



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia de Eletrónica e Telecomunicações e de
Computadores**



Jogo Social com o Propósito de Aproximar Pessoas

LUÍS ALEXANDRE MARTINS PINTO

(Licenciado)

Projeto Final para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Prof. Doutor Rui Jesus

Júri:

Presidente: Prof. Doutor Nuno Cruz

Vogal: Prof. Doutor André Lourenço

Setembro, 2021



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia de Eletrónica e Telecomunicações e de
Computadores**



Jogo Social com o Propósito de Aproximar Pessoas

LUÍS ALEXANDRE MARTINS PINTO

(Licenciado)

Projeto Final para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Prof. Doutor Rui Jesus

Júri:

Presidente: Prof. Doutor Nuno Cruz

Vogal: Prof. Doutor André Lourenço

Setembro, 2021

À minha mãe, Amélia que sempre esteve lá para mim nos maus e bons momentos. Pelo teu apoio e sacrifício sem ti não era possível tamanho feito.

Aos meus avós maternos, Hermínia e João por cuidarem de mim e me ensinarem lições fundamentais da vida que me ajudaram a guiar na verdadeira direção.

Aos meus avós paternos, Lourdes e António pelo apoio e suporte do qual não seria possível sem o vosso dispor.

Ao meu pai, José Luís de quem eu pouco conheci mas que tanto me marcou.

Ao meu melhor amigo Miguel a tua unicidade constante é um pilar de suporte na minha vida.

Agradecimentos

Ao Professor Doutor Rui Jesus por me ter acolhido e apoiado no cumprimento deste projeto. Sem a sua orientação tal não seria possível.

Ao ISEL - Instituto Superior de Engenharia de Lisboa que marcou a minha vida pela sua formação profissional e pelos seus docentes que me fizeram crescer como aluno e ser humano. Através da Licenciatura e Mestrado aprendi várias lições de vida e como continuar em frente mesmo quando parecia impossível.

Aos meus amigos de Licenciatura, André Luz, António Simões, Catarina Vaz, Diogo Afonso, Nuno Sousa, Miguel Serejo, Tiago Costa e Ricardo Duarte cujas amizades serão sempre nutridas com carinho no meu coração.

Aos meus amigos de Mestrado, Daniel Rodrigues e Diogo Horta cuja amizade fortuita resultou como um marco de boas memórias a conservar.

Aos meus amigos David Vilhena e Ricardo Oliveira por me iluminarem etapas da vida e que por muito distante que estejam a vossa amizade continuará sempre ao meu lado.

À minha família por todo o suporte, sacrifício e apoio dado. Sem vocês não me tinha levantado quando caía e não me conseguia levantar. Sem a vossa visão e luz guia não encontraria o caminho a marcar. Este será sempre um marco da minha vida que sem a vossa presença nunca teria sido possível.

Resumo

Num mundo em que a tecnologia é cada vez mais predominante na sociedade, a apatia social tem vindo a erguer-se como principal causa da falta de interação entre as pessoas. A necessidade de conhecer as pessoas fisicamente é ofuscada pelas conversas online. Hoje em dia o telemóvel passou de um utensílio para uma necessidade básica, desde navegar pela Internet a pagar contas, o *smartphone* é o canivete suíço moderno. Apesar de apresentar múltiplas vantagens para o cidadão comum, a utilização exagerada do *smartphone* pode abstrair a pessoa do mundo real. Desde responder a mensagens enquanto se está sentado à mesa de refeição a falar com outras pessoas, via uma aplicação de *chat*, enquanto se está num café com os amigos, o *smartphone* trouxe consigo a cativante porta de facilidade de comunicação sem deslocação.

O objetivo deste trabalho é o desenvolvimento de um jogo para *smartphones*, com o objetivo de fazer deslocar os jogadores para um local físico de modo a promover o contacto social entre os jogadores. É utilizado o potencial do *smartphone* e o seu poder de computação para cativar o jogador a sair e explorar o mundo moderno através do seu ecrã. Através das mecânicas implementadas no jogo, são criados locais de interesse para todos os jogadores de maneira a criar um ponto de interesse comum. Assim, é possível juntar pessoas num mesmo local e simplificar o processo de contacto social físico através da troca de opiniões e experiências de jogo. O jogador tem à sua disposição a exploração do mundo em formato gráfico que apresenta monstros para captura e duelos para combater. A captura de monstros serve para completar a coleção e capturando o mesmo monstro um número exato de vezes, causa a evolução de nível deste aumentando os seus poderes. Para combater, o jogador precisa de configurar o seu esquadrão com um monstro de ataque, de defesa e de sabotagem dos seus monstros capturados. O combate de duelos recompensa o jogador com experiência e consequentemente a evolução de nível. Quando o jogador evolui de nível este ganha pontos de habilidades que servem para ativar ou melhorar as habilidades. As habilidades fornecem efeitos aos monstros do esquadrão e são utilizadas durante o combate múltiplas vezes. É também objetivo deste projeto a avaliação da experiência proporcionada pelo jogo ao jogador incluindo a usabilidade. A avaliação da experiência do jogador é realizada com testes com 14 jogadores e utilizando o *Game Experience Questionnaire* (GEQ) [10]. Para medir a usabilidade é utilizado o questionário, *System Usability Scale* (SUS) [29]. Este documento descreve o jogo, todo o processo de avaliação realizado e apresenta a análise aos resultados obtidos.

Palavras-chave: Contacto social, Mecânicas de Jogo, GUI, *Smartphone*, *Gameplay*.

Abstract

In a world where technology is increasingly prevalent in society, social apathy has emerged as the main cause of lack of interaction between people. The need to get to know people physically is overshadowed by online conversations. Nowadays the mobile phone has gone from being a tool to a basic necessity, from surfing the Internet to paying bills, the smartphone is the modern Swiss army knife. Despite having multiple advantages for the common citizen, the exaggerated use of the smartphone can abstract the person from the real world. From answering messages while sitting at the meal table to talking to other people, via a chat app, while at a cafe with friends, the smartphone brought with it the captivating ease of communication without having to travel.

The objective of this work is the development of a game for smartphones, with the objective of moving players to a physical location in order to promote social contact between players. It uses the potential of the smartphone and its computing power to captivate the player to go out and explore the modern world through its screen. Through the mechanics implemented in the game, places of interest are created for all players in order to create a common point of interest. Thus, it is possible to bring people together in the same place and simplify the process of physical social contact through the exchange of opinions and game experiences. The player has at his disposal the exploration of the world in a graphic format that presents monsters to capture and duels to fight. The capture of monsters serves to complete the collection and capturing the same monster an exact number of times causes its level to evolve by increasing its powers. To fight, the player needs to configure his squad with an attack, defense and sabotage monster of his captured monsters. Duel combat rewards the player with experience and consequently level evolution. When the player level up, he earns skill points that are used to activate or improve skills. Abilities provide effects to monsters in the squad and are used during combat multiple times. It is also the objective of this project to evaluate the experience provided by the game to the player, including usability. The evaluation of player experience is carried out with tests with 14 players and using *Game Experience Questionnaire* (GEQ) [10]. To measure usability is used the questionnaire, *System Usability Scale* (SUS) [29]. This document describes the game, the entire evaluation process carried out and presents the analysis of the results obtained.

Keywords: Social contact, Game Mechanics, GUI, *Smartphone*, *Gameplay*.

Índice

Lista de Figuras	xvii
Lista de Tabelas	xxi
Lista de Abreviaturas e Siglas	xxiii
Glossário	xxv
1 Introdução	1
1.1 Motivação	1
1.2 Definição do Problema	2
1.3 Análise de Jogos	2
1.4 Objetivos e Contribuições	2
1.5 SquadUp	3
1.6 Organização do Documento	3
2 Trabalho Relacionado	5
2.1 Jogos com um propósito	6
2.2 Jogos para <i>Smartphones</i>	6
2.2.1 Pokémon GO	7
2.2.2 Follow JC Go	10
2.3 Browser Games	12
2.3.1 Farmville	13
2.4 Aplicações Sociais	14

3 Solução Proposta	17
3.1 Objetivos do jogo	18
3.1.1 Objetivo Principal	18
3.1.2 Objetivos Secundários	19
3.2 Tecnologias para implementação da solução	19
3.3 Arquiteturas do jogo SquadUp	21
3.3.1 Arquitetura do cenário Autenticação	22
3.3.2 Arquitetura do cenário Mapa-mundo	23
3.3.3 Arquitetura do cenário Captura	26
3.3.4 Arquitetura do cenário Duelo	28
4 Interface Utilizador do Jogo	33
4.1 Permissões	34
4.2 Cenário Autenticação	35
4.2.1 Definições	36
4.2.2 Iniciar Sessão	36
4.2.3 Registo	37
4.3 Cenário Mapa-mundo	38
4.3.1 Monstros no Mapa-mundo	40
4.3.2 Duelos no Mapa-mundo	41
4.3.3 Tutoriais	42
4.3.4 Menu Principal	43
4.3.4.1 Menu Definições	44
4.3.4.2 Menu Bestiário	46
4.3.4.3 Menu Monstro	49
4.3.4.4 Menu Jogador	50
4.3.4.5 Menu Esquadrão	52
4.3.4.6 Menu Habilidades	54
4.3.4.7 Menu Linha de Habilidades	55
4.4 Cenário Captura	58
4.5 Cenário Duelo	61

5	Mecânicas do Jogo SquadUp	67
5.1	Mecânicas de Monstro	67
5.1.1	Tipo	68
5.1.2	Subtipo	69
5.1.3	Raridade	69
5.2	Mecânica de Habilidades	70
5.3	Mecânicas de Evolução	72
5.4	Mecânicas de Captura de Monstro	73
5.5	Mecânica de Duelo	73
6	Módulos Técnicos	75
6.1	Tutoriais	75
6.1.1	Funcionamento Programático dos Tutoriais	75
6.2	Mapa Mundo	76
6.3	Módulo de Habilidades	77
6.3.1	Estrutura	77
6.3.1.1	Componentes da Habilidade	80
6.3.1.2	Gestor de Habilidades	81
6.4	Módulo Monstro	82
6.4.1	Estrutura	83
6.4.1.1	Gestor	85
6.4.1.2	Gerador	86
6.5	Módulo Player	86
6.5.1	Estrutura	86
6.5.1.1	Esquadrão	88
6.6	Módulo Capture	88
6.6.1	Introdutor	88
6.6.2	Gestor	89
6.6.2.1	Auxiliares de Gestão	90
6.6.3	Encerrador	92
6.7	Módulo Duelo	92
6.7.1	Gestor	93
6.7.1.1	Gestor de Combatentes	95
6.7.1.2	Gestor de Esquadrão em Combate	96

7 Design Funcional	99
7.1 Painel de Habilidades	99
7.2 Painel de Monstro	101
7.3 Cabeçalho	103
7.3.1 Cabeçalho Padrão	104
7.3.2 Cabeçalho Incapacitado	104
7.3.3 Cabeçalho Contador	105
7.4 Menu Principal	105
8 Avaliação	107
8.1 Metodologia	107
8.2 Participantes	108
8.3 Questionário	108
8.4 Resultados	109
9 Conclusão	115
9.1 Trabalho Futuro	116
9.2 Considerações Finais	116
Bibliografia	117
A Respostas dos Participantes nos Testes de Usabilidade	i
A.1 Caracterização	i
A.2 Tarefas de Usabilidade	iv
A.3 SUS	v
B Avaliação da Experiência do Jogador	vii
B.1 Módulo <i>Core</i>	vii
B.2 Módulo <i>Post Game</i>	xi
C Arquivo de Protótipos	xiii
C.1 Protótipos de <i>layouts</i> de várias GUIs e HUDs	xiii

Lista de Figuras

2.1	Jogos para <i>smartphones</i> no dia a dia.	6
2.2	Transição de cenas na captura de um <i>Pokémon</i> no Pokémon GO.	7
2.3	Perfis e inventário do jogador no jogo Pokémon GO.	8
2.4	Contacto social promovido pelo agrupamento.	9
2.5	Exemplo do mapa-mundo gráfico e ginásio do Pokémon GO.	9
2.6	Personagens do jogo Follow JC Go e a sua agregação.	10
2.7	Perfil do jogador e os seus recursos no jogo Follow JC Go.	11
2.8	Referência a diversos jogos <i>online(browsers games)</i>	12
2.9	Demonstração das várias ações de jogabilidade no jogo Farmville.	13
2.10	Perfis, páginas e murais de noticias das redes sociais.	15
3.1	Cenários do jogo SquadUp.	17
3.2	Estrutura do jogo SquadUp e a arquitetura dos seus cenários.	21
3.3	Arquitetura do cenário Autenticação.	22
3.4	Composição de elemento Gestores GUI da arquitetura do cenário Autenticação.	23
3.5	Arquitetura do cenário Mapa-mundo.	24
3.6	Composição do GameManager.	25
3.7	Composição do elemento gestores GUI da arquitetura do cenário Mapa-mundo.	25
3.8	Arquitetura do cenário Captura.	26
3.9	Composição do elemento gestores GUI da arquitetura do cenário Captura.	27
3.10	Arquitetura do cenário Duelo.	28
3.11	Composição do elemento gestores GUI da arquitetura do cenário Duelo.	29
3.12	Composição do elemento gestores de batalha da arquitetura do cenário Duelo.	30

4.1	Mensagem para acesso à localização do <i>smartphone</i>	34
4.2	Exemplo de autenticação no jogo SquadUp.	35
4.3	Exemplo de registo no jogo SquadUp.	37
4.4	Exemplo de registo no jogo SquadUp.	38
4.5	Exemplos do mapa de jogo do SquadUp em localizações de Portugal.	39
4.6	Representação gráfica dos monstros no jogo SquadUp.	40
4.7	Representação gráfica de um Duelo no mapa de jogo.	41
4.8	Exemplo de vários tutoriais de várias partes do jogo SquadUp.	42
4.9	GUI do Menu Principal.	43
4.10	Demonstração do ícone do botão Mute.	44
4.11	Exemplo de alteração de idioma Definições.	45
4.12	GUI do Menu Bestiário.	46
4.13	Botões de filtragem por tipo de poder.	47
4.14	Botões de filtração por tipo de raridade.	47
4.15	Exemplo de painéis de monstro.	48
4.16	Variações do GUI do Menu Monstro.	49
4.17	GUI do Menu Jogador.	52
4.18	Tutorial e variações do GUI do Menu Esquadrão.	53
4.19	Tutorial e GUI do Menu Habilidades.	54
4.20	Variações do GUI do Menu Linha de Habilidades.	55
4.21	Cabeçalho do Menu Linha de Habilidades de Sabotagem.	56
4.22	GUI de uma Habilidade com apenas uma Habilidade Combo.	56
4.23	Composição da GUI do painel de Habilidade.	57
4.24	Tutorial do Cenário Captura.	58
4.25	Exemplificação das ações possíveis no cenário Captura.	59
4.26	GUI do Menu Resultado do Cenário Captura.	60
4.27	GUI dos Menus do Cenário Duelo.	61
4.28	Exemplo do painel de análise do Ataque e Defesa.	63
4.29	Exemplo do painel de análise de Resultado de Dano.	65
4.30	GUI dos Menus Resultado do Cenário Duelo.	66
5.1	Árvore de habilidades de Age of Empires I.	70
5.2	Árvore de habilidades de League of Legends.	71

5.3	Árvore de habilidades de League of Legends.	72
6.1	Estrutura de uma Habilidade.	78
6.2	Derivação da classe Ability pelas várias Habilidades.	80
6.3	Estrutura de AbilitiesContainer.	82
6.4	Estrutura da classe Monster.	83
6.5	Estrutura de MonsterPowers.	85
6.6	Estrutura de Player.	87
6.7	Estrutura de PlayerPreferences.	88
6.8	Estrutura de CaptureSceneManager.	90
6.9	Estrutura de OverrideOrb.	91
6.10	Classes responsáveis pelo preenchimento do painel Ataque e Defesa.	95
6.11	Estrutura de FightersManager e ligação com FightStages.	95
6.12	Estrutura de FightStages.	96
7.1	Evolução do design do painel de habilidade.	100
7.2	Exemplo de ergonomia das mãos.	100
7.3	Identificação dos elementos do Painel de Monstro.	101
7.4	Símbolos dos poderes do jogo SquadUp.	102
7.5	Texturas para identificação do subtipo de Monstro.	102
7.6	Cores para identificação da raridade de Monstro.	103
7.7	Exemplos das variações do cabeçalho de menu.	104
7.8	Exemplificação do Botão externo Ajuda.	105
8.1	Escala de Pontuação do SUS.	110
A.1	Análise do género da população do questionário.	i
A.2	Análise da idade da população do questionário.	ii
A.3	Análise da quantidade de tempo diário no <i>smartphone</i> , da população do questionário.	ii
A.4	Análise da quantidade de tempo diário a jogar no <i>smartphone</i> , da população do questionário.	iii
A.5	Análise do tempo diário a jogar em outras plataformas, da população do questionário.	iii
C.1	Desenhos dos vários <i>layouts</i> de organização do conteúdo do painel de habilidade.	xiii
C.2	Desenho de protótipos de <i>layout</i> do painel de habilidade.	xiv
C.3	Desenho de protótipo do menu Jogador.	xv

C.4	Desenho de protótipo final do layout do menu Jogador.	xvi
C.5	Desenho de protótipo do menu Bestiário.	xvii
C.6	Desenho de protótipo do menu Monstro.	xviii
C.7	Desenho de protótipo final do layout do menu Monstro.	xix
C.8	Desenho de protótipo do menu de resultado do cenário Captura.	xx
C.9	Desenho de protótipo final de layout para o HUD do cenário Duelo.	xxi
C.10	Desenho de protótipo do menu de ataque e defesa do cenário Duelo.	xxii
C.11	Desenho de protótipo do menu de resultado do cenário Duelo.	xxiii

Lista de Tabelas

8.1	Avaliação dos resultados referentes à secção Tarefas do questionário.	109
8.2	Avaliação dos componentes do módulo <i>Core</i>	111
8.3	Avaliação dos componentes do módulo <i>Post Game</i>	112
A.1	Avaliação dos resultados referentes à secção Tarefas do questionário.	iv
A.2	Avaliação dos resultados referentes à secção SUS do questionário.	v
A.3	Análise e pontuação do SUS.	v
B.1	Avaliação do componente Competence do módulo <i>Core</i>	vii
B.2	Avaliação do componente Sensory do módulo <i>Core</i>	viii
B.3	Avaliação do componente Flow do módulo <i>Core</i>	viii
B.4	Avaliação do componente Tension do módulo <i>Core</i>	ix
B.5	Avaliação do componente Challenge do módulo <i>Core</i>	ix
B.6	Avaliação do componente Negative Affect do módulo <i>Core</i>	x
B.7	Avaliação do componente Positive Affect do módulo <i>Core</i>	x
B.8	Avaliação do componente Positive Experience do módulo <i>Post Game</i>	xi
B.9	Avaliação do componente Negative Experience do módulo <i>Post Game</i>	xi
B.10	Avaliação do componente Tiredness do módulo <i>Post Game</i>	xii
B.11	Avaliação do componente Returning to Reality do módulo <i>Post Game</i>	xii

Lista de Abreviaturas e Siglas

API	<i>Application Programming Interface.</i> 20
GEQ	<i>Game Experience Questionnaire.</i> ix, xi, 108
GUI	<i>Graphical User Interface.</i> ix, xi, xvi, xvii, xviii, 18, 23, 25, 26, 27, 28, 29, 33, 38, 42, 43, 44, 46, 49, 51, 52, 53, 54, 55, 56, 57, 59, 60, 61, 65, 66, 82, 86, 88, 89, xiii
HUD	<i>Heads-up display.</i> xvi, xx, 19, 28, xiii, xxi
IDE	<i>Integrated development environment.</i> 19, 20
ISEL	Instituto Superior de Engenharia de Lisboa. 8, 9, 15
MOBA	<i>Multiplayer Online Battle Arena.</i> 2
RPG	<i>Role Playing Game.</i> 2
RTS	<i>Real Time Strategy.</i> 2
SDK	<i>Software Development Kit.</i> 20
SUMI	<i>Software Usability Measurement Inventory.</i> 108
SUS	<i>System Usability Scale.</i> ix, xi, xix, xxi, 108, 110, 115, v

Glossário

- combo** Uma combinação de ações que perfazem uma ação conjunta..
56, 57, 78, 79, 80, 81, 100, 105
- jogabilidade** A maneira específica de como os jogadores interagem com o
jogo.. xvii, 3, 4, 5, 7, 8, 9, 11, 13, 14, 16, 18, 20, 23, 26, 107

1

Introdução

Este capítulo apresenta a motivação e introduz o jogo desenvolvido para aproximar pessoas e a análise realizada.

1.1 Motivação

No mundo moderno, o cidadão comum tem cada vez mais tarefas a desempenhar no seu dia-a-dia e cada vez menos tempo pessoal. A carga horária de trabalho estende-se para além da definida seguindo a pessoa para dentro do seu lar. Com o escalar de trabalho o cidadão deparou-se com a necessidade de aumentar o seu desempenho e conseqüentemente equilibrar a sua distribuição de tempo efetivo/lazer.

Com o aumento de pedidos por tecnologias para a assistência da resolução de problemas e comunicação, o mercado dirigiu a sua atenção e evolui várias tecnologias constantes na vida da pessoa. Sendo estas o computador e os dispositivos móveis. Destes aparelhos, o *smartphone* teve uma evolução escalar devido à sua integração na comunicação social das pessoas e a possibilidade de tratar de várias tarefas com o mínimo esforço possível. Isto permitiu que as pessoas conseguissem aumentar a sua produtividade na sua área de trabalho e pessoal de uma maneira *multitasking*. Infelizmente com a adaptação das novas tecnologias e redes sociais para comunicação, as pessoas ficaram cada vez mais abstraídas da socialização pessoal e física, levando a um aproximar digital e afastamento pessoal.

Com este conceito de comunicação digital integrado desde raiz, as novas gerações tornaram-se especialmente abstraídas do mundo exterior criando problemas na interação de pessoa para pessoa. A transmissão de pensamentos, desabafos e sentimentos foram substituídas por *gifs*, *emojis* e abreviações de frases, levando a comunicação pessoal a um nível de insensibilidade devido à falta de "cara à cara".

Observando este problema como uma constante diária, decidiu-se usar as novas tecnologias como leme de navegação para um foco na socialização física entre as pessoas.

