduction-virtual-reality/lecture/ra7jh/break-of-presence>. Acesso em: 26/09/2022
- [3] SYLAIYOU, Stella; MANIA, Katerina; KAROULIS, Athanasios; WHITE, Martin. Exploring the relationship between presence and enjoyment in a virtual museum. **International Journal of Human-Computer Studies**, vol. 68, Issue 5, 2010. Disponível em: <https://doi.org/10.1016/j.ijhcs.2009.11.002>. Acesso em: 26/09/2022
- [4] USOH, Martin; ARTHUR, Kevin; WHITTON, Mary C., BASTOS, Rui; STEED, Anthony; SLATER, Mel; BROOKS, Frederick P., Jr. Walking > Walking-in-place > Flying, in Virtual Environments. In **Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99)**, 1999. Disponível em: <https://doi.org/10.1145/311535.311589>. Acesso em: 26/09/2022
- [5] IBM 29 Card Punch. **IBM Archives**. Disponível em: [https://www.ibm.com/ibm/history/exhibits/vintage/vintage\\_4506VV4002.html](https://www.ibm.com/ibm/history/exhibits/vintage/vintage_4506VV4002.html). Acesso em: 26/09/2022
- [6] OTHMAN, Mohd Kamal; PETRIE, Helen; POWER, Christopher. Engaging Visitors in Museums with Technology: Scales for the Measurement of Visitor and Multimedia Guide Experience. **Human-Computer Interaction - INTERACT 2011**, 2011. Disponível em: [https://doi.org/10.1007/978-3-642-23768-3\\_8](https://doi.org/10.1007/978-3-642-23768-3_8). Acesso em: 26/09/2022

[7] ALTIERI, Alex; CECCACCI, Silvia; GIRALDI, Luca; LEOPARDI, Alma; MENGONI, Maura; TALIPU, Abudukaiyoumu. Affective Guide for Museum: A System to Suggest Museum Paths Based on Visitors' Emotions. **Universal Access in Human-Computer Interaction. Design Methods and User Experience**. 2021. Disponível em: [https://doi.org/10.1007/978-3-030-78092-0\\_35](https://doi.org/10.1007/978-3-030-78092-0_35). Acesso em: 26/09/2022

[8] MOESGAARD, Tomas Gislason; WITT, Mass; FISS, Jonathan; WARMING, Cecilie; KLUBIEN, Jonathan; SCHONAU-FOG, Henrik. Implicit and Explicit Information Mediation in a Virtual Reality Museum Installation and its Effects on Retention and Learning Outcomes. **Proceedings of The 9th European Conference on Games-Based Learning : ECGBL 2015**, 2015. Disponível em: <https://vbn.aau.dk/en/publications/implicit-and-explicit-information-mediation-in-a-virtual-reality-> [y-](https://vbn.aau.dk/en/publications/implicit-and-explicit-information-mediation-in-a-virtual-reality-). Acesso em: 26/09/2022

[9] Virtual Reality In Gaming Market Size, Share & Trends Analysis Report By Component, By Device, By User (Commercial Space, Individual), By Region, And Segment Forecasts, 2020 - 2027. **Grand View Research**. 2018. Disponível em: <https://www.grandviewresearch.com/industry-analysis/virtual-reality-in-gaming-market/toc>. Acesso em: 26/09/2022

[10] RAAEN, Kjetil; Kjellmo, Ivar. Measuring Latency in Virtual Reality Systems. **Entertainment Computing - ICEC 2015**. 2015 Disponível em: [https://doi.org/10.1007/978-3-319-24589-8\\_40](https://doi.org/10.1007/978-3-319-24589-8_40). Acesso em: 26/09/2022

[11] NORMAN, Donald. The Design of Everyday Things. 1988

[12] Museum Definition. **International Council of Museums**. 2022. Disponível em: <https://icom.museum/en/resources/standards-guidelines/museum-definition/> Acesso em: 26/09/2022



