

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

TXAI DURIGON WIESER

**REcolorindo o Reino: Desenvolvimento de
jogo infantil utilizando detecção de cores.**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Profa. Dra. Renata Galante

Porto Alegre
2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^ª. Patricia Pranke

Pró-Reitora de Graduação: Prof^ª. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^ª. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Rodrigo Machado

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“What we see changes what we know.
What we know changes what we see.”*

— JEAN WILLIAM FRITZ PIAGET

AGRADECIMENTOS

Gostaria de agradecer primeiramente à meus pais, Luciano e Raquel, por acreditarem que educação pode mudar o mundo, sempre me incentivado na busca de conhecimento e por terem me apoiado tanto durante esses quase 10 anos de vida misturando trabalho, estudos e lazer.

Agradeço também ao resto da minha família por sempre acreditarem em meu potencial, em especial à minha tia Luciane que me acolheu em sua casa nos primeiros anos de faculdade, ao meu irmão Tayhú pelas risadas e brincadeiras que me ajudaram relaxar durante longos períodos de estudo e à minha namorada Mariana que me apoiou e incentivou enquanto eu escrevia este trabalho.

E por fim, um agradecimento especial à professora Renata Galante, que me deu aula lá no começo do curso e que agora no final me acolheu e aceitou orientar o desenvolvimento este trabalho.

Muito obrigado.

RESUMO

Sabemos que a introdução de dispositivos móveis cada vez mais cedo na rotina de crianças vem acontecendo desde sua grande popularização, hoje em dia é comum que crianças ainda não alfabetizadas já utilizem *tablets* e *smartphones*, permanecendo ociosas por grande parte do seu dia e assim perdendo interações com o mundo real ímpares para seu desenvolvimento motor. Foi diante deste problema que surgiu o desafio desse trabalho: Criar um jogo infantil que estimule o movimento, mesclando o contato com o mundo real e o mundo virtual. Um jogo infantil que possibilita o aprendizado através da diversão e da brincadeira, promovendo o aperfeiçoamento das habilidades iniciais com uma maior plenitude. Na envolvente história de REcolorindo o Reino, a criança tem um papel desafiador: ajudar os personagens a salvarem o reino de uma terrível maldição que o descoloriu. O desafio do jogo consiste em fazer a criança achar as cores verdadeiras dos objetos do reino imaginário no seu mundo real, recolorindo o reino e acabando com a maldição. REcolorindo o Reino apresenta uma jogabilidade diferenciada e desafiadora em relação à maioria dos jogos existentes, pois exige que a criança movimente-se e explore o espaço físico, relacionando a sua realidade com a do jogo, incentivando o desenvolvimento do raciocínio lógico e de habilidades motoras essenciais para uma criança de 3 à 6 anos. Isso se faz possível graças a diferente forma de interação proposta no jogo, o reconhecimento das cores através da câmera do dispositivo. Durante a exploração do ambiente ao seu redor, a criança aponta o dispositivo para diversos objetos enquanto caminha, e em tempo real, imagens da câmera são capturadas e processadas pelo jogo, e através de um algoritmo de predominância de cores o jogo detecta as cores principais dos objetos que estão na direção do dispositivo, assim o jogo conecta informações dos dois mundos, criando uma dinâmica que faz com que as crianças se divirtam e exercitem diferentes habilidades do seu corpo, trazendo inúmeros benefícios. Através dos experimentos aplicados com crianças e seus responsáveis adultos foi percebido uma forte aceitação de jogos com temáticas infantis e que incentivem movimentação e interação com o mundo real.

Palavras-chave: Jogo, Infantil, Cores, Imagens, App, iOS.

REcolorindo o Reino: Children's game development using color detection.

ABSTRACT

We know that the introduction of mobile devices earlier and earlier in the routine of children has been happening since its great popularity, nowadays it is common for children who are not yet literate to use tablets and smartphones, remaining idle for a long part of the day and thus losing unique real-world interactions for their motor skill development. It was because of this problem that the challenge of this project was chosen: Create a children's game that encourages movement, mixing contact with the real world and the virtual world. A children's game that enables learning through fun and play, promoting the improvement of initial skills with greater fullness. In the immersive story of *REcolorindo o Reino*, the child has a challenging role: help the characters save the kingdom from a terrible curse that discolored it. The game's challenge is to make the child find the true colors of the objects of the imaginary realm in their real-world, recoloring the realm and ending the curse. *REcolorindo o Reino* presents different and challenging gameplay compared to most existing games as it requires the child to move and explore the physical space, relating their reality to the game, encouraging the development of logical reasoning and motor skills essential for a 3 to 6-year-old child. This is made possible thanks to the different forms of interaction proposed in the game, the recognition of colors through the device's camera. During the exploration of the environment around them, the child points the device at various objects while walking, and in real-time, images from the camera are captured and processed by the game, and through a color predominance algorithm, the game detects the primary colors of the objects that are in the direction of the device, so the game connects information from the two worlds, creating a dynamic that makes children have fun and exercise different body skills, bringing countless benefits. Through the experiments applied with children and their adult guardians, a strong acceptance of children-themed games that encourage movement and interaction with the real world was noticed.

Keywords: Game, Children, Colors, Images, App, iOS.

LISTA DE FIGURAS

Figura 2.1	Exemplo de código declarativo em <i>SwiftUI</i>	21
Figura 2.2	Diagrama da arquitetura <i>MVC</i>	22
Figura 2.3	Diagrama da arquitetura <i>MVVM</i>	23
Figura 2.4	Percepção humana por receptor.	25
Figura 2.5	Representação do gamut de sRGB comparado em CIE1931.	27
Figura 2.6	Representação do espaço CIELAB vista de cima	28
Figura 3.1	Estética e desafios do jogo <i>Peek-a-Zoo</i>	32
Figura 3.2	Deteção de objetos do jogo <i>Emoji Scavenger Hunt</i>	33
Figura 3.3	Integração com a realidade do jogo <i>ARrrrrgh</i>	34
Figura 3.4	Interação do jogo <i>Pokemon GO</i> com a realidade.....	35
Figura 4.1	Cronograma de desenvolvimento.	38
Figura 4.2	Representação gráfica da Cena de Apresentação (wireframe).	45
Figura 4.3	Representação gráfica da Cena de Introdução (wireframe).....	45
Figura 4.4	Representação gráfica da Cena do Mapa do Reino (wireframe).	46
Figura 4.5	Representação gráfica da Cena da Região: O Bosque (wireframe).....	47
Figura 4.6	Representação gráfica da Cena de Desafio (wireframe).	48
Figura 4.7	Representação gráfica do Cena de Opções (wireframe).	49
Figura 4.8	Fluxograma da Cena de Desafio.	50
Figura 4.9	Diagrama das portas de acesso entre as cenas.	52
Figura 4.10	Fluxograma de navegação.	53
Figura 4.11	Diagrama arquitetural da cena de desafio.	60
Figura 4.12	Resultado de testes unitários com imagens da camera.	64
Figura 4.13	Exemplo de 3 cenas que foram desenvolvidas	65
Figura 4.14	Conjunto de todos personagens desenvolvidos.....	66
Figura 4.15	Conjunto de todos itens desenvolvidos.....	66
Figura 4.16	Conjunto de todas as regiões desenvolvidas	67
Figura 4.17	À esquerda: cena da introdução. À direita: cena do encerramento.....	67
Figura 5.1	Pirâmide da necessidade dos jogadores.....	70
Figura 5.2	Tarefas realizadas durante os testes:	74
Figura 5.3	Respostas PD1	74
Figura 5.4	Respostas PD2	75
Figura 5.5	Respostas PD3	75
Figura 5.6	Respostas PD4	75
Figura 5.7	Respostas PD5	76
Figura 5.8	Respostas PD6	76
Figura 5.9	Respostas PA1.....	78
Figura 5.10	Respostas PA2.....	78
Figura 5.11	Respostas PA3.....	78
Figura 5.12	Respostas PA4.....	79
Figura 5.13	Respostas PA5.....	79

LISTA DE TABELAS

Tabela 1.1	Características do desenvolvimento corporal	13
Tabela 3.1	Comparação de funcionalidades.....	36
Tabela 4.1	Metas e Principais Tarefas da Sprint 1	39
Tabela 4.2	Metas e Principais Tarefas da Sprint 2	39
Tabela 4.3	Metas e Principais Tarefas da Sprint 3	40
Tabela 4.4	Metas e Principais Tarefas da Sprint 4	40
Tabela 4.5	Metas e Principais Tarefas da Sprint 5	41
Tabela 4.6	Metas e Principais Tarefas da Sprint 6	41
Tabela 4.7	Metas e Principais Tarefas da Sprint 7	42
Tabela 4.8	Benefícios de ações do jogo	44
Tabela 4.9	Objetos de cada nível e região	51
Tabela 4.10	Dispositivos compatíveis	55
Tabela 4.11	Chaves e descrições dos itens armazenados	56
Tabela 4.12	Valores de comparação do CIE Delta E	63
Tabela 5.1	Aspectos de usabilidade em jogos e aplicações.	70

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
APP	Application
AR	Augmented Reality
CBL	Challenge Based Learning
CIE	International Commission on Illumination
FPS	Frames Per Second
GDD	Game Development Design
HTTP	Hypertext Transfer Protocol
MVC	Model-View-Controller
MVVM	ModelView-View-Controller
NPC	Non-Player Character
RGB	Red Green Blue
SDK	Software Development Kit
SSD	Solid State Drive
UFRGS	Universidade Federal do Rio Grande do Sul
UI	User Interface
UX	User Experience
VGA	Video Graphics Array

SUMÁRIO

1 INTRODUÇÃO	12
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 Challenge Based Learning	15
2.1.1 Engage	15
2.1.2 Investigate	16
2.1.3 Act	16
2.2 Métodos Ágeis	17
2.2.1 <i>Backlog</i> do Produto	17
2.2.2 <i>Sprint</i>	18
2.2.3 Ciclo de Vida de Lançamento	18
2.3 Tecnologias	19
2.3.1 Plataforma <i>iOS</i>	19
2.3.2 Linguagem <i>Swift</i>	19
2.3.3 Framework <i>SwiftUI</i>	20
2.4 Arquitetura	21
2.4.1 Model-View-Controller	22
2.4.2 Model-View-ViewModel	23
2.5 Colorimetria	24
2.5.1 Percepção	25
2.5.2 Representações	26
2.5.2.1 <i>sRGB</i> - Red-Green-Blue	26
2.5.2.2 <i>CIELAB L*a*b*</i>	27
2.5.3 Comparações	28
2.5.3.1 Distância Euclidiana	28
2.5.3.2 <i>CIELAB</i> Delta E	29
3 TRABALHOS RELACIONADOS	31
3.1 Peek-a-Zoo - Animal Games	31
3.2 Emoji Scavenger Hunt - Google	32
3.3 ARrrrrgh - A magical treasure adventure	33
3.4 Pokemon GO	34
3.5 Análise comparativa	35
4 DESENVOLVIMENTO	37
4.1 Idealização	37
4.2 Gerenciamento	38
4.2.1 Cronograma	38
4.2.1.1 Sprint 1	39
4.2.1.2 Sprint 2	39
4.2.1.3 Sprint 3	40
4.2.1.4 Sprint 4	40
4.2.1.5 Sprint 5	41
4.2.1.6 Sprint 6	41
4.2.1.7 Sprint 7	42
4.3 Game Design Document	42
4.3.1 Enredo	42
4.3.1.1 Personagens principais	43
4.3.1.2 História	43
4.3.2 <i>Gameplay</i>	44

4.3.3	Composição do jogo	44
4.3.3.1	Cena de Introdução	45
4.3.3.2	Cena de Mapa do Reino	46
4.3.3.3	Cena de Região	46
4.3.3.4	Cena de Desafio	47
4.3.3.5	Cena de Encerramento	48
4.3.3.6	Cena de Opções.....	48
4.3.4	Funcionamento dos desafios	49
4.3.4.1	Descrição da Cena de Desafio.....	50
4.3.4.2	Lista dos níveis	51
4.3.5	<i>Estrutura e Progressão</i>	51
4.3.5.1	Hierarquia e acesso das cenas	52
4.4	Implementação	53
4.4.1	Visão geral	54
4.4.2	Arquitetura geral	56
4.4.2.1	Models.....	56
4.4.2.2	Views.....	57
4.4.2.3	ViewModels	58
4.4.2.4	Managers.....	59
4.4.3	Arquitetura da cena	60
4.4.4	Detecção de Cores.....	61
4.4.4.1	Captura de Imagens.....	61
4.4.4.2	Predominância de Cores	61
4.4.4.3	Comparação de Similaridade	62
4.5	Resultado	64
4.5.1	Funcionalidades do Aplicativo.....	65
4.5.2	Material Gráfico	65
4.5.3	Material de Áudio	68
4.5.4	<i>Cheat Code</i>	68
4.5.5	Atribuições	68
5	AVALIAÇÃO COM USUÁRIOS	69
5.1	Experimento	69
5.1.1	Análise de Uso	71
5.1.1.1	Lista de tarefas esperadas:	71
5.1.2	Questionários	72
5.1.2.1	Perguntas para crianças:.....	72
5.1.2.2	Perguntas para adultos:	72
5.1.2.3	Perguntas demográficas:	72
5.2	Resultados.....	73
5.2.1	Análise de uso	73
5.2.1.1	Perguntas demográficas:	74
5.2.2	Perguntas para crianças.....	76
5.2.3	Perguntas para adultos	77
6	CONCLUSÃO	81
	REFERÊNCIAS.....	83

1 INTRODUÇÃO

Devido a rápida evolução de dispositivos eletrônicos móveis, como *smartphones* e *tables*, e como previsto por MYERS; BEIGL em 2003, vivemos hoje numa época onde eles se tornaram o paradigma dominante da computação pessoal. Assim, da mesma maneira como mudaram a forma que adultos se comunicam, trabalham e se divertem, também estão influenciando de diversas maneiras o desenvolvimento infantil.

Atividades tradicionais que eram muito presentes no dia-a-dia de crianças durante décadas passadas, como pega-pega, queimada, amarelinha e taco, hoje já são mais difíceis de encontrar e os presentes preferidos migraram de bolas, bonecos, bicicletas e patins para qualquer que seja o eletrônico do momento. É fato que hoje crianças passam bastante tempo interagindo com interfaces digitais enquanto permanecem sentadas (ou até deitadas) e isso implica diretamente em quanto elas estão se estimulando fisicamente e interagindo com o mundo real.

Nesse projeto, são explorados mecanismos tecnológicos, práticas pedagógicas e brincadeiras lúdicas para entregar um *Jogo Sério* que estimula o desenvolvimento físico e mental de crianças, apresentando uma jogabilidade diferenciada e desafiadora em relação à maioria dos jogos existentes, que incentiva o movimento e exploração do espaço físico, relacionando a realidade do jogador com a do jogo.

Em 2015, PAIVA; COSTA descreve que as crianças do século XXI nasceram em um período no qual a tecnologia é o alicerce da manutenção das relações sociais, e que por isso, tornaram-se indispensáveis. Assim, crianças aprendem a utilizar dispositivos eletrônicos muito antes de serem alfabetizadas e essa condição pode provocar dificuldades no processo de aprendizagem e no desenvolvimento corporal.

Esses novos paradigmas contribuem para o aumento do sedentarismo entre as crianças, que trocaram brincadeiras de rua por sofás e eletrônicos. Segundo Guerra, Farias e Florindo (2016) as causas do comportamento sedentário são diversas, mas entre as mais frequentes está o tempo de tela, que combina os indicadores tempo de televisão, computador, videogame, *smartphones* e *tablets*. Desde então, diversas alternativas estão sendo estudadas para estimular hábitos saudáveis em crianças.

Durante as idades de 0-6 anos, crianças enfrentam seu período mais importante de descobrimento corporal. Segundo Piaget (1994), consciência corporal é algo que se desenvolve naturalmente na infância, se esta tiver permissão de conhecer seu corpo, o que implica experimentar os movimentos, utilizá-los com desenvoltura e ter a sensação

de domínio deste corpo.

A ciência que tem como objeto de estudo o homem através do seu corpo em movimento e em relação ao mundo é a psicomotricidade e está relacionada ao processo de maturação, onde o corpo é a origem das aquisições cognitivas, afetivas e orgânicas. (PSICOMOTRICIDADE, 2015)

Durante o desenvolvimento infantil, algumas das principais características são: Percepção espacial, percepção visual, motricidade ampla, e motricidade fina. Como pode ser visto na tabela 1.1.

Tabela 1.1 – Características do desenvolvimento corporal

Característica	Descrição
Percepção Espacial	Orientação e interação no espaço com seu corpo e objetos.
Percepção Visual	Percepção e distinção de ilustrações e objetos reais, e suas formas e cores.
Motricidade Ampla	Equilíbrio do corpo e coordenação motora.
Motricidade Fina	Integração entre os gestos manuais, a visão e os movimentos do corpo.

Fonte: Psicomotricidade (2015)

Evidências apontam que jogos são utilizados no aprendizado desde a Roma antiga onde possuíam temáticas similares a tarefas realizadas por adultos e eram criados para que crianças treinassem para seu futuro trabalho. (KISHIMOTO, 2011)

Nos dias atuais, a grande maioria dos jogos possuem o propósito único de entretenimento, por isso, uma nova categoria chamada *Jogos Sérios* foi criada para abranger todo jogo que possui propósitos além do entretenimento, como educação, treinamento profissional, saúde, publicidade, e políticas públicas. Jogos sérios para crianças pequenas que estão tendo seu primeiro contato com dispositivos eletrônicos são raros. Isso leva essas crianças a consumir jogos de uma categoria destinada a outra faixa etária.

O objetivo desse projeto é ocupar esta lacuna educacional de valor ímpar ao desenvolvimento infantil, sendo elaborado pensando nas características desta fase. O desenvolvimento explora o estímulo ao movimento, mesclando o contato com o mundo real e o mundo virtual, possibilitando o aprendizado através da diversão e da brincadeira, promovendo o aperfeiçoamento das habilidades iniciais com uma maior plenitude. A diferente forma de interação proposta no jogo é reconhecer cores através da câmera do dispositivo, combinando uma história envolvente, uma arte atraente, narrações e efeitos sonoros que estabelecem uma ligação entre a realidade e a fantasia, onde o jogador é desafiado a sair em busca de cores do mundo real para completar a história, criando uma dinâmica que faz com que as crianças se divirtam e exercitem diferentes habilidades do seu corpo, trazendo

inúmeros benefícios.

Para o desenvolvimento completo e eficiente desse trabalho, foram utilizadas diversas tecnologias, técnicas e padrões utilizados pelo mercado de Engenharia de *Software*, incluindo métodos ágeis, padrões de arquitetura de código para aplicações móveis, armazenamento de informações *on-device*, *frameworks* de interface, padrões de *design* de jogos, e implementação das representações, detecções e comparações de cores capturadas através da câmera de dispositivo.