1.2 Definição do Problema

Na escola, na universidade, no trabalho, muitas tarefas são realizadas por várias pessoas de forma colaborativa. No entanto, o relacionamento entre as pessoas nem sempre é o melhor. Vários estudos mostram que os jogos sociais (e.g., Pokémon GO) em muitas ocasiões ajudam a aproximar pessoas em alguns contextos (e.g., vizinhos, pessoas que trabalham na mesma empresa ou na universidade) [16].

1.3 Análise de Jogos

A proposta baseia-se na análise das características de vários jogos sociais, principalmente no jogo Pokémon GO devido à sua imensa popularidade e sucesso como jogo social. Apesar de o jogo Pokémon GO ter sido usado como base de criação da proposta, analisaram-se mais jogos para que houvesse uma diversidade em mecânica de jogo, funcionalidade, apresentação e consequentemente anexando uma unicidade ao jogo por este partilhar tantos estilos diversos.

Os estilos de jogos usados foram:

- *Real Time Strategy* (RTS);
- *Role Playing Game* (RPG);
- *Multiplayer Online Battle Arena* (MOBA).

As mecânicas de jogo usadas pelos estilos de jogo analisados são cruciais para a criação da mecânica de jogo da proposta. O jogo *Age Of Empires* [1] apresenta ao jogador uma árvore tecnológica que permite evoluir a sua civilização. Para este fim, o jogador precisa de aceder à árvore, selecionar a tecnologia e gastar os recursos necessários para a sua pesquisa. Esta mecânica foi utilizada em combinação com a antiga mecânica de mestria do jogo *League of Legends* [17].

A mecânica de mestria disponibiliza ao jogador três linhas baseadas nas categorias "Ataque", "Defesa", "Utilidade" e cada linha é estruturada por várias habilidades. Estas habilidades concedem melhorias aos poderes/características base da personagem do jogador assim como novas capacidades. Assim, foram utilizadas estas duas mecânicas combinadas para desenvolver a mecânica de habilidades do jogo SquadUp, proporcionando ao jogo mais profundidade em termos de detalhe e ao jogador mais conteúdo de entretenimento.

1.4 Objetivos e Contribuições

O objetivo deste trabalho é desenvolver um jogo para *smartphones* que possa aproximar as pessoas e ajude a criar melhores relações entre amigos, colegas de trabalho ou até vizinhos. É um jogo social, mas com o propósito de ajudar a aproximar pessoas [9].

Para desenvolver um jogo para *smartphones* com o objetivo de promover a socialização entre pessoas que não se conhecem, é utilizada a estratégia de ligação de desafios virtuais a pontos geográficos reais de modo a proporcionar a deslocação a esses locais e possibilitar/facilitar a interação entre jogadores no mesmo local físico. Criando uma ligação entre dois mundos e proporcionando ao jogador uma nova

experiência de jogabilidade, ou seja, o jogador irá encarnar a sua personagem na vida real. Para que a personagem consiga deslocar-se para os objetivos no mundo virtual é necessário que o jogador se desloque também no mundo real. Este tipo de experiência origina uma ligação única entre o jogador e a sua personagem e consequentemente uma experiência única.

Assim, este projecto apresenta as seguintes contribuições:

1. Jogo SquadUp;
2. Avaliação do Jogo SquadUp;
3. Discussão dos resultados obtidos.

1.5 SquadUp

O jogo desenvolvido neste projeto para *smartphones* com sistema operativo *Android* é designado por SquadUp. No jogo SquadUp o jogador precisa de se deslocar a locais físicos para superar desafios de modo a fazer evoluir a sua personagem e a melhorar as qualidades do seu esquadrão. O esquadrão é composto por três monstros com habilidades que ajudam a superar desafios virtuais em locais físicos específicos. É possível que vários jogadores no mesmo local físico possam interagir. É nesta fase que a componente social do jogo é utilizada para promover a comunicação/socialização. O jogo proposto apresenta as seguintes características principais:

1. Combinação/mistura entre o mundo virtual do jogo com o mundo real;
2. Jogabilidade ao nível da simplicidade de um jogo casual;
3. A evolução do jogador no jogo influencia diretamente os seus monstros;
4. Promove a interação entre jogadores que se deslocam pelo mundo real enquanto jogam no mundo virtual. Ao combinar os dois mundos o jogador experiência um estilo diferente de jogabilidade;
5. Combinação de monstros variada e as suas características afetadas diretamente pelas habilidades pertencentes ao jogador;

Para observação da solução implementada ver [Jogo SquadUp \[11\]](#), esta fonte também disponibiliza a transferência e um tutorial para implementação nas plataformas de desenvolvimento utilizadas.

1.6 Organização do Documento

Para além deste capítulo de introdução, a dissertação é composta por mais oito capítulos, tendo no total nove:

Capítulo 2: Trabalho Relacionado - Capítulo de enquadramento, onde são apresentados vários jogos e aplicações com características que suportam o conceito de um jogo social.

Capítulo 3: Solução Proposta - Neste capítulo é apresentada a arquitetura da solução proposta e os objetivos do jogo.

Capítulo 4: Interface Utilizador do Jogo - Neste capítulo são apresentadas as várias interfaces gráficas de utilizador que compõem o jogo. A constituição destas, o seu funcionamento e razão de existência.

Capítulo 5: Mecânicas do Jogo SquadUp - Neste capítulo são apresentadas as várias mecânicas de jogo que esboçam as regras e jogabilidade do jogo. Assim como as sinergias que estas geram entre si e o que implicam para o jogador.

Capítulo 6: Módulos Técnicos - Neste capítulo são apresentados os módulos técnicos mais importantes do jogo e a sua contribuição para a estrutura programática do jogo.

Capítulo 7: Design Funcional - Neste capítulo são apresentados os vários conceitos criados em torno dos *designs* para vários elementos gráficos do jogo. São explicados a sua razão de existência e o motivo da aplicação destes.

Capítulo 8: Avaliação - Neste capítulo são apresentados as análises dos dados resultantes do questionário aos utilizadores e o significado que estes resultados implicam na estrutura e implementação do jogo.

Capítulo 9: Conclusão - Neste capítulo são apresentadas as conclusões sobre os resultados obtidos da análise de dados posteriormente feita e melhorias a explorar em trabalhos futuros.

2

Trabalho Relacionado

Para melhor enquadramento do jogo social proposto serão apresentados vários jogos e aplicações relacionadas com as características do jogo que suportam o conceito de aproximar e promovem a socialização de pessoas que não se conheciam. O modelo dos jogos apresentados contém vários conceitos de psicologia social [7] introduzidos na jogabilidade que promovem a envolvência do jogador. Ou seja, os jogos possuem vários elementos tanto diretos como indiretos que influenciam/cativam o jogador.

Estes criam uma relação de necessidade/envolvimento exponencialmente crescente, do jogador para com o jogo em si, o que pode ser constatado como um fator negativo pois implica vício, mas ao mesmo tempo um fator positivo quando o jogo promove o envolvimento social do jogador para com outras pessoas (ver Figura 2.1). Todos estes conceitos e o modelo que os gera são demonstrados através de vários jogos de diversas plataformas em seguida.

O jogo proposto pode ser considerado um jogo social, na medida em que promove a socialização [20]. No entanto, também pode ser considerado um jogo baseado na localização (*Location-based Game*) [19] uma vez que a evolução do jogo está dependente da localização, ou neste caso, da deslocação física de jogadores a determinados locais. Neste contexto é um jogo baseado em localização com um propósito [8], aproximar pessoas.

Na secção seguinte são apresentados alguns jogos com um propósito relacionados com os objetivos do jogo proposto. Na secção seguinte são apresentados os jogos para *smartphones* com ênfase nos jogos baseados em localização. O capítulo termina com os jogos online e com as aplicações sociais que pela sua natureza promovem socialização à distância.

2.1 Jogos com um propósito

Os jogos digitais não são apenas ferramentas de entretenimento. Vários jogos que têm sido propostos, foram desenvolvidos com outras finalidades além de serem apenas jogos. Alguns dos exemplos são os jogos sérios para saúde [24] ou educação [25], jogos para resolver problemas científicos [31], ou mesmo aplicações para turismo ou cultura com características de gamificação [15].

Em relação aos jogos com o propósito de aproximar pessoas, os trabalhos propostos em [23, 26, 33] usaram jogos para ajudar a aproximar e manter a ligação entre pessoas. Essas propostas baseiam-se na aproximação das pessoas de uma comunidade. Nas propostas [26] e [23], o contexto utilizado foi baseado em comunidades acadêmicas por meio de estudos em escolas ou em conferências. Em um contexto de negócios para aumentar a confiança de colegas que trabalham na mesma empresa, mas em locais diferentes, o estudo realizado em [33] mostra que os jogos sociais são melhores do que os quebra-gelos sociais nesta tarefa. Essas propostas foram desenvolvidas e avaliadas em ambientes que favorecem as relações entre as pessoas. Em uma configuração semelhante, mas sem usar um jogo, em [13] foi proposto o *Opinionizer* que se baseava em exibições públicas localizadas em eventos sociais como festas. Esta aplicação encoraja as pessoas na multidão a colocarem os seus comentários e opiniões em um espaço social virtual (exibição pública) e essa ação permite que elas iniciem uma conversa com pessoas localizadas em volta. Novamente, o estudo foi conduzido em um contexto de conferência científica.

2.2 Jogos para Smartphones



Figura 2.1: Jogos para *smartphones* no dia a dia.

Os jogos apresentados baseiam-se na realidade virtual/realidade aumentada utilizando o sistema de posicionamento global/geolocalização para mover a personagem do jogador no ambiente de jogo. Este tipo de movimento é idêntico ao movimento do cursor que identifica a posição do utilizador de uma

aplicação de navegação, ou seja, o utilizador ao mover-se irá fazer com que o cursor se mova também para a posição atual do utilizador.

Este tipo de jogabilidade consegue transmitir a sensação de realismo ao jogador visto que o ambiente de jogo é um sítio real trazendo consigo duas vantagens:

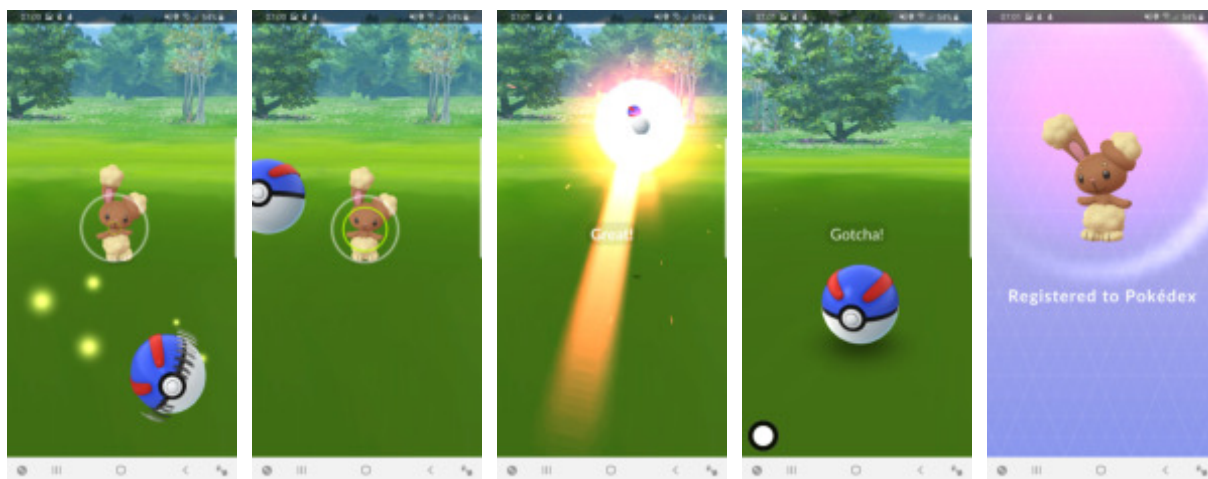
1. Não é necessário criar nenhum ambiente de jogo reduzindo o custo/esforço laboral;
2. O jogador sente-se mais envolvido pessoalmente devido às localizações reais estabelecendo uma relação mais pessoal com a sua personagem.

Apesar do mercado de jogos para *smartphones* ser recente, os jogos deste, oferecem estilos de jogabilidade bastante conhecidos e utilizados em outras plataformas que não aproveitam as várias características que um dispositivo móvel oferece, sem ser o poder computacional e os comandos através de toque. Assim o conceito de um jogo utilizar a localização em tempo real do dispositivo móvel e combinar a transposição do ambiente real para uma versão gráfica deste, introduziu uma onda de inovação há muito esperada num mercado estagnado.

Vários jogos que utilizaram este conceito com base em vários temas são apresentados e explorados em detalhe, em seguida.

2.2.1 Pokémon GO

Um dos jogos de sucesso com as características anteriores foi o Pokémon GO [21, 22]. No Pokémon GO o jogador é encarado como um treinador de criaturas *Pokémon* e o seu objetivo é apanha-los todos. Para isto o jogador necessita de explorar o mundo à sua volta para encontrar os *Pokémons* e ao estabelecer contacto este irá utilizar uma *Pokébola* para o capturar. A Figura 2.2 mostra algumas imagens que ilustram o jogo para ação de captura de um *Pokémon*.



(a) Agarrar a *Pokébola*. (b) Atirar a *Pokébola*. (c) Acertar a *Pokébola*. (d) Captura Sucedida. (e) Registo.

Figura 2.2: Transição de cenas na captura de um *Pokémon* no Pokémon GO.

Após alguma evolução em termos de experiência do jogador, ditada a partir de níveis, este é inquirido sobre qual a equipa a que deseja pertencer, sendo estas três e distinguidas pelas suas cores e o *Pokémon* que cada uma ostenta. Tanto o jogador (ver Figura 2.3a) como qualquer *Pokémon* evoluem e ganham

novas características. É possível observar na Figura 2.3b que o *Pokémon Pikachu* pode tanto evoluir na sua força, como para um novo *Pokémon* como aprender um novo ataque.

O jogador possui uma mochila para armazenar os seus recursos, desde vários tipos de Pokébolas, em que cada uma possui um nível distinto de maior probabilidade de captura do *Pokémon*, a vários tipos de alimentação que ajudam o jogador a diminuir a resistência de captura do *Pokémon* e poções para recuperar a saúde dos *Pokémons* do jogador. Estes recursos são demonstrados na Figura 2.3c e Figura 2.3d.

O *Pokémon GO* promove o contacto social devido às várias características de jogabilidade competitiva, mas também ao exercício físico do jogador, devido à necessidade de se deslocar para ir de encontro com novos *Pokémons*.

Um dos pontos principais no foco social são os locais designados de *Pokéstops*(ver Figura 2.5d). Estes locais são utilizados pelos jogadores para receber recursos, como por exemplo *Pokébolas*, mas também para utilizar um item dedicado a aumentar a probabilidade de aparecimento de *Pokémons* naquele local. Este conceito fez com que muitos jogadores se reunissem num local destes para cumprir o objetivo principal do jogo, mas também promoveu o contacto social devido à aglomeração de jogadores no mesmo local.



(a) Perfil do jogador.

(b) Perfil de um Pokémon.

(c) *Items* para captura.(d) *Items* para cura.

Figura 2.3: Perfis e inventário do jogador no jogo *Pokémon GO*.

É bastante comum observar determinados sítios possuírem vários *Pokéstops* seguidos, como por exemplo um paredão e consequentemente um grande foco de pessoas situadas em redor desses *Pokéstops* ou como podemos observar na Figura 2.4 um grupo de pessoas a jogarem perto do ginásio localizado na entrada principal do ISEL.

O ginásio no *Pokémon GO* fornece lutas entre jogadores entre os seus *Pokémons*. Os ginásios são locais disputados por jogadores de equipas contrárias, quando os jogadores derrotam os oponentes que ocupam o ginásio com os seus *Pokémons* estes podem habitar o ginásio cada um com um *Pokémon* para reclamar o ginásio para a sua equipa e defende-lo.

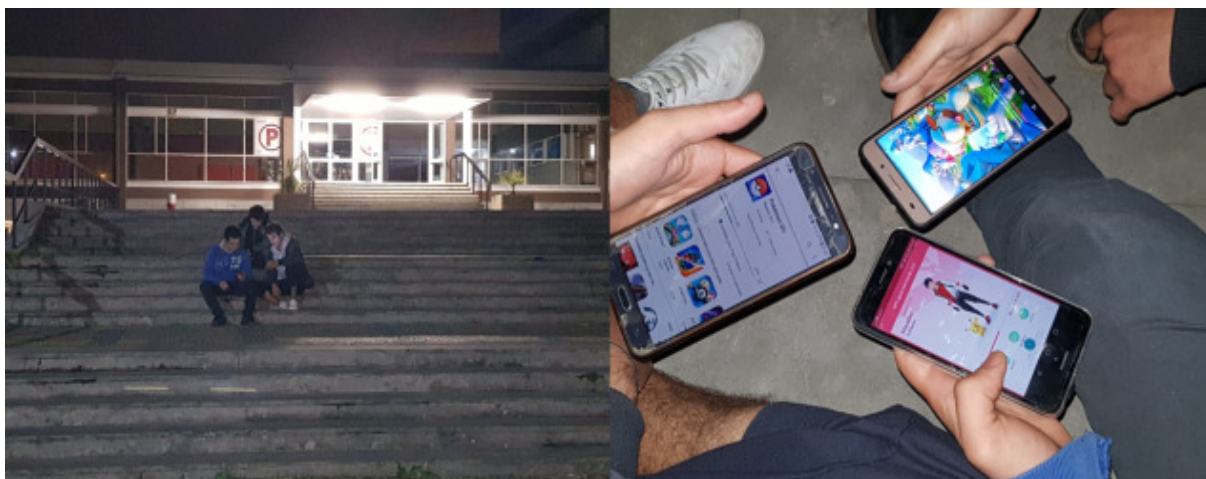
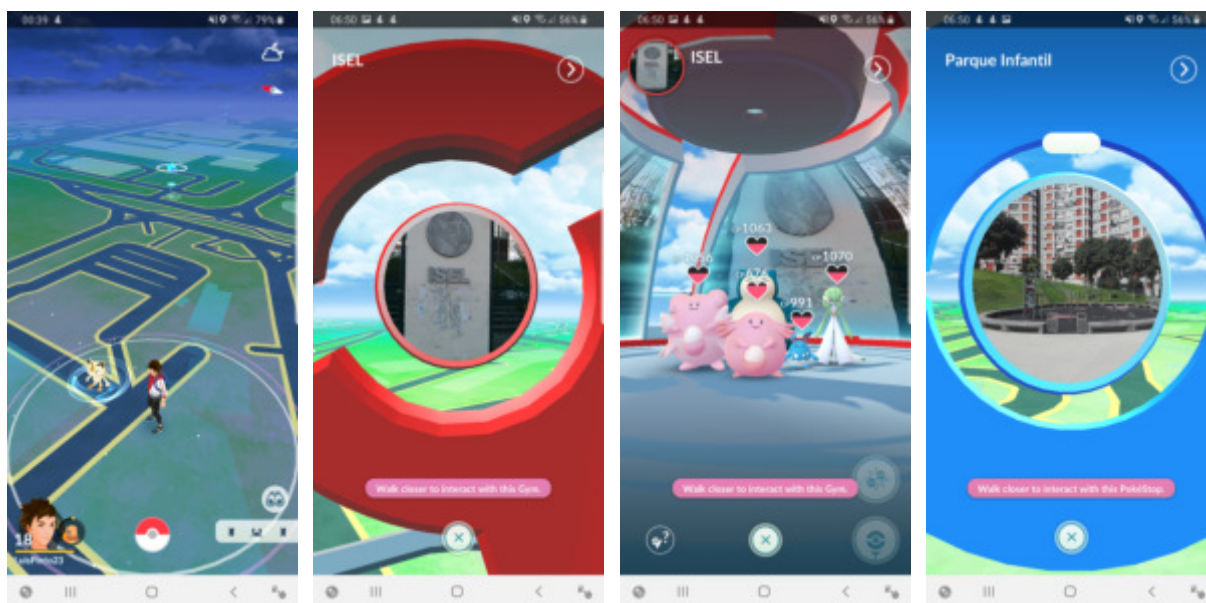


Figura 2.4: Contacto social promovido pelo agrupamento.

Por cada dia que os jogadores defendem o seu ginásio, estes recebem dinheiro virtual que pode ser usado para comprar recursos. Pode-se observar na Figura 2.5 que existe um ginásio, localizado na entrada principal do ISEL, a ser dominado por jogadores da equipa vermelha.



(a) ISEL em mapa gráfico. (b) Página do ginásio ISEL. (c) Conteúdo do ginásio. (d) Pokéstop.

Figura 2.5: Exemplo do mapa-mundo gráfico e ginásio do Pokémon GO.

O contacto social entre jogadores desconhecidos uns dos outros foi de nível elevado, mesmo que este fosse mínimo o diálogo era estabelecido. Desde à comparação da coleção de cada jogador até à combinação de estratégias para derrotar um ginásio de uma equipa adversária, o contacto social era estabelecido com muito menor dificuldade devido à aproximação física de ambos os jogadores motivados pelos objetivos/características do jogo. Este tipo de jogabilidade é o que possui mais ênfase na construção do jogo proposto, devido à facilidade em promover o contacto social assim como a facilidade que promove em diversificar o sistema de jogo em si, pois em vez de ser só um sítio de recolha de recursos pode ser um sítio de confronto entre jogadores ou um sítio de confronto entre o jogador e um inimigo virtual.

2.2.2 Follow JC Go

O Follow JC Go [5] é outro jogo com características interessantes. Tem como base a educação religiosa sendo que o seu objetivo consiste em procurar personagens históricas da bíblia, vocações marianas, santos, beatos e pessoas reais espalhadas pelo mapa de jogo.



(a) Personagem Bíblica no mapa. (b) Personagem Mariana no mapa. (c) Coleção de personagens Bíblicas. (d) Coleção de personagens Marianas. (e) Agregação de personagem à equipa.

Figura 2.6: Personagens do jogo Follow JC Go e a sua agregação.

O objetivo principal de jogo é possuir a maior pontuação pessoal, da equipa de evangelização e do país a que pertence, mas o principal objetivo dos criadores é promover o Cristianismo através do diálogo/interação com as pessoas e consequentemente das pessoas para com outras pessoas.

Quando o jogador encontra alguém do lote de procura este tem que convidar a pessoa para ingressar na sua equipa de evangelização. A ingressão da personagem é conseguida com sucesso respondendo a uma série de questões à cerca da personagem fazendo com que o jogador adquira conhecimento sobre esta, caso não o tenha. Este método não é aplicado às pessoas reais, ou seja, outros jogadores. Neste caso ambos os jogadores precisam de dialogar e no final decidirem, se querem fazer parte ou não da equipa de evangelização do jogador contrário.