- [13] GARNER, Robert. Restoring The IBM 1401. **Computer History Museum**, 2013. Disponível em: <https://computerhistory.org/blog/restoring-the-ibm-1401/>. Acesso em: 26/09/2022
- [14] STYLIANI, Sylaiou; FOTIS, Liarokapis; KOSTAS, Kotsakis; PETROS, Patias. Virtual museums, a survey and some issues for consideration. **Journal of Cultural Heritage**, vol. 10, Issue 4, 2007. Disponível em: <https://doi.org/10.1016/j.culher.2009.03.003>. Acesso em: 26/09/2022
- [15] INSKO, Brent E. Measuring Presence: Subjective, Behavioral and Physiological Methods. **Being there: Concepts, effects and measurement of user presence in synthetic environments** 5, 2003. Disponível em: <https://www.cs.ryerson.ca/aferworn/courses/CPS841/CLASSES/CPS841CL09/MeasuringPresence.pdf>. Acesso em: 26/09/2022
- [16] OWENS, Brent. Forward Rendering vs. Deferred Rendering. **Envato Tuts**, 2013. <https://gamedevelopment.tutsplus.com/articles/forward-rendering-vs-deferred-rendering--gamedev-12342> Acesso em: 24/09/2022
- [17] PSOTKA, Joseph. Immersive training systems: Virtual reality and education and training. **Instructional Science**, vol. 23, 1995. Disponível em: <https://doi.org/10.1007/BF00896880>. Acesso em: 08/10/2022
- [18] SCHWIND, Valentin; KNIERIM, Pascal; HAAS, Nico; HENZE, Niels. Using Presence Questionnaires in Virtual Reality. **Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems**, 2019. Disponível em: <https://doi.org/10.1145/3290605.3300590>. Acesso em: 19/11/2022
- [19] PAN, Sylvia Xueni. “What is VR and How is it Different from Other Media?”. Introduction to Virtual Reality, **Coursera**. University of London. Disponível em:

<https://www.coursera.org/learn/introduction-virtual-reality/lecture/PvDZ4/what-is-vr-and-how-is-it-different-from-other-media>. Acesso em: 05/12/2022

[20] JERALD, Jason J. Scene-motion-and latency-perception thresholds for head-mounted displays. Diss. The University of North Carolina at Chapel Hill, 2009. Disponível em: <https://www.proquest.com/openview/869308a96bbd47cda7382acdb5bf1955/1?pq-origsite=scholar&cbl=18750>. Acesso em: 09/01/2023

[21] UNITY. Create Gameplay. **Unity Manual**. Disponível em: <https://docs.unity3d.com/Manual/CreatingGameplay.html>. Acesso em: 09/01/2023

[22] BLENDER. **Blender Reference Manual**. Disponível em: <https://docs.blender.org/manual/en/latest/>. Acesso em: 09/01/2023

## APÊNDICE A - QUESTIONÁRIO PRELIMINAR

Os itens numerados de 4 a 6 são respondidos em uma escala Likert de 1 a 5.

1. Nome completo;
2. Idade;
3. Gênero;
4. Qual é a sua experiência com realidade virtual?
5. Tenho interesse na computação.
6. Tenho interesse em visitar um museu sobre computação.

## APÊNDICE B - QUESTIONÁRIO IPQ

O questionário original não é disponibilizado em Português. Essa versão é uma tradução do autor deste trabalho. Os itens dos questionários são afirmações ou perguntas que devem ser respondidas em uma escala Likert de 1 a 5. Os itens são divididos em três: presença espacial, envolvimento, e realismo experienciado. Além disso, existe um item que não se encaixa nessas divisões.

Item que não se encaixa:

- Eu tive uma sensação de “estar lá” no mundo gerado por computadores;

Presença espacial:

- Eu senti que o mundo virtual me rodeava;
- Eu senti como se estava somente olhando para fotos;
- Eu não me senti presente no espaço virtual;
- Eu tive uma sensação de agir dentro do espaço virtual, ao invés de agir operando algo fora do espaço;
- Eu senti presença no espaço virtual;

Envolvimento:

- O quão ciente você estava de que o mundo real o cercava enquanto navegava o mundo virtual?

- Eu não estava ciente do mundo real;
- Eu ainda prestava atenção ao mundo real;
- Eu estava completamente cativado pelo mundo virtual;

Realismo experienciado:

- O quão realista o mundo virtual pareceu para você?
- O quanto sua experiência no mundo virtual foi consistente com sua experiência no mundo real?
- O mundo virtual pareceu mais realista do que o mundo real.

## APÊNDICE C - QUESTIONÁRIO DA QUALIDADE

Todos os itens são respondidos em uma escala Likert de 1 a 5.

Relação com a experiência:

- Eu gostei da experiência.
- A experiência foi interessante.
- Minha experiência foi inspiradora.
- Eu senti que estava participando da experiência, ao invés de somente a assistindo.

Realismo da experiência:

- O ambiente no qual estava era realista.
- Eu fiquei impressionado pelo aspecto estético da exibição.
- Foi fácil me movimentar pelo ambiente virtual.
- A interação com o ambiente era natural.
- Eu pude produzir os resultados desejados intuitivamente.

Aprendizagem:

- Eu pude entender tudo que foi visto e feito na exibição.
- Minha experiência enriqueceu meu conhecimento.
- Após realizar a experiência, eu estou mais interessado em aprender sobre seu tópico.
- Após realizar a experiência, eu estou mais inclinado a visitar um museu que a ofereça.