Este trabalho apresenta a seguinte organização. No segundo capítulo (Fundamentação Teórica), são aprofundados os conteúdos e tecnologias utilizadas para na concepção e desenvolvimento do projeto, como as metodologias utilizadas, detalhes da interpretação, arquitetura, colorimetria e como a percepção humana se relaciona com as diversas representações de cores. No terceiro capítulo (Trabalhos Relacionados), são apresentados jogos que compartilham características com esse projeto e uma breve comparação das funcionalidades que possuem. No quarto capítulo (Desenvolvimento), é apresentado o processo de desenvolvimento do jogo, apresentando partes do *Game Design Document*, arquiteturas utilizadas e o resultado do final do jogo. No quinto capítulo (Avaliação com Usuários), é detalhada a pesquisa conduzida com usuários e a análises dos resultados obtidos. Por fim, o sexto capítulo (Conclusão) contém uma retrospectiva do que foi aprendido com este trabalho e são analisados pontos de melhoria e possíveis futuras funcionalidades para o jogo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, consta o referencial teórico para o embasamento do projeto. Aqui, foram explorados conteúdos como colorimetria, métodos de captura e processamento de imagens, tecnologias de desenvolvimento de software e arquiteturas de código. Também descreveremos duas metodologias principais que foram utilizadas nesse projeto, uma para idealização e especificação, e outra para o desenvolvimento. Os assuntos são dispostos aqui em ordem que foram estudados e de acordo com a necessidade durante o projeto.

2.1 Challenge Based Learning

Para idealização e especificação do projeto, ou seja, decidir qual a área de atuação, qual o problema estudado e a correspondente proposta de solução, foi utilizado o *framework Challenge Based Learning* ou *CBL*. A estrutura da *CBL* é colaborativa e prática, pedindo a todos os participantes que identifiquem ideias (*Big Ideas*), façam boas perguntas (*Essential Questions*), identifiquem e resolvam desafios (*Challenges*), adquiram conhecimento profundo da área temática, desenvolvam habilidades do século XXI e compartilhem seus pensamentos com o mundo. (CBL.ORG, 2021)

A *CBL* foi criada como resultado de estudos e análises de metodologias realizados por educadores exemplares de todo mundo durante o "*Apple Classrooms of Tomorrow-Today*", um projeto que tinha como princípio identificar métodos essenciais do *design* do ambiente de aprendizagem do século XXI. *CBL* é dividida em três principais partes: *Engage*, *Investigate* e *Act*. Cada uma dessas atividades é interconectada e precisa da anterior para ser completada.

2.1.1 Engage

A primeira etapa do processo, participantes devem se familiarizar com o conteúdo de estudo, definindo uma grande ideia de área de pesquisa, e anotar diversas questões essenciais sobre o problema, chegando num desafio específico de escolha.

Big Idea: Um conceito amplo que pode ser explorado de diversas maneiras, geralmente relata a área a ser abordada no desafio. Alguns exemplos são: Comunidade, Saúde, Criatividade e Democracia.

Essential Questions: Com base na *Big Idea* surgem diversas perguntas e questionamentos. O processo define que se crie muitos questionamentos, pensando divergente e convergente-mente. Alguns exemplos são: "Por que pessoas se alimentam mal?" e "Quais as motivações de uma pessoa ao abrir um negócio próprio?".

Challenge: Através das *Essential Questions* devemos extrair um só desafio, que será o alvo da tentativa de solução do trabalho. Algo que seja imediato, possível de ser resolvido e que crie entusiasmo. Alguns exemplos são: "Como facilitar a rotina de alimentação saudável?" e "Como aproximar empresas para colaboração através de parcerias?".

2.1.2 Investigate

Na segunda etapa do processo, o *Challenge* obtido na primeira etapa é estudado e aprofundado com pesquisas, artigos, sites e entrevistas, para criar um melhor entendimento dos caminhos possíveis para resolvê-lo.

Guiding Questions: Produzir uma quantidade grande de perguntas convergentes para o desafio. Essas perguntas devem nos guiar a uma solução e refinar ainda mais o problema. Alguns exemplos são: "Como parcerias entre empresas funcionam?" e "Quais os benefícios das parcerias entre empresas?".

Guiding Activities: Listagem de atividades, pessoas e tudo o que pode ajudar durante a fase de investigação. Alguns exemplos são: Cursos, experimentos, livros, questionários e jogos, *papers*.

Synthesis: Documentação de tudo aprendido nessa etapa do processo.

2.1.3 Act

É a etapa final da metodologia onde conceitos e soluções são concebidas. A partir da documentação anterior, uma lista de possibilidades de soluções é criada, para então ser avaliada e finalmente uma das soluções escolhida para ser desenvolvida.

Solution Concepts: Deve-se criar conceitos de soluções bem claras, definidas e detalhadas serem analisadas e possivelmente serem desenvolvidas. Um exemplo é: "Desenvolver uma aplicação no formato rede social que conecte empreendedores com

objetivos similares e conhecimentos diferentes para colaborar em projetos inter-empresariais”.

Solution Development: Um dos conceitos concebidos na etapa anterior é escolhido para desenvolvimento. Nessa etapa, podemos utilizar diversas metodologias, como por exemplo, métodos ágeis, modelo cascata e *Extreme Programming*.

2.2 Métodos Ágeis

Métodos Ágeis é um conjunto de práticas e processos utilizados no desenvolvimento e entrega de software. Seu nome foi escolhido por tratar de uma abordagem menos burocrática em comparação à outros modelos aplicados na época. No Manifesto para Desenvolvimento Ágil de Software (BECK et al., 2001), valores e princípios foram documentados, tendo como principal as seguintes ideias:

A priorização de:

- Indivíduos e iterações mais que processos e ferramentas;
- Software funcional mais que documentação abrangente;
- Colaboração do cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Os projetos ágeis são divididos em iterações, cada uma com seus próprios processos, levantamentos de requisitos, desenvolvimento e testes. O foco deve ser entregar um projeto com o máximo de valor agregado, por isso, partes secundárias, como documentação, devem ser minimalistas, dando prioridade para os requisitos funcionais. Outra característica é que o cliente fica mais envolvido no processo, e no final de cada iteração podem ser feitas correções na direção do projeto, assim diminuindo muito o tempo desperdiçado. Nas próximas seções, são detalhadas algumas práticas ágeis importantes para o desenvolvimento esse trabalho.

2.2.1 Backlog do Produto

O *backlog* do produto é uma divisão do trabalho a ser feito, cada item pode ter diferentes formatos, como histórias de usuário ou casos de uso. Esses itens definem

funcionalidades, correções, requisitos funcionais e não funcionais, qualquer tarefa que contribua para um projeto melhor. Os itens são priorizados pelo time, ou pelo *product owner* e muitas vezes também ganham um valor associado a dificuldade ou tempo de implementação.

2.2.2 Sprint

Sprints, também conhecidas como iterações, são caixas de tempo de geralmente 1-4 semanas onde certa quantidade de itens do *backlog* geral serão realizadas. No começo de cada *sprint* é feito um planejamento onde é definido um objetivo, e tarefas do *backlog* geral são selecionadas e o *backlog* específico da *sprint* é criado. No final de cada *sprint* acontece uma cerimônia onde os resultados são analisados e apresentados para o *Stakeholder*. Antes da próxima *sprint*, uma cerimônia de retrospectiva também pode ser realizada, onde são discutidas melhorias e as lições aprendidas durante esse período.

2.2.3 Ciclo de Vida de Lançamento

O ciclo de vida de lançamento (*Release Lifecycle*) define algumas entregas em diferentes estágios do desenvolvimento, cada uma com um nível de maturidade do produto. Essas fases podem variar de protótipos até correções de *bugs* da versão final.

Existem diversas fases definidas. A seguir, são descritas as mais utilizadas:

Alpha: Primeiro artefato de software que vai ser testado por pessoas fora do ambiente de desenvolvimento, geralmente usando técnicas *Black-box*. É muito útil para validações de funcionalidades com usuários e se o software resolve o problema proposto, mesmo que ainda não completamente ou com pequenos erros.

Beta: Esse lançamento geralmente acontece quando o software já tem todas suas funcionalidades completas, mas que ainda não foram vastamente testadas, isso implica que muitas podem ainda conter *bugs* e problemas de performance.

Release Candidate: É uma versão beta que já passou por algumas iterações de testes e correções e que agora tem o potencial de ser estável o suficiente para ser lançada oficialmente. Como é apenas uma candidata, essa versão ainda pode ser rejeitada e substituída por uma nova.

Stable Release: Também chamada versão de produção, é uma versão candidata que passou por todos testes e verificações e foi aprovada para lançamento. Essa versão é lançada publicamente ou entregue para o cliente final.

2.3 Tecnologias

Para o desenvolvimento desse trabalho são necessárias diversas tecnologias, descreveremos nessa seção algumas delas, como sistema operacional, a linguagem de programação, e *framework* gráfico.

2.3.1 Plataforma *iOS*

O sistema operacional *iOS*, desenvolvido pela *Apple*, foi anunciado para o mundo em 2007 juntamente com o primeiro dispositivo rodando, entregando uma revolução no mercado de dispositivos móveis como manipulação direta da interface através de controles multi-toque. O destino do *iOS* são os dispositivos eletrônicos próprios da empresa como *iPhones*, *iPads* e *iPods* e é hoje um dos sistemas operacionais móveis mais utilizados.

Proporcionando uma plataforma avançada juntamente com *SDKs* de alta qualidade e um sistema de venda e assinaturas, *iOS* se tornou a plataforma de escolha para desenvolvedores ao redor do mundo. Criando um ecossistema de aplicativos que nunca antes tinham tido um ambiente nutritivo para crescerem na escala mundial.

2.3.2 Linguagem *Swift*

Anunciada em 2014, a linguagem de programação *Swift* tem rapidamente se tornado uma das linguagens que cresceu mais rapidamente em sua história. *Swift* torna fácil a escrita de software que é rápido e seguro por padrão. (APPLE, 2008)

Além de ter seu código aberto, *Swift* é desenvolvida abertamente, ou seja, todas as funcionalidades e mudanças são discutidas pela comunidade, passam por um processo de estudo, análise e aprovação para só depois serem incluídas na linguagem. *Swift* funciona oficialmente para todos dispositivos *Apple* bem como *Linux* e *Windows*, e possui soluções ainda não oficiais para *Android* e microcontroladores.

Embora *Swift* seja uma linguagem de propósito geral, é considerada altamente

opinativa e foi construída utilizando abordagens modernas em relação a segurança, performance e padrão de design de software. Algumas das principais características de *Swift* são descritas a seguir:

Segura: O jeito mais óbvio de escrever código deve também se comportar de uma maneira segura. Comportamento indefinido é o principal inimigo da segurança, e os erros de desenvolvedores devem ser detectados antes do software estar em produção. Escolher por segurança algumas vezes significa que *Swift* vai parecer mais estrita, mas clareza e concisão salvam tempo longo prazo.

Rápida: *Swift* é destinada como um substituto para linguagens baseadas em C (C, C++ e Objective-C). Por isso, *Swift* deve ter performance comparável a essas linguagens na maioria das tarefas. Performance precisa também ser previsível e consistente. Existem diversas linguagens com muitas *features* novas e interessantes, mas ser também rápida é uma característica rara.

Expressiva: *Swift* se beneficia de décadas de avanço na ciência da computação para oferecer uma sintaxe que é prazerosa de usar, com características modernas que desenvolvedores esperam. As funcionalidades de *Swift* são desenvolvidas para funcionar juntamente e criar uma linguagem que é poderosa e divertida de usar.

Algumas das principais funcionalidades são:

- *Closures* são unificadas com ponteiros de função.
- Tuplas e retorno de múltiplos valores.
- Rápida e concisa iteração em coleções e *Ranges*.
- *Structs* que suportam métodos, extensões e protocolos.
- Paradigmas de programação funcional, como *map*, *filter* e *reduce*.
- Sistema de tratamento de erros robusto.
- Controle de fluxo avançado, com *do*, *guard*, *defer*, e *repeat*.

2.3.3 Framework *SwiftUI*

SwiftUI é uma ferramenta de criação de interface que proporciona se criar design de aplicativos com código em uma maneira declarativa (HUDSON, 2020). Isso é, uma maneira de dizer ao *framework* como queremos que a interface apareça e funcione, e os detalhes exatos de aparência são definidos pela plataforma que essa interface está sendo

executada. Por exemplo, ao rodar o mesmo código de um botão no *iPhone* e no *Mac* o botão poderá ter uma aparência diferente, respeitando os padrões de cada dispositivo.

A diferença entre uma interface declarativa é melhor compreendida quando comparada com uma interface imperativa. Basicamente, numa interface imperativa, ao clicar em um botão por exemplo, podemos criar um novo componente e adicionar ele no canto da tela, e depois ativar a opacidade de um outro componente. Na interface declarativa declaramos todos os estados possíveis da nossa interface e as funções de transição, e ao clicar no botão somente acontece uma mutação do estado e a *view* vai para o estado previamente declarado.

A Figura 2.1 ilustra um exemplo de código declarativo de uma *view* em *SwiftUI*:

Figura 2.1 – Exemplo de código declarativo em *SwiftUI*

```
import SwiftUI

struct Content : View {

    @State var model = Themes.listModel

    var body: some View {
        List(model.items, action: model.selectItem) { item in
            Image(item.image)
            VStack(alignment: .leading) {
                Text(item.title)
                Text(item.subtitle)
                    .color(.gray)
            }
        }
    }
}
```

Fonte: Apple (2020)

2.4 Arquitetura

São diversas as arquiteturas existentes para jogos e a utilização de cada uma é relacionada a preferência do time e principalmente a escolha do *framework*/plataforma de desenvolvimento. Grandes plataformas, como *Unity* e *Unreal*, tendem a utilizar uma arquitetura baseada em componentes, como a *Entity Component System*, já para aplicativos móveis que usam *frameworks* de interface declarativos as mais utilizadas são variações de *MVC* e/ou *MVVM*.

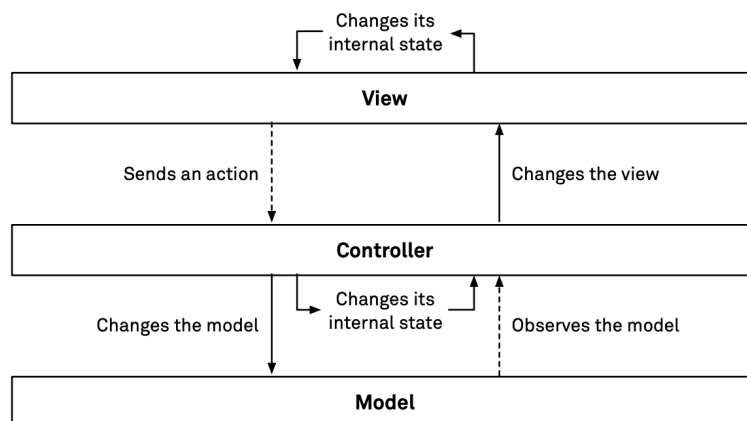
2.4.1 Model-View-Controller

É um dos mais influentes padrões arquiteturais, mas sua história de criação não é muito clara. Segundo Mitchell (2014), *MVC* foi inventado no Parque Xerox nos anos 70, supostamente por *Trygve Reenskaug* e sua primeira aparição pública foi na linguagem *Smalltalk-80*, mas por muito tempo existia pouquíssima documentação sobre seu funcionamento.

Durante a introdução do primeiro *SDK* para *iPhones* pela *Apple* em 2008 a recomendação de arquitetura para desenvolvimento de aplicações foi *MVC*, com apenas algumas adaptações para funcionar com o *framework* de *UI* mais popular da época (*UIKit*).

MVC possui 3 componentes principais com funções distintas, o *Model*, a *View* e a *Controller* e as três trabalham interligadas para responder as ações de uma aplicação, como pode ser visto na Figura 2.2.

Figura 2.2 – Diagrama da arquitetura *MVC*



Fonte: Eidhof Florian Kugler (2018)

MVC é baseada em princípios clássicos de *Orientação a Objetos*: Objetos gerenciam seus comportamentos e estado internamente e se comunicam através de interfaces e protocolos; *Views* são geralmente auto contidas e altamente reusáveis; e objetos do *Model* são independentes de apresentação e evitam qualquer dependência com o resto do programa. É responsabilidade da camada *Controller* combinar as outras duas camadas em um sistema funcional.

Model: Responsável pela lógica de negócio e dados da aplicação.

View: Responsável pela representação gráfica mostrada na tela e pelas interações do usuário.

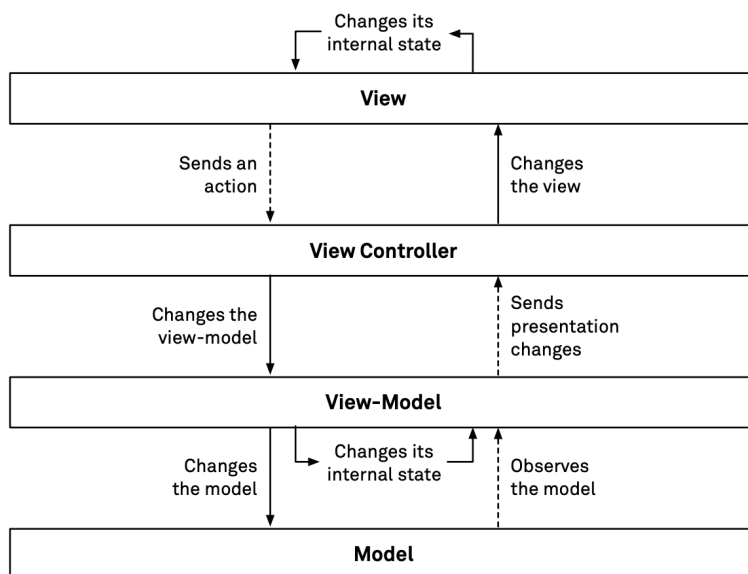
Controller: Coordena ações da *View*, aplicando ao *Model*, e altera visualizações na *View* de acordo com mudanças no *Model*.

No desenvolvimento *iOS*, a utilização de *MVC* fica um pouco mais complicada, pois a *Controller* acaba recebendo também parte da responsabilidade de controle do ciclo de vida das *Views*. Por isso, se torna menos testável. Para isso, podemos buscar variações e melhorias na arquitetura, como mostra a próxima seção sobre *MVVM*.

2.4.2 Model-View-ViewModel

MVVM é uma variação do padrão de design *Presentation Model* (por *Martin Fowler*) criada pela *Microsoft* e anunciada em 2005 para ser incorporada no sistema gráfico do *.NET*. Com essa arquitetura, a *Controller* perde algumas responsabilidades, que são passadas para a *ViewModel*, como estado da *view* e regras de negócio. Como pode ser observado na Figura 2.3, essa camada também pode ser a responsável por realizar efeitos colaterais externos, como requisições *HTTP*, tornando a *Controller* testável e reutilizável, como descrito por Eidhof Florian Kugler (2018).

Figura 2.3 – Diagrama da arquitetura *MVVM*



Fonte: Eidhof Florian Kugler (2018)

MVVM é também baseada em princípios clássicos de *Orientação a Objetos*: Objetos gerenciam seus comportamentos e estado internamente e se comunicam através de interfaces e protocolos; *Views* são geralmente auto contidas e altamente reusáveis; e obje-

tos do *Model* são independentes de apresentação e evitam qualquer dependência com o resto do programa. É responsabilidade da camada *ViewController* receber ações da *View* e configura-la a pedidos. A *ViewModel* então ganha o papel de combinar as camadas *ViewController* e *Model* formando um sistema funcional.

Model: Responsável pela lógica de negócio e dados da aplicação.