Como é possível observar, o jogo baseia-se muito em ensinar matéria religiosa ao jogador, mas também se foca no fator social ao fazer com que o jogador tenha que dialogar com outros jogadores para realizar o objetivo principal.

Cada jogador compete com a sua equipa evangélica contra os restantes, sendo que para isto o jogador terá que recrutar mais jogadores ou até mesmo pessoas que não joguem o jogo.

No ponto de vista social é um passo extremamente importante, pois o jogador tendo um objetivo em mente consegue dialogar com novas pessoas sem ficar retido por alguma incapacidade social, tanto dele como da pessoa para com quem ele se está a dirigir.

Apesar de não ser um objetivo do jogo, este em si é um objetivo principal do fator social, ou seja, para quem ambiciona desenhar e criar um jogo que promova o diálogo necessita de ter em mente que as pessoas têm de se sentir abstraídas dos seus problemas ou inseguranças para que tal aconteça. Sendo através dos objetivos requeridos pelo jogo uma das várias maneiras possíveis.

Cada jogador possui quatro recursos que necessita de gerir sendo as seguintes:

1. Espiritualidade (ver Figura 2.7d);
2. Pão (ver Figura 2.7b);
3. Água;
4. Moedas de prata (ver Figura 2.7c).

Todos estes elementos são exemplificados na Figura 2.7 que demonstra que o jogador necessita de se deslocar aos locais marcados com os recursos e em seguida compra-los com moedas de prata. Este conceito introduz uma micro-economia no jogo implicando que o jogador necessita de gerir todos os seus recursos com mais cuidado devido ao seu acesso ser limitado. Este conceito é bastante comum nos jogos cuja jogabilidade não é complexa mas para compensar a simplicidade de jogo aplica um mecanismo de dependência ao jogador.

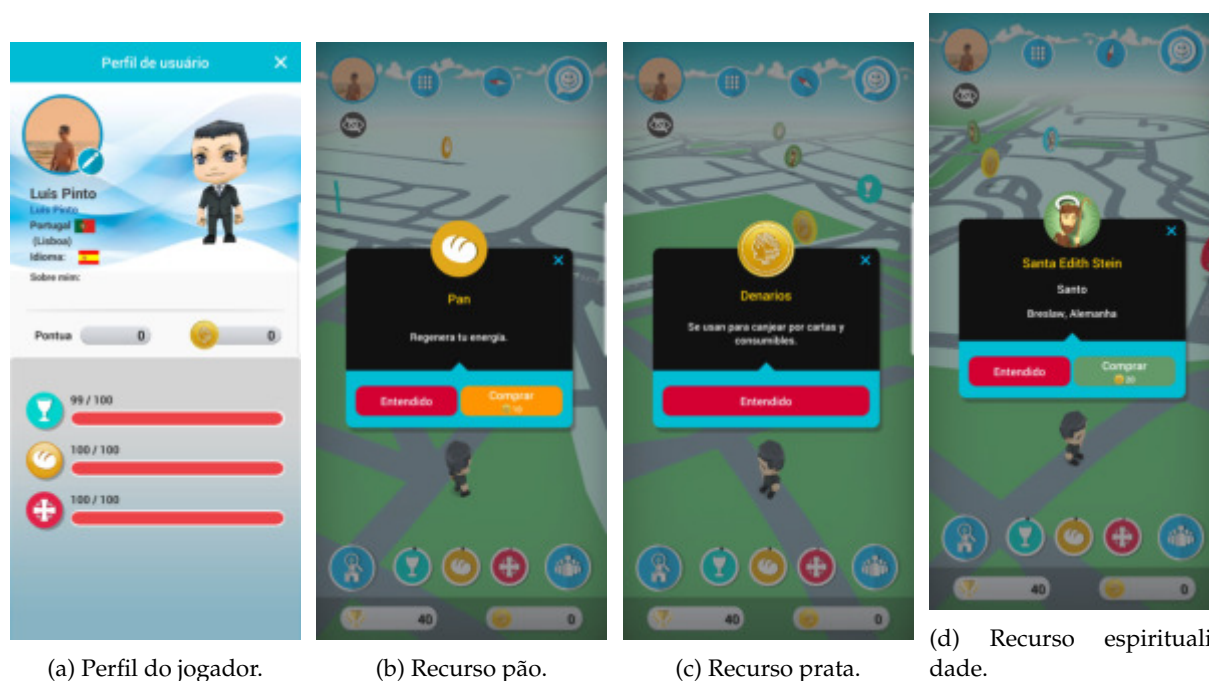


Figura 2.7: Perfil do jogador e os seus recursos no jogo Follow JC Go.

O jogador necessita de encontrar os locais que disponibilizam estes recursos para sobreviver. Os três primeiros recursos aumentam a pontuação pessoal, da equipa e do país, enquanto o último recurso o jogador pode ser usado quando lhe é pedido pelo jogo.

Estes elementos servem para que o jogador tenha a sensação/necessidade de voltar ao jogo para gerir a personagem. Este é um fator clássico nos jogos dedicados aos *smartphones* para manter a continuidade/participação do jogador no jogo. Devido ao jogador ter que voltar ao jogo intervaladamente para que o seu personagem não faleça é imposto sobre ele a constante preocupação, logo o jogo passa a fazer parte do dia a dia do jogador.

Estes tipos de estratégias de aliciamento ao jogador já eram usados muito antes de haver jogos para *smartphones*, os melhores exemplos e os mais parecidos com estes são os famosos “*browsers games*” (ver Figura 2.8).

2.3 Browser Games

Os jogos *online* ou *browser games* são jogos que adquiriram a sua fama através de vários fatores:

1. Simplicidade;
2. Gratuito;
3. Acessível em termos:
 - (a) Poder de processamento computacional;
 - (b) Utilizar um *website* dedicado, não implicando transferência/instalação de software.
4. Promovido pelos seus jogadores devido às razões acima referidas.

Estes jogos tornaram-se bastante famosos devido a estes fatores, conseguindo uma enorme fama e adesão pela comunidade jovem que apesar de já ter algum acesso a computadores, não possuía um computador pessoal ou o seu computador não oferecia poder de processamento computacional para jogos com gráficos avançados. Os jogadores podiam criar "clãs" e recrutar outros jogadores para os seus clãs. Um clã é uma equipa com uma estrutura de comando implementada para organização/liderança deste. Com este conceito os *browser games* disponibilizaram aos jogadores um meio de competição entre estes e conseqüentemente uma maior dependência para com o jogo. O jogador dirigia-se a uma página web



Figura 2.8: Referência a diversos jogos *online*(*browsers games*).

dedicada, registava-se no jogo e eram-lhe introduzidas as várias regras do jogo. As regras não são nem extensas nem complexas e a dificuldade de realizar ações no jogo é de grau baixo. O que permite que o jogador obtenha uma satisfação rápida para com as suas ações no jogo. Este tipo de experiência de jogo tem muito sucesso nos jogadores que possuem uma vida diária atarefada/complexa, logo a realização das suas ações sem quaisquer dificuldades apresentadas permite que o jogador se sinta realizado e conseqüentemente aumente a sua moral para consigo próprio. Mas à medida que o jogador vai jogando as várias ações que este realiza com facilidade tem um período de conclusão cada vez maior e é aqui que o jogo irá apresentar ao jogador a possibilidade de retirar esta dificuldade através de consumíveis digitais via a loja interna do jogo.

2.3.1 Farmville

Em relação aos jogos online ou os jogos de rede sociais, o jogo Farmville [5] foi um dos jogos mais marcantes. Baseia-se em gerir uma quinta, isto é feito através da cultivação de vegetais/frutos e também do tratamento de animais. Existem dois tipos de dinheiro virtual no jogo, um que é dado através da execução de tarefas/outras ações designadas de *Farm Coins* e outro dinheiro virtual que só pode ser adquirido com dinheiro real via a loja virtual designado de *Farm Cash*.

O jogador necessita de gastar dinheiro virtual (qualquer um servirá) para preparar a terra para plantação após isto o jogador planta e espera o tempo necessário para a colheita. O jogador vai evoluindo de nível de experiência ao perfazendo várias ações e ficando assim com acesso a mais *items*. Existem também outros *items* que o jogador pode adquirir no mercado do jogo através do seu dinheiro.

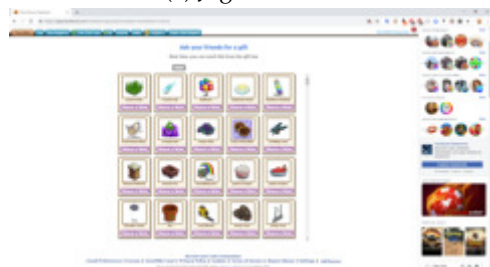
Este jogo em si não possui nenhuma dificuldade como já mencionado acima, em respeito aos jogos de navegadores web, mas devido a ser um jogo da rede social Facebook conseguiu promover grande um contacto social online.



(a) Jogabilidade.



(b) Utilização das redes sociais.



(c) Interação com amigos para recursos.



(d) Ajuda de amigos com recursos.

Figura 2.9: Demonstração das várias ações de jogabilidade no jogo Farmville.

Ao possibilitar que o jogador melhore a sua quinta com maior prontidão através de convites, solicitando a ajuda de outros jogadores ou a receber recompensas ajudando-os. O Farmville tornou o contacto social entre o jogador para com outras pessoas constante.

Promove a extensão da lista de amizade devido à necessidade de o jogador procurar e adicionar mais pessoas à sua lista de amizade para conseguir solicitar mais ajuda no jogo. O jogo mais tarde permitia que o jogador adicionasse outros jogadores como amigos dentro do jogo mesmo que estes não fossem na rede social.

Toda esta atividade é feita com o objetivo de aumentar o número de recompensas/ajuda, mas também aumentou o contacto social. Apesar do diálogo entre o jogador com outras pessoas fosse dado com interesses secundários não deixava de ser diálogo social. Juntando ao fato de se utilizar uma das redes

sociais mais famosas do mundo que ainda crescia exponencialmente, este jogo pode ser encarado como um grande promotor social da era digital.

Mais tarde foi substituído pelo Farmville 2 [4], mas também teve concorrência de vários outros jogos construídos sobre o mesmo modelo de jogabilidade e interação do jogador para com a sua lista de amigos.

2.4 Aplicações Sociais

A interação entre pessoas ocorre todos os dias de maneira mundana. Este tipo de contacto social dá-se devido às necessidades das pessoas, como por exemplo uma pessoa que trabalhe em um supermercado irá ser abordado várias vezes ao dia por clientes que necessitam de informações sobre a localização de produtos, se existe um produto em específico e outros casos referentes ao assunto. Mas este tipo de interação social não está ao nível de aproximar pessoas, logo apesar do dia-a-dia estar repleto de oportunidades, estas não são utilizadas devido a um número de fatores de constrangimento/limitações temporais.

É aqui que entram as aplicações sociais. Estas aplicações servem para estabelecer contacto social entre pessoas através de vários meios desde a troca de mensagens até à troca de imagens. Podemos até dizer que este tipo de comunicação tenha começado com a troca de mensagens de texto entre contactos telefónicos, mas este tipo de comunicação é muito restringido devido a não possibilitar a pessoa de poder visualizar nada mais a não ser o texto que recebe e que escreve.

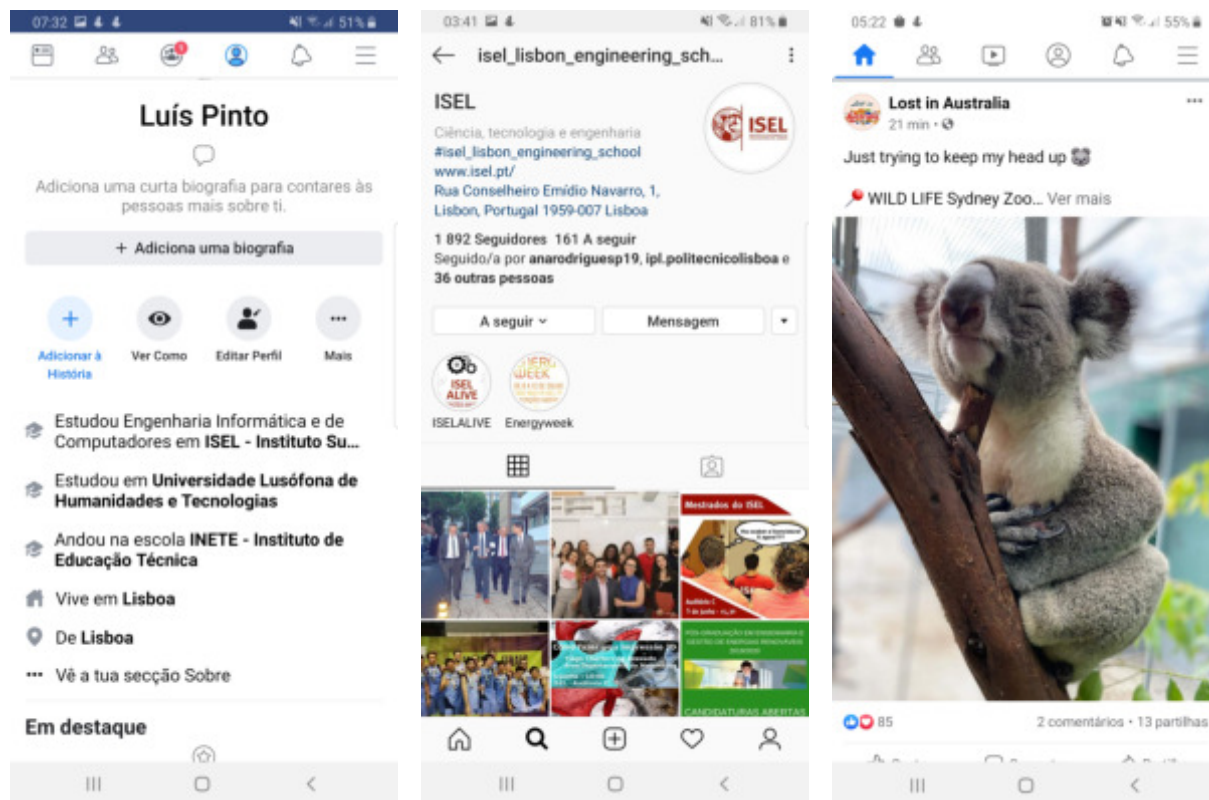
A vantagem das aplicações em comparação aos meios tradicionais de conversação é que estas permitem a exposição de características pessoais tanto em texto como visualmente tendo sempre em respeito o nível de privacidade que o utilizador selecionou. Este tipo de traços pessoais podem ser dos mais básicos como a data de nascimento, sítio onde a pessoa trabalha/estuda até aos mais pessoais como os interesses/gostos da pessoa e a aparência visual desta. A importância de um utilizador conseguir visualizar o aspeto físico de outro é uma das características mais cobijadas pelas aplicações de redes sociais de tal maneira que a aplicação Instagram baseia-se em representar o utilizador via imagens e não via informação textual.

As aplicações Facebook, Instagram, Snapchat, Twitter [6, 14, 27, 34] abriram muitas portas de comunicação social entre pessoas com pessoas assim como entre entidades e pessoas. A importância destas é de tal forma que qualquer empresa, seja qual for o seu nível de importância, necessita de estar representada em pelo menos uma das mencionadas acima.

Isto deve-se ao facto de que as pessoas focam cada vez mais a sua atenção nas redes sociais como sala de conversa, mas também como um sítio onde se podem atualizar a nível cultural e conhecimento geral.

As gerações mais novas substituíram o ritual de ligar a televisão para informação/entretenimento pelo ritual de aceder à aplicação de rede social para conseguir a mesma informação, ou seja, o objetivo principal é o mesmo, mas a plataforma é que muda. Apesar de tudo é possível as tecnologias coexistirem umas com as outras, pois o rádio apesar de ser substituído pela televisão continuou a existir a ser usado, não tanto nos domicílios, mas mais nas viaturas de transporte pessoal.

Apesar de haver tantas barreiras na interação social quebradas, continuou a haver um certo constrangimento entre as pessoas e o mais importante a falta de privacidade que estas geraram. É aqui que entram



(a) Perfil no Facebook.

(b) Página da entidade ISEL.

(c) News feed do Facebook.

Figura 2.10: Perfis, páginas e murais de notícias das redes sociais.

as aplicações de interação social com o propósito de encontros pessoais, pois não há nada mais pessoal que uma pessoa revelar determinadas características pessoais. Logo a privacidade é um fator de alta importância.

O Tinder, Grindr, Bumble [2, 12, 32] são algumas das múltiplas aplicações de encontros existentes e estas aplicações focam-se em quebrar as barreiras de constrangimento de interação social através de mecanismos tanto psicológicos como simplificadores. O Tinder baseia-se em extrair informação do perfil de Facebook do utilizador para construção do seu perfil e análise de maneira a conseguir localizar pessoas que possuam características/gostos em comum. Como é possível verificar o utilizador não precisa de se preocupar em construir o seu perfil ou em procurar as pessoas. Ao não obrigar o utilizador a fazer a configuração do seu perfil, o processo de automatismo evita que o utilizador se sinta enfiado e possivelmente desista antes de sequer começar. Em seguida são apresentados ao utilizador os vários sujeitos em comum para que este possa apreciar e caso o utilizador goste desliza o elemento para a direita indicando que gosta à aplicação, caso não goste o deslize é executado para a esquerda. Todos estes passos são feitos de maneira anónima, ou seja, o utilizador só irá revelar à pessoa que indicou que gostava se esta pessoa executar o mesmo gesto no utilizador em si.

Com a aparição de novas metodologias sociais em cooperação com as tecnologias móveis deu-se também uma atualização nas aplicações de jogos para *smartphones*, ou seja, até a uma certa altura a maioria dos jogos seriam jogos que as pessoas poderiam jogar, mas o máximo de interação social seria através de algum modo multi jogador online. Não havia a possibilidade de o utilizador sair para o exterior e enquanto jogava estabelecer também contacto social.

Logo a introdução de um jogo para *smartphones* que faz com que o utilizador necessite de deslocar para alcançar os objetivos e conseqüentemente possa estabelecer contacto social com outros jogadores foi de facto uma nova ramificação no sector de jogos. Apesar de o conceito ser totalmente novo não foi suficiente para que este tivesse sucesso, foi também necessário que houvesse alguma influência exterior. Foi a partir da aplicação Pokémon GO que este novo tipo de jogo começou a ganhar mais atenção do público, devido ao seu método de jogabilidade aliado ao *franchise* Pokémon que por si só possui uma popularidade enorme.

Apesar da influência não ser direta, as aplicações sociais tornaram este tipo de jogo real, pois o mercado tem atenção ao que o consumidor deseja. Logo se o mercado observa que o público está cada vez mais envolvido em interligar-se diariamente via aplicações sociais, então a necessidade de haver um jogo que envolva o jogador em interações com o mundo exterior é algo de espetar.

Solução Proposta

Nesta secção é apresentada a solução proposta e a sua composição por várias arquiteturas. O jogo SquadUp é composto por quatro cenários e cada um destes cenários possui uma arquitetura única em comparação com os restantes. Isto deve-se ao facto de que cada cenário é focado em um conjunto de ações pertencentes ao jogo. Por exemplo, o cenário Autenticação é focado no conjunto de ações de registar e autenticar e só neste cenário é que são possíveis encontrar estas ações. É possível observar os cenários que compõem o jogo SquadUp e as suas respectivas ações na Figura 3.1.

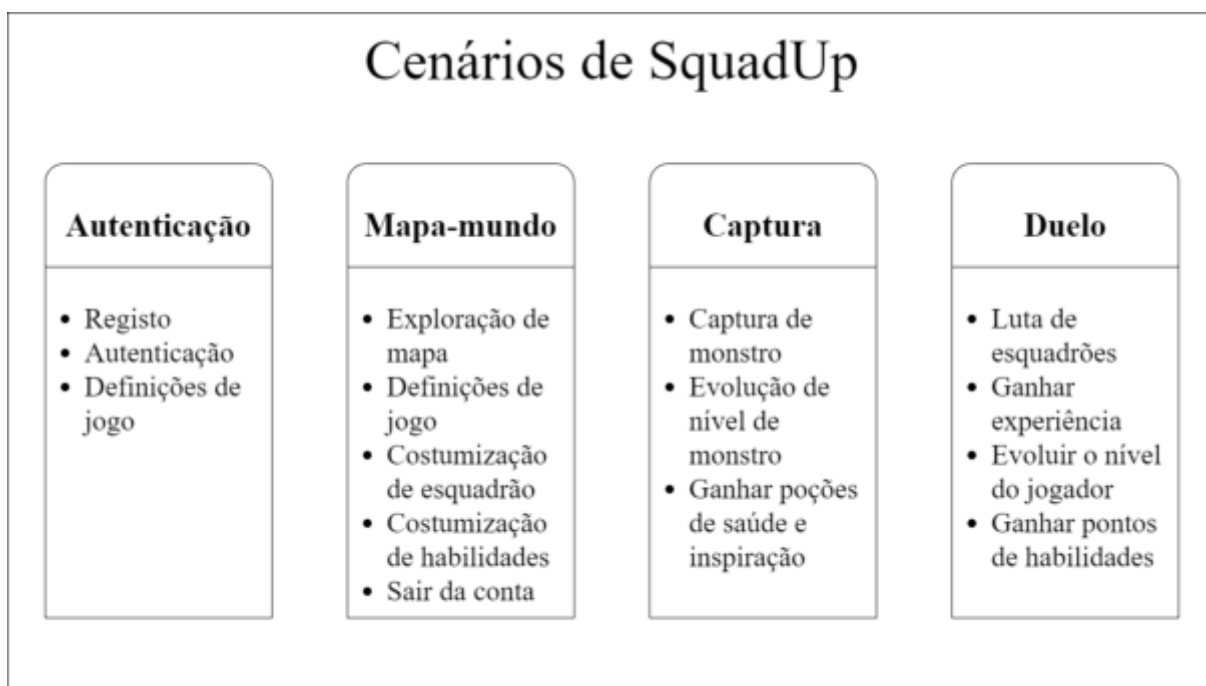


Figura 3.1: Cenários do jogo SquadUp.

Os objetivos do jogo SquadUp também são introduzidos assim como o funcionamento deste. Para que haja uma melhor ligação do conteúdo informativo ao conteúdo técnico são apresentadas as várias arquiteturas em detalhe e as suas respetivas características que as tornam únicas.

Para que o conteúdo do jogo seja apresentado em pormenor, foram criados vários capítulos cada um responsável por apresentar várias áreas importantes do jogo, como por exemplo toda a *Graphical User Interface* (GUI) dos vários menus e cenários que é apresentada no capítulo "Interface Utilizador do Jogo". Este capítulo também explica os funcionamentos dos cenários e os resultados visuais das várias funcionalidades distribuídas pelos menus, painéis e objetos interativos.

O capítulo "Mecânicas do Jogo SquadUp" explica em detalhe as várias mecânicas implementadas no jogo SquadUp, o funcionamento destas e a sua contribuição para a jogabilidade do jogo.

O capítulo "Módulos Técnicos" introduz o funcionamento de vários componentes programáticos, da solução implementada. O objetivo deste capítulo é apresentar o funcionamento programático do jogo SquadUp e dos vários elementos pertencentes aos componentes não gráficos.

3.1 Objetivos do jogo

O jogo SquadUp tem como foco transportar o jogador para uma aventura que utiliza o mundo real mas em formato gráfico, ou seja, tornando os locais que este frequenta no seu dia-a-dia em portas para uma nova captura de monstro ou para uma nova batalha a combater. Ao aplicar novos significados aos locais que o jogador considera mundanos e comuns, é possível proporcionar inspiração ao jogador para explorar estes locais com um novo ponto de vista.

Esta abordagem é conseguida através da aplicação de objetivos em vários locais, que tem como consequência a deslocação do jogador para os ditos locais, proporcionando o objetivo principal do jogo.

3.1.1 Objetivo Principal

Ao criar estes locais comuns é possível criar um ponto de encontro comum a vários jogadores, um ponto de encontro que possa ser considerado por muitos como um local de confraternização e socialização.

Como os locais de captura de monstro ou de duelo são comuns a todos os jogadores, estes locais servirão de ponto de encontro para estes. Assim, é possível proporcionar o objetivo principal de aproximar pessoas fisicamente através do jogo, ou seja, através do senso comum e dedução, um jogador pode aperceber-se que existem mais jogadores na área através da sua localização e utilização do *smartphone*. A partilha de experiências de jogo, de estratégias aplicáveis via as habilidades, a configuração do esquadrão e quais os monstros que combinam melhor uns com os outros, podem servir de motivo para iniciar uma conversa. É assim que o jogo SquadUp atinge o seu objetivo principal ao utilizar a sua jogabilidade como meio de deslocação do jogador para locais comuns a outros jogadores e servindo como ponto de interesse comum para iniciação de uma conversa, possibilitando a hipótese de amizades ou parcerias. Apesar de tudo, o jogo também pode ser jogado individualmente e a experiência de jogo não focada em socialização continua a ser a mesma.

3.1.2 Objetivos Secundários

Apesar de o jogo não ter um fim definido haverá sempre novos desafios a ultrapassar que permitem evoluir a personagem. O jogo baseia-se num jogo de ocupação de tempo e persistência. No entanto, existem outros objetivos secundários para que o jogo consiga transmitir e impor no utilizador uma ligação forte entre estes. Estes objetivos secundários são os seguintes:

1. Evolução constante dos monstros em relação ao seu nível;
2. Evolução constante do jogador em relação ao seu nível e pontos de habilidades;
3. Aumentar a quantidade de poções de inspiração e saúde;
4. Completar a coleção de monstros do jogo.

Estes objetivos secundários complementam-se, originando um *hobby* para o jogador passar o seu tempo. Na Figura 2.1 é exemplificado uma consequência do jogo Pokémon GO, sendo esta a socialização indireta através da partilha de experiências proporcionadas pela experiência de jogo por cada pessoa para os restantes.

3.2 Tecnologias para implementação da solução

Para implementação do jogo SquadUp foram utilizadas duas tecnologias, o Unity para a implementação de toda a estrutura do jogo SquadUp e o Mapbox como serviço de mapas para estabelecer o mapa geográfico em formato gráfico [18, 35].

O Unity é um motor de jogo desenvolvido pela Unity Technologies, que abrange diversas plataformas, como por exemplo jogos para o sistema operativo Windows, para o sistema operativo Android e outros sistemas operativos. O Unity disponibiliza a importação de vários *assets* da loja integrada e de outros serviços de terceiras partes. Através deste sistema foi possível obter os elementos gráficos de certos componentes do jogo como:

1. Monstros;
2. Ícone do Duelo no cenário Mapa-mundo;
3. Mapa de jogo do cenário Duelo;
4. Personagens de jogador;
5. Ícones das habilidades.

O Unity disponibiliza um *Integrated development environment* (IDE) integrado, que foi utilizado para implementação dos seguintes elementos:

1. Os menus dos vários cenários;
2. Os *Heads-up display* (HUD) dos vários cenários;
3. As caixas de colisão nos monstros;

4. A órbita de captura;
5. O mapa de jogo do cenário Duelo;
6. Parte do mapa de jogo do cenário Captura;
7. Criação e posicionamento dos monstros;
8. A animação aplicada na transição entre cenários;
9. A reprodução das músicas e efeitos sonoros.

Todos os componentes gráficos foram implementados através do IDE do motor de jogo, mas os *scripts* responsáveis por gerir toda a estrutura programática da solução foram implementados através do IDE Visual Studio [36]. Na implementação dos *scripts* foi utilizada a *Application Programming Interface* (API) de *scripting* do Unity, sendo que esta API está dividida em vários Namespaces por uma ordem hierárquica, em que cada Namespace fornece acesso a várias classes escritas pelos programadores da Unity Technologies. Desta API foi utilizada a classe *MonoBehaviour* que permite agregar os *scripts* aos *GameObjects* no IDE do Unity, ou seja, qualquer elemento no IDE do Unity é um *GameObject* que pode agregar várias características/funcionalidades, mas para se poder controlar este *GameObject* é necessário associar-lhe um *script* que só é possível caso este derive da classe *MonoBehaviour*. No caso do jogo SquadUp foi necessária a implementação de múltiplos *scripts* com foco na gestão, aplicação de eventos e afetação de vários *GameObjects* de maneira a criar uma interação entre o jogador e os elementos gráficos.

O Mapbox é uma plataforma de dados de localização que fornece vários tipos de mapas e serviços de localização. O Mapbox fornece um SDK para implementação no Unity, disponibilizando um conjunto de *scripts* e vários *prefabs*. Estes *prefabs* são *GameObjects* reutilizáveis/configuráveis, ou seja, um *prefab* é um *template* que possibilita a criação de *GameObjects* com as propriedades e componentes pré-configuradas neste. Através da importação e implementação do *Software Development Kit* (SDK) fornecido pelo Mapbox foi possível utilizar vários *prefabs* para a implementação do mapa de jogo do cenário Mapa-mundo e do mapa de jogo do cenário Captura.

O *LocationBasedGame* é o *prefab* disponibilizado pelo SDK do Mapbox que fornece o mapa gráfico. Este *prefab* é composto por vários *GameObjects*, em que cada um destes é gerido por um *script*. Cada *GameObject* é responsável por contribuir para a gestão de localização do *smartphone*. O mapa é desenhado conforme as características que foram pré-selecionadas no editor de mapas disponibilizados pelos serviços do Mapbox. Este editor permite criar um mapa conforme os requisitos necessários para o jogo. No caso do jogo SquadUp é necessário que os edifícios apareçam no mapa mas que não se transponham em altura ofuscando/cobrindo os monstros gerados. Assim, os edifícios receberam uma altura mínima de renderização para que a sua presença seja simbolizada no mapa mas não estraguem a jogabilidade ao jogador.

Também foi associado um *script* à personagem do jogador no cenário Mapa-mundo para que este desloque-se automaticamente através dos dados de localização disponibilizados pelo *smartphone* do jogador ao serviço de localização do Mapbox. O cenário Captura utiliza um *prefab* que recria um fundo de ambiente citadino de maneira a correlacionar os jogadores de localizações citadinas aos mapa de jogo virtual.

3.3 Arquiteturas do jogo SquadUp

Na implementação da solução proposta foi essencial dividir os vários cenários por blocos para melhor organização e estruturação do funcionamento destes. Para atingir este objetivo, a implementação teve início na parte gráfica devido a esta ser a parte essencial para captar a estrutura programática necessária. Seguindo as recomendações da metodologia de desenvolvimento de jogos centrada na experiência do jogador [3].

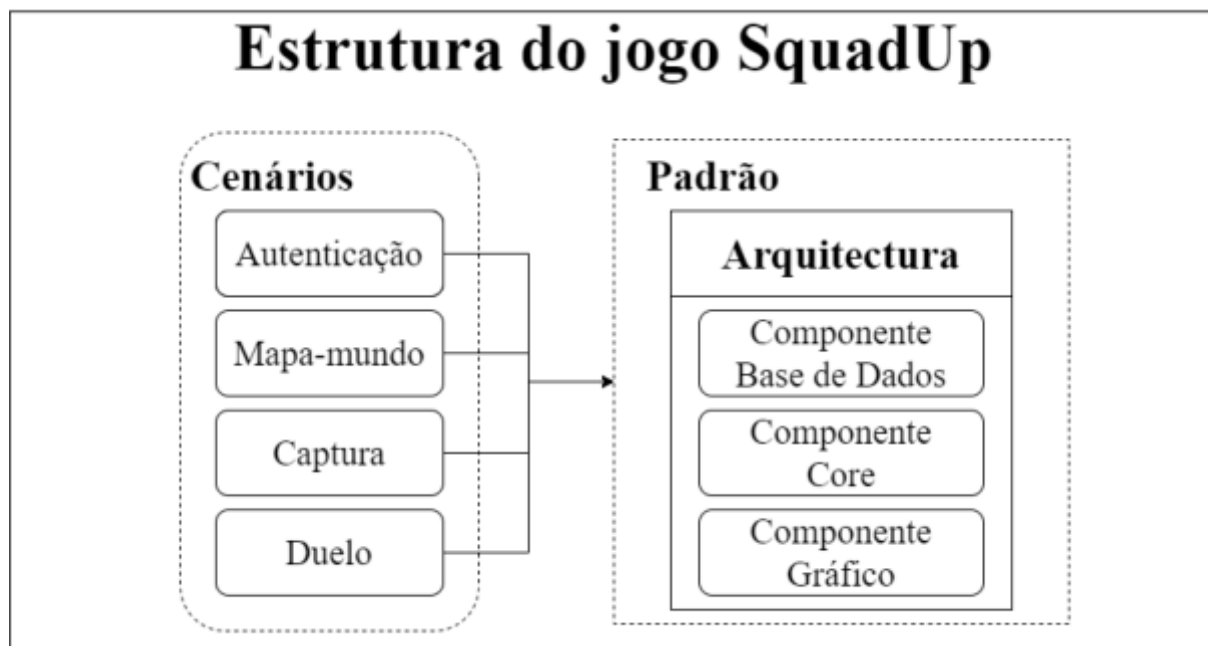


Figura 3.2: Estrutura do jogo SquadUp e a arquitetura dos seus cenários.

Esta abordagem originou um padrão de implementação para as arquiteturas do jogo SquadUp, em que a construção de cada cenário partia sempre do fator visual e concluía com a implementação de vários elementos programáticos no bloco central que tratam de gerir e fornecer os dados. Como é possível observar pela Figura 3.2 este padrão é constituído por três blocos de componentes, sendo estes:

1. Base de Dados - Este componente é referente aos conjuntos de dados escritos, lidos e guardados pelo jogo no *smartphone* do jogador;
2. Core - Este componente é referente aos gestores, geradores e mecânicas implementadas nos vários *scripts* da solução. O componente Core também fornece os recursos gráficos e textuais aos elementos do componente Gráfico. Todas as afetações executadas aos dados armazenados pelo componente Base de Dados são geridos pelo componente Core;
3. Gráfico - Este componente é referente a todos os elementos visuais que o jogo SquadUp apresenta ao jogador, por exemplo os monstros, a personagem do jogador, os menus.

Em seguida irão ser introduzidas as arquiteturas dos cenários e a composição dos seus componentes. Apesar de cada cenário ser distinto um do outro, os seus componentes partilham vários elementos, por motivos de reutilização ou apenas de minimização da figura em si. No último caso é apresentada uma figura expandido o elemento e a sua composição interna.

3.3.1 Arquitetura do cenário Autenticação

O cenário Autenticação tem como função disponibilizar a opção de registo ao jogador e consequentemente a autenticação deste para o jogo. Devido a estas funções, não são apresentados elementos gráficos em formato 3D, pois todo o conteúdo necessário de apresentar ao jogador é viável através de menus em formato 2D.

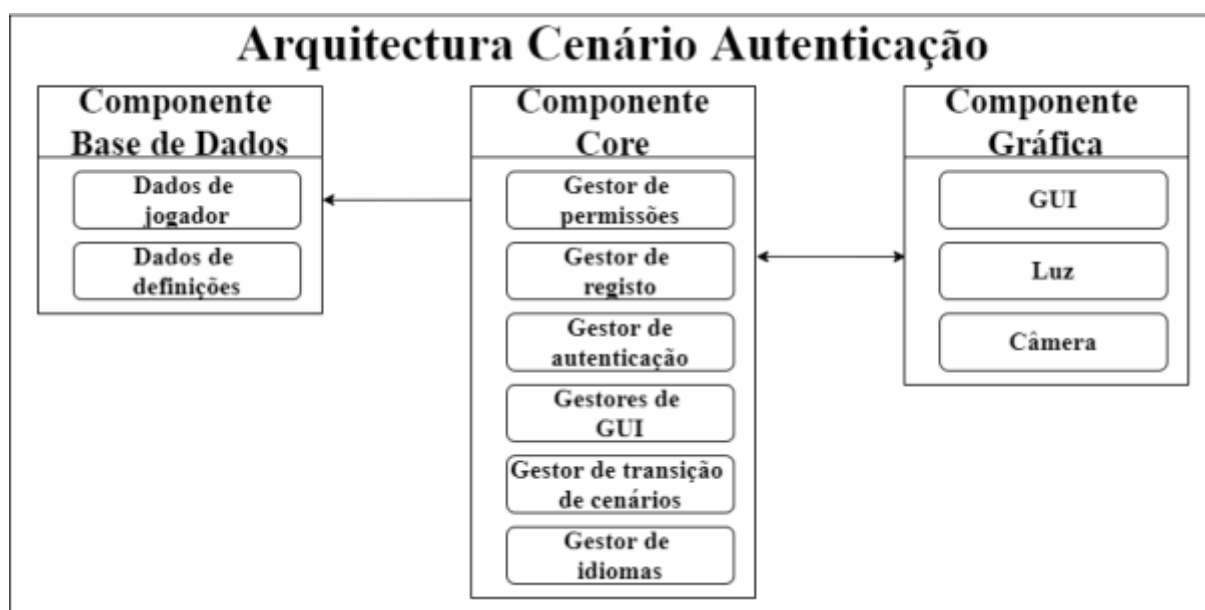


Figura 3.3: Arquitetura do cenário Autenticação.

Assim, o componente Gráfico é constituído por poucos elementos como se pode observar na Figura 3.3. Esta figura representa a arquitetura do cenário Autenticação e como se pode observar o componente Core é o composto por mais elementos, daí o nome ser Core, pois é uma peça central para o funcionamento de todos os elementos da arquitetura.

Quando o jogo SquadUp é iniciado pela primeira vez é apresentada uma mensagem de permissão para acesso à localização do *smartphone* do jogador. Apenas no caso do jogador conceder a permissão é que o jogo SquadUp apresenta o menu para autenticação ou registo. Para o jogador criar a sua conta necessita de fornecer um nome para a sua personagem, um *email* e uma *password*. Após introduzido estes dados é apresentado ao jogador para escolha, um conjunto de bustos que representam as várias personagens que o jogador pode encarnar no jogo. A autenticação é executada quando o jogador confirma a criação da sua conta ou quando o jogador reinicia o jogo, após ter encerrado a aplicação no seu *smartphone*.

Todas estas ações são geridas pelos vários gestores representados na figura e todas estas opções são apresentadas pelos vários gestores englobados no elemento "Gestores GUI" pertencente ao componente Core. Como é possível observar na Figura 3.4, o menu de registo e autenticação são partilhados por um gestor devido à partilha do mesmo painel para os dois efeitos.

Foi escolhida esta abordagem para um melhor aproveitamento de recursos computacionais mas também para proporcionar uma experiência de simplicidade e com poucos passos para o caso de registo.

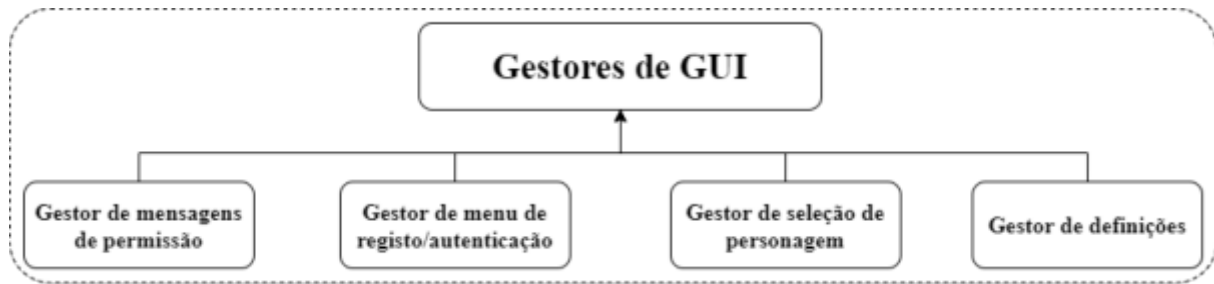


Figura 3.4: Composição de elemento Gestores GUI da arquitetura do cenário Autenticação.

O menu definições possui um gestor dedicado por motivos de reutilização do mesmo *script* utilizado no menu definições do cenário Mapa-mundo mas também para evitar múltiplos canais de acesso aos dados das definições. Apesar da partilha no menu de registo e autenticação, a abordagem aplicada no jogo SquadUp consiste em incluir em cada menu, um gestor responsável para tratar de todos os seus elementos interativos. No caso dos elementos textuais, estes são geridos e preenchidos pelo gestor de idiomas.

Este gestor é usado em todas as arquiteturas e está preparado com uma extensão de métodos para preenchimento de todos os elementos textuais de todas as GUI do jogo SquadUp. É verificado no início do jogo, através da leitura dos dados das definições, o idioma que o jogador escolheu. Cada idioma no jogo é representado por um tipo e todos estes tipos herdam de um tipo abstrato. Este tipo abstrato serve como mapa para que os seus descendentes implementem todos os elementos de texto necessários para o preenchimento.

O gestor de transição de cenários é como o nome indica, o gestor responsável por transitar o jogo entre os seus vários cenários. Devido a ser um elemento comum e necessário para todos os cenários, este é invocado no cenário Autenticação e mantém-se vivo até que o jogo seja encerrado por parte do jogador. Para além de fazer a transição de cenários que são despoletados por eventos pré-carregados, este gestor também faz a animação de *fade in/out* aplicando um intervalo temporal. Esta animação permite que o jogador tenha a noção que o jogo está a transitar entre cenários e cria um ritmo necessário no fluxo da jogabilidade.

3.3.2 Arquitetura do cenário Mapa-mundo

O cenário Mapa-mundo é um cenário principal do jogo SquadUp, pois é deste cenário que o jogador passa para o cenário Captura, Duelo e Autenticação. É também neste cenário que o jogador pode visualizar através dos vários sub-menus disponibilizados pelo menu principal, as suas estatísticas, os seus monstros, o seu esquadrão, aplicar pontos nas habilidades para melhorar o seu esquadrão e outras funções.

O jogador pode visualizar o local onde se situa no preciso momento em formato gráfico, sendo este o mapa de jogo e ele próprio através da sua personagem. Esta transposição do presente real para um presente gráfico é uma das principais atrações do jogo SquadUp, devido a proporcionar uma sensação de exploração de um mundo novo através do mundo real já conhecido pelo jogador. Todos estes elementos gráficos em formato 3D ou 2D estão listados no componente Gráfico, apresentados na Figura 3.5.

Foram construídos várias estruturas de dados por motivos de simplicidade e modularidade. Assim é possível separar os vários tipos de dados aos seus respetivos gestores sem que estes tenham acesso

aos restantes, evitando complexidade na escrita/leitura destes. Todos estes dados são estruturados em *scripts* serializáveis, o que permite criar a estrutura através de tipologia e armazenar os dados respeitando esta.

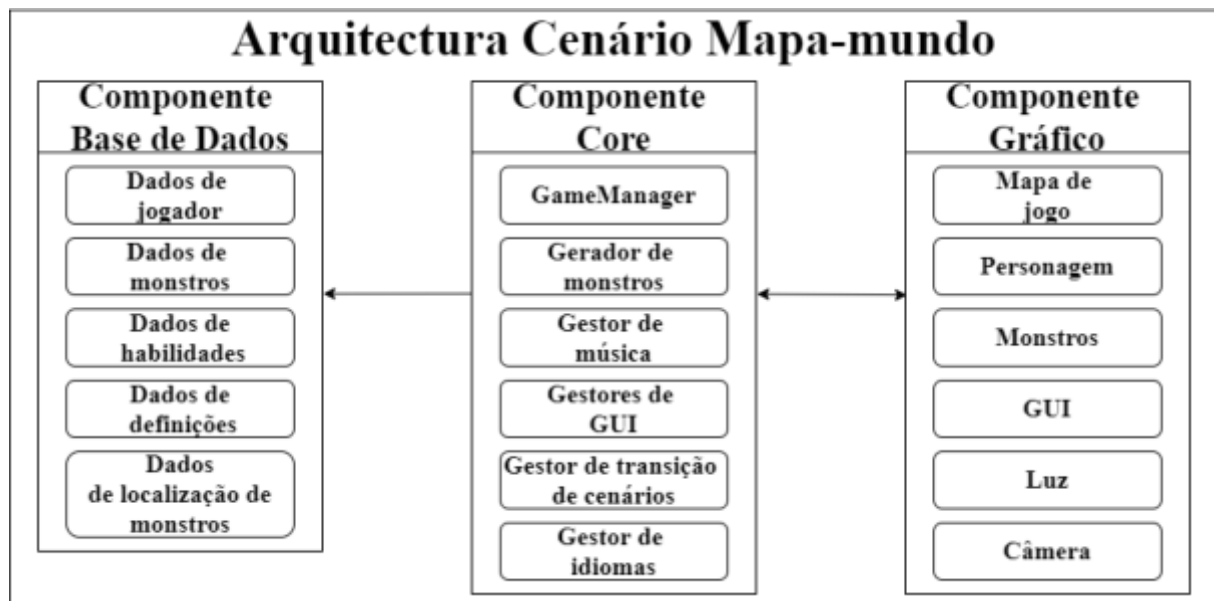


Figura 3.5: Arquitectura do cenário Mapa-mundo.