View: Responsável pela representação gráfica mostrada na tela e pelas interações do usuário.

ViewController: Cordena ações da *View*, aplicando ao *Model*, e altera visualizações na *View* de acordo com mudanças no *Model*.

ViewModel: Aplica e observa mudanças ao *Model*, e quando necessário, manda alterações de visualizações para *ViewController*.

Dividindo as responsabilidades como mostrado acima, temos um padrão muito mais isolado e com melhores definições de comunicação se tornando muito mais fácil de testar e fazer suposições sobre o funcionamento do sistema.

2.5 Colorimetria

Cores foram sempre um grande fascínio para a humanidade e seus fenômenos sempre foram sujeitos de investigações ao decorrer dos séculos. Em 1666, Newton definiu os fundamentos da ciência das cores quando descobriu que a luz branca do sol era composta de uma mistura de todas as cores do espectro, e essa descoberta foi o começo natural para o estudo também das representações de cores.

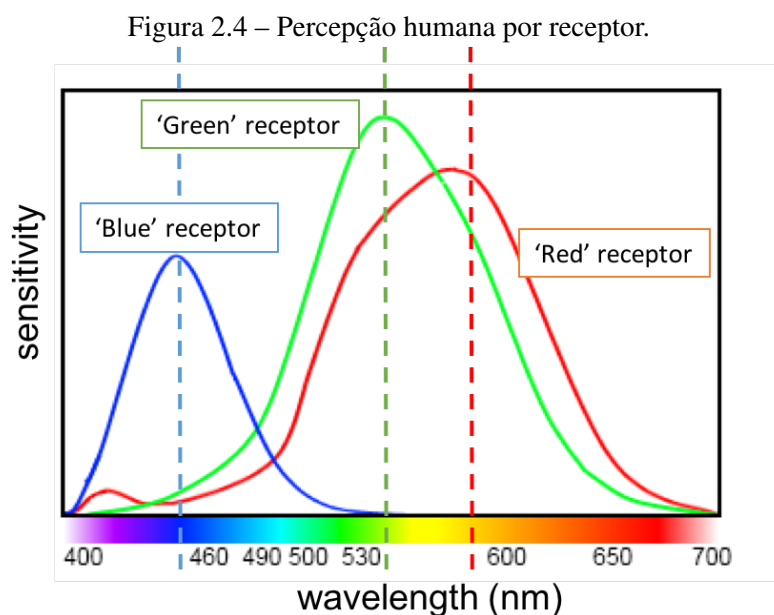
No livro *The Reproduction of Colour* (2005), HUNT dá o exemplo de que ao analisarmos as cores das folhas e frutos de um tomateiro iluminado pelo sol, percebemos que as folhas são verdes e os frutos vermelhos, pois ao receberem a luz do sol as superfícies absorvem alguns comprimentos de ondas e refletem o restante, no caso das folhas os comprimentos de onda refletidos são o que conhecemos por verde e no caso do fruto, vermelho.

Além de analisar os fenômenos naturais das cores, para realizarmos esse trabalho, precisamos compreender como uma cor é representada digitalmente, o que isso significa, e como podemos comparar diferentes cores e extrair informações úteis e corretas sobre elas.

2.5.1 Percepção

Nossa percepção de cores começa com os 3 receptores localizados nos nossos olhos, cada um sensível a uma faixa de comprimento de onda. O resultado combinado desses 3 receptores é interpretado pelo nosso cérebro e nos dá a possibilidade de discernir cores, diferenciando sua cromaticidade.

Para reproduzirmos cores digitalmente basta então criarmos maneiras de excitar esses 3 receptores de maneira controlada. Foi descoberto que cada um desses receptores tem foco em uma faixa de comprimento de onda, e suas correspondentes cores foram chamadas de Cores Primárias, vermelho, verde e azul, mostrado graficamente na Figura 2.4. Utilizando mecanismos similares, podemos replicar essas cores digitalmente, misturando diferentes proporções das cores primárias. Assim, a proporção de combinação dessas 3 cores primárias nos dá uma representação em 3 dimensões. (JOBLOVE; GREENBERG, 1978)



Fonte: Spectral Ray Tracing (<https://ceciliavision.github.io/graphics/a6/>)

Existem diversas formas de se especificar uma cor, e cada um desses modelos tenta representar de uma forma diferente esse fenômeno físico através de um modelo matemático. Cada um desses modelos de cor consiste em parâmetros (geralmente entre 3 e 4, segundo Ibraheem et al. (2012)) representando os componentes das cores. Podemos pensar que esses parâmetros criam um espaço dimensional conhecido como espaço da cor.

2.5.2 Representações

É importante ressaltar a diferença entre "Espaço de cores" e "Modelo de cores". Um espaço de cores é uma organização específica de cores que pode ser arbitrária, como por exemplo a associação de cores físicas em uma paleta de cores, ou seguir um modelo matemático. O espaço de cores é útil para entender a capacidade de representação de um dispositivo ou até mesmo de um arquivo digital. Espaços de cores podem nos mostrar se é possível reproduzir uma cor de um dispositivo em outro, se é possível reter detalhes de uma imagem e quanto será comprometido em uma conversão.

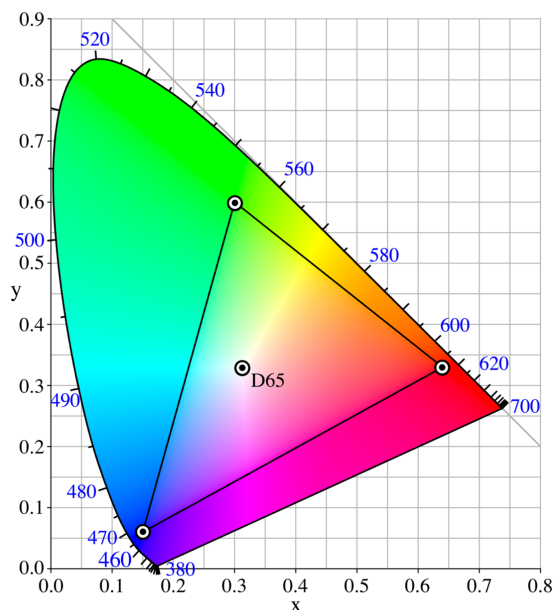
Já um modelo de cores é um modelo matemático abstrato que descreve como cores podem ser representadas através de tuplas de números. Sem associação à um espaço de cores, um modelo de cores é arbitrário e sem conexão com nenhum outro entendimento ou interpretação de cor conhecido. Por isso, é adicionada uma função de mapeamento entre um modelo de cor e um espaço de cor estabelecendo um escopo para o espaço de cor, conhecido como *gamut*. Os espaços de cores *Adobe RGB* e *sRGB* são totalmente diferentes embora sejam os dois baseados no mesmo modelo de cores, o *RGB*.

As maiores referências de espaço de cores, usadas como padrões, são o *CIELAB* e *CIEXYZ*, que foram criadas com intuito de abranger todas as cores que uma pessoa alterações na visão consegue enxergar. Geralmente, é interessante definir espaços de cores diferentes, com sistemas de coordenadas diferentes, para atingir diferentes objetivos de representação, manipulação ou entendimento, buscando uma aproximação da representação com o propósito de uso. A seguir, apresentamos 2 espaços de cores, o *sRGB* e o *CIELAB*. (IBRAHEEM et al., 2012)

2.5.2.1 sRGB - Red-Green-Blue

O espaço *sRGB* foi criado pela HP e Microsoft em 1996 para padronizar cores usando o modelo *RGB*. A ideia geral do *RGB* é que podemos gerar qualquer cor usando 3 cores primárias, ou seja, utilizando uma abordagem aditiva através da superposição das cores vermelho, verde e azul. A adição do vermelho ao verde nos dá o amarelo, por exemplo. A representação do *gamut* de *sRGB* pode ser visto na Figura 2.5.

Figura 2.5 – Representação do gamut de sRGB comparado em CIE1931.



Fonte: sRGB CIExy1931 (https://commons.wikimedia.org/wiki/File:CIExy1931_sRGB.png)

A escolha das cores primárias é relacionada a fisiologia do olho humano. Essas primárias estimulam uma variação máxima nos diferentes cones da nossa retina e criam o maior triângulo possível na representação. Esses espaços aditivos são muito usados em monitores, e dispositivos eletrônicos em geral, e se tornaram os espaços mais encontrados hoje.

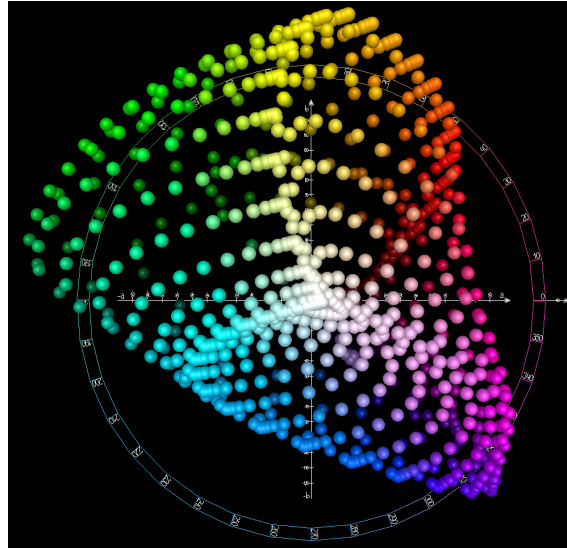
2.5.2.2 CIELAB $L^*a^*b^*$

O espaço de cor *CIELAB*, também conhecido como $L^*a^*b^*$, foi definido pela Comissão Internacional de Iluminação (*CIE*) em 1976 com a intenção de estabelecer um espaço uniforme onde uma mudança numérica deveria corresponder a uma mudança perceptível por humanos de mesma magnitude. É expressada por 3 valores: L^* indicando a percepção de luminosidade, e a^* e b^* indicando as 4 cores da visão humana: vermelho, verde, azul, e amarelo. Esse espaço deriva de outro conhecido como *CIEXYZ*, que é baseado no modelo *Standard Observer*.

O espaço *CIELAB* é tri-dimensional e cobre inteiramente a faixa de cores perceptíveis pelos humanos (*gamut*). O conceito de pares opostos é usado na definição de a^* e b^* , onde a^* corresponde ao par verde-vermelho e b^* representa o par azul-amarelo, os valores variam sem limite, negativos correspondem ao primeiro elemento do par e positivos ao segundo. Para cobrir a faixa visível variam geralmente até 150. O espaço *CIELAB*

pode ser visto na Figura 2.6.

Figura 2.6 – Representação do espaço CIELAB vista de cima



Fonte: CIELAB Space (https://commons.wikimedia.org/wiki/File:CIELAB_color_space_top_view.png)

2.5.3 Comparações

Comparação entre cores, ou distância entre cores, é um valor que indica a proximidade de duas cores. Essa medida possibilita colocar em valor a percepção que nós humanos temos quando vemos duas cores. Porém, nem sempre uma comparação reflete a percepção humana, que é muito complexa e não linear. Os espaços e modelos de cores tentam sempre se aproximar da percepção humana ou resolver problemas específicos, a mesma coisa acontece com os padrões para comparações entre cores. Diversos padrões vem sendo propostos e estudados para que se tenha comparações que se aproximam da nossa compreensão.

2.5.3.1 Distância Euclidiana

O jeito mais direto e simples de calcularmos a diferença entre cores é usar a proximidade dos pontos no seu espaço. Isso pode ser feito de maneira trivial usando a distância euclidiana. Ao compararmos duas cores *RGB* utilizamos cada valor como uma coordenada coordenada num espaço 3D, assim, para calcular a distância euclidiana temos:

$$\text{distance} = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

Embora não seja o método de cálculo de distância de cores mais próximo da percepção humana, a distância euclidiana é muito utilizada em diversas aplicações por sua facilidade de entendimento e baixo custo operacional para seu cálculo. (HUNT, 2005)

2.5.3.2 CIELAB ΔE^*

Para medir distância entre cores representadas em $L^*a^*b^*$ o CIE desenvolveu a medida ΔE^* que denota a diferença (delta) da sensação (E - *Empfindung*) das cores. A primeira fórmula foi concebida em 1976, porém ao longo dos anos foram desenvolvidas melhorias nas fórmulas devido a não-uniformidade do espaço de cor (CIELAB), ficaram conhecidas como fórmulas de 1994 and 2000. As não-uniformidades são importantes porque a visão humana é mais sensível a algumas cores do que outras.

2.5.3.2.1 CIE76 Foi a primeira fórmula para calcular diferença de cor a partir de um conjunto de coordenadas $L^*a^*b^*$ e como podemos ver, é uma também uma distância euclidiana, porém em outras coordenadas. (HUNT, 2005) Sua definição segue abaixo:

A partir de 2 cores (L_1^*, a_1^*, b_1^*) e (L_2^*, a_2^*, b_2^*) :

$$\Delta E_{ab}^* = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2}.$$

2.5.3.2.2 CIE94 Em 1994 a fórmula original foi estendida para levar em conta não-uniformidades do espaço de cor com a introdução de pesos específicos derivados de testes com casos reais de uso no setor automotivo. ΔE é definida então no espaço de cor $L^*C^*h^*$ com diferenças na luminosidade, cromaticidade e *hue*, calculadas a partir de coordenadas $L^*a^*b^*$.

Como visto em *The Reproduction of Color* (HUNT, 2005), a partir de 2 cores (L_1^*, a_1^*, b_1^*) e (L_2^*, a_2^*, b_2^*) temos as seguintes fórmulas para transformação:

$$\Delta E_{94}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2}$$

onde:

$$\begin{aligned}
\Delta L^* &= L_1^* - L_2^* \\
C_1^* &= \sqrt{a_1^{*2} + b_1^{*2}} \\
C_2^* &= \sqrt{a_2^{*2} + b_2^{*2}} \\
\Delta C_{ab}^* &= C_1^* - C_2^* \\
\Delta H_{ab}^* &= \sqrt{\Delta E_{ab}^{*2} - \Delta L^{*2} - \Delta C_{ab}^{*2}} = \sqrt{\Delta a^{*2} + \Delta b^{*2} - \Delta C_{ab}^{*2}} \\
\Delta a^* &= a_1^* - a_2^* \\
\Delta b^* &= b_1^* - b_2^* \\
S_L &= 1 \\
S_C &= 1 + K_1 C_1^* \\
S_H &= 1 + K_2 C_1^*
\end{aligned}$$

kC e kH são pesos das unidades e kL , $K1$ e $K3$ constantes dependendo da aplicação.

graphic	arts	textiles
k_L	1	2
K_1	0.045	0.048
K_2	0.015	0.014

2.5.3.2.3 **CIEDE2000** No ano 2000 mais melhorias na definição foram feitas para que se adequasse a percepção humana, foram feitas 5 mudanças no modelo anterior:

- Adicionado um termo de rotação (RT) de *hue* (matiz) para lidar com o problema na região azul.
- Compensação para cores neutras.
- Compensação para luminosidade. (SL)
- Compensação para cromaticidade. (SC)
- Compensação para matiz (*hue*). (SH)

Esse novo método é muito mais preciso e semelhante às percepções humanas do que os anteriores, porém se tornou mais custoso no quesito processamento. Também contém fórmulas mais complexas e com mais passos para o cálculo, tendo uma implementação mais trabalhosa do que as anteriores. Por esses motivos, as fórmulas desse método não serão dispostas nesse trabalho. (HUNT, 2005)

3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados quatro jogos que compartilham características com esse projeto, seja por possuírem uma dinâmica inovadora, foco em público infantil ou abordagens educacionais. O primeiro é o *Peek-a-Zoo*, um jogo infantil que ensina atividades do cotidiano através de personagens representando animais. O segundo é o *Emoji Scavenger Hunt* um jogo que estimula o jogador a procurar objetos no mundo real que correspondem aos *Emoji* selecionados. O terceiro é o *ARrrrrgh*, um jogo divertido de caça ao tesouro em realidade aumentada. O quarto é o mais famoso deles, *Pokemon Go*, uma versão da clássica franquia em realidade aumentada. Por fim, são realizadas avaliações e comparações entre esses jogos e comentadas as características que possuem em comum com esse projeto.

3.1 Peek-a-Zoo - Animal Games

Através de uma variedade de atividades divertidas, crianças aprendem a identificar animais, emoções, ações, sons e muito mais. *Peek-a-Zoo* é um jogo que ganhou prêmios por seu foco em crianças de 2-5 anos e foi adquirido pela *Khan Academy*, empresa reconhecida globalmente pelos esforços em disseminar conhecimento.

Segundo Moose (2017), as principais atividades do jogo são: Identificar animais (como jacarés, gatos e galinhas), emoções (como felicidade, surpresa e tristeza), ações (como piscando os olhos, abanando e bocejando), posições (como de lado, de costas e de cabeça para baixo) e sons dos animais. A Figura 3.1 mostra a estética e os desafios do jogo.

Figura 3.1 – Estética e desafios do jogo *Peek-a-Zoo*.

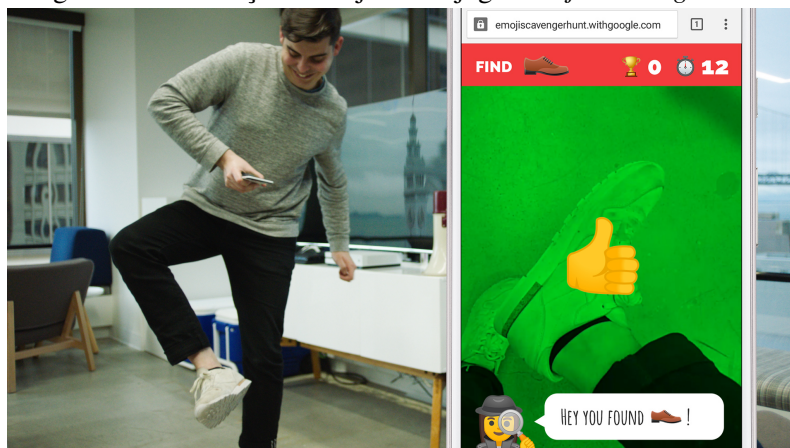


Fonte: Moose (2017)

Esse jogo foi escolhido para compor a lista de trabalhos relacionados por compartilhar de uma característica essencial deste projeto: Entreter crianças enquanto incentivam o aprendizado.

3.2 Emoji Scavenger Hunt - Google

Desenvolvido pelo *Google* como prova de conceito para testar demonstrar novas tecnologias nesse jogo um *emoji* é sorteado de forma aleatória e o jogador deve procurar em tempo real um objeto ao seu redor cujo *emoji* representa. Por exemplo, ao sortear o *emoji* de um carro, deve apontar o celular para um carro ou carro de brinquedo para completar o desafio. *Emoji Scavenger Hunt* utiliza a câmera do celular e algoritmos de *Machine Learning* para adivinhar o que está enxergando. (GOOGLE, 2019)

Figura 3.2 – Detecção de objetos do jogo *Emoji Scavenger Hunt*

Fonte: Google (2019)

Esse jogo foi escolhido para compor a lista de trabalhos relacionados por utilizar uma interação interessante e diferenciada: Utilizar a câmera para reconhecer objetos do mundo real. A Figura 3.2 mostra um usuário interagindo com o jogo e a interface correspondente.