É no cenário Mapa-mundo que o GameManager é iniciado. Este elemento tem como objetivo servir de contentor para vários elementos necessários pelos cenários Mapa-mundo, Captura e Duelo. Na Figura 3.6 pode-se observar os vários elementos que compõem o GameManager. Estes elementos são utilizados ao longo do jogo pelos vários cenários permitindo o acesso constante à sua informação, sendo que as suas funções são as seguintes:

1. Gestor de jogador - disponibiliza o acesso aos vários dados do jogador assim como o seu esquadrão. Também é este gestor que está encarregue de guardar todos os elementos de dados do componente Base-de-dados;
2. Gestor de monstros - disponibiliza todos os monstros do jogo, contabiliza todos os monstros que o jogador capturou e gere as suas evoluções;
3. Gestor de preferências do jogador - responsável por configurar as definições de idioma do jogo, conforme os dados das definições. Disponibiliza o acesso ao gestor de idiomas;
4. Gestor de monstro a capturar - responsável por guardar a informação do monstro a capturar que foi selecionado pelo jogador no mapa de jogo do cenário Mapa-mundo. Este gestor é usado para confirmação de captura e para transmissão dos dados necessários para contabilização do gestor de monstros;
5. Gestor de habilidades - responsável por iniciar todas as habilidades do jogo SquadUp e servir como contentor destas. Disponibiliza diversas maneiras de seleção de habilidades por meios de parametrização e também aplica a ativação destas.



Figura 3.6: Composição do GameManager.

Como já referido, o cenário Mapa-mundo possui um vasto menu composto por diversos sub-menus. Devido aos menus serem bastante diferentes uns dos outros na sua interface foi necessário aplicar um gestor para cada um destes. Com a implementação destes menus, também foi necessário aplicar um tutorial para guiar o jogador através das várias características informativas e interativas que cada menu disponibiliza, logo também foi necessário aplicar um gestor para cada tutorial. Assim resultou em um dos elementos mais compostos no cenário Mapa-mundo e também dos outros cenários, sendo possível observar na Figura 3.7.

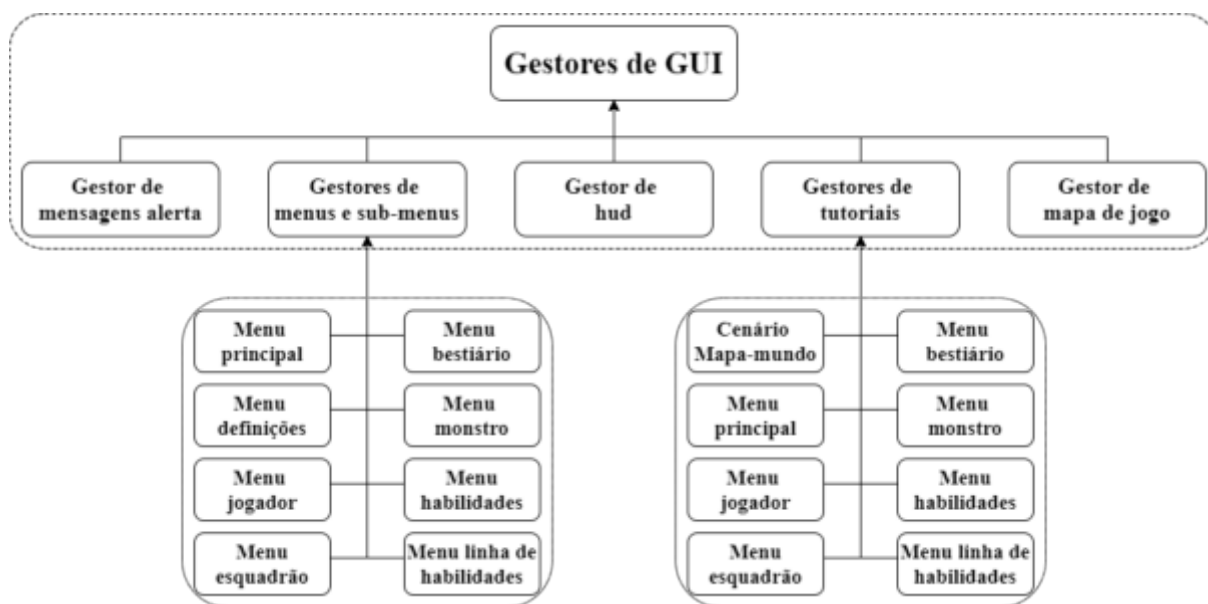


Figura 3.7: Composição do elemento gestores GUI da arquitetura do cenário Mapa-mundo.

O componente Core do cenário Mapa-mundo também introduz dois elementos novos, o gerador de monstros e o gestor de música. O gerador de monstros é o elemento que trata de criar e posicionar os monstros em redor da personagem do jogador. Este gerador utiliza as coordenadas do jogador para colocar o monstro numa área limitada em volta do jogador, mas em uma posição aleatória e sem sobreposição sobre os monstros restantes. O monstro é fornecido aleatoriamente pelo gestor de monstros. O gerador de monstros também está encarregue de destruir os monstros criados no mapa de jogo através de um limite temporal. Tanto a criação como a destruição são executadas com base temporal estabelecendo ao jogador um ritmo de captura de monstros.

O gestor de música trata de reproduzir as músicas pré-estabelecidas a este quando o jogador entra no cenário. Este elemento é usado no cenário Mapa-mundo, Captura e Duelo, tendo músicas diferentes

para cada um destes cenários. O objetivo na diversidade musical é estabelecer uma ligação entre o nível de pressão da jogabilidade de cada cenário à seriedade e expressão rítmica que cada música transpõem.

3.3.3 Arquitetura do cenário Captura

O cenário Captura foca-se na captura de monstros por parte do jogador, proporcionando um cenário com uma baixa quantidade de menus, sub-menus e elementos GUI. Podemos observar na Figura 3.8 que a arquitetura deste cenário foca-se em gerir todo o processo de captura de um monstro, assim como a mecânica à volta da órbita de captura, este último é o utensílio usado pelo jogador para capturar o monstro.

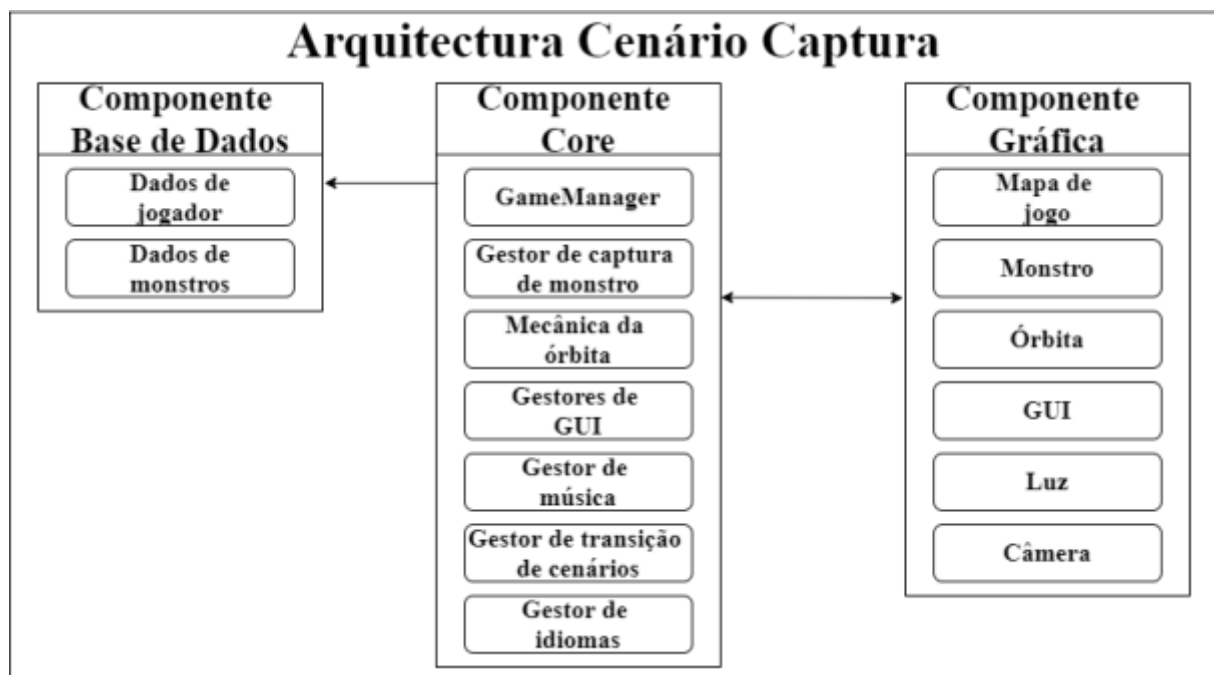


Figura 3.8: Arquitetura do cenário Captura.

Como é possível observar pela figura existem elementos que também aparecem em outras arquiteturas, isto é devido às suas funcionalidades que abrangem, como é o caso do Gestor de idiomas que disponibiliza todas as informações textuais de todos os elementos GUI de todos os cenários.

Apesar da partilha de elementos pelas várias arquiteturas, o cenário Captura possui os seus elementos personalizados, estes são o gestor de captura de monstro e mecânica de órbita. O elemento gestor de captura é responsável por iniciar o cenário Captura em vários elementos, sendo estes:

1. Extrair a informação através do Gestor de monstro a capturar disponibilizado pelo elemento GameManager para criação e posicionamento do monstro no mapa de jogo;
2. Calcula, gere e marca a quantidade de vezes que o jogador pode tentar capturar o monstro.
3. Cria as órbitas de captura sempre que estas não entram em colisão com o monstro;
4. Através de várias condições deteta e avança para o menu de resultado, para ambos os casos de sucesso na captura do monstro por parte do jogador.

O elemento mecânica de órbita trata de criar toda a física por detrás do arremesso desta, ou seja, desde a deteção de toque, arrastar a órbita pelo ecrã e o soltar desta em direção do monstro é gerido por este elemento.

Este elemento trata de invocar o gestor de captura quando a órbita não colide com o monstro, dando início ao processo de contagem da quantidade de vezes que o jogador ainda tem para capturar o monstro e caso esta seja igual a zero, a ativação do menu de resultado. O mesmo processo para o caso da órbita ter entrado em colisão com o monstro é invocado pelo monstro em si e o gestor de captura trata de adicionar o monstro ao gestor de monstros encapsulado no elemento GameManager, avançado em seguida para o menu resultado.

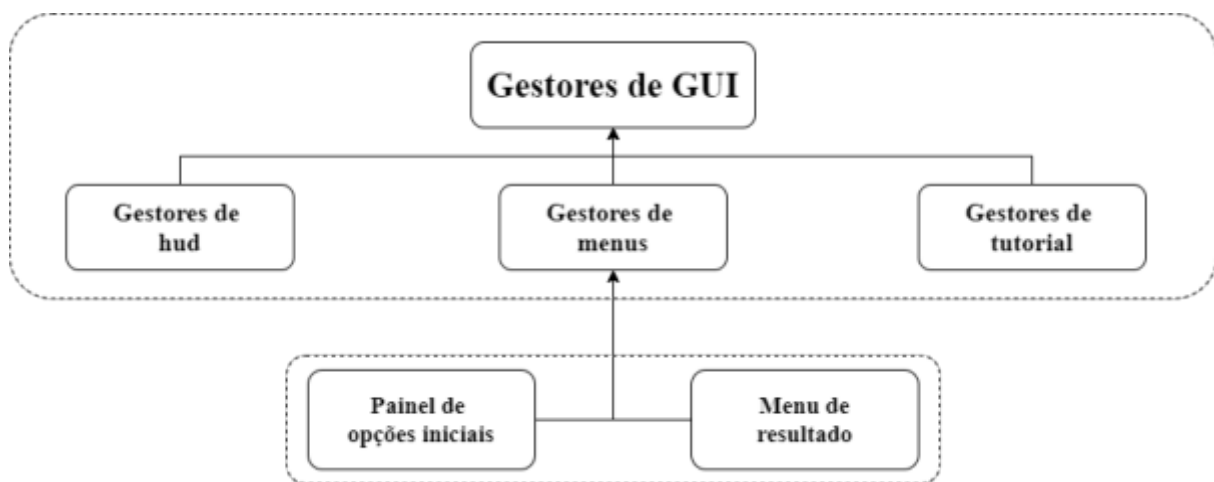


Figura 3.9: Composição do elemento gestores GUI da arquitetura do cenário Captura.

Na Figura 3.9 podemos observar que o cenário Captura apenas possui dois menus, um é referente a um painel que apresenta duas opções de procedimento para com o cenário, sendo estas fugir e voltar para o cenário Mapa-mundo e a outra continuar para a captura do Monstro. Estas opções foram disponibilizadas ao jogador para caso em que tenha selecionado o monstro por engano e deseje sair do cenário sem ter que passar pela processo de captura.

O menu de resultado informa o jogador do seu sucesso ou insucesso na captura do monstro, também irá apresentar as estatísticas do monstro, como por exemplo, a quantidade de vezes que já o viu e capturou. Como parte do resultado final, o jogador pode ou não receber poções de saúde/inspiração, estas poções servem para ajudar a restaurar a saúde dos monstros no esquadrão do jogador e para aumentar as chances destes na ativação de certas habilidades. A disponibilização de poções ao jogador através da tentativa de captura de um monstro é um meio de reforçar o jogador a tentar capturar mais monstros e também um meio de motivação para os casos de insucesso na captura do monstro.

A componente Base de dados tem como composição os elementos dados de jogador e monstros devido às várias modificações nos dados aplicadas pelo processo em si. Ou seja, sempre que o jogador tenta capturar um monstro ele irá no mínimo observar no monstro, o que implica modificar a contagem deste nos dados de monstros. Seja qual for o resultado da captura, o jogador irá receber poções o que implica modificar também os dados deste.

3.3.4 Arquitetura do cenário Duelo

O cenário Duelo tem como foco o combate entre o esquadrão do jogador e um esquadrão feito no momento. Para atingir este objetivo foi necessário criar um mapa de jogo focado em fornecer um ambiente adequado de combate, um HUD que indicasse ao jogador as afetações aplicadas aos seus monstros e do inimigo. Para além dos fatores gráficos, também foi necessário implementar uma estrutura para gestão e aplicação dos vários passos necessários a efetuar em cada turno do combate.

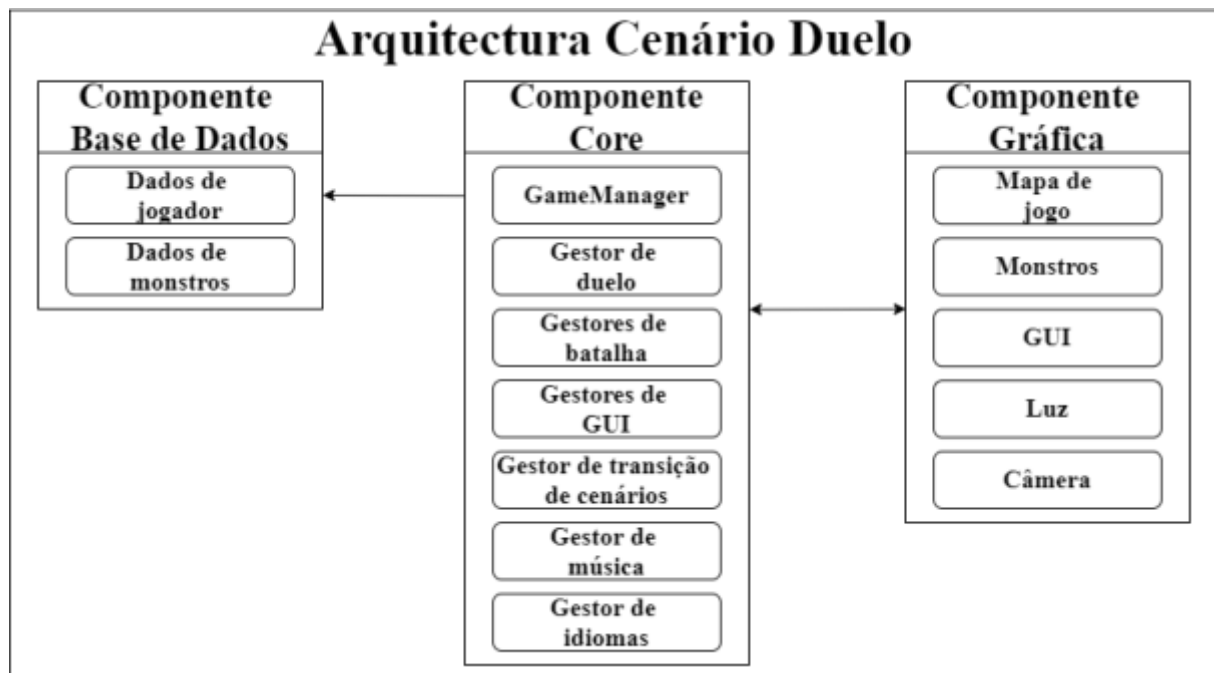


Figura 3.10: Arquitetura do cenário Duelo.

A Figura 3.10 demonstra a arquitetura do cenário Duelo e podemos observar que o componente Core é o componente com mais elementos de todos os componentes. O elemento gestor de Duelo pode ser considerado o elemento mais importante do componente Core, devido a servir como motor de arranque para os vários elementos que compõem a arquitetura deste cenário.

Muitos dos elementos que o gestor de Duelo trata de iniciar e atribuir funcionalidades pertencem aos elementos "Gestores de batalha" e "Gestores de GUI". Estes elementos irão ser apresentados em forma expandida para melhor percepção da sua composição e as várias funcionalidades de que são incumbidos.

As funções do elemento gestor de duelo são as seguintes:

1. Verifica e apresenta o tutorial referente ao cenário;
2. Inicia o gestor de inimigo;
3. Inicia o gestor de esquadrões;
4. Instância os monstros e posiciona-os no mapa de jogo;
5. Inicia o gestor de combate;
6. Inicia o gestor de painéis de saúde;

7. Aplica os eventos dedicados a cada botão do Menu opções de combate;
8. Inicia o gestor de ataque;
9. Inicia o gestor de defesa;
10. Inicia o gestor de painéis de batalha.

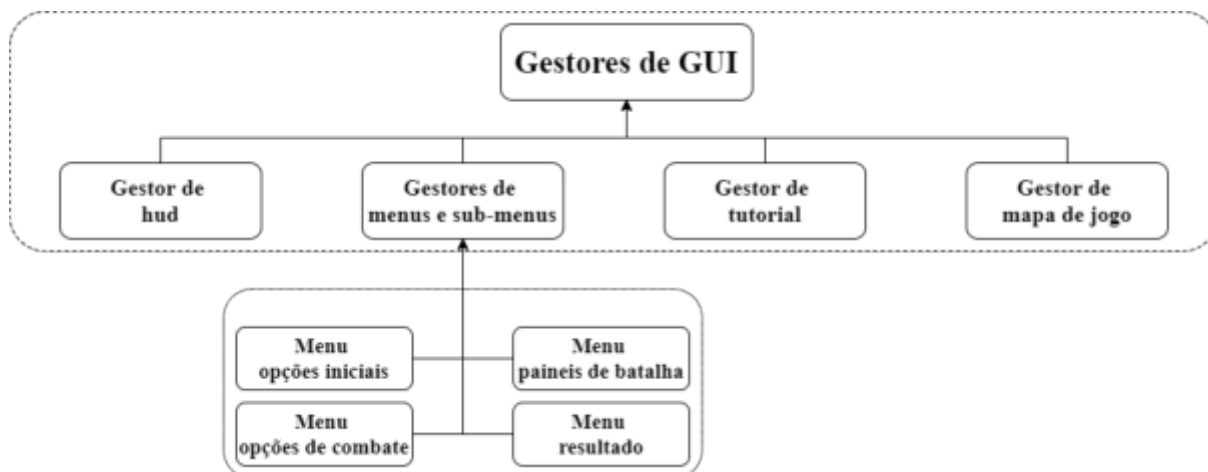


Figura 3.11: Composição do elemento gestores GUI da arquitetura do cenário Duelo.

Na Figura 3.11 podemos observar os vários elementos que compõem o elemento “Gestores de GUI”, este elemento é responsável por gerir o elemento GUI da componente Gráfico. Devido ao combate do jogo SquadUp envolver muitas variantes dos poderes dos monstros e as habilidades aplicadas a estes, foi necessário criar painéis adequados para apresentação dos valores destas variantes.

Para que o jogador assimile a informação necessária do que aconteceu no momento e ao mesmo tempo seja guiado pelo processo de combate, foi necessário implementar vários painéis informativos e interativos. Os painéis são os seguintes:

1. Menu opções iniciais - Quando o jogador inicia um Duelo são lhe introduzidas duas opções iniciais, a de fugir e a de continuar. Estas opções foram implementadas neste cenário para que o jogador tenha a opção de retirar do Duelo sem ter que completar o processo de combate caso deseje;
2. Menu opções de combate - Se o jogador decidir proceder para o combate, as opções fugir e continuar são substituídas pelas opções:
 - (a) Poção de saúde - O jogador pode aplicar uma poção de saúde para recuperar a saúde dos monstros do seu esquadrão. Apenas é aplicável uma vez por fase;
 - (b) Poção de inspiração - O jogador pode aplicar uma poção de inspiração para aumentar a chance (probabilidade) das habilidades (que estão ativas) que recorrem ao mecanismo de chance para serem ativadas sobre os poderes dos monstros do esquadrão;
 - (c) Atacar/defender - Procede para a próxima fase de combate, quer seja ataque ou defesa.
3. Menu painéis de batalha - É responsável por apresentar ao jogador a informação de combate de cada fase. Estes painéis são repartidos por dois painéis, em que o primeiro painel informa o jogador sobre os valores ofensivos e defensivos monstros, assim como a influência das habilidades

nestes. O segundo painel apresenta ao jogador a aplicação dos vários tipos de danos aos monstros defensivos;

4. Menu resultado - Apresenta ao jogador várias estatísticas do combate, o resultado de sucesso e a experiência ganha. Caso o jogador evolua de nível também é informado do seu progresso, a quantidade de pontos de habilidade que ganhou e a quantidade destes.

O elemento "Gestor de HUD" é referente ao gestor dos painéis de saúde, este gestor trata de apresentar a saúde dos monstros de cada esquadrão. Os monstros dos esquadrões são identificados nos painéis de saúde através de uma imagem e a sua saúde é uma barra progressiva que irá diminuir ou aumentar, conforme o valor da saúde do monstro em combate. Este gestor é invocado sempre que um ataque é executado e o dano aplicado ao esquadrão defensor. O seu processo trata de receber o esquadrão defensor e atualizar os painéis referente aos monstros do esquadrão.

O elemento "Gestor de tutorial" é equivalente aos outros gestores de tutoriais nos outros cenários, sendo que neste caso é dedicado a apresentar a informação referente ao cenário Duelo.

O elemento "Gestor de mapa de jogo" é pertencente ao gestor de Duelo que trata de criar os monstros do esquadrão do jogador e o inimigo. No final do processo posiciona os monstros nas posições definidas para cada monstro conforme o seu tipo.

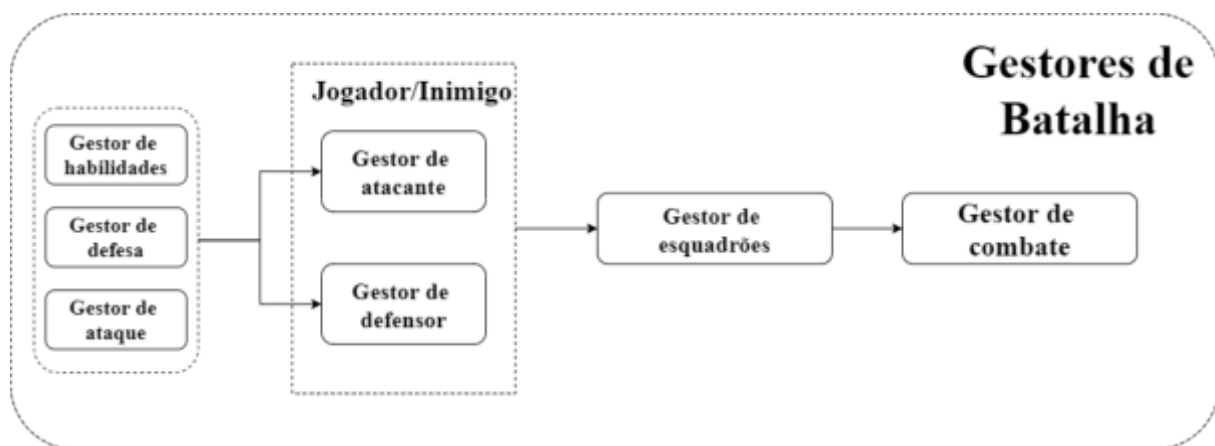


Figura 3.12: Composição do elemento gestores de batalha da arquitetura do cenário Duelo.

Na Figura 3.12, é possível observar a estrutura implementada para executar uma batalha entre esquadrões. Um duelo termina quando o estado de saúde de todos os monstros de um esquadrão atinja o estado de desmaiado, ou seja, os esquadrões irão combater entre si várias rondas até atingirem estas condições. Uma ronda para um esquadrão é composta por duas fases a de ataque e a de defesa. Assim, foi necessário implementar uma estrutura para gerir os vários passos, tanto do ataque como da defesa e em si tornar esta estrutura aplicável a ambos os esquadrões sem restrições.

O elemento "Gestor de esquadrões" trata de disponibilizar o acesso ao gestor de jogador e de inimigo. Também gere a condição de ataque, que informa se o esquadrão alvo já passou pela fase de ataque ou não. E utilizando esta informação contabiliza o número de rondas executadas.

Os elementos "Gestor de atacante" e "Gestor de defensor" são responsáveis por:

1. Iniciar e disponibilizar o acesso ao gestor de ataque do esquadrão;
2. Iniciar e disponibilizar o acesso ao gestor de defesa do esquadrão;
3. Disponibilizar o acesso ao gestor de habilidades do esquadrão;
4. Ativar as habilidades de ataque e aplicar aos poderes dos monstros do esquadrão;
5. Ativar as habilidades de defesa e aplicar aos poderes dos monstros do esquadrão;
6. Aplicar o dano nos monstros do esquadrão;
7. Verificar por monstros no estado de saúde desmaiado e aplicar habilidades referentes a esta condição (caso estejam ativas);
8. Recuperar uma porcentagem da saúde dos monstros através de habilidades dedicadas à regeneração de saúde.

Os elementos "Gestor de ataque" e "Gestor de defesa" são responsáveis pela execução de cálculos sobre os vários poderes e habilidades dos monstros dos esquadrões que representam. Todos estes cálculos são executados por uma ordem definida pelas várias restrições sobre-aplicadas, devido aos efeitos das habilidades sobre os monstros do esquadrão a que pertencem, assim como sobre os monstros do esquadrão inimigo.

O elemento "Gestor de ataque" é responsável por gerir e calcular os seguintes elementos:

1. Os valores resultantes das habilidades dedicadas ao ataque sobre os poderes de ataque dos monstros do esquadrão;
2. O valor total de ataque de cada monstro após a influência das habilidades dedicadas ao esquadrão como as habilidades do esquadrão inimigo;
3. Os valores dos vários tipos de dano ofensivo a aplicar sobre o esquadrão defensivo.

O elemento "Gestor de defesa" é responsável por gerir e calcular:

1. Os valores resultantes das habilidades dedicadas à defesa sobre os poderes de defesa dos monstros do esquadrão;
2. O valor total de defesa de cada monstro após a influência das habilidades dedicadas ao esquadrão como as habilidades do esquadrão inimigo;
3. O valor percentual de aplicação de dano a cada monstro do esquadrão;
4. O valor total de dano defendido;
5. Quantos monstros estão com um estado de saúde desmaiado.

O elemento "Gestor de combate" é o elemento responsável por aplicar os vários passos de luta de uma fase, ou seja, quando os esquadrões combatem um contra o outro, é necessário invocar vários gestores para processamento de certos passos.

Para uma luta ser desencadeada, este elemento trata de designar o atacante e o defensor através do gestor de esquadrões. O gestor de esquadrões irá designar se o atacante é o jogador ou o inimigo e consequentemente o defensor. Após este passo, são ativadas as habilidades e aplicados os efeitos destas sobre os poderes dos monstros de cada esquadrão. Após a aplicação dos efeitos, são invocados os gestores de ataque e defesa para cálculo dos poderes ofensivos e defensivos. Com os valores ofensivos e defensivos calculados, são calculados as percentuais de dano respectivas a cada monstro defensivo e aplicado o dano sobre estes. Como último passo, é verificado se o estado de saúde de todos os monstros defensivos é equivalente ao de desmaiado, caso seja, é atingida a condição de vitória.

4

Interface Utilizador do Jogo

Este capítulo descreve o funcionamento do jogo e um conjunto de funções adjacentes ao jogo que são úteis para jogar o jogo tais como, as permissões de sistema necessárias, o tutorial para aprender a jogar, o sistema de autenticação e as definições do jogo.

O jogo SquadUp é composto por quatro cenários, o cenário Autenticação, Mapa-mundo, Captura e Duelo. Cada cenário introduz ao jogador as suas interações, informação contextual e visual através das suas *Graphical User Interface* (GUI). Cada GUI é composta por vários menus, painéis e representações gráficas de certos elementos.

O cenário Autenticação irá ser composto por vários menus dedicados ao registo e autenticação do jogador para o jogo.

O cenário Mapa-mundo apresenta ao jogador o mundo real através de uma representação gráfica. O jogador também consegue aceder aos outros cenários e aceder a vários menus cada um com uma representação gráfica única e apropriada para o seu funcionamento.

O cenário Captura demonstra a representação gráfica do monstro que o jogador selecionou para capturar. O jogador irá visualizar o menu de opções e resultado assim como uma representação gráfica da órbita de captura.

O cenário Duelo é composto por uma GUI que apresenta imensos detalhes ao jogador através de vários painéis. O cenário também possui vários menus para que o jogador saiba as suas opções de interação inicial e resultado final do Duelo.

4.1 Permissões

Quando o jogo SquadUp é iniciado pela primeira vez é apresentado ao jogador a opção de permitir ou recusar o acesso à localização do dispositivo móvel.

Caso o jogador recuse é lhe apresentado uma mensagem a explicar que o jogo precisa da localização para funcionar e se este quer tentar outra vez, como é possível de observar na Figura 4.1.



Figura 4.1: Mensagem para acesso à localização do *smartphone*.

Caso o jogador recuse outra vez, a aplicação irá fechar e quando abrir de novo irá disponibilizar a mesma mensagem até que o jogador aceite. Em certas versões do sistema operativo *Android* a recusa da permissão irá ser permanente o que faz com que o jogo não passe da verificação de permissão ou inicie de todo.

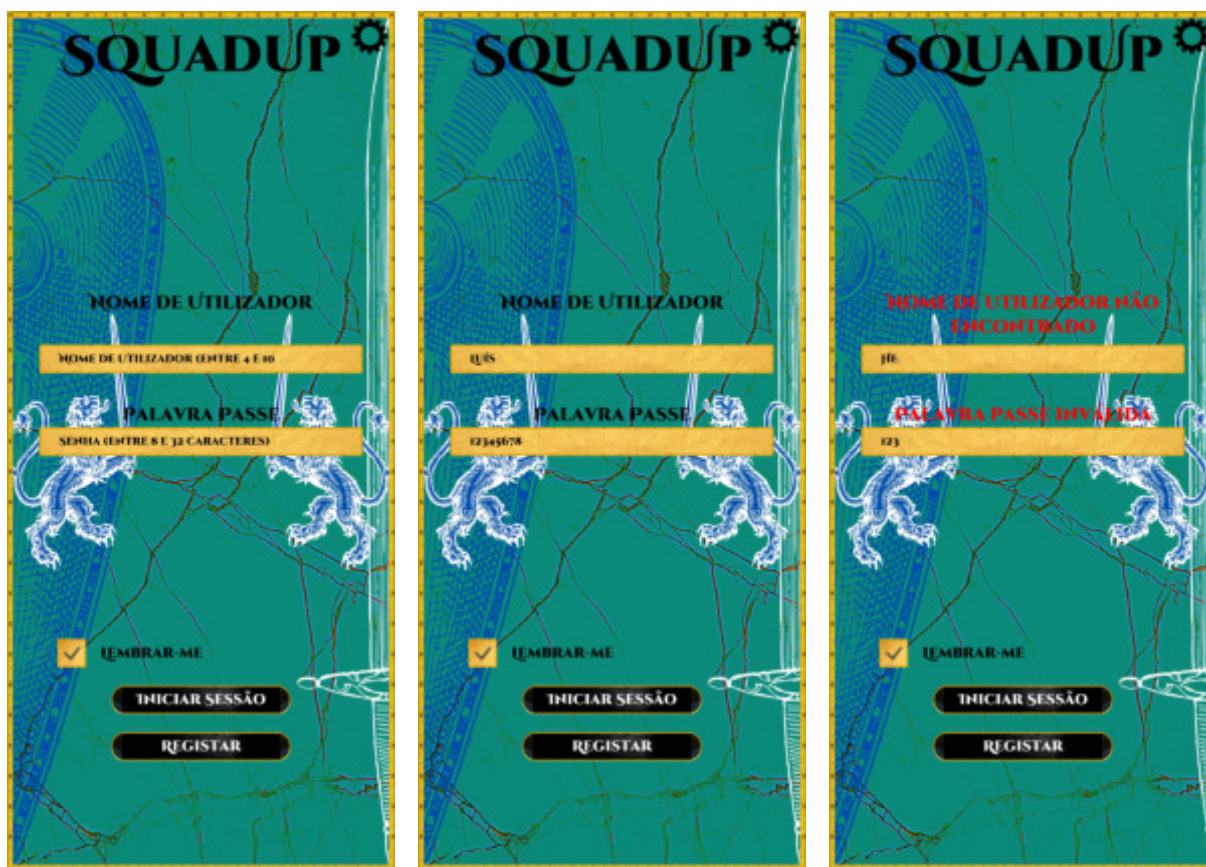
Para este tipo de casos é necessário que o jogador altere a permissão de acesso à localização do dispositivo móvel através das definições do seu dispositivo móvel ou desinstale e reinstale o jogo. Após este processo e o processo de registo serem executados com sucesso, se o jogador ativou a opção de "Lembrar-me", que é explicada no capítulo Cenário Autenticação, não é apresentado mais nenhum destes ecrãs de permissão, mas sim um ecrã de "Loading" antes da passagem para o cenário "Mapa-mundo".

4.2 Cenário Autenticação

Após o jogador aceitar a permissão de acesso à localização do dispositivo móvel é transportado para o cenário Autenticação. O menu apresenta quatro opções, como é possível observar na Figura 4.2a:

1. Iniciar a Sessão;
2. Registrar;
3. Definições;
4. Lembrar-me.

Estas opções são explicadas em seguida, com exceção da quarta opção devido a ser um mecanismo de automatização do processo de autenticação. Esta opção fica associada à opção "Iniciar Sessão" e "Registrar", por fazer parte do processo de ambas.



(a) Painel de Autenticação.

(b) LogIn Correto.

(c) LogIn Incorreto.

Figura 4.2: Exemplo de autenticação no jogo SquadUp.

4.2.1 Definições

O jogo SquadUp inicia com um idioma predefinido, mas o jogador pode não possuir conhecimento deste idioma. Sendo assim, é disponibilizado o acesso às “Definições” do jogo, para que o jogador consiga alterar a língua de jogo e assim tenha um processo de “Registo” adequado. Também é disponibilizado a alteração ao volume da música ambiente assim como ao volume dos efeitos sonoros.

Este menu de “Definições” é o mesmo menu de “Definições” que o jogador tem acesso quando acede ao menu “Principal” disponibilizado apenas no cenário “Mapa-mundo”. Foi escolhido reutilizar o menu em ambas os cenários, para não variar nenhuma opção de definições em ambos os menus, para evitar singularidades complexas e opções de definições inalcançáveis sem necessidade. Assim consegue-se ambientar o jogador a um único menu de definições e ao mesmo tempo simplificar a estrutura de menus implementados no jogo SquadUp.

O acesso às definições é feito através de um botão no canto superior direito. Para melhor percepção do jogador é utilizado um símbolo de uma roda dentada que é associada ao acesso às definições nas aplicações para *smartphones*. Pode ser observado este botão na Figura 4.2 e a interface do menu “Definições” no capítulo “Menu Definições” na Figura 4.11.

4.2.2 Iniciar Sessão

Para o jogador poder iniciar a sessão é necessário que este introduza o seu nome de jogador e a sua *password*. Os dados a inserir estão limitados em quantidade mínima e máxima de caracteres. Para que o jogador tenha noção dos limites são disponibilizados os limites como mensagens sublineares. Estas mensagens são colocadas dentro das caixas de texto correspondentes aos dados a serem inseridos, no momento em que o jogador inicia a inserção dos dados as mensagens desaparecem, sendo substituídas pelo conteúdo inserido. Pode-se observar as mensagens de limite na Figura 4.2a. Para facilitar o acesso ao jogo é disponibilizada a opção de “Lembrar-me”. Esta opção permite ao jogador que entre diretamente para o cenário de jogo principal sem ter que inserir os seus dados sempre que inicia o jogo SquadUp. Na mesma figura a opção “Lembrar-me” é representada por um *Toggle* que começa ativada por predefinição.

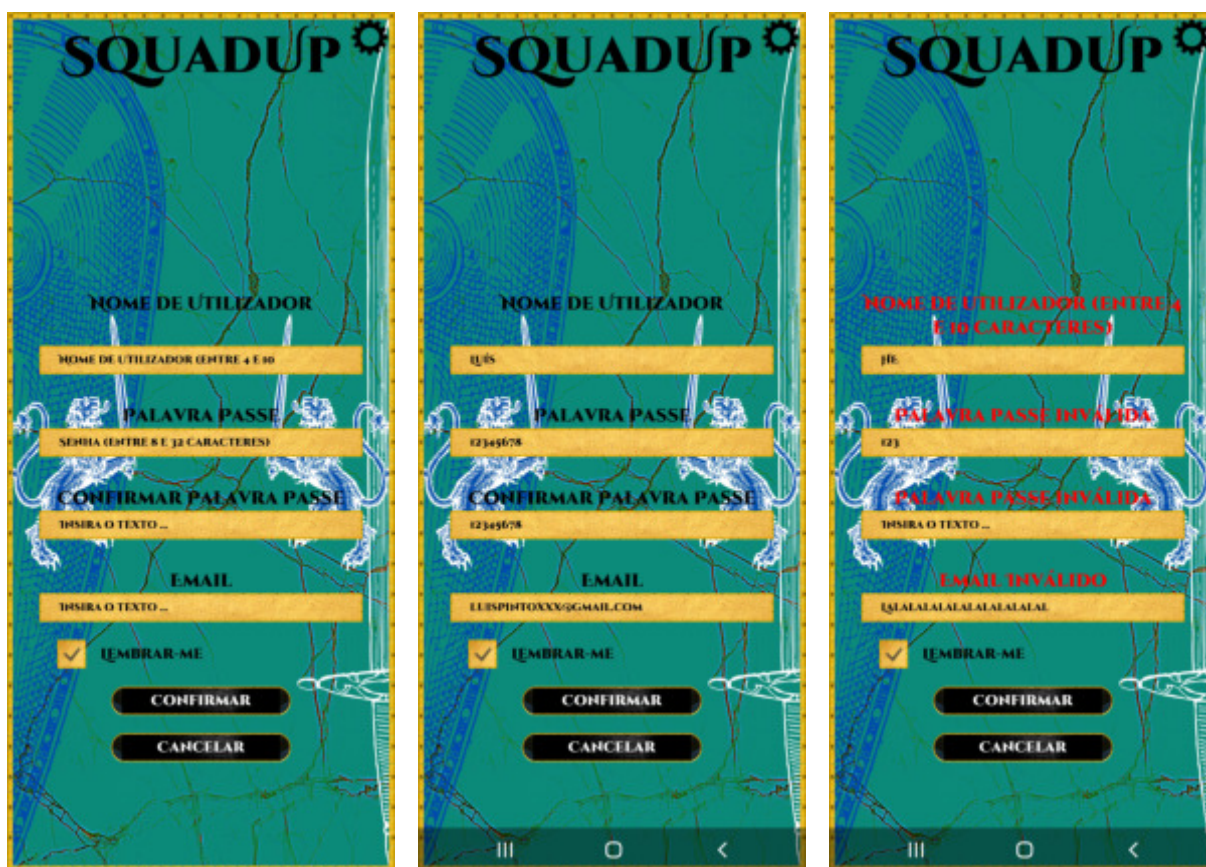
Internamente o jogo verifica os dados inseridos pelo jogador com os dados armazenados. São comparados o nome e a *password* introduzidos, para que possam ser apresentadas mensagens de dados inseridos incorretamente, de dados não correspondentes com a conta a que o jogador tenta aceder ou se a conta é inexistente. É possível de observar estas mensagens na Figura 4.2c. As mensagens de erro variam conforme os dados inseridos pelo jogador, ou seja, se o jogador tentar aceder a uma conta que não existe irá ser reportado esse erro, se o jogador tentar aceder a uma conta existente, mas errou na *password* irá ser reportada a mensagem correspondente a esse erro.

Se o jogador decidir proceder ao registo de conta em vez de iniciar sessão todos os dados que este tenha inserido nas respetivas caixas de texto, serão mantidos evitando que o jogador tenha que repor os dados novamente.

4.2.3 Registo

Para que o jogador consiga jogar o jogo SquadUp, é preciso criar uma conta. Sendo assim o jogador terá que proceder ao registo de uma conta para que possa jogar, para isto é necessário inserir os dados requisitados:

1. Nome;
2. Password;
3. Password;
4. Email.



(a) Painel de Registo.

(b) Registo Correto.

(c) Registo Incorreto.

Figura 4.3: Exemplo de registo no jogo SquadUp.

O nome que o jogador inserir no processo de registo é o nome que o jogador terá associada à sua personagem dentro do jogo SquadUp. É necessário que o jogador insira a sua *password* em duplicado para evitar que o jogador se tenha enganado na *password* que inseriu inicialmente, evitando ao jogador erros de preenchimento.

Como é possível observar na Figura 4.3a caso o jogador não insira os dados corretamente são apresentadas mensagens de erro, substituindo o título acima de caixa de texto pela mensagem de erro colorida a vermelho. Seja um ou vários erros, estes são apresentados singularmente para cada tipo de dado incorretamente inserido 7.1.



(a) Painel de Bustos.

(b) Mensagem de Alerta.

(c) Confirmação de Escolha.

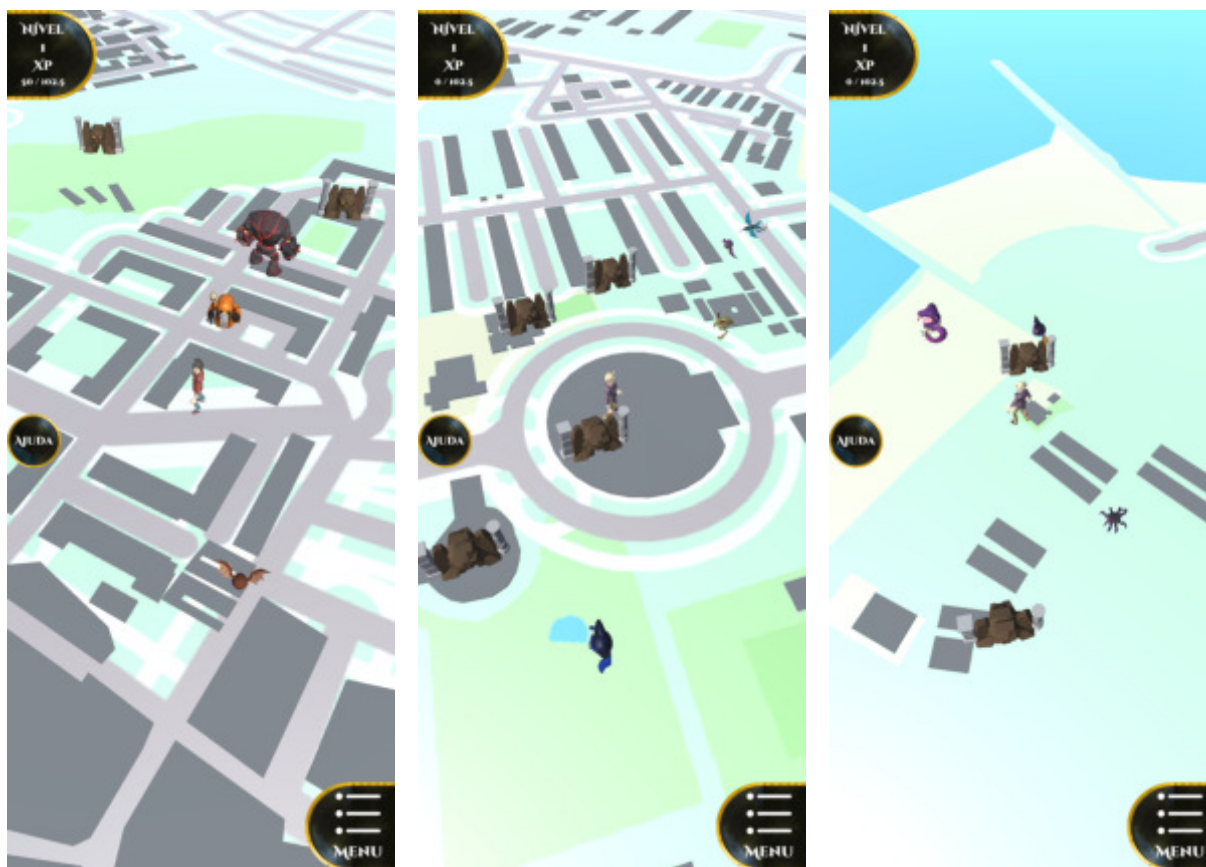
Figura 4.4: Exemplo de registo no jogo SquadUp.