3.3 ARrrrrgh - A magical treasure adventure

ARrrrrgh é um jogo que traz a brincadeira clássica de esconde-esconde para os tempos modernos. Dois jogadores se transformam em piratas onde o primeiro é responsável por esconder um tesouro virtual, utilizando realidade aumentada, em algum lugar do ambiente em que se encontra sem que o segundo jogador veja. Após isso, o segundo jogador tem a tarefa de encontrar o tesouro com a ajuda de um personagem que dá dicas de localização. O jogador percorre o ambiente tentando escavar virtualmente o chão na busca do tesouro. (MEDIA, 2017)

Figura 3.3 – Integração com a realidade do jogo *ARrrrrgh*

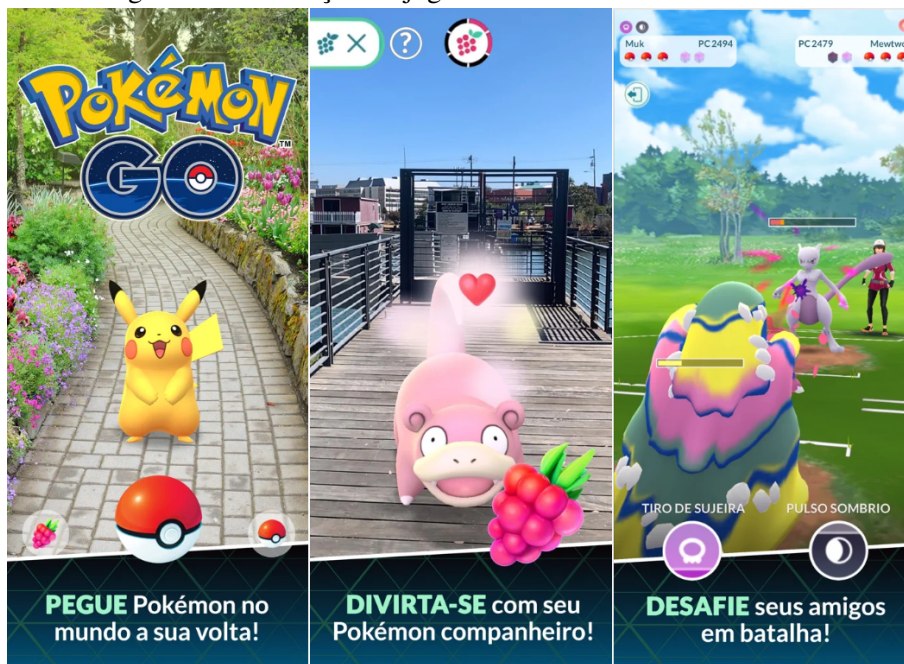
Fonte: Media (2017)

Esse jogo foi escolhido para compor a lista de trabalhos relacionados por também utilizar uma interação muito interessante e combina-la com elementos lúdicos e histórias para envolver crianças. A Figura 3.3 mostra 3 cenas do jogo que mostram a integração dos elementos no mundo real.

3.4 Pokemon GO

Um dos jogos com maior número de *downloads* e mais rentáveis de 2016, *Pokemon GO* foi o primeiro jogo (ou aplicativo) que realmente popularizou a tecnologia baseada em localização e realidade aumentada. No jogo, você é um treinador de *Pokemons* e usa o dispositivo móvel para capturar, treinar e lutar contra criaturas virtuais, projetadas com realidade aumentada em algum local do mundo real. O jogo chegou a ter 147 milhões de usuários ativos mensais. (NIANTIC, 2016)

Figura 3.4 – Interação do jogo *Pokemon GO* com a realidade.



Fonte: Niantic (2016)

Esse jogo foi escolhido para compor a lista de trabalhos relacionados por ser um marco nos jogos com interações que aumentam a realidade do usuário. A Figura 3.4 mostra 3 cenas que demonstram a interface do jogo e sua integração com o mundo real.

3.5 Análise comparativa

Analisando os jogos propostos anteriormente, percebe-se características semelhantes que podem ser comparadas diretamente, mas também encontramos muitas maneiras diferentes de alcançar objetivos similares, por isso, ao comparar esses jogos temos que usar também uma análise mais holística, analisando o produto final e a experiência do usuário. As funcionalidades específicas podem ser visualizadas e comparadas na Tabela 3.1.

Ao comparar a estética e a apropriação para o público infantil de 2-5 anos, vemos que *Peek-a-Zoo* tem destaque, oferecendo uma interface muito simples e amigável, com animais ilustrados também de maneira simples e oferecendo muita interações ao toque, além de áudio, vozes e animações. Já *ARrrrrgh*, entrega uma experiência interessante porém um pouco mais madura, provavelmente mais adequada para crianças acima dos 5 anos. *Pokemon Go* por sua vez pode ser considerado um jogo com estética e funcionalidades mais voltadas para um público do final da categoria infantil para cima.

Ao considerarmos o uso de tecnologias para criar uma mecânica de interação inovadora e envolvente, *Emoji Scavenger Hunt*, *ARrrrrgh* e *Pokemon Go* se destacam, aumentando a realidade do jogador e a integrando na história do jogo. Porém, como descrito anteriormente, entre esses três, somente *ARrrrrgh* entrega uma experiência apropriada para crianças pequenas, onde a realidade é expandida através recursos lúdicos que encantam o jogador.

Outro ponto analisado é a disponibilidade dos aplicativos na língua portuguesa, dos propostos somente *Pokemon Go* tem tradução para o português, porém o jogo contém somente conteúdo escrito traduzido, não possui interações sonoras e vozes.

Dos jogos propostos, somente *Peek-a-Zoo* explora o quesito educação ou estimulação do aprendizado infantil, criando pequenos desafios para serem resolvidos pelas crianças.

Tabela 3.1 – Comparação de funcionalidades

Característica	Peel-a-Zoo	Scavenger Hunt	ARrrrrgh	Pokemon GO
App Mobile	Sim	Não	Sim	Sim
Interação com Realidade	Não	Sim	Sim	Sim
Temática Infantil	Sim	Não	Sim	Não
Estimula Aprendizado	Sim	Não	Não	Não
Interações em Português	Não	Não	Não	Parcial

Fonte: Autor

Neste trabalho, é explorado como proposta esses três principais pilares analisados: interações que aumentam a realidade, estética e funcionamento lúdicos para primeira infância e conteúdos que estimularam o aprendizado utilizando diálogos e desafios em português.

4 DESENVOLVIMENTO

Neste capítulo, é descrito todo o processo de desenvolvimento do projeto REcolorindo o Reino, começando pela idealização e gerenciamento, seguido pelo *Game Design Document*, implementação e finalizando com os resultados finais.

4.1 Idealização

A metodologia escolhida na idealização deste projeto foi a *Challenge Based Learning* (2.1), pois acredita-se oferecer mecanismos e processos que incentivam a criatividade e pensamento não convencionais, possibilitando a criação de propostas inovadoras para ajudar na solução de problemas reais do nosso mundo.

Para começar, foi escolhida uma *Big Idea*, um tema geral que abrange a área a ser trabalhada, como visto em na seção 2.1.

Big Idea: Educação/Desenvolvimento Infantil

Com a definição da área pronta, vários questionamentos foram feitos, e o principal escolhido para aprofundamento foi:

Essential Question: Como levar a diversão dos jogos para fora dos dispositivos, proporcionando um melhor desenvolvimento motor e espacial para as crianças?

Através dessa pergunta, foi construído um desafio, que é o objetivo de resolução desse projeto.

Challenge: Desenvolver um jogo que estimule a interação de crianças com o ambiente a sua volta.

A partir do desafio escolhido, foram criadas diversas possíveis tentativas de chegar numa solução. A solução escolhida para tentar mitigar esse desafio foi:

Solution: Um jogo que utiliza um sistema de detecção de cores através da câmera do dispositivo para tornar possível a interação entre a história, a criança, o mundo real e os desafios de encontrar as cores de um reino descolorido.

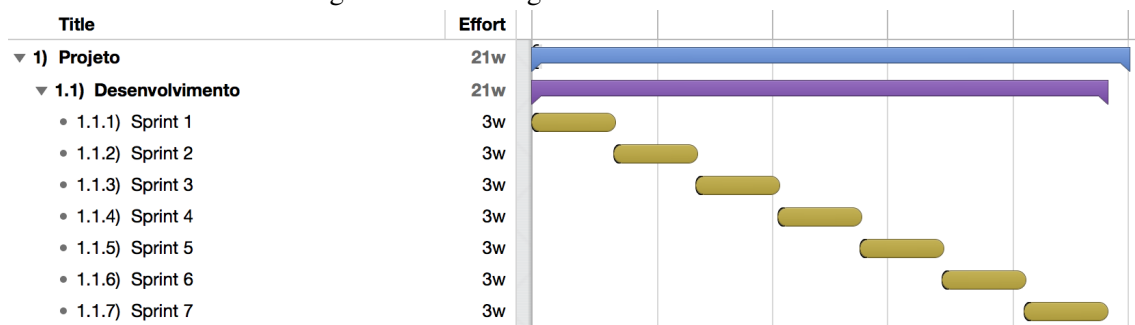
4.2 Gerenciamento

Para gerenciamento desse projeto foram utilizadas algumas metodologias ágeis, pois manter uma organização das tarefas, análise de progresso e cumprimento dos prazos eram de alta importância, além de propiciar ao projeto um ambiente dinâmico que propiciasse experimentação.

4.2.1 Cronograma

Para o desenvolvimento do REcolorindo o Reino foram definidas 7 *Sprints* de 3 semanas, pois acreditou-se que 3 semanas seria um bom período para finalizar a quantidade de tarefas propostas para cada iteração, levando em conta a quantidade de horas disponíveis. O cronograma total de desenvolvimento previsto foi de 21 semanas, como pode ser visto na figura na Figura 4.1 :

Figura 4.1 – Cronograma de desenvolvimento.



Fonte: Autor

Dividimos os requisitos do projeto em 7 etapas (*Sprints*), assim conseguimos manter uma análise do progresso e corrigir qualquer problema para nos mantermos no cronograma. A divisão em alto nível das tarefas pode ser vista nas Tabelas 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 e 4.7, cada uma relativa a uma das sprints.

4.2.1.1 Sprint 1

Tabela 4.1 – Metas e Principais Tarefas da Sprint 1

Metas	Principais Tarefas
<ul style="list-style-type: none"> - Design e identidade visual inicial. - Estrutura do projeto de software. - Protótipos de detecção de cores. 	<p>Programador: Criação da estruturação de código e relação de classes e cenas.</p> <p>Programador: Experimentos com os algoritmos de reconhecimento de cores.</p> <p>Ilustrador: Criação dos personagens principais e <i>NPCs</i>.</p> <p>Designer: Interface da cena de desafio.</p>

Fonte: Autor

4.2.1.2 Sprint 2

Tabela 4.2 – Metas e Principais Tarefas da Sprint 2

Metas	Principais Tarefas
<ul style="list-style-type: none"> - Cena de jogo parcialmente implementada (sendo possível colorir os objetos). - Design final dos elementos e personagens do jogo. 	<p>Programador: Codificação da cena de desafio a partir dos conceitos feitos pelo designer na etapa anterior.</p> <p>Programador: Codificação da interface da cena de desafio a partir dos conceitos feitos pelo designer na etapa anterior.</p> <p>Ilustrador: Criação dos elementos de desafio.</p> <p>Designer: Criação da interface das cenas (menus, seletor de nível, etc.).</p>

Fonte: Autor

4.2.1.3 *Sprint 3*

Tabela 4.3 – Metas e Principais Tarefas da Sprint 3

Metas	Principais Tarefas
<ul style="list-style-type: none"> - Ilustrações do jogo prontas aguardando finalização. - Interface do jogo e menus funcionais. 	<p>Programador: Codificação de navegação e interface.</p> <p>Programador: Codificação dos personagens e suas interações.</p> <p>Designer: Finalização digital dos personagens.</p> <p>Animador: Animação dos personagens da história inicial e final.</p>

Fonte: Autor

4.2.1.4 *Sprint 4*

Tabela 4.4 – Metas e Principais Tarefas da Sprint 4

Metas	Principais Tarefas
<ul style="list-style-type: none"> - Navegação do jogo entre cenas e menus funcional. - Mecânica do seletor de regiões e de níveis. - Versão Alpha: Versão parcial mas funcional do jogo, deve ser testado a estrutura e principalmente a cena de desafio. 	<p>Programador: Codificação das cenas de seleção de nível.</p> <p>Programador: Implementação das <i>cutscenes</i>.</p> <p>Designer: Finalização dos elementos.</p> <p>Animador: <i>Cutscenes</i> da história.</p> <p>Áudio: Produção da narração dos personagens e história.</p>

Fonte: Autor

4.2.1.5 Sprint 5

Tabela 4.5 – Metas e Principais Tarefas da Sprint 5

Metas	Principais Tarefas
<ul style="list-style-type: none"> - Todas as mecânicas de jogo, navegações da interface e arte implementadas. - Versão Beta: Versão com todas <i>features</i> funcionais do jogo, deve ser testado geral, embora erros são esperados. 	<p>Programador: Codificação de gerenciador de controle de dados do jogador.</p> <p>Programador: Implementação dos efeitos visuais.</p> <p>Programador: Integração da narração aos personagens.</p> <p>Designer: Finalização de <i>cutscenes</i>.</p> <p>Animador: Efeitos visuais da cena de desafio.</p> <p>Áudio: Trilha sonora das cenas.</p>

Fonte: Autor

4.2.1.6 Sprint 6

Tabela 4.6 – Metas e Principais Tarefas da Sprint 6

Metas	Principais Tarefas
<ul style="list-style-type: none"> - Design final completamente integrado ao jogo. - Animações e áudio final integrados. - Versão Release Candidate: Versão candidata a lançamento. Tudo está pronto porém mais testes devem ser realizados para comprovar a qualidade do produto. 	<p>Programador: Calibrar implementação de reconhecimento de cores.</p> <p>Programador: Integrar trilha sonora ao jogo.</p> <p>Programador: Implementar efeitos visuais nas cenas.</p> <p>Designer: Melhorias gerais.</p> <p>Ilustrador: Melhorias gerais.</p> <p>Animador: Melhorias gerais.</p> <p>Áudio: Efeitos sonoros gerais.</p>

Fonte: Autor

4.2.1.7 Sprint 7

Tabela 4.7 – Metas e Principais Tarefas da Sprint 7

Metas	Principais Tarefas
- Testes e calibrações. - Versão Stable Release : O produto final, testado e pronto para o lançamento.	Testes. Correções. Ajustes. Acabamento.

Fonte: Autor

4.3 Game Design Document

Nesta seção, é detalhado o documento de *Design* do jogo REcolorindo o Reino. É descrito seu enredo, *Gameplay*, composição do jogo, entre outros detalhes de cenas e demais conteúdo. O *Game Design* é a parte mais importante de um jogo, pois documenta todas as decisões feitas na sua criação. Isso é insumo direto para a fase de implementação, facilitando a comunicação entre os envolvidos e a oficialização das decisões.

4.3.1 Enredo

O enredo do jogo REcolorindo o Reino se divide em 3 partes principais:

- Cena de Introdução
- Jogo (Cena do Mapa do Reino, Cenas de Região e Cenas de Desafio)
- Cena de Encerramento

Na Cena de Introdução temos a história inicial, na qual o reino e os personagens são apresentados ao jogador. Ao final da introdução, temos um conflito na história e surge um problema a ser resolvido. Com essa quebra da estabilidade, cabe aos personagens e ao jogador solucionarem o desafio proposto, recolorir o reino.

Assim, é introduzida a segunda etapa, o jogo propriamente dito (Cena de Jogo), que evolui em um mapa do reino (Cena do Mapa do Reino), onde cada região (Cena de Região) deve ser completada. Ao finalizar todas as regiões disponíveis na Cena de Mapa do Reino, inicia a Cena de Encerramento e o desfecho da história.

4.3.1.1 *Personagens principais*

Lindamaria: (protagonista) Irmã de Léopetro e filha do Corão. Juntamente com seu irmão tem como missão recolorir o reino e reunir todas as partes do medalhão colorido.

Léopetro: (protagonista) Filho do Corão e irmão de Lindamaria. Juntamente com ela tem como missão recolorir o reino e reunir todas as partes do medalhão colorido.

Magotávio: (antagonista) Ao ser libertado da maldição que lhe aprisionava na caverna Magotávio sobrevoa o reino e o descolore se vingando pelos seus anos presos na caverna.

4.3.1.2 *História:*

Existia um reino fascinante nas montanhas ao sul da América do Sul, lá as cores eram tão vibrantes que transformavam tudo em fantasia. Era um reino alegre e colorido onde até o sol brilhava com mais intensidade. Todos os dias a princesa Lindamaria e o príncipe Léopetro corriam e brincavam, desfrutando do bom dia que se apresentava.

Um dia, distraídos pelas lindas cores do arco-íris, chegaram até a caverna proibida, curiosos entraram, e sem querer quebraram o medalhão mágico que aprisionava o malvado Magotávio. Para castigá-los, Magotávio sobrevoou o reino e descoloriu tudo o que encontrava pela frente, deixando o tudo sem graça, sem cor e sem vida.

Sem saber o que fazer, Lindamaria e Léopetro correram para pedir a ajuda a seu pai, o Corão que lhes disse: É preciso encontrar amigos no mundo real para nos ajudar a recolorir o nosso reino e consertar o medalhão.

Nesse momento, o jogador entra como protagonista da história. Ele deve correr pela sua casa, buscar em seu jardim, em seus brinquedos e no seu dia a dia as cores verdadeiras do reino para o salvar.

A cada etapa concluída do jogo, uma parte do medalhão se reconstrói e, quando completo, voltará a adormecer o Magotávio, fazendo que todo o reino volte a viver colorido e feliz para sempre.

Assim o jogador e seus novos amigos, Lindamaria e Léopetro, poderão se movimentar e desfrutar deste lindo reino.

4.3.2 *Gameplay*

O desafio do jogo consiste em recolorir objetos descoloridos pelo antagonista da trama. Para isso, o jogador deve deslocar-se no ambiente físico ao seu redor e capturar cores de objetos reais que possuam a cor do objeto descolorido. Para realizar esta captura, a criança deve apontar a câmera do dispositivo a um objeto que tenha a cor desejada. Tecnicamente, esta cor é inferida a partir de um algoritmo que interpreta as imagens da câmera em tempo real, e aplica a cor ao objeto no jogo.

Esta jogabilidade diferenciada possui um grande potencial educacional, pois exercita diferentes funções psicomotoras da criança. O ato de apontar a câmera exercita a percepção espacial; identificar as cores exercita a percepção visual; a locomoção manejando o dispositivo e observando os objetos a sua volta exercitam a motricidade ampla e fina, como pode ser visto na Tabela 4.8.