Após a inserção dos dados corretamente, são apresentados 14 bustos de personagens para o jogador escolher, esta personagem será a representação do jogador no mapa de jogo. Caso este não tenha selecionado nenhuma e tente proceder para o jogo é apresentada uma mensagem temporária de que não escolheu ainda a sua personagem. Após o registo com sucesso o jogo passa para o cenário “Mapa-mundo”. Na Figura 4.4a é possível observar os bustos e na Figura 4.4b a mensagem temporária referida previamente.

4.3 Cenário Mapa-mundo

Após o cenário “Autenticação” o jogador procede para o cenário “Mapa-mundo”. Este é o principal cenário do jogo SquadUp. É a partir deste cenário que o jogador consegue aceder ao cenário “Captura”, ao cenário “Duelo” e ao cenário “Autenticação”. A GUI deste cenário é composta por:

1. Um mapa gráfico que representa a localização geográfica em tempo real do jogador;
2. Um painel que apresenta o nível e experiência/experiência necessária para evoluir;
3. Botão “Ajuda” que disponibiliza o acesso ao tutorial referente ao cenário “Mapa-mundo”;
4. Botão “Menu” que disponibiliza o acesso ao menu “Principal”.



(a) Prior Velho, Loures.

(b) Portela, Lisboa.

(c) Costa da Caparica, Almada.

Figura 4.5: Exemplos do mapa de jogo do SquadUp em localizações de Portugal.

O mapa é preenchido por vários elementos geográficos e gráficos:

1. Terrenos;
2. Relvados;
3. Parques;
4. Prédios;
5. Passeios;
6. Estradas;
7. Monstros;
8. Duelos;
9. Personagem do jogador.

Os três primeiros elementos referidos são sempre representados pela cor verde, mas conforme o seu tipo mais acentuada será a escuridão da cor associada, ou seja, o terreno é o elemento com a cor verde mais leve enquanto o parque é o elemento com a cor verde mais escura.

Os prédios são representados por uma cor cinzento-escuro e a sua representação gráfica foi limitada em altura para que não bloqueasse o acesso aos monstros. Os passeios foram associados à cor branca e as estradas à cor cinzento-claro. É possível observar a representação gráfica dos elementos acima referidos na Figura 4.5. Todos os monstros têm uma representação gráfica respetiva e a sua criação é explicada no capítulo “Monstros no Mapa-mundo”. Os Duelos têm uma representação gráfica única, sendo esta uma entrada para uma gruta em formato de caveira, como é possível de observar na Figura 4.7.

O jogo inicialmente iria ter a interação com os monstros e os locais de Duelo limitados por raio de distância ao jogador, mas com o acontecimento da pandemia foi implementado um sistema sem limites para que o jogador pudesse testar o jogo sem ter que pôr a sua saúde em risco. Após os vários confinamentos aplicados, foi necessário implementar o jogo sem limites para aceder ao modo de Captura e de Duelo, pois o jogador poderia estar confinado à sua residência e não ter a possibilidade de sair para locais públicos. A personagem que o jogador selecionou é posicionada no local onde este se situa geograficamente em tempo real.

4.3.1 Monstros no Mapa-mundo



Figura 4.6: Representação gráfica dos monstros no jogo SquadUp.

Quando o jogador entra no cenário “Mapa-mundo” são criados e posicionados monstros com respeito à localização da personagem do jogador. Existe dois métodos de criação de monstros, sendo estes os seguintes:

1. Monstros novos;
2. Monstros já existentes.

Os monstros já existentes são criados através da leitura dos dados de localização e identificação destes. Pelo jogo SquadUp partir do princípio em que o jogador precisa de capturar monstros para conseguir fazer Duelos e/ou evoluir os monstros que já tem capturados, foi necessário implementar o registo das localizações dos monstros previamente criados, para que o jogador não perdesse nenhum monstro sempre que fosse para outro cenário.

Esta abordagem é limitada por duas restrições sendo uma por data de calendário e a outra por um intervalo de tempo, ou seja, quando é feita a leitura dos dados dos monstros previamente criados para a sua criação, estes dados são removidos singularmente se não respeitarem as restrições referidas acima.

A criação de um monstro novo é feita após a criação dos monstros existentes, sendo a seleção do monstro a ser criado aleatória e a posição deste escolhida de maneira aleatória. Esta posição está limitada a um raio em redor da localização da personagem do jogador.

Os limites de posição evitam que os monstros fiquem em cima do jogador e evitam que fiquem a uma grande distância deste. Após o processo de verificação da posição com respeito à posição da personagem do jogador são verificadas as posições dos monstros já criados para que não existam posições duplicadas, evitando duas representações gráficas sobrepostas.

Quando o jogador pressiona um monstro ele é transportado para o cenário “Captura” para que possa tentar capturar o monstro. Caso o jogador não tenha sucesso na captura do monstro, o monstro será recriado à mesma no cenário “Mapa-mundo”. Caso contrário os dados de localização correspondentes ao monstro são retirados.

Após a criação inicial de monstros é também posta em execução duas rotinas temporárias, sendo uma correspondente à criação de um monstro e a outra à destruição de um dos monstros criados. Com estas rotinas é possível disponibilizar ao jogador uma diversidade/rotação de monstros e evitar o elevado preenchimento do mapa-mundo.

4.3.2 Duelos no Mapa-mundo



Figura 4.7: Representação gráfica de um Duelo no mapa de jogo.

As representações gráficas dos Duelos são criadas via o serviço MapBox. Através deste serviço é possível nomear certos tipos de locais geográficos como posições de interesse, disponibilizando uma representação gráfica para que estas posições tenham marcos visuais.

O Duelo foi criado através deste processo e a sua representação gráfica foi criada com um visual agressivo para representar o ambiente que este invoca, como é possível observar na Figura 4.7.

Foi escolhido os parques com relvados como tipo de local geográfico, para associar a posição de interesse ligada ao Duelo. Foi feita esta escolha porque, em geral, os parques com relvados proporcionam momentos de lazer e abstração dos problemas às pessoas. Tornando-se assim sítios ideais para que os jogadores possam passear, relaxar e ao mesmo tempo sentir a interação do mundo virtual do jogo SquadUp com o mundo real.

4.3.3 Tutoriais

Ao passar pela primeira vez para o cenário Mapa-mundo é apresentado um tutorial que introduz ao jogador os aspetos principais do jogo e do seu funcionamento. Estes aspetos são aprofundados em outros tutoriais que são disponibilizados no cenário “Mapa-mundo”, “Captura” e “Duelo”. Existem outros tutoriais que ajudam o jogador a utilizar/interpretar os dados que lhe são apresentados nas várias opções do menu “Principal” situado no cenário “Mapa-mundo”.



(a) Cenário Mapa-mundo.

(b) Menu Linha de Habilidades.

(c) Cenário Captura.

Figura 4.8: Exemplo de vários tutoriais de várias partes do jogo SquadUp.

Pode-se observar na Figura 4.8 vários exemplos de tutoriais disponibilizados no jogo SquadUp. Estes tutoriais estão sempre disponíveis ao jogador através do botão “Ajuda”. Sempre que o jogador aceder a um tutorial ele pode avançar/retroceder pelos vários tópicos apresentados.

O GUI aplicado aos tutoriais apresenta um conjunto de funcionalidades na base, estas são:

1. Botão “Fechar”;
2. Toggle “Desligar Ajuda”;
3. Botão “Anterior”;
4. Botão “Próximo”.

A opção “Desligar Ajuda” serve para o jogo não apresentar ao jogador o tutorial, sempre que entra no menu ou cenário com um tutorial associado. A opção vem ativa por predefinição. Os botões “Anterior” e “Próximo” servem para navegar pelos múltiplos ecrãs que cada tutorial tem. Quando o jogador chega ao último ecrã o botão “Próximo” muda a sua legendagem para “Fechar”, tendo a mesma funcionalidade que a primeira opção mencionada na lista acima. A diferença é que o botão “Fechar” é dedicado à função de encerrar o tutorial seja em qualquer ecrã.

4.3.4 Menu Principal

Quando o jogador acede ao menu “Principal” é apresentado um tutorial que descreve o objetivo de cada opção na lista de opções do menu “Principal” e também o funcionamento do cabeçalho localizado em cada um dos menus secundários.



Figura 4.9: GUI do Menu Principal.

O menu “Principal” é o menu que permite o acesso aos outros menus e submenus. Através deste menu também são iniciados vários fatores programáticos que influenciam a transição de painel para painel, ou seja, o menu principal serve de raiz para todos os outros menus e ao mesmo tempo mantém o controlo de transição pelos menus. Com a opção “Sair da Sessão” o jogador consegue sair do cenário “Mapa-mundo” e voltar para o cenário “Autenticação”. Ao utilizar esta opção do menu, o jogador irá desligar a conta que utilizou para entrar no jogo e poderá voltar a aceder utilizando a mesma ou criar uma conta nova através do processo já explicado. Todas as outras opções do Menu Principal irão ser descritas em seguida, pela ordem do topo para a base, começando pelo Menu Definições e acabando no Menu Linha de Habilidades.

A GUI do menu “Principal” é composta por:

- Cinco botões de acesso a outros menus;
- O botão “Sair Da Sessão” para desligar a conta do jogador ao jogo SquadUp;
- O botão “Ajuda” para aceder ao tutorial referente ao menu “Principal”;
- O botão “Voltar” para retornar ao mapa de jogo.

Apesar de haver cinco opções acessíveis via o menu “Principal”, algumas destas opções representam entradas para submenus. A estrutura de menus é a seguinte:

- Menu Principal,
 - Menu Definições,
 - Menu Bestiário,
 - * Menu Monstro.
 - Menu Jogador,
 - Menu Esquadrão,
 - Menu Habilidades,
 - * Menu Linha de Habilidades.

Em seguida é descrito o propósito de cada menu e o motivo da existência dos submenus.

4.3.4.1 Menu Definições

O jogo SquadUp disponibiliza o menu de definições que permite controlar o volume da música ambiente, dos efeitos sonoros e escolher o idioma. Para este menu não existe nenhum tutorial, mas a interface do menu é bastante simplificada com o objetivo de criar uma interação fácil e intuitiva para o jogador. Este menu é apresentado tanto no cenário “Mapa-mundo” como no cenário “Autenticação”.

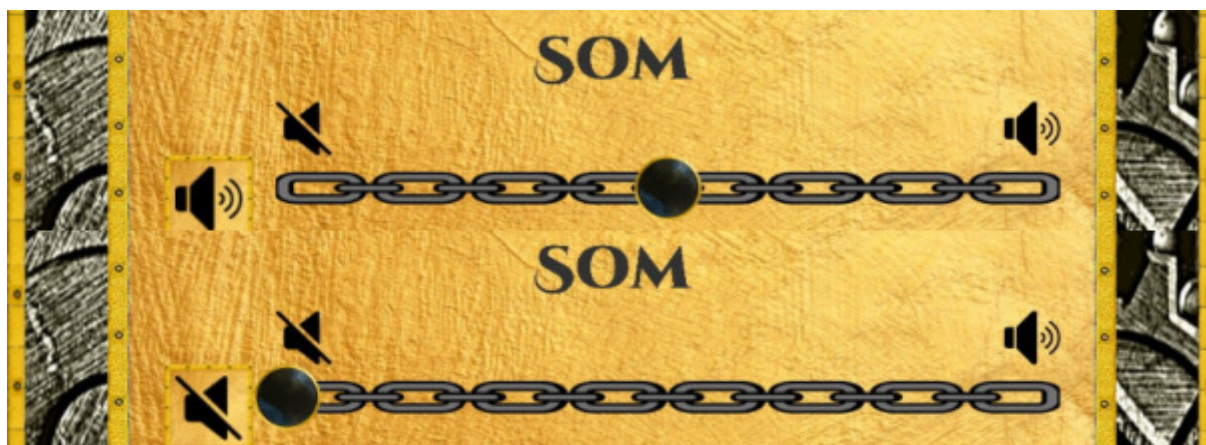


Figura 4.10: Demonstração do ícone do botão Mute.

Para o jogador controlar o volume de música ambiente ou efeitos sonoros são disponibilizados dois mecanismos deslizantes. Ambos os mecanismos têm na ponta esquerda um ícone representativo de volume mínimo e na ponta direita um ícone representativo de volume máximo. Assim o jogador consegue associar a direção de movimento do botão deslizante à modificação de volume sonoro.

Caso o jogador simplesmente queira retirar o som tanto da música ambiente e/ou dos efeitos sonoros é também disponibilizado a opção de *Mute* à esquerda dos mecanismos. Este botão irá mudar de ícone sempre que o jogador o utilizar, dando ao jogador o conhecimento do estado de volume sonoro para a música ambiente e/ou para os efeitos sonoros. É possível observar na Figura 4.10 os dois ícones do botão *Mute*, representando se este está ou não ativado assim como os mecanismos deslizantes.



(a) Idioma Inglês.

(b) Seleção de Idioma.

(c) Idioma Português.

Figura 4.11: Exemplo de alteração de idioma Definições.

Para o jogador alterar o idioma de jogo, tem que carregar na caixa que apresenta a língua utilizada no momento. Ao ser pressionada a caixa irá expandir-se como uma lista, apresentando ao jogador os idiomas suportados pelo jogo. O jogador pressiona outra vez no idioma que deseja selecionar, se este não quiser mudar a linguagem basta apenas voltar para o menu principal que a linguagem não será mudada da definida no momento. O processo de alteração de idioma do jogo SquadUp é demonstrado pela Figura 4.11.

4.3.4.2 Menu Bestiário

Quando a opção “Bestiário” é escolhida pela primeira vez é apresentado o tutorial referente à navegação da lista de monstros dentro do menu “Bestiário”, à utilização de filtros para uma procura de monstros mais refinada e também ao significado dos *layouts* introduzidos nos painéis dos monstros. Estes *layouts* foram introduzidos para que o jogador conseguisse reconhecer o tipo, subtipo e raridade do monstro, observando os detalhes do painel de monstro.



Figura 4.12: GUI do Menu Bestiário.

A interface do Menu Bestiário é composta por:

- Quatro botões dedicados aos tipos de poder;
- Cinco botões dedicados às raridades de monstro;
- O botão “Ajuda” para aceder ao tutorial referente ao menu “Bestiário”;
- O botão “Voltar” para retornar ao menu “Principal”;
- Uma lista preenchida com painéis de monstro de todos os monstros no jogo.

Como é possível observar na Figura 4.12 a interface gráfica do Menu Bestiário disponibiliza ao jogador nove botões dedicados à filtragem da lista de monstros. Esta possibilidade de filtrar tem como objetivo excluir todos os monstros da lista que não partilhem o tipo de poder selecionado e/ou a raridade selecionada.

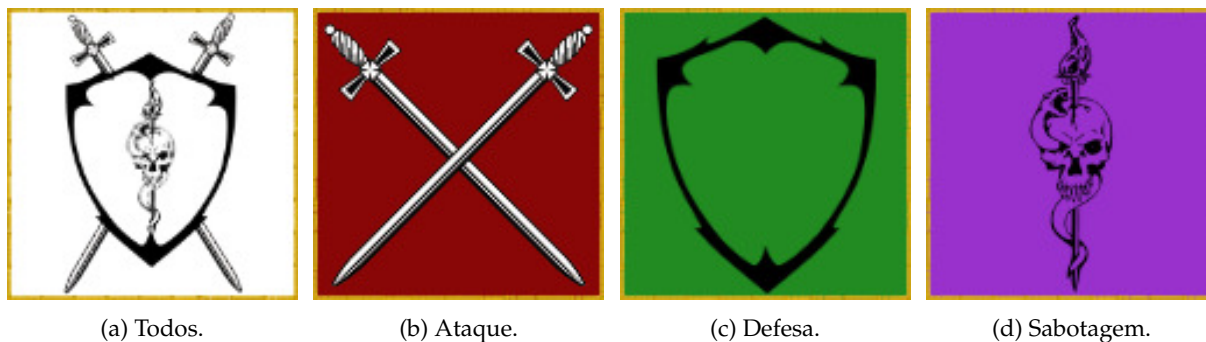


Figura 4.13: Botões de filtragem por tipo de poder.

A primeira linha de botões é dedicada à filtragem por tipos de poder. Como podemos observar na Figura 4.12, cada botão tem um ícone que corresponde ao tipo de poder que representa. Em seguida, é apresentada a descrição de cada botão (em que 1 representa o botão mais à esquerda e 4 o botão mais à direita):

1. Todos os tipos de monstros;
2. Só os monstros do tipo de ataque;
3. Só os monstros do tipo de defesa;
4. Só os monstros do tipo sabotagem.

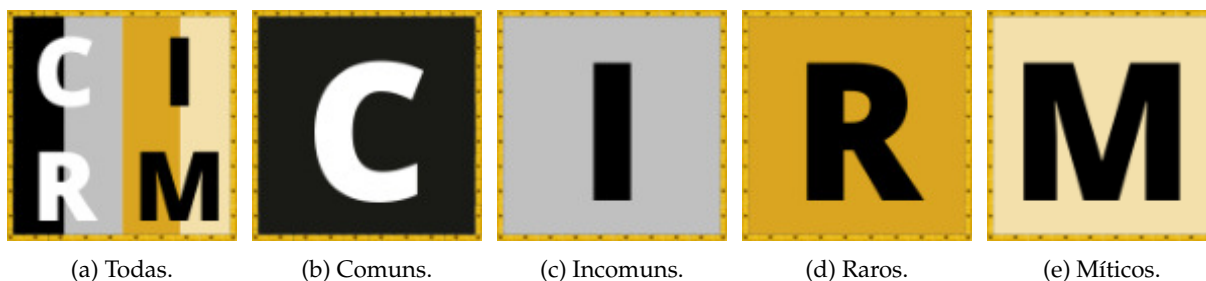


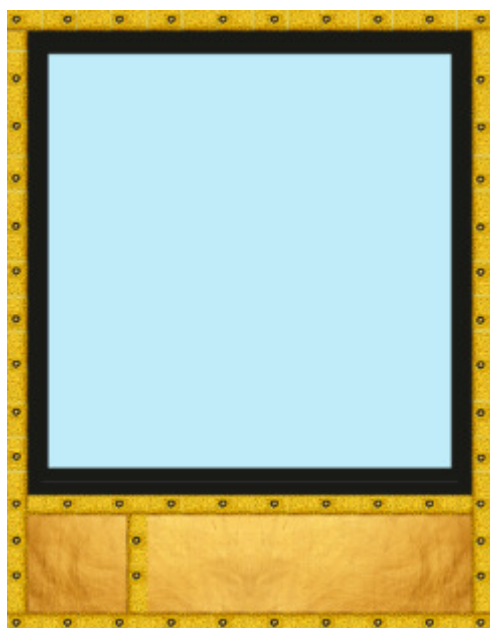
Figura 4.14: Botões de filtração por tipo de raridade.

A segunda linha de botões é dedicada à filtragem por tipos de raridade. Como podemos observar na Figura 4.14, cada botão tem um ícone que corresponde ao tipo de raridade que representa. A descrição da filtragem de cada botão é descrita em seguida (em que 1 representa o botão mais à esquerda e o 5 o botão mais à direita):

1. Todas as raridades de monstros;
2. Só os monstros com raridade comum;
3. Só os monstros com raridade incomum;
4. Só os monstros com raridade rara;
5. Só os monstros com raridade mítica.

O menu “Bestiário” foi criado para que o jogador possa visualizar os monstros que capturou e os que observou. O jogador também pode ver as características dos monstros que capturou clicando no busto destes. Todos os monstros são embutidos em uma lista que o jogador pode navegar, cada monstro é representado através de um painel de monstro que apresenta várias características do monstro como referido no capítulo “Painel de Monstro”.

Todos os painéis de monstro estão espaçados para que o jogador possa navegar entre a lista e não efetue cliques sem intenção. O jogador ao carregar no ícone do monstro no painel de monstro abre o submenu “Monstro” que irá apresentar as características do monstro.



(a) Painel de Monstro Visto/Capturado.



(b) Painel de Monstro não visto/capturado.

Figura 4.15: Exemplo de painéis de monstro.

Como é possível observar na Figura 4.15a os monstros que são vistos ou capturados pelo jogador irão ser apresentados por este painel de monstro, enquanto todos os outros monstros que não tenham sido ainda vistos serão mantidos anónimos utilizando o painel de monstro, representado pela Figura 4.15b. O jogador só consegue aceder ao menu “Monstro” dos monstros que capturou clicando no busto destes. Todos os outros monstros que só viu ou estão anónimos não é permitido o acesso.

A estrutura gráfica do painel de monstro é explicada em mais detalhe no capítulo “Painel de Monstro”, para melhor perceção das várias camadas de cores e símbolos utilizados neste com o objetivo de identificação do monstro e a sua tipologia.

É através do menu “Bestiário” que o jogador pode começar por planear a sua estratégia de combate para Duelo verificando a quantidade/diversidade de monstros que possui e as melhores combinações de monstros dos três tipos.

4.3.4.3 Menu Monstro

Este menu é apenas acessível através do “Bestiário” como já referido anteriormente, isto deve-se ao facto de que a interface gráfica do menu “Monstro” é reutilizada para cada Monstro na lista do menu “Bestiário”, mudando apenas a imagem de fundo e o painel de características do monstro conforme o seu tipo.

A GUI do menu “Monstro” é composta por:

- O botão “Ajuda” para aceder ao tutorial referente ao menu “Monstro”;
- O botão “Voltar” para retornar ao menu “Bestiário”;
- O botão “Adicionar” para adicionar o monstro ao esquadrão do jogador;
- O painel com as características do monstro seleccionado.



(a) Variação Ataque.