Tabela 4.8 – Benefícios de ações do jogo

Função Psicomotora	Benefícios
Apontar/Mirar dispositivo para as cores	Percepção espacial
Identificar as cores no dispositivo e nos objetos ao redor	Percepção visual
Locomoção com o dispositivo em mãos	Motricidade ampla
Locomoção com o dispositivo, observação dos objetos ao redor e interação no jogo	Motricidade Fina

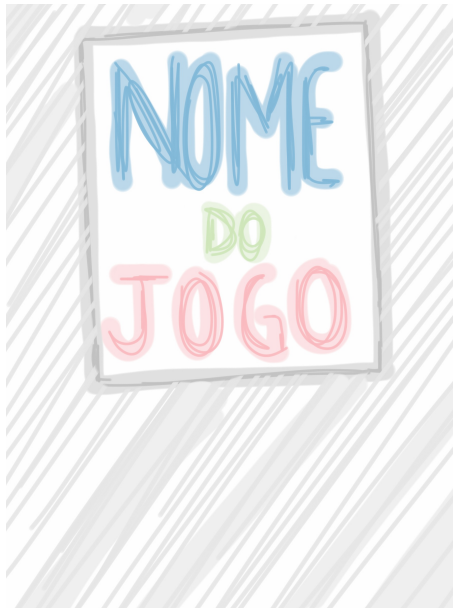
Fonte: Autor

4.3.3 **Composição do jogo**

A interface de usuário de REcolorindo o Reino é muito importante, pois é parte imprescindível na experiência do jogo. Ela é composta por diferentes cenas (correspondendo a diferentes páginas do aplicativo) que formam a estrutura de navegação. Os caminhos de navegação e suas condições de acesso estão detalhadas na seção Hierarquia e acesso das cenas (4.3.5.1). Abaixo estão brevemente descritos os principais tipos de cenas que compõem o jogo.

Cabe observar que, as imagens abaixo são meramente ilustrativas (wireframe). Têm o objetivo de transmitir a ideia de organização e dimensão dos elementos do jogo e não representam o produto final deste projeto. A Figura 4.2 mostra o protótipo da “Cena de Apresentação” do jogo.

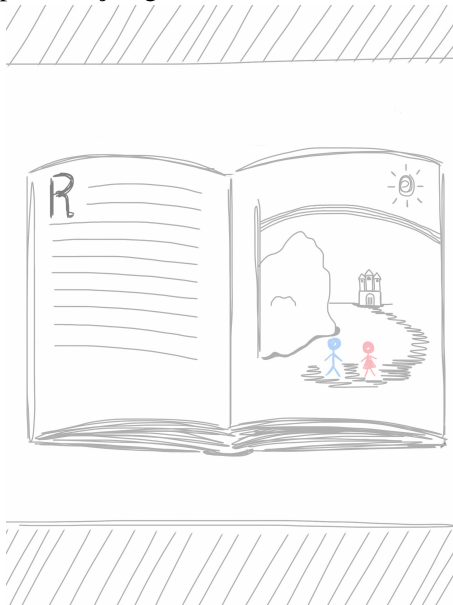
Figura 4.2 – Representação gráfica da Cena de Apresentação (wireframe).



4.3.3.1 Cena de Introdução

A ideia da cena introdutória é fazer com que a criança seja contagiada pela fantasia de um conto de fadas. A abordagem escolhida é a utilização de uma animação em vídeo que contará a história inicial do reino. A cena começa com a narração da história descrita anteriormente (4.3.1.2), enquanto uma animação de um livro com os personagens é apresentada. A Figura 4.3 mostra o protótipo da “Cena de Introdução” do jogo.

Figura 4.3 – Representação gráfica da Cena de Introdução (wireframe).

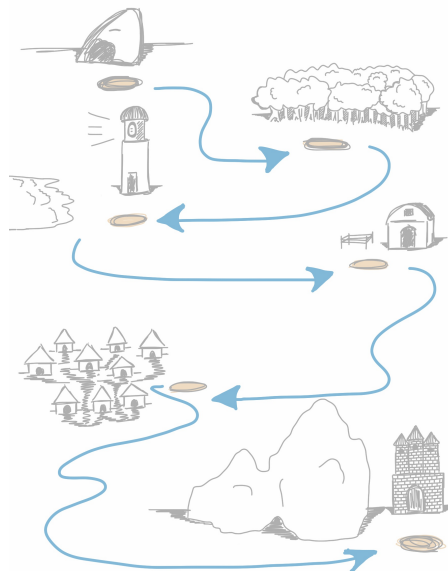


Fonte: Autor

4.3.3.2 Cena de Mapa do Reino

Após a cena introdutória, esta cena é apresentada, trata-se de um mapa geral do reino e nela é possível visualizar cinco atalhos que dão acesso as cinco Cenas de Região do jogo: o Bosque, a Fazenda, o Mercado, o Jardim e o Castelo. Se o jogo já foi usado anteriormente, a última região jogada é destacada no mapa, caso contrário a primeira é recomendada. Cabe ao jogador selecionar a fase. Este processo se repete toda vez que a Cena de Mapa do Reino é apresentada. Além das regiões, existem também os botões de navegação e o personagem protagonista escolhido estará presente para interagir com o jogador. A Figura 4.4 mostra o protótipo da “Cena do Mapa do Reino” do jogo.

Figura 4.4 – Representação gráfica da Cena do Mapa do Reino (wireframe).



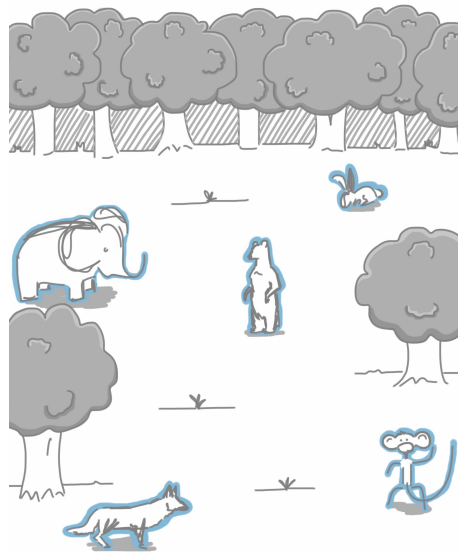
Fonte: Autor

4.3.3.3 Cena de Região

Existem cinco cenas deste tipo, o Bosque, a Fazenda, o Mercado, o Jardim e o Castelo. Estas cenas são acessadas através da Cena de Mapa do Reino. Como introduzido anteriormente, estas são compostas por objetos a serem recoloridos dispostas em um plano de fundo, um NPC e os botões de navegação. Inicialmente, os objetos atalhos para Cena de Desafio estão com um aspecto de descolorido e quando são completados ganham cor assim passando a ideia de item concluído.

Ao acessar este tipo cena, um NPC recebe o jogador e pede ajuda para recolorir os objetos em forma de texto e narração. Os objetos se destacam e cabe o jogador selecionar o desafio que deseja jogar. A Figura 4.5 mostra o protótipo da “Cena da Região” do jogo.

Figura 4.5 – Representação gráfica da Cena da Região: O Bosque (wireframe).



Fonte: Autor

Os itens e *NPCs* são específicos para cada região e estão listados na seção Lista dos Níveis (4.3.4.2).

4.3.3.4 Cena de Desafio

Principal cena onde se caracteriza o *gameplay* e o nível design do jogo. É composta principalmente pelo objeto a ser recolorido e um *NPC* específico que estará presente para interagir e prestar ajuda ao jogador. Ao escolher um desafio (objeto a ser colorido), o objeto é disposto no centro da tela. Um *NPC* aparece na cena e interage com o jogador, explicando o desafio e dando dicas para completar. A Figura 4.6 mostra o protótipo da “Cena de Desafio” do jogo.

Figura 4.6 – Representação gráfica da Cena de Desafio (wireframe).



Fonte: Autor

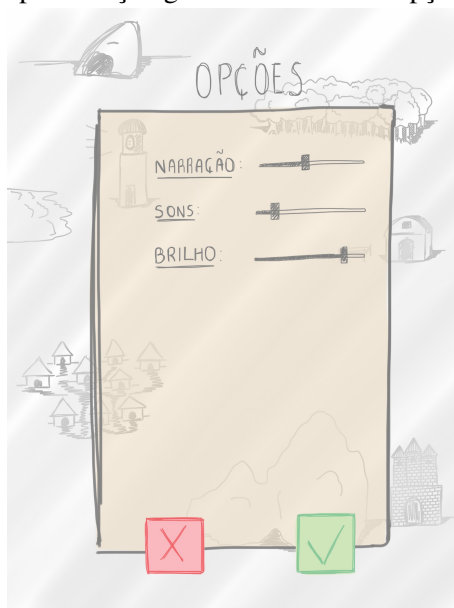
4.3.3.5 Cena de Encerramento

Cena apresentada quando o jogador concluí todos os desafios. Ela vem para finalizar a história iniciada por Lindamaria e Léopoldo e vivenciada pelo jogador. Com a ajuda do medalhão, o Magotávio é aprisionado novamente na caverna trazendo paz, alegria e cores de volta ao reino.

4.3.3.6 Cena de Opções

Tela apresentada ao entrar no menu de opções do jogo. A Figura 4.7 mostra o protótipo da “Cena de Opções” do jogo.

Figura 4.7 – Representação gráfica do Cena de Opções (wireframe).



Fonte: Autor

4.3.4 Funcionamento dos desafios

Como visto brevemente em Composição do jogo (4.3.3), o projeto é composto por diferentes cenas (Cena de Mapa do Reino, Cena de Região, Cena de Desafio, entre outras). A seguir, é descrito o funcionamento da principal cena do jogo, a Cena de Desafio, onde propriamente se caracteriza o *gameplay*.

Ao ser apresentada a Cena de Desafio ao jogador, um personagem introduzirá o desafio explicando para o jogador que ele deve procurar a cor do item central da tela. Então, a criança terá que circular nos ambientes próximos para encontrá-la e, dessa maneira, a criança é induzida a exercitar-se.

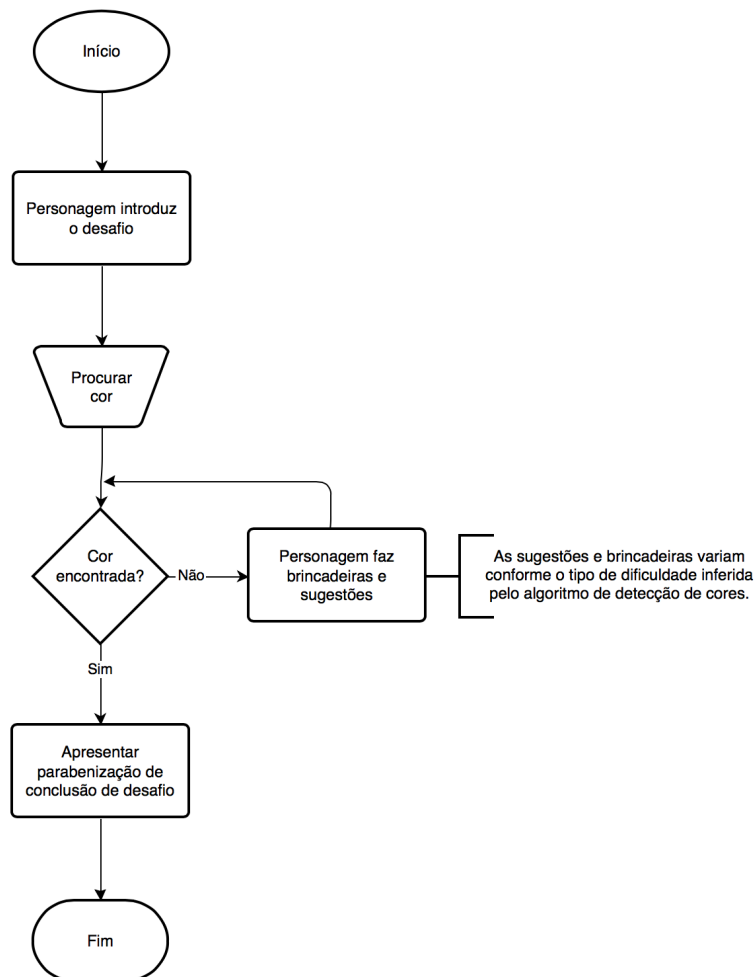
Ao encontrar um objeto no mundo real que julga ter a cor do item, o jogador deverá apontar a câmera para este artefato. Logo que o jogador permanece por um curto período de tempo apontando, a cor do objeto é inferida pelo sistema de detecção que também verifica se ela corresponde ao conjunto de cores as quais o objeto deve ser colorido, ou seja, se a cor foi encontrada. Caso a cor não pertença a esse conjunto, o personagem irá interagir com o jogador fazendo brincadeiras e dando sugestões para que o jogador continue a procurar em outros lugares.

Para evitar que o usuário fique preso no mesmo desafio, além da ajuda do NPC, ele terá a opção de pular de desafio. Já no caso de encontrar a cor correta, o jogador será recompensado com uma experiência audiovisual contagiante que o estimule a seguir

buscando mais desafios.

A Figura 4.8 ilustra o fluxograma para auxiliar no entendimento do fluxo da Cena de Desafio:

Figura 4.8 – Fluxograma da Cena de Desafio.



Fonte: Autor

4.3.4.1 Descrição da Cena de Desafio

Esta cena é mostrada toda vez que um objeto vai ser colorido. Detalharemos aqui essa cena de um modo geral, e fica subentendido que ela será personalizada de acordo com a região e ao desafio selecionado. Os principais elementos contidos na cena são descritos a seguir, e podem ser localizados na Figura 4.6.

Item à colorir: Destacado no centro da cena, muda de cor conforme a cor interpretada pelo algoritmo de detecção de cores das imagens da câmera. Uma animação de resposta ao jogador é exibida após a cor correta ser detectada;

NPC: Logo abaixo do objeto principal, juntamente com a versão escrita da mensagem narrada, estará o personagem selecionado pelo jogador. Ele é responsável por guiar o jogo, dando dicas e avisos conforme visto no Funcionamento do desafio (4.3.4);

Botão voltar: Localizado no canto superior esquerdo da cena. Volta para a Cena de Região, ou seja, por onde os desafios são acessados;

Circulo Mágico: Localizado no canto superior direito da cena. Circulo colorido com a cor que deve ser encontrada pelo jogador.

4.3.4.2 Lista dos níveis

A temática dos níveis é dada pelo conteúdo da região na qual está contido e é representado por um item a ser recolorido. O jogo é composto por cerca de 25 níveis distribuídos em 5 regiões. Na Tabela 4.9 são listados as diferentes regiões e seu conteúdo:

Tabela 4.9 – Objetos de cada nível e região

Região	NPC	Objetos
O Bosque	Índia	Mico, Garça, Tamanduá, Arara e Jacaré.
A Fazenda	Fazendeiro	Porco, Abobora, Cachorro, Melancia e Pintinho.
A Feira	Vendedora	Banana, Cenoura, Limão, Maçã e Uva
O Jardim	Jardineira	Girassol, Joanelha, Regador, Borboleta e Mico
O Castelo	Rei Corão	Cavalo, Piano, Gato, Telescópio e Barco

Fonte: Autor

4.3.5 Estrutura e Progressão

A progressão do jogo é direta, dividida em dois níveis: Região e Desafio. Existem cinco regiões e cada uma delas possui entre cinco e oito desafios, ou seja, objetos a serem recoloridos. Para completar o jogo, o jogador deve ter completado todos os desafios e reconstruído o medalhão com as suas cinco partes.

Ao completar um desafio, o objeto que o representa aparece colorido e com aparência de concluído na Cena de Região. Completados todos os desafios de uma região, o atalho que dá acesso a essa região na Cena de Mapa do Reino adquire um aspecto de concluído e o jogador ganha o pedaço de medalhão correspondente.

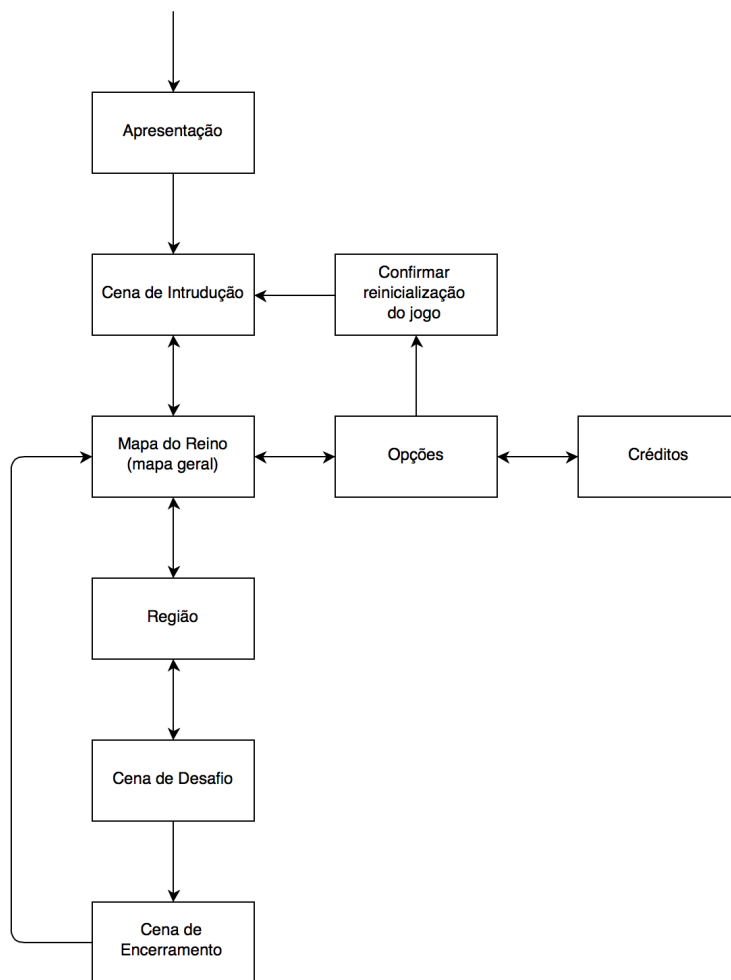
O jogador não será forçado a seguir uma ordem de desafios, ele poderá selecionar livremente qual desafio da Região deseja jogar. Entretanto, para liberar uma próxima

Região, o jogador deverá ter colorido todos os objetos da última Região disponível. No momento em que todas as Regiões do Mapa do Reino estiverem concluídas, ou seja, quando as cinco partes do medalhão estiverem juntas, é iniciada a Cena de Encerramento da história.

4.3.5.1 Hierarquia e acesso das cenas

O diagrama da Figura 4.9 ilustra os sentidos de navegação possíveis entre as cenas:

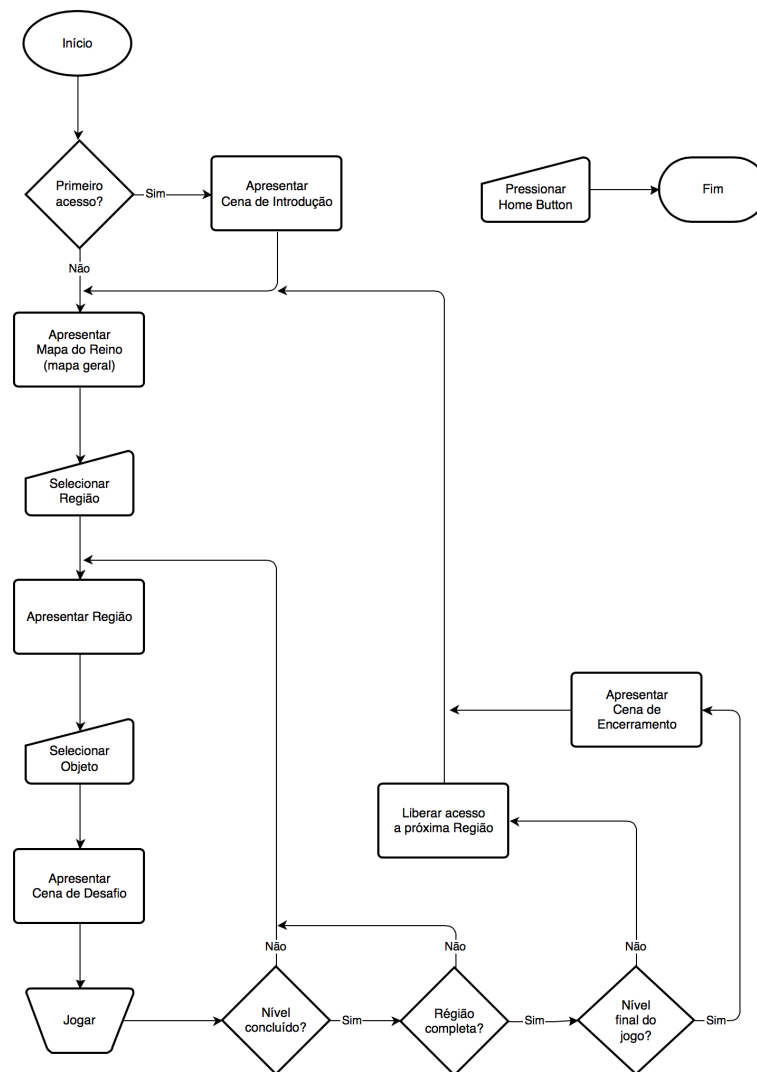
Figura 4.9 – Diagrama das portas de acesso entre as cenas.



Fonte: Autor

O fluxograma da Figura 4.10 demonstra as condições de acesso entre as diferentes cenas:

Figura 4.10 – Fluxograma de navegação.



Fonte: Autor

4.4 Implementação

Esta seção apresenta detalhes sobre a implementação do *Game Design Document* de REcolorindo o Reino. Começando com uma visão geral sobre o projeto e na sequência um detalhamento dos padrões de arquitetura, armazenamento, *frameworks* e componentes principais.

4.4.1 Visão geral

A primeira decisão de implementação do projeto foi a escolha da plataforma para que o jogo fosse desenvolvido. A plataforma (Sistema Operacional) escolhida foi a *iOS*, na versão 15 e os motivos principais para essa escolha foram a familiaridade do autor com a plataforma e a disponibilidade de dispositivos móveis para testes com os requisitos necessários. Já a versão mínima de suporte (15) foi escolhida pois oferece grandes avanços nas ferramentas de desenvolvimento, componentes de UI, linguagens suportadas e *frameworks* disponíveis nativamente.

Com a plataforma escolhida, podemos listar os dispositivos que rodam o sistema operacional e que possuem os requisitos mínimos de hardware para rodar o jogo. Como trata-se de um jogo que não possui uma demanda grande de processamento gráfico, os requisitos de hardware são apenas possuir uma câmera traseira no dispositivo, podendo ser jogado tanto em *Tablets* como em *Smartphones*. Assim, os dispositivos suportados no momento de escrita desse trabalho podem ser vistos na Tabela 4.10.

Existem diversas maneiras e tecnologias para criação de jogos para plataforma *iOS*, como a utilização de suítes de desenvolvimento de jogos como *Unity* e *Unreal* ou o desenvolvimento utilizando bibliotecas gráficas mais simples e leves, como *Sprite Kit*, *Cocoa2D*, *UIKit* e *SwiftUI*. Já como linguagem de programação nativa, as mais utilizadas são *Swift*, *Objective-C* e *C++*. *Swift* foi a linguagem escolhida por apresentar as melhores características ao serem comparadas com as outras, como foi visto na seção 2.3.2 da fundamentação teórica, simplicidade, rapidez, segurança e expressividade são os principais diferenciais de *Swift*.

Devido às características desse projeto, como navegações lineares, interface 2D, e estética que lembram ilustrações em um livro, foi escolhido desenvolver o jogo utilizando uma biblioteca de interface que seja declarativa pois apresentam diversas vantagens a bibliotecas imperativas, como visto em 2.3.3, por isso a escolha de usar *SwiftUI* foi feita.

Como REcolorindo o Reino é um jogo *single-player*, local e *off-line*, não houve necessidade de nenhuma comunicação com servidores. Porém, para o bom funcionamento da progressão do jogador durante o jogo, precisamos persistir alguns dados da memória volátil (RAM) para a memória não-volátil (SSD) do dispositivo, assim, mesmo depois de encerrar o jogo e desligar o dispositivo temos o estado preservado. Os dados principais que necessitam ser armazenados são descritos na Tabela 4.11.

Como os requisitos de armazenamento são bem simples e os dados são estruturas

Tabela 4.10 – Dispositivos compatíveis

Smartphones	Tablets
iPhone 13	iPad Pro 12.9-inch (5th generation)
iPhone 13 mini	iPad Pro 11-inch (3rd generation)
iPhone Xs Max	iPad Pro 12.9-inch (4th generation)
iPhone 13 Pro	iPad Pro 11-inch (2nd generation)
iPhone 13 Pro Max	iPad Pro 12.9-inch (3rd generation)
iPhone 12	iPad Pro 11-inch (1st generation)
iPhone 12 mini	iPad Pro 12.9-inch (2nd generation)
iPhone 12 Pro	iPad Pro 12.9-inch (1st generation)
iPhone 12 Pro Max	iPad Pro 10.5-inch
iPhone 11	iPad Pro 9.7-inch
iPhone 11 Pro	iPad (9th generation)
iPhone 11 Pro Max	iPad (8th generation)
iPhone Xs	iPad (7th generation)
iPhone Xr	iPad (6th generation)
iPhone X	iPad (5th generation)
iPhone 8	iPad mini (6th generation)
iPhone 8 Plus	iPad mini (5th generation)
iPhone 7	iPad mini 4
iPhone 7 Plus	iPad Air (4th generation)
iPhone 6s	iPad Air (3rd generation)
iPhone 6s Plus	iPad Air 2
iPhone SE (1st generation)	
iPhone SE (2nd generation)	
iPod touch (7th generation)	

Fonte: Apple (2021)

Tabela 4.11 – Chaves e descrições dos itens armazenados

Chave	Descrição
GAME.ALREADY.OPPENED	Indica se o jogo já foi aberto. Útil para realizar operações somente na primeira vez que o software for aberto.
DID.WATCH.INITIAL.STORY	Indica se a história inicial já foi assistida. Útil para pular a cena da história inicial se ela já foi assistida.
CHOSEN.CHARACTER	Indica qual o personagem selecionado pelo usuário como principal. É o personagem que acompanhará o usuário na cena de Mundo e na cena de Desafio.
LAST.FOCUSED.REGION	Indica qual a última região do mundo que o usuário estava. Útil para recarregar a cena de Mundo sempre mantendo estado preservado.
ITEM.COMPLETED. \$\$\$	Indica se o item \$\$\$ foi completado. \$\$\$ atua como um <i>placeholder</i> para o nome do item.
REGION.COMPLETED. \$\$\$	Indica se a região \$\$\$ foi completada. \$\$\$ atua como um <i>placeholder</i> para o nome da região.
WORLD.COMPLETED. \$\$\$	Indica se o mundo \$\$\$ foi completado. \$\$\$ atua como um <i>placeholder</i> para o nome do mundo.

Fonte: Autor

simples e leves foi escolhido utilizar um componente *KeyValueStore* disponibilizado pelo sistema operacional, chamado *UserDefaults*. Trata-se de um armazenamento de valor-chave persistente hierárquico (opcionalmente distribuído), otimizado para armazenar configurações e informações leves.

4.4.2 Arquitetura geral

Como visto em na seção 2.4 da fundamentação teórica, são diversas as arquiteturas usadas para jogos e a escolha depende principalmente dos *frameworks* ou plataformas utilizadas no desenvolvimento. Como utilizamos *SwiftUI* como *framework* de interface optamos por utilizar conceitos de orientação a objetos num padrão *MVVM*, separando responsabilidade entre *Models*, *ViewModels*, *Views* e *Managers*.

4.4.2.1 Models

Podemos categorizar como modelos os tipos usados pela aplicação para organizar a lógica de negócio e os dados gerais da aplicação. Os principais modelos utilizados nesse

projeto são brevemente descritos a seguir:

Character Enumeração que representa os personagens do jogo. Além de identificar o personagem por um identificador único, possui o nome gráfico, se o personagem é principal ou secundário, *assets* correspondentes e de qual região esse personagem pertence.

World Enumeração que representa os mundos do jogo. Além de identificar o mundo por um identificador único, possui o nome gráfico e quais as suas regiões.

Region Enumeração que representa as regiões do jogo. Além de identificar a região por um identificador único, possui o nome gráfico, personagem da região e quais os itens de desafio.

ChallengeItem Enumeração que representa os itens de desafio do jogo. Além de identificar o item por um identificador único, possui o nome gráfico, *assets* correspondentes e qual sua cor alvo.

Dialog Estrutura que representa a fala de um personagem no jogo. Possui o nome do recurso de áudio e legenda bem como o tipo de dialogo e de qual personagem pertence.

Story Enumeração que representa as histórias do jogo. Além de identificar a história por um identificador único, possui o nome gráfico, e *assets* de video e imagem correspondentes para a cena *StoryScene*.

4.4.2.2 Views

São principalmente responsáveis por estruturar e mostrar a interface para o usuário, também atuam como mediadores entre os inputs através da tela, como botões, e as *ViewModels*. Exemplo das principais *Views* utilizadas nesse projeto são brevemente descritas abaixo:

WorldSceneView Apresenta os *assets* do reino, posicionando os botões de acesso as regiões nos seus devidos locais, também é responsável por configurar a navegação entre os botões das regiões. Também contém a *CommunicationView* centralizada na parte de baixo da tela.

RegionSceneView Apresenta os *assets* da região, posicionando os botões de acesso aos desafios nos seus devidos locais. Também contém a *CommunicationView* centralizada na parte de baixo da tela.

ChallengeSceneView Apresenta os *assets* do desafio, bem como a *ChallengeItemView* configurada no meio da tela, também contém a *CommunicationView* centralizada na parte de baixo da tela.

ChallengeItemView É a *view* que contém o item de desafio, é responsável por mostrar o indicador de porcentagem de completude da cor.

CommunicationView Responsável por apresentar as legendas e personagem dos diálogos correspondentes da cena.

CameraAccessView Mostra a interface de permissão de acesso a câmera do dispositivo, também explica que os dados capturados pela câmera são utilizados somente de forma local e para detecção de cores.

StoryView É responsável por mostrar as animações das histórias e apresentar os botões de interação quando necessário.

RootView É a *view* raiz do jogo, serve somente como *container* para apresentar a *view* correta após o carregamento.

GameButtonView Apresenta *assets* dos botões e coordena as interações e animações de pressionado e bloqueado.

MenuView Apresenta as opções do menu e seus respectivos botões de acesso.

4.4.2.3 ViewModels

Fazem a conexão das *views*, *models* e *managers*, dando estrutura e funcionamento para a aplicação. Os principais *ViewModels* utilizados nesse projeto são brevemente descritos abaixo:

WorldViewModel Controla o funcionamento da *WorldScene*. É responsável por apresentar o mundo permitindo que o jogador navegue entre as regiões correspondentes. Também é responsável por controlar quais regiões estão disponíveis para o jogador entrar e quais já foram completadas com sucesso. Também coordena, juntamente com a *CommunicationViewModel*, as narrações dos personagens.

RegionViewModel Controla o funcionamento da *RegionScene*. É responsável por apresentar a região contendo os itens de desafio correspondentes, controlando quais já foram completados com sucesso. Também coordena, juntamente com a *CommunicationViewModel*, as narrações dos personagens.

ChallengeViewModel A *ViewModel* mais importante, já que controla o funcionamento

da *ChallengeScene* e coordenar o desafio. Utiliza o componente *PredominantColorVideoAnalyzer* para obter informações da cor predominante e aplica-la ao item desafio. Também coordena, juntamente com a *CommunicationViewModel*, as narrações dos personagens.

CommunicationViewModel Controla o funcionamento das *views* de comunicação, que envolvem um personagem, uma área de legendas e o áudio correspondente. Sempre trabalha em conjunto com uma *ViewModel* de cena para executar os diálogos no momento certo, também seleciona de tempo em tempo diálogos apropriados para deixar o jogador envolvido.

CameraAccessViewModel É responsável por controlar a *view* correspondente e pedir permissão de acesso a câmera do dispositivo ao usuário. Também explica que os dados capturados pela câmera são utilizados somente de forma local e para detecção de cores.

StoryViewModel É responsável por controlar o conteúdo de vídeo e imagens da cena de história, também possui o controle de interação para seleção de personagens durante a história inicial.

RootViewModel É a *ViewModel* inicial do jogo. Ao ser ativada, irá carregar do repositório informações básicas de uso do jogo para decidir qual cena deve ser apresentada primeiramente.

4.4.2.4 Managers

São responsáveis por gerenciar um recurso do sistema e realizar a comunicação com o resto da aplicação. Exemplo dos principais *Managers* utilizados nesse projeto são brevemente descritos a seguir:

SoundManager Controla a trilha sonora do jogo. Recebe instruções das *ViewModels* de cena quando há transições de cenas na navegação e a partir dessa informação realiza a troca da trilha se necessário. Também escuta notificações do dispositivo para parar a trilha em caso de fechar o jogo, mudar para outro aplicativo e ou desligar o dispositivo.

RepositoryManager Controla o acesso e persistência de informações de jogo como se o jogo já foi aberto, qual o personagem selecionado pelo usuário e quais itens e regiões estão disponíveis e completos.

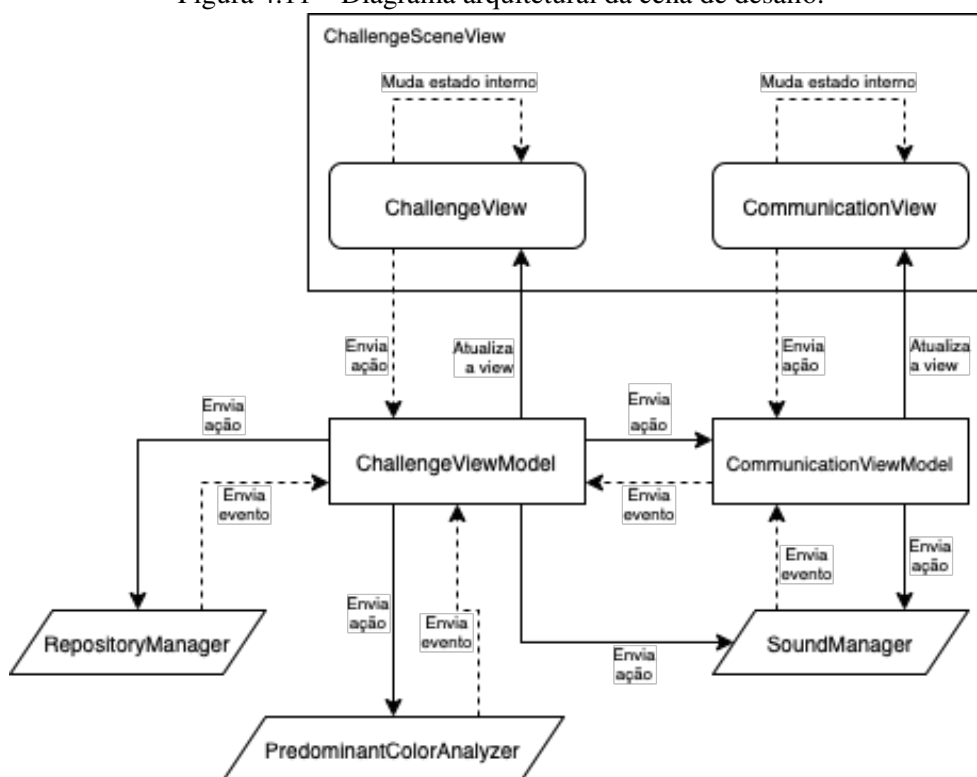
PredominantColorAnalyzer É responsável pela captura de imagens e análise da predo-

minância de cores. Esse componente é de extrema importância para o projeto, por isso será descrito mais profundamente na seção 4.4.4.

4.4.3 Arquitetura da cena

A cena de desafio foi escolhida como exemplo para descrição do funcionamento da arquitetura e relacionamento entre os principais objetos por ser a cena mais complexa e com maior número de relacionamento entre diferentes objetos.

Figura 4.11 – Diagrama arquitetural da cena de desafio.



Fonte: Autor

Na Figura 4.11 são observadas duas *views* (*ChallengeView* e *CommunicationView*) que compõem a *view* *ChallengeSceneView*. Essas possuem estado interno próprio mínimo, somente estado de botões e estado de animações, e seguindo os padrões *MVVM* são conectadas cada uma a seu *ViewModel*. A *ChallengeViewModel*, responsável por coordenar o desafio, se comunica com 3 *Managers* além de se comunicar com a segunda *CommunicationViewModel* que por sua vez se comunica somente com o *SoundManager* para controlar o áudio dos diálogos e sincronizar com as legendas e personagens. Os outros 3 *Managers* utilizados são *RepositoryManager*, *PredominantColorAnalyzer* e *SoundManager*. No diagrama da Figura 4.11 as setas indicam a direção da comunicação,

sendo que as de linha continua indicam chamadas de métodos e funções diretas, já as com tracejado indicam *callbacks* e *delegates*.

Com essa arquitetura, podemos ver que as responsabilidades são bem divididas entre os componentes assim respeitando os princípios do bom desenvolvimento de software.

4.4.4 Detecção de Cores

Nessa seção, será detalhada a implementação do sistema de detecção de cores, começando pela captura de imagens através da câmera do dispositivo, passando pelo algoritmo de predominância de cores utilizados, e por fim pela detecção de similaridade de cores.

4.4.4.1 Captura de Imagens

A implementação da captura de imagens começa pela classe *PredominantColorAnalyzer* que é considerada um *Manager* na nossa arquitetura. Esse objeto é responsável por gerenciar a sessão de captura de vídeo da câmera traseira do dispositivo. Isso é realizado criando um objeto chamado *AVCaptureSession*, que faz parte de uma biblioteca do sistema operacional *iOS*, que é configurando através de um *AVCaptureDeviceInput* com as características necessárias. É também configurada uma saída para a sessão através de um *AVCaptureVideoDataOutput* que notifica o *PredominantColorAnalyzer* pelo mecanismo de *callback/delegate*.

O *manager* então é chamado pelo *callback* 5 vezes por segundo, pois a sessão foi configurada para capturar o vídeo com *frame-rate* de 5 *fps*, já que através de testes práticos foi encontrado ser um valor ótimo para termos boa vazão no algoritmo de predominância de cores e também uma experiência adequada na interface do jogo. Como parâmetro do *callback* é recebido um *frame* do vídeo que é uma matriz de triplas representando as componentes *RGB*. Como as câmeras dos dispositivos entregam cada vez mais qualidade e resolução, foi percebido que para as análises de cor não eram necessários *frames* com tanta resolução, através de testes foi detectado que uma qualidade *VGA* com resolução de 640x480 era suficiente.

4.4.4.2 Predominância de Cores

Descrição em alto nível dos passos do algoritmo de predominância de cores:

1. Criar um objeto chamado *colorsCountedSet* do tipo *CountedSet* que será responsável por armazenar elementos RGB correspondentes a cada pixel da imagem. *CountedSet* é uma coleção mutável, sem ordem e que guarda apenas objetos distintos mas que armazena o número de repetições quando duplicados forem adicionados. É a estrutura perfeita já que o acesso e adição de elementos tem um custo médio constante $O(1)$.
2. Uma interação sobre a matriz de pixel é realizada (custando $O(w*h)$) e os valores de *RGB* são adicionados um por um na variável *colorsCountedSet* definida anteriormente. Ao fim desse passo temos na variável *colorsCountedSet* uma lista de todas as diferentes cores da contidas na imagem e quantas vezes cada uma apareceu na imagem.
3. Os elementos de *colorsCountedSet* são percorridos e todas cores que contem uma repetição menor do que 0.1% do tamanho da imagem são removidas. Isso serve para otimizar os passos seguintes e descartar cores que não são relevantes.
4. Agora o conteúdo de *colorsCountedSet* é ordenado em ordem decrescente pela quantidade de repetições. Logo após a ordenação somente as 500 primeiras são selecionadas e o resto descartado, novamente, por questões de otimização.
5. Nesta etapa vamos agrupar cores da lista de saída da etapa anterior com base na diferença entre elas usando o algoritmo *CIE94 ΔE* . A primeira cor a ser adicionada no grupo é a cor alvo, assim será feita a comparação dando prioridade para ela. Após essa etapa temos um grupo de cores distintas entre si com uma contagem de frequência de aparição na imagem.
6. Como as cores podem estar desordenadas devido a etapa anterior, é realizada uma nova ordenação em ordem decrescente pela quantidade de repetições. Logo após a ordenação somente as 8 primeiras são selecionadas e o resto descartado, novamente, por questões de otimização.
7. Dividimos o número de quantidade de repetições de cada cor pela soma total para obter uma porcentagem de predominância e após isso temos uma lista de 8 cores distintas dominantes da imagem ordenadas pela porcentagem de aparição.

4.4.4.3 Comparação de Similaridade

Com a lista ordenada de cores predominantes da imagem, comparamos o primeiro item com a cor alvo (a cor que sabemos que o usuário está procurando) usando o algoritmo

Tabela 4.12 – Valores de comparação do CIE Delta E

Valor	Significado
0.0	Idêntico (<i>Identical</i>)
(0.0, 1.0]	Similar (<i>Similar</i>)
([1.0, 2.0]	Perto (<i>Close</i>)
(2.0, 10.0]	Próximo (<i>Near</i>)
(10.0, 50.0]	Diferente (<i>Different</i>)
(50.0, *]	Longe (<i>Far</i>)

Fonte: Autor

$CIE94 \Delta E$ e temos como resposta a distância da cor predominante na imagem com a cor alvo. A partir desse valor, utilizamos as constantes da Tabela 4.12 para saber se a proximidade das cores.

Nos testes realizados durante a implementação desse sistema de detecção de cores foram usados os 3 algoritmos de similaridade de cores, Distância Euclidiana RGB, $CIE76 \Delta E$, $CIE94 \Delta E$. O algoritmo $CIE94 \Delta E$ foi o escolhido por apresentar maior precisão na comparação. Por mais que ele seja mais custoso a performance dos dispositivos hoje em dia são mais que suficientes suportar essa computação com a frequência necessária.

Alguns dos resultados dos testes unitários que foram desenvolvidos durante o desenvolvimento do sistema são mostrados na Figura 4.12. Podemos ver através dessas 6 imagens que dado uma cor alvo (mostrada no topo esquerdo das imagens) o sistema de predominância e identificação de cores identificou corretamente as cores principais da imagem (mostradas na lateral direita da imagem), priorizando as cores mais próximas da cor alvo.

Figura 4.12 – Resultado de testes unitários com imagens da camera.



Fonte: Autor

4.5 Resultado

Nesta seção, será mostrado um pouco do resultado ao final do desenvolvimento do projeto. Foram produzidos 6 cenários diferentes (5 regiões e 1 mapa geral), 8 personagens, 3 trilhas sonoras, narrações das falas dos personagens, efeitos sonoros, efeitos de comemoração, além de todo código e documentação da implementação. Todo material foi produzido exclusivamente para este jogo.

A seguir, na Figura 4.13 seguem imagens finais de 3 cenas do jogo: Cena do Mapa do Reino, Cena de Região e Cena de Desafio.

Figura 4.13 – Exemplo de 3 cenas que foram desenvolvidas



Fonte: Autor

4.5.1 Funcionalidades do Aplicativo

- Detecção de cores através da câmera do dispositivo móvel;
- História envolvente que explora a fantasia;
- Mapa geral do reino com 5 regiões onde cada uma possui 5 objetos para recolorir;
- 9 personagens, sendo 3 principais (Léopetro - Lindamaria - Magotávio);
- Trilha sonora e arte específicas para o público infantil, tornando a experiência mais completa e cativante;
- Desafios estimulam o movimento e que motivam a resolução de problemas com diversão;
- Jogo criado especificamente para crianças de 2 a 5 anos. Que combina interações do mundo virtual com o mundo real, possibilita o aprendizado através da diversão e da brincadeira.

4.5.2 Material Gráfico

Nas próximas seções, é apresentado o material produzido exclusivamente para o jogo infantil.

A Figura 4.14 contém as ilustrações dos personagens do jogo: Léopetro, Lindamaria, Magotávio Fazendeiro, Jardineira, Vendedora e Corão.

Figura 4.14 – Conjunto de todos personagens desenvolvidos



Fonte: Autor

A Figura 4.15 contém as ilustrações dos 25 objetos a serem coloridos no jogo:

Figura 4.15 – Conjunto de todos itens desenvolvidos



Fonte: Autor

A Figura 4.16 contém as ilustrações das 5 regiões e do Mapa do Reino:

Figura 4.16 – Conjunto de todas as regiões desenvolvidas



Fonte: Autor

Foram produzidas duas animações para contar a história do jogo, apresentadas na Figura 4.17, são elas:

Introdução: Introduz a história do jogo e seus personagens. No final desta animação é proposto um desafio para os jogadores: ajudar os personagens a achar as cores do reino que foi descolorido pelo malvado vilão. Duração: 02m02s.

Encerramento: Finaliza a história iniciada por Lindamaria e Léopoldo e vivenciada pelo jogador: com a ajuda do medalhão, o Magotávio é aprisionado novamente na caverna trazendo paz, alegria e cores de volta ao reino. Duração: 00m42s.

Figura 4.17 – À esquerda: cena da introdução. À direita: cena do encerramento.



Fonte: Autor

4.5.3 Material de Áudio

Para uma experiência mais rica e prazerosa, torna-se essencial a utilização de recursos de áudio e música. Portanto, o jogo possui a narração da história, falas dos personagens, trilhas musicais e outros efeitos. Foram produzidas 3 músicas para as animações e cenas do jogo. Além das trilhas, todas as falas dos personagens são narradas por interpretadores que deixam o jogo mais atrativo e também mais acessível a todas idades e crianças portadoras de necessidades especiais.

4.5.4 Cheat Code

A funcionalidade de liberação de todos os níveis do jogo, necessária para testes, foi escondida dentro do menu de reinicialização do jogo. Este menu é acessado através do menu principal do jogo que pode ser acessado através do botão com a engrenagem no Mapa do Reino. Ao digitar a sequência de números: **1 - 9 - 3 - 7 - 5 - 6**, todos os níveis do jogo tornam-se abertos.

4.5.5 Atribuições

Sob encomenda do Autor, as ilustrações de REcolorindo o Reino foram produzidas por *Júlia Albertin*, a trilha sonora foi composta por *Sergio Olivé* e a narração dos personagens foram feitas por *Heitor Schmidt*, *Theo Schmidt*, *Eduarda Schmidt*, *Mariah Schmidt* e *Râmisa Schmidt*. Por fim, o *game design* que deu origem a esse projeto foi inicialmente proposto em parceria com *Albert Kolberg* e *Fernando Donini*, em 2014, quando também foi desenvolvida uma versão protótipo do jogo que serviu como base na re-escrita do projeto em novas tecnologias, linguagens e *frameworks*. A versão inicial do jogo que deu a origem para esse trabalho foi premiada no projeto do governo federal *INOVAapps*, premiando os melhores projetos de jogos sérios do país.

Muito obrigado a todos por sua participação nesse projeto.

5 AVALIAÇÃO COM USUÁRIOS

O objetivo principal desse trabalho foi criar um jogo infantil que envolva, divirta e entretenha, além de incentivar o movimento das crianças. Assim, o objetivo principal desse capítulo é realizar avaliações com possíveis usuários do jogo com o intuito de verificar se essas premissas foram satisfeitas. Como o público alvo do jogo são crianças de 2-5 anos, além de realizar testes com elas também foram realizados alguns testes com adultos, para assim avaliarmos a aceitação do jogo na família, já que muitas vezes os pais que acabam propondo atividades para crianças dessa idade, além de tomarem conta dos dispositivos eletrônicos.

Esse capítulo é dividido em 2 seções, a primeira detalha as atividades realizadas e os métodos, ferramentas e protocolos utilizados na execução. Por fim, na segunda seção, os resultados são apresentados e analisados.

5.1 Experimento

Segundo Barendregt (2006), se o jogo não é agradável e divertido, se não desperta o interesse do usuário, se é cansativo, ou se é muito difícil ou muito fácil, o jogador não consegue se entreter de forma plena. O que um jogo precisa conter para satisfazer esses fatores depende de muitas características do público alvo, como faixa etária, desenvolvimento intelectual, condições psicológicas e sociais, e claro, o objetivo do time de criação do jogo, geralmente um conjunto de atributos objetivos subjetivos.

É muito difícil isolar os problemas de usabilidade dos problemas de entretenimento, já que são relacionados e dependem um do outro, então para esse conjunto de fatores se deu o nome jogabilidade. Essa característica é o que se procura avaliar em uma avaliação de jogos com usuários. Na Figura 5.1, Jordan (2000) coloca em uma pirâmide as necessidades dos usuários de forma hierárquica e é muito usada para guiar avaliação de jogos.

Figura 5.1 – Pirâmide da necessidade dos jogadores.



Fonte: (JORDAN, 2000)

Segundo Lazzaro e Keeker (2004), não se deve analisar os mesmos aspectos de usabilidade ao testar jogos e aplicações, pois o objetivo dos mesmos são distintos. Na Tabela 5.1 são mostradas alternativas para os aspectos mais comuns de análise.

Tabela 5.1 – Aspectos de usabilidade em jogos e aplicações.

Aplicação	Jogo
Realização da tarefa	Conclusão da tarefa
Eliminar erros	Diversão ao vencer obstáculos
Recompensa externa	Recompensa pessoal
Considera que a tecnologia deve ser humanizada	Considera que humanos precisam ser desafiados
Intuitividade	Novas coisas devem ser aprendidas
Reduzir a carga de trabalho	Aumentar a dificuldade

Fonte: (LAZZARO; KEEKER, 2004)

Na avaliação da jogabilidade de REcolorindo o Reino foram selecionadas 7 crianças acompanhadas de seus pais para participar dos experimentos, as crianças por motivos de serem o público alvo do jogo e os adultos as acompanhando para que possamos ter uma noção de quão bem aceito o jogo seria na família, além de podermos realizar os questionários comportamentais e demográficos separados para as crianças e adultos. Essa combinação propicia uma avaliação mais completa, pois é difícil de realizar alguns tipos de perguntas diretamente para as crianças. O teste é dividido em duas etapas, a primeira é a análise de uso e a segunda é a realização dos questionários.

Antes da execução dos experimentos, é lida a política de privacidade e uso de dados de forma anônima para os testadores, criança e adulto responsável. O teste só inicia quando os dois concordam em participar e que não tenham nenhuma dúvida sobre o uso dos dados.

5.1.1 Análise de Uso

Durante a análise de uso, a criança utilizaram o jogo de forma não-guiada, os usuários foram somente instruídos a testarem o jogo da maneira que achassem melhor, durante um período de aproximadamente 10 minutos. Como o REcolorindo o Reino interage diretamente com o ambiente externo, foi criada uma simulação de ambiente com objetos e cores pré-selecionadas, assim todos os testadores realizam o teste em condições similares. Devido a pandemia de *COVID-19*, os testes foram realizados em ambiente aberto, e com todas as medidas de prevenção recomendadas.

É encorajado que durante o teste os usuários falem o que estão pensando, sentindo e tentando realizar, como sugere o método *Think Aloud*, assim o avaliador pode tomar notas dos comentários para análise. Além das anotações do avaliador, é mantida uma lista de tarefas esperadas, onde são anotadas quais as tarefas que o usuário conseguiu realizar durante o tempo proposto. Não é esperado que o usuário realize todas as tarefas, apenas usaremos para medir a diferença na exploração em cada teste.

5.1.1.1 Lista de tarefas esperadas:

1. Abrir o jogo.
2. Assistir história inicial.
3. Escolher personagem principal.
4. Navegar pelo mapa do reino.
5. Entrar em uma das regiões.
6. Entrar em mais de uma região.
7. Entrar em mais de duas regiões.
8. Entrar em item de desafio.
9. Completar um item de desafio.
10. Completar mais de um item diferente de desafio.
11. Completar mais de dois itens diferentes de desafio.
12. Completar mais de cinco itens diferentes de desafio.
13. Completar todos os itens de desafio da última etapa.
14. Assistir o encerramento da história.

5.1.2 Questionários

Após a análise de uso, um breve questionário foi realizado para criança e logo em seguida o acompanhante também respondeu algumas questões sobre a criança e sua impressão sobre a análise. Assim, podemos personalizar as perguntas e obter informações mais relevantes de cada grupo.

5.1.2.1 Perguntas para crianças:

PC1 Você achou esse jogo fácil, médio ou difícil de entender?

PC2 Você gostaria de jogar esse jogo até o final?

PC3 O que você mais gostou no jogo?

PC4 O que você achou de um jogo que você precisa levantar, caminhar, se mexer para jogar?

5.1.2.2 Perguntas para adultos:

O questionário para adultos utiliza a escala proposta por Likert (1932) para respostas, com 5 opções: (1) Discordo totalmente, (2) Discordo parcialmente, (3) Neutro, (4) Concordo parcialmente, (5) Concordo totalmente.

PA1 Jogos como esses são importantes para o desenvolvimento de minha filha/meu filho.

PA2 Minha filha/Meu filho gostaria desse jogo.

PA3 Seria muito difícil para minha filha/meu filho jogar esse jogo no dia a dia.

PA4 Minha filha/Meu filho seria mais ativo se jogasse mais jogos assim.

PA5 Minha filha/Meu filho preferiria um jogo desses comparado com um jogo normal que não precisaria se mexer.

COMENTÁRIOS Comentários sobre as perguntas anteriores.

5.1.2.3 Perguntas demográficas:

As perguntas demográficas são muito úteis para reconhecer padrões de comportamento e gostos dependendo diretamente de fatores da vida do usuário. As perguntas que foram realizadas são:

PD1 Qual a idade de sua filha/seu filho?

PD2 Qual o gênero de sua filha/seu filho?

PD3 Qual a escolaridade de sua filha/seu filho?

PD4 Qual o "tempo de tela" de sua filha/seu filho? (TV, computador, video-game, celular, etc.)

PD5 Qual a frequência de utilização de *smartphones* e *tablets* de sua filha/seu filho?

PD6 Qual a frequência de utilização de jogos digitais?

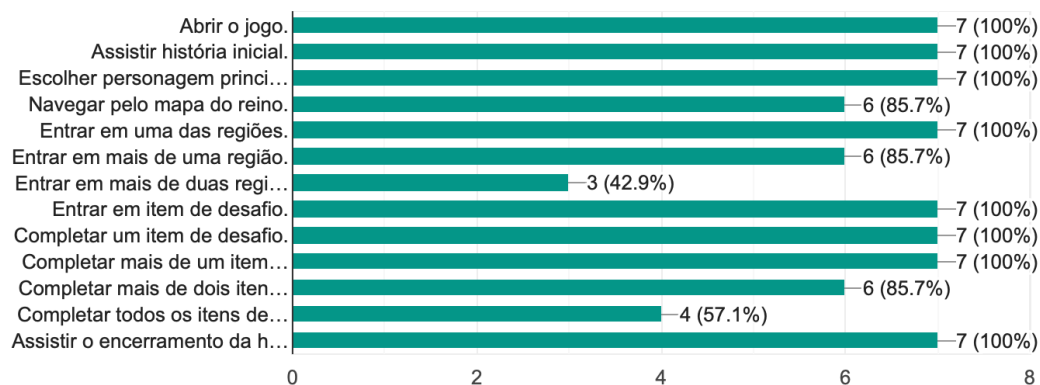
5.2 Resultados

Nessa seção, são resumidos e analisados os resultados obtidos durante a análise de uso, perguntas para crianças, perguntas para adultos e perguntas demográficas. Além disso, também seguem algumas anotações e comentários que foram capturados durante o teste.

5.2.1 Análise de uso

No gráfico da Figura 5.2, temos um resumo das tarefas realizadas por cada usuário durante o uso do jogo. Podemos analisar que o uso do jogo foi bastante satisfatório, indicado pelo fato de que quase todos os usuários cumpriram quase todas as tarefas. Algumas das tarefas como "Entrar em mais de duas regiões" e "Completar todos os itens da última etapa" foram concluídas por uma média de 50% dos usuários. Essa métrica era esperada no momento de criação das tarefas e é um indicador que algumas crianças tiveram mais facilidade e exploraram mais o jogo durante o teste do que outras. Realizando uma análise juntamente com os dados demográficos e observações do aplicador, percebe-se que crianças menores tiveram uma progressão mais lenta no jogo durante o tempo proposto, porém, também percebemos que crianças mais velhas mostraram menor interesse em desbravar as diferentes áreas do jogo. Acredita-se que isso não seja um problema, pois é esperado que diferentes crianças progridam diferentemente no jogo, e como indicado no gráfico, todas as crianças conseguiram completar a maioria das tarefas.

Figura 5.2 – Tarefas realizadas durante os testes:



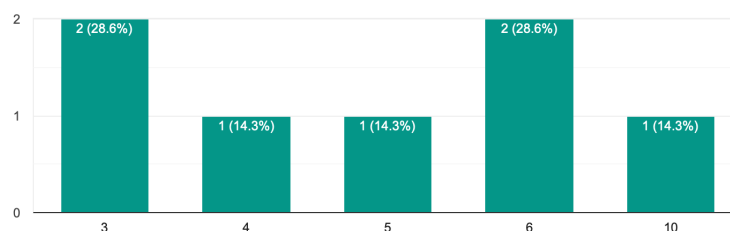
Fonte: Autor

Uma observação realizada durante os testes é relacionada à reação das crianças ao descobrirem a dinâmica do jogo pela primeira vez, apresentando surpresa e encanto ao perceber que podem capturar as cores com a câmera.

5.2.1.1 Perguntas demográficas:

[PD1] Qual a idade de sua filha/seu filho?

Figura 5.3 – Respostas PD1

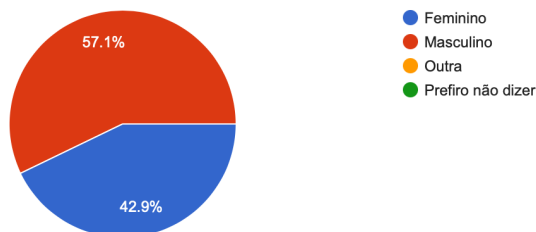


Fonte: Autor

Para os testes de uso, foram selecionadas crianças de diferentes idades, como pode ser observado no gráfico da Figura 5.3. O foco foram crianças de 3-6 anos, porém também realizamos o teste com uma criança maior, de 10 anos.

[PD2] Qual o gênero de sua filha/seu filho?

Figura 5.4 – Respostas PD2

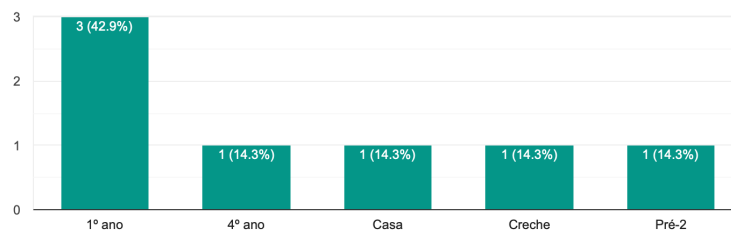


Fonte: Autor

Também foi tentado equilibrar o número de crianças de cada gênero, como pode ser visto no gráfico da Figura 5.4, tivemos uma diferença de 1 criança a mais com gênero masculino.

[PD3] Qual a escolaridade de sua filha/seu filho?

Figura 5.5 – Respostas PD3

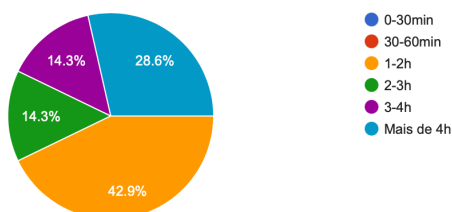


Fonte: Autor

A respeito da escolaridade, como nossos testadores tem idades diferentes, também observamos uma distribuição, como pode ser visto na Figura 5.5. Inclusive com uma criança que não está ainda na escola.

[PD4] Qual o "tempo de tela" de sua filha/seu filho? (TV, computador, video-game, celular, etc.)

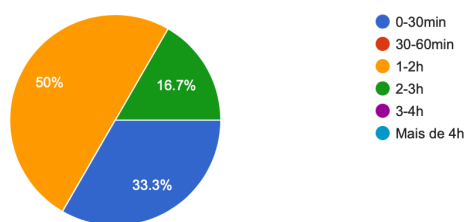
Figura 5.6 – Respostas PD4



Fonte: Autor

[PD5] Qual a frequência de utilização de *smartphones* e *tablets* de sua filha/seu filho?

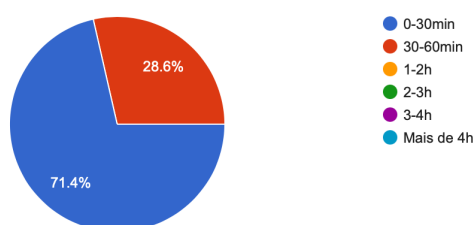
Figura 5.7 – Respostas PD5



Fonte: Autor

[PD6] Qual a frequência de utilização de jogos digitais?

Figura 5.8 – Respostas PD6



Fonte: Autor

Foi perguntado também, através das 3 perguntas das Figuras 5.6, 5.7 e 5.8, o "tempo de tela" total de cada criança, quanto desse tempo é em *smartphones* e *tablets*, e quanto é em jogos digitais. Pode-se analisar que a maioria das crianças dessa idade passa menos de 30 minutos jogando jogos, e nenhuma delas passa mais de 60 minutos. Segundo comentários após a pergunta de tempo de jogos digitais pais comentaram que a preferência era dada para visualização de vídeos na plataforma *Youtube* através do celular ou *tablet* e para programas de TV.

5.2.2 Perguntas para crianças

Como esperado, as respostas das crianças para a maioria das perguntas foram muito simples, quanto mais abertas as perguntas mais elaboradas as respostas, porém, mesmo assim, bem simples, principalmente nas crianças menores.

Muitas respostas sim e não. Poucas Perguntas que se destacaram:

[PC1] Você achou esse jogo fácil, médio ou difícil de entender?

A maioria das crianças achou o jogo fácil, porém algumas crianças responderam que acharam médio/difícil. Isso pode ser devido a dificuldade de capturar as cores corretas

para alguns objetos, principalmente tons da cor verde. Isso pode ser devido ao tamanho do espectro do verde nos espaços de cores. Algumas crianças também confundiram cores como vermelho e laranja, o que é esperado para crianças que ainda estão desenvolvendo seu senso de cores, e acredita-se que o jogo poderia ajudar nesse desenvolvimento também e como visto anteriormente, ao desenvolver um jogo não podemos deixá-lo muito fácil nem muito difícil, mas ser desafiante é uma característica positiva na jogabilidade.

[PC2] Você gostaria de jogar esse jogo até o final?

Todas os usuários responderam que gostariam de jogar até o final.

[PC3] O que você mais gostou no jogo?

Nessa pergunta obtivemos respostas muito interessantes, provavelmente por ser uma pergunta mais aberta. Algumas das respostas que mais se destacaram:

- "Das cores."
- "Do *tablet* e das cores."
- "Gostei da história, que as cores sumiram."
- "A história das cores."
- "Gostei que é um jogo diferente."
- "Achar as cores."
- "Os desenhos."

[PC4] O que você achou de um jogo que você precisa levantar, caminhar, se mexer para jogar?

Nessa pergunta também obtivemos respostas interessantes, algumas das respostas que mais se destacaram:

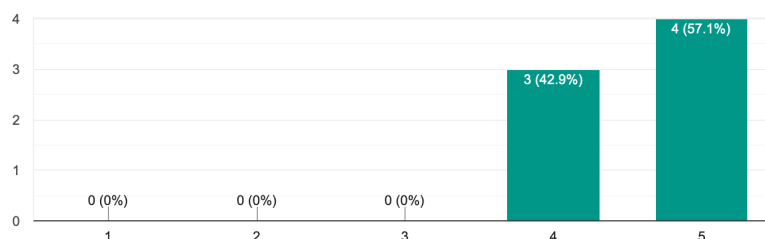
- "Gostei que tem que sair pra procurar as cores."
- "Achei legal, eu gosto de jogos que tem que se mexer."Citou o jogo *JustDance* em comparação.
- "Gostei, igual *Pokemon* que tem que sair procurando.

5.2.3 Perguntas para adultos

Para os adultos as perguntas foram elaboradas usando a escala de LIKERT variando de 1 a 5. A maioria das respostas foram bem satisfatórias, segue abaixo os graficos de respostas e comentários sobre algumas respostas interessantes.

[PA1] Jogos como esses são importantes para o desenvolvimento de minha filha/meu filho.

Figura 5.9 – Respostas PA1

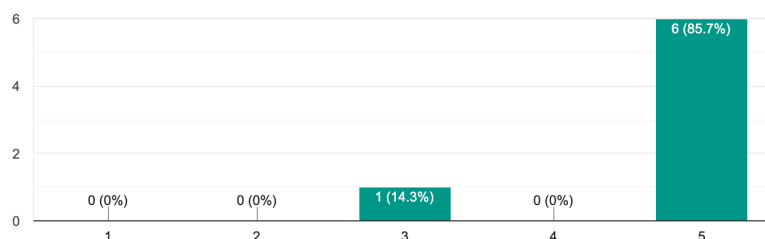


Fonte: Autor

Podemos analisar a partir da Figura 5.9 que todos os adultos concordaram que jogos com uma dinâmica diferente e que influencia o movimento são importantes ou muito importantes para o desenvolvimento infantil.

[PA2] Minha filha/Meu filho gostaria desse jogo.

Figura 5.10 – Respostas PA2

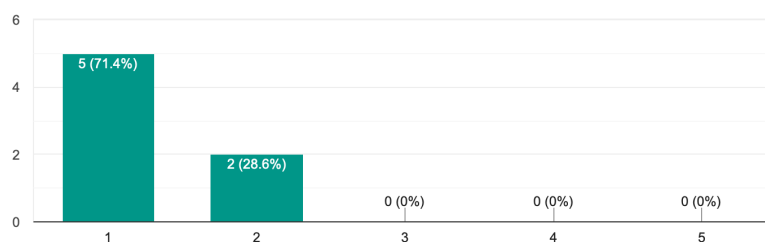


Fonte: Autor

A Figura 5.10 evidencia que a maioria dos adultos acredita que seus filhos gostariam de jogar o jogo. Apenas um dos entrevistados disse que não tinha certeza e disse que seu filho já estava acostumado com jogos mais avançados no computador e que não saberia se esse jogo seria ideal pra sua faixa etária, o que era esperado, pois esse usuário foi a criança mais velha que participou do teste.

[PA3] Seria muito difícil para minha filha/meu filho jogar esse jogo no dia a dia.

Figura 5.11 – Respostas PA3

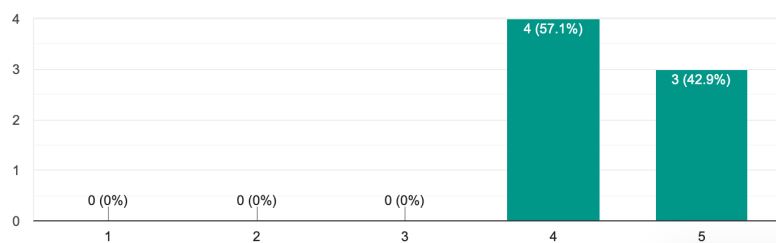


Fonte: Autor

Todos os adultos acreditam que seus filhos não teriam dificuldades para jogar esse jogo no seu dia a dia, como pode ser analisado na Figura 5.11.

[PA4] Minha filha/Meu filho seria mais ativo se jogasse mais jogos assim.

Figura 5.12 – Respostas PA4

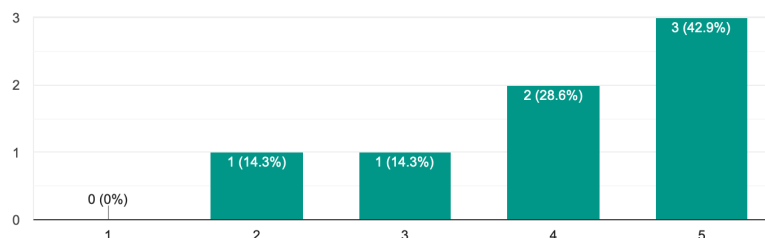


Fonte: Autor

Já através da Figura 5.12 podemos concluir que todos os adultos acreditam que seus filhos seriam mais ativos se jogassem mais jogos assim.

[PA5] Minha filha/Meu filho preferiria um jogo desses comparado com um jogo normal que não precisaria se mexer.

Figura 5.13 – Respostas PA5



Fonte: Autor

Por fim, analisando a Figura 5.13, podemos concluir que a maioria dos adultos responderam que acreditam que seus filhos prefeririam jogos com uma proposta diferente do que o jogo normal, incentivando o movimento e a dinâmica de interagir com o mundo real. Porém, alguns adultos acreditam que seus filhos não deixariam de jogar seus jogos tradicionais em prol de jogos que exigem muito movimento e esforço físico. Analisando as perguntas demográficas, essas crianças, cujo pais responderam com mais dúvidas sobre a afirmação, são as crianças mais velhas que entrevistamos e que jogam muitos outros jogos mais avançados.

[COMENTÁRIOS] Comentários sobre as perguntas anteriores.

- "Ela adorou, ela não consegue ficar muito tempo parada, então seria perfeito pra conseguir se entreter."

- "Achei bem legal o jogo, interessantes que os jogos possibilitem movimento nesta idade de formação. Também a história e o envolvimento das cores são muito legais."
- "Acho que ele não trocaria os jogos de computador por um jogo desse tipo, mas talvez jogaria os dois."
- "Acho super importante este tipo de jogo que faz a criança se exercitar, acho que no início devo ajudar a entender como se joga."
- "Interessante, ele gostava muito do *Pokemon*, mas acho que passou a fase agora. Não "sei se a história ele ia gostar tanto pois já ta acostumado com jogos tipo *Minecraft* e tal."
- "Com certeza, ela não gosta muito de jogos no *tablet*, mas acho que se fosse pra se movimentar e brincar ela gostaria mais."
- "Seria bem bom, pois fica bastante assistindo tv e se puder jogar um jogo que faça se movimentar e aprender ao mesmo tempo seria perfeito."

6 CONCLUSÃO

Este trabalho apresentou o desenvolvimento completo de REcolorindo o Reino, um jogo criado com foco na primeira infância com o objetivo de ser utilizado no cotidiano de crianças, propiciando entretenimento e estimulando o desenvolvimento lógico/motor. Em REcolorindo o Reino, a utilização da câmera como ferramenta de jogo não é simplesmente um facilitador, é também uma inovação no modo de interação com os recursos do jogo, com o intuito de ajudar a resolver o real problema da ociosidade no desenvolvimento infantil com diversão e muito movimento.

Para realização desse trabalho de maneira plena, aprofundamentos técnicos foram necessários em diversos temas. Criação de jogos, design, desenvolvimento *iOS*, captura de imagens, *Swift* e *frameworks* de interface foram estudados, tudo isso utilizando práticas de métodos ágeis provendo eficiência e uma certa flexibilidade para o desenvolvimento do projeto. Para sua idealização, pesquisas sobre desenvolvimento infantil -motor e lógico- foram necessárias para que a proposta pudesse mitigar melhor o problema que o excesso de tempo de tela e ociosidade causam. Durante a implementação, algumas dificuldades com a detecção de cores foram enfrentadas e através de pesquisas foi entendido melhor como o ser humano percebe esse fenômeno, como existem diferentes métodos de representarmos cores, e como diferentes formas de analisar e comparar essas cores implicam na proximidade com o compreendido pelo ser humano.

Esse trabalho forneceu uma visão muito interessante sobre percepção humana, o quanto computadores e dispositivos eletrônicos devem ser pensados e calibrados para nossas sensações, e o quanto eles impactam nossa vida. A versão final produzida pelos esforços descritos neste projeto conseguiu atender os requisitos estabelecidos e está disponível gratuitamente na AppStore (2021) (2021).

Todo trabalho tem espaço para melhorias, em REcolorindo o Reino não é diferente, mais itens de desafio, mais interações com personagens e mais animações são apenas algumas das melhorias de conteúdo possíveis. Já em relação a melhorias no funcionamento, as principais seriam no algoritmo de detecção de predominância de cores, que poderia ser evoluído para detectar objetos, assim isolando o fundo da imagem, e dessa maneira o resultado da detecção poderia se aproximar ainda mais da expectativa do jogador.

Acredita-se também que seria muito interessante a realização de uma nova versão de REcolorindo o Reino, dessa vez em Realidade Aumentada, similar ao jogo *Pokémon*

Go, conectando ainda mais o mundo virtual do jogo com o mundo real do jogador. Essa foi uma das propostas de soluções criadas na etapa de *CBL*, porém pela dificuldade no desenvolvimento seria impraticável para esse projeto, por isso foi descartada e deixada como uma ideia para uma exploração futura.

REFERÊNCIAS

- APPLE. **Swift Org**. 2008. Disponível em: <<https://swift.org>>.
- APPLE. **SwiftUI Overview**. 2020. Disponível em: <<https://developer.apple.com/xcode/swiftui/>>.
- APPLE. **Apple iOS 15**. 2021. Disponível em: <<https://www.apple.com/ios/ios-15/>>.
- APPSTORE. 2021. Disponível em: <<https://apps.apple.com/br/app/recolorindo-o-reino/id924580292>>.
- BARENDREGT, W. Evaluating fun and usability in computer games with children. **Electronics Letters - ELECTRON LETT**, 01 2006.
- BECK, K. et al. Manifesto for agile software development. 2001.
- CBL.ORG. **Challenge Base Learning Org**. 2021. Disponível em: <<https://www.challengebasedlearning.org/>>.
- EIDHOF FLORIAN KUGLER, M. G. C. **App Architecture: iOS Application Design Patterns in Swift**. [S.l.]: CreateSpace Independent, 2018.
- GOOGLE. **Emoji Scavenger Hunt**. 2019. Disponível em: <<https://emojiscavengerhunt.withgoogle.com/>>.
- GUERRA, P. H.; FARIAS, J. C. d.; FLORINDO, A. A. Comportamento sedentário em crianças e adolescentes brasileiros: revisão sistemática. **Revista de Saúde Pública, SciELO Brasil**, v. 50, 2016.
- HUDSON, P. **Hacking with Swift**. 2020. Disponível em: <<https://www.hackingwithswift.com/quick-start/swiftui/what-is-swiftui>>.
- HUNT, R. **The Reproduction of Colour**. Wiley, 2005. (The Wiley-IS&T Series in Imaging Science and Technology). ISBN 9780470024263. Disponível em: <<https://books.google.com.br/books?id=Cd\FVeuO10gC>>.
- IBRAHEEM, N. A. et al. Understanding color models: a review. **ARNP Journal of science and technology**, Citeseer, v. 2, n. 3, p. 265–275, 2012.
- JOBLOVE, G. H.; GREENBERG, D. Color spaces for computer graphics. **SIGGRAPH Comput. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 12, n. 3, p. 20–25, ago. 1978. ISSN 0097-8930. Disponível em: <<https://doi.org/10.1145/965139.807362>>.
- JORDAN, P. W. **Designing pleasurable products: An introduction to the new human factors**. [S.l.]: CRC press, 2000.
- KISHIMOTO, T. M. **O Brincar e Suas Teorias**. [S.l.]: Cengage Learning, 2011.
- LAZZARO, N.; KEEKER, K. What's my method?: a game show on games. In: . [S.l.: s.n.], 2004, p. 1093–1094.

LIKERT, R. **A Technique for the Measurement of Attitudes**. Archives of Psychology, 1932. (A Technique for the Measurement of Attitudes, Nº 136-165). Disponível em: <<https://books.google.com.br/books?id=9rotAAAAYAAJ>>.

MEDIA, W. **ARrrrrgh**. 2017. Disponível em: <<https://apps.apple.com/us/app/arrrrgh/id1299548311>>.

MITCHELL. **The Model View Controller History**. 2014. Disponível em: <<https://wiki.c2.com/?ModelViewControllerHistory>>.

MOOSE, D. D. **Peek-a-Zoo**. 2017. Disponível em: <<https://apps.apple.com/app/id477766317?xs=1>>.

MYERS, B.; BEIGL, M. Handheld computing. **Computer**, v. 36, n. 9, p. 27–29, 2003.

NIANTIC. **Pokemon GO**. 2016. Disponível em: <<https://apps.apple.com/us/app/pokémon-go/id1094591345>>.

PAIVA, N. M. N. D.; COSTA, J. A influência da tecnologia na infância: desenvolvimento ou ameaça. **Psicologia. pt**, v. 1, p. 1–13, 2015.

PIAGET, J. **O juízo moral na criança**. [S.l.]: Summus Editorial, 1994.

PSICOMOTRICIDADE, A. B. de. **Definição de Psicomotricidade**. 2015. Disponível em: <<https://psicomotricidade.com.br/sobre/o-que-e-psicomotricidade/>>.