(b) Variação Defesa.

(c) Variação Sabotagem.

Figura 4.16: Variações do GUI do Menu Monstro.

Como é possível observar na Figura 4.16 o jogador pode adicionar o monstro ao Esquadrão, utilizando o botão dedicado a esta ação posicionado no canto superior direito. O fundo de ambiente do menu “Monstro” muda conforme o tipo do monstro, como é possível ver pelas figuras Figura 4.16a, Figura 4.16b, Figura 4.16c.

Os dados do monstro são disponibilizados e apresentados através do painel, estando separados pelos dados Pessoais referentes à tipologia e os dados de combate referentes aos valores dos poderes do monstro. Os dados são os seguintes:

- Dados Pessoais,
 - Tipo,
 - Subtipo,
 - Quantidade de vezes que foi visto,
 - Quantidade de vezes que foi apanhado,
 - Quantidade de vezes que precisa de ser apanhado para evoluir .

- Dados de Combate,
 - Poder Principal,
 - * Valor Base,
 - * Valor Extra conforme o nível.
 - Poder Secundário,
 - * Valor Base,
 - * Valor Extra conforme o nível.
 - Poder Secundário fictício (só para monstros do tipo sabotagem),
 - * Valor Base,
 - * Valor Extra conforme o nível.
 - Vida,
 - * Valor Base,
 - * Valor Extra conforme o nível.

Estas são as características que definem um monstro, é a partir destes dados que o jogador consegue determinar quantos monstros mais terá que capturar para o evoluir para o nível seguinte. Com os valores dos poderes do monstro apresentados, o jogador consegue determinar e avaliar se o monstro se aplica à estratégia que concebeu e se consegue criar uma boa sinergia com os restantes monstros do esquadrão.

4.3.4.4 Menu Jogador

Quando acede à opção Jogador pela primeira vez é apresentado um tutorial que demonstra ao jogador os vários dados que lhe são apresentados e como os interpretar.

Este menu foi criado com o propósito de fornecer ao jogador uma opção onde pode ver as suas características pessoais, os seus feitos assim como os seus pertences dentro do jogo SquadUp. Esta informação é transmitida ao jogador pelo painel que apresenta por tópicos os vários dados do jogador.

O menu não disponibiliza nenhuma interação sem ser a predefinida pelo cabeçalho, reforçando o objetivo principal do menu "Jogador" ser um menu de acesso a dados pessoais do jogador referente ao jogo.

A GUI do menu Jogador é composta por:

- O botão "Ajuda" para aceder ao tutorial referente ao menu "Jogador";
- O botão "Voltar" para retornar ao menu "Principal";
- O painel com as características do jogador.

O painel do menu "Jogador" é constituído pelos vários campos de informação:

- Experiência,
 - Experiência atual do jogador,
 - Experiência a ganhar para o jogador evoluir de nível.
- Pontos,
 - Pontos de habilidades por usar,
 - Total de pontos de habilidades que o jogador possui.
- Monstros,
 - Monstros que o jogador já visualizou,
 - Monstros que o jogador não capturou,
 - Total de monstros que o jogador capturou,
 - * Monstros capturados de raridade comum,
 - * Monstros capturados de raridade incomum,
 - * Monstros capturados de raridade rara,
 - * Monstros capturados de raridade mítica.
- Duelos,
 - Total de duelos que o jogador fez,
 - * Quantidade de duelos que o jogador fugiu,
 - * Quantidade de duelos que o jogador perdeu,
 - * Quantidade de duelos que o jogador ganhou.
 - Rácio de vitória do jogador possui em duelos,
- Poções,
 - Quantidade de poções de vida que o jogador tem,
 - Quantidade de poções de inspiração que o jogador tem.

EXPERIÊNCIA	
ACTUAL	50
PARA EVOLUIR	52.5
PONTOS	
POR USAR	2
TOTAL	3
MONSTROS	
VISTOS	11
NÃO CAPTURADOS	2
CAPTURADOS	9
COMUM	1
INCOMUM	3
RARO	2
MÍTICO	3
RÁCIO DE CAPTURA	0
BATALHAS	
TOTAL	3
QUE FUGIU	0
PERDIDAS	2
GANHAS	1
RÁCIO DE VITÓRIA	33
POÇÕES	
VIDA	23
INSPIRAÇÃO	20

Figura 4.17: GUI do Menu Jogador.

Com acesso a esta informação disponibilizada pelo menu “Jogador”, o jogador consegue verificar se a sua estratégia está a ter sucesso ou não verificando o seu rácio de vitória em comparação ao valor que previamente possuía. Também consegue verificar a quantidade de poções que possui e se precisa de obter mais poções antes de tentar fazer novos Duelos.

Pode observar também se aplicou todos os pontos de habilidades ou se falhou algum e assim completar a aplicação de pontos pelas habilidades, não deixando para trás a possível vantagem que obtém das habilidades para os Duelos. O cabeçalho também apresenta o nome que o jogador adotou para o jogo SquadUp e o nível em que este se encontra.

Logo este menu foi implementado com o propósito de ser um acesso à informação pessoal referente ao jogo SquadUp, não possui nenhuma interação sem ser a que o cabeçalho do menu disponibiliza.

4.3.4.5 Menu Esquadrão

Quando o jogador acede ao menu “Esquadrão” a interface apresenta três painéis, um para cada tipo de poder, como é possível observar na Figura 4.18b, que apresenta os três painéis vazios sem qualquer preenchimento de dados. O menu “Esquadrão” é composto pelos seguintes elementos:

- O botão “Ajuda” para aceder ao tutorial referente ao menu “Esquadrão”;
- O botão “Voltar” para retroceder para o menu “Principal”;
- Três painéis para os três monstros associados ao Esquadrão.



(a) Tutorial do Menu.

(b) Com Esquadrão Vazio.

(c) Com Esquadrão Cheio.

Figura 4.18: Tutorial e variações do GUI do Menu Esquadrão.

Um Esquadrão é composto por um monstro do tipo “Ataque”, um do tipo “Defesa” e um do tipo “Sabotagem”. Assim quando o Jogador adicionar um monstro ao Esquadrão este irá aparecer no painel referente ao tipo de poder deste, sendo possível observar um exemplo na Figura 4.18c.

Cada painel apresenta as várias características do monstro, assim como os valores dos seus poderes e vida. Estes valores são apresentados em vários formatos, para que o jogador possa observar o que constitui a totalidade de cada poder.

O tutorial referente ao menu “Esquadrão” apresenta ao jogador que tipo de informação é apresentada no menu, como é possível observar na Figura 4.18a que apresenta uma das partes do tutorial do menu “Esquadrão”.

Os valores apresentados e os monstros usados no tutorial são predefinidos para exemplificação, mas caso o jogador tenha monstros adicionados ao esquadrão, serão estes os usados para exemplificação no tutorial. Esta abordagem permite ao jogador ter melhor percepção dos dados que deseja compreender devido à associação do exemplo com a escolha de monstros que este fez.

Esta mudança no tutorial é conseguida através da modularidade aplicada a cada painel e às suas tabelas. Cada tabela tem um gestor de valores dedicado e cada painel gere todos os gestores de tabelas, passando-lhe os dados necessários.

4.3.4.6 Menu Habilidades

O menu “Habilidades” é o menu responsável pelo acesso à linha de habilidades dos três tipos de poderes do jogo SquadUp. O menu “Habilidades” é composto por:

- O botão “Ajuda” para aceder ao tutorial referente ao menu “Habilidades”;
- O botão “Voltar” para retroceder para o menu “Principal”;
- Três botões para acesso ao Menu Linha de Habilidades para cada tipo de poder.



(a) Tutorial do Menu.



(b) Menu Habilidades.

Figura 4.19: Tutorial e GUI do Menu Habilidades.

Cada tipo de poder é representado por um logótipo e o seu nome dentro do logótipo como é possível verificar na Figura 4.19b. Cada logótipo é também um botão responsável por dar o acesso ao menu “Linha de Habilidades”, ou seja, se o jogador pressionar o logótipo “Ataque” o jogador irá aceder ao menu “Linha de Habilidades” de “Ataque”.

Este menu “Linha de Habilidades” é apenas acessível através desta cadeia de eventos, o jogador não consegue ter acesso às linhas de habilidades sem aceder ao menu “Habilidades”. Quando o jogador acede pela primeira vez ao menu Habilidades, é-lhe introduzido uma tutorial que o introduz ao que cada botão representa e o que acontece ao ser pressionado. Podemos observar na Figura 4.19a a descrição do efeito do botão com a legenda Ataque, no tutorial referente ao Menu Habilidades.

4.3.4.7 Menu Linha de Habilidades

Quando o menu “Linha de Habilidades” é acedido pela primeira vez, é apresentado um tutorial que introduz ao jogador as funcionalidades do cabeçalho e como interpretar/interagir com o conteúdo da interface da habilidade. O GUI do menu é utilizado para as três variações, mudando apenas o título e as habilidades atribuídas à lista. A implementação de uma GUI mutável para as três variações foi aplicada com o objetivo de manter as alterações, nos valores das habilidades, executadas pelo jogador sem ter que as guardar de uma forma permanente. Assim é possível diminuir o processamento computacional e ao mesmo tempo evitar problemas de registo de dados.



(a) Variação Ataque.

(b) Variação Defesa.

(c) Variação Sabotagem.

Figura 4.20: Variações do GUI do Menu Linha de Habilidades.

Como é possível observar na Figura 4.20 o menu “Linha de Habilidades” é composto pelos seguintes elementos:

- O botão “Ajuda” para aceder ao tutorial referente ao menu “Linha de Habilidades”;
- O botão “Voltar” para retroceder ao menu “Habilidades”;
- Uma lista de habilidades.

O menu “Linha de Habilidades” possui um cabeçalho diferente de todos os outros menus, pois o botão “Ajuda” foi substituído por quatro contadores. Cada contador apresenta o número de pontos aplicados seguido do número máximo de pontos que se pode aplicar. Como é possível observar na Figura 4.21



Figura 4.21: Cabeçalho do Menu Linha de Habilidades de Sabotagem.

as cores dos contadores indicam a que linha de habilidades é que o contador pertence. O contador com fundo a preto é o que indica a quantidade de pontos que o jogador aplicou e o máximo que este pode aplicar. A quantidade máxima de pontos que o jogador pode aplicar irá aumentar à medida que o jogador evolui de nível.

Cada linha de habilidades é composta por dez habilidades, logo as variações do menu “Linha de Habilidades” irão sempre demonstrar dez habilidades na lista. Esta lista é interativa permitindo ao jogador percorre-la através dos movimentos de *swipe down/up*. Cada habilidade é representada por um painel dedicado, como é possível observar na Figura 4.23.

No jogo SquadUp a maioria das habilidades fazem combo com duas habilidades, mas existem habilidades que só fazem combo com uma habilidade, isto implica que o painel tem que ajustar a apresentação de dados referente às habilidades combo da habilidade que este representa. Para isto é posicionada uma imagem com um “X” no lugar da segunda habilidade combo, passando como mensagem que a habilidade não contém uma segunda habilidade combo. Como é possível observar na Figura 4.22 a imagem ocupa o espaço por inteiro de uma habilidade combo, ou seja, o seu ícone e contador.



Figura 4.22: GUI de uma Habilidade com apenas uma Habilidade Combo.



Figura 4.23: Composição da GUI do painel de Habilidade.

Como é possível observar na Figura 4.23 a interface de uma habilidade é composta por vários elementos:

1. Botão para diminuir pontos aplicados;
2. Botão para aumentar pontos aplicados;
3. Título da habilidade;
4. Descrição da habilidade;
5. Ícone associado à habilidade;
6. Contador de pontos aplicados/pontos máximos que se pode aplicar;
7. Indicador de combo;
8. Ícone da habilidade combo;
9. Contador de pontos da habilidade combo.

A habilidade apresenta ao jogador o efeito desta através da descrição. Ao serem aplicados ou retirados pontos à habilidade utilizando os elementos 1 e 2, os valores apresentados na descrição mudam conforme a quantidade de pontos aplicados. A habilidade tem um valor de efeito e caso recorra ao mecanismo de chance (probabilidade) para ser ativada também possui um valor de chance. A descrição da habilidade poderá ter até três valores apresentados, o de efeito, o de chance e o de combo.

Como referido acima, cada habilidade faz combo com outras habilidades, esta mecânica de jogo é explicada no capítulo "Mecânica de Habilidades". Para que o jogador possa saber quais as habilidades que fazem "combo", são incluídos os ícones destas e os respetivos contadores de pontos (elementos 8 e 9). Quando o "combo" é ativado a palavra "Combo" (elemento 7) na interface da habilidade muda de cor para representar a ativação do mecanismo. Também é acrescentado um excerto de texto à descrição indicando o benefício extra que o combo fornece à habilidade.

4.4 Cenário Captura

Ao pressionar um monstro no mapa de jogo dentro do cenário “Mapa-mundo” o jogador é transportado para o cenário “Captura”. Este cenário é responsável por dar a hipótese de capturar o monstro, através do arremesso de uma órbita ao monstro. O cenário “Captura” não faz ligação com mais nenhum cenário sem ser o cenário “Mapa-mundo” ao qual é dependente em termos de acesso. Toda a informação necessária para o jogador saber como interagir com o cenário é disponibilizada e apresentada ao jogador de forma automática aquando do primeiro acesso deste ao cenário “Captura”.

O tutorial deste cenário introduz ao jogador as duas opções iniciais, “Fugir” ou “Continuar”, que lhe são disponibilizadas sempre que entra no cenário “Captura”, como é possível de observar na Figura 4.24a.



(a) Introdução do menu inicial.

(b) Introdução do contador.

(c) Introdução da órbita.

Figura 4.24: Tutorial do Cenário Captura.

Inicialmente o jogador pode fugir e voltar ao cenário Mapa-mundo, mas se o jogador desejar continuar para poder capturar o monstro, ele não pode sair do cenário até gastar todas as órbitas de captura ou conseguir capturar o monstro. É também introduzido o contador de órbitas (Figura 4.24b) que a interface disponibiliza, o que a órbita de captura (Figura 4.24c) representa e como é que o jogador consegue capturar o monstro.

A GUI do cenário Captura é composta pelos seguintes elementos:

- Um mapa de jogo;
- O monstro selecionado pelo jogador;
- Uma órbita de captura;
- Um contador de órbitas de captura;
- Três menus.



(a) Opções iniciais.

(b) Exemplo da opção Fugir.

(c) Tentativa de captura.

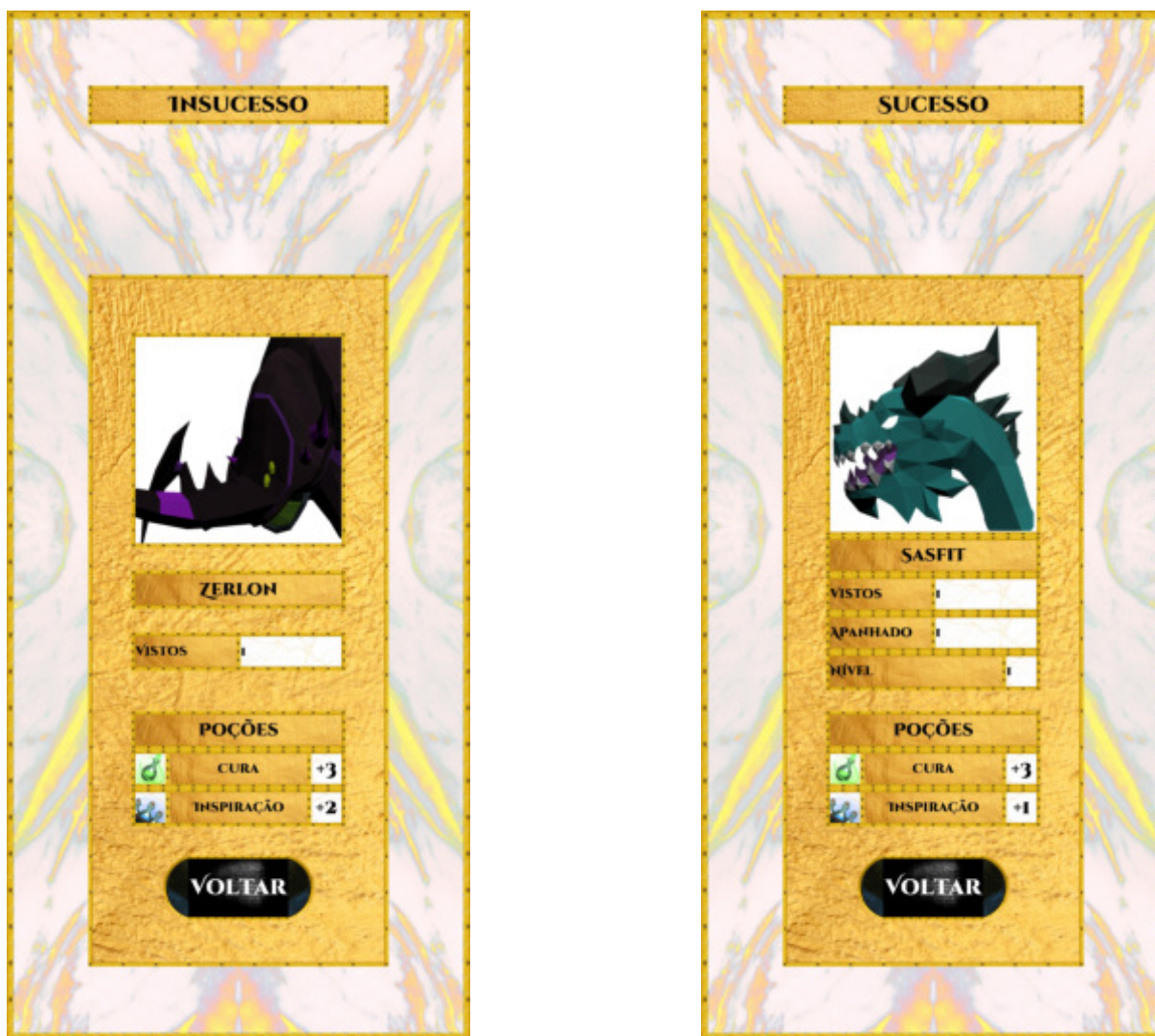
Figura 4.25: Exemplificação das ações possíveis no cenário Captura.

Quando o jogador tenta capturar um monstro, é necessária uma cadeia de movimentos físicos da parte deste para que o arremesso da órbita de captura tenha um ângulo ideal em que a zona de aterragem da órbita seja o topo do monstro. Esta mecânica é mais aprofundada no capítulo “Mecânicas de Captura de Monstro”.

Caso o jogador consiga acertar no monstro, é apresentado ao utilizador um ecrã de sucesso e em seguida é retornado ao cenário “Mapa-mundo”, caso o utilizador falhe também é apresentado um ecrã, mas o de insucesso e este é retornado de volta ao cenário “Mapa-mundo”.

O cenário “Captura” disponibiliza três menus ao jogador sendo o primeiro o menu com as opções “Fugir” e “Continuar” e os outros dois menus são os menus de resultado de captura.

Os menus de resultados apresentam ao jogador o resultado da tentativa de captura do monstro assim como outros dados, como é possível observar na Figura 4.26.



(a) Insucesso de Captura.

(b) Sucesso de Captura.

Figura 4.26: GUI do Menu Resultado do Cenário Captura.

No caso de insucesso de captura do monstro, o jogador é informado da quantidade de vezes que já avisou o monstro que tentou capturar, enquanto no caso de sucesso o jogador recebe a mesma informação e mais a quantidade de vezes que já capturou o monstro e o nível do monstro pertencente ao “Bestiário” do jogador.

A descrição do nível do monstro é dada para que o jogador consiga aperceber-se se o monstro que possui evoluiu de nível com a captura que conseguiu ou não.

Em ambos os casos (sucesso ou insucesso) o jogador recebe sempre uma quantidade aleatória dos dois tipos de poções, sendo que no caso de insucesso a quantidade aleatória tem um limite máximo inferior ao limite máximo designado para o caso de sucesso de captura.

4.5 Cenário Duelo



(a) Opções iniciais.

(b) Fase de defesa.

(c) Fase de ataque.

Figura 4.27: GUI dos Menus do Cenário Duelo.

Quando o jogador pressiona o símbolo do Duelo no mapa de jogo dentro do cenário “Mapa-mundo” e se o seu esquadrão estiver completo com um monstro de cada tipo de poder, o jogador é transportado para o cenário “Duelo”. É neste cenário que o jogador pode utilizar o seu esquadrão para enfrentar outros esquadrões e ganhar a experiência necessária para evoluir de nível.

O tutorial referente ao cenário “Duelo” é apresentado ao jogador quando este é transportado pela primeira vez para o Duelo. O tutorial explica ao jogador a utilidade das poções que ganhou nas capturas, as fases de “Ataque” e de “Defesa” que compõem a estrutura de uma ronda e o significado dos dois painéis que aparecem por cada fase.

O GUI do cenário “Duelo” é composta pelos seguintes elementos:

- Um mapa de batalha;
- O botão “Ajuda” para aceder ao tutorial referente ao cenário “Duelo”;
- Seis monstros;
- Dois painéis cada um com três barras de vida para os respetivos monstros do esquadrão representante;

- Duas caixas de texto de identificação de esquadrão para os painéis referidos acima;
- Quatro menus;
- Dois painéis de análise do combate.

No cenário “Duelo” o esquadrão do jogador é colocado à esquerda do mapa de jogo e o esquadrão inimigo é colocado à direita do mapa de jogo, posicionando os monstros frente a frente uns dos outros.

O inimigo que o jogador enfrenta sempre que entra no cenário “Duelo” é um inimigo virtual e composto por um esquadrão de monstros e uma lista de habilidades própria. As habilidades que o inimigo pode utilizar são as mesmas que o jogador pode utilizar, mas a atribuição de pontos é feita aleatoriamente assim como a quantidade de pontos a serem colocados. A quantidade de pontos que o inimigo pode aplicar nas habilidades é a mesma quantidade que o jogador pode aplicar. Após a aplicação de pontos pelas habilidades é feita a construção do esquadrão através da seleção dos monstros. Os monstros pertencentes ao esquadrão inimigo são escolhidos aleatoriamente para cada tipo de poder. Cada monstro do esquadrão inimigo é configurado com o mesmo nível que o monstro do mesmo tipo, do esquadrão do jogador.

O primeiro menu a aparecer ao jogador é o menu com as opções de “Fugir” e “Continuar”, idêntico em funcionamento ao menu inicial do cenário “Captura”. Se o jogador decidir continuar o Duelo, o menu inicial é substituído pelo menu de combate.

Como é possível observar na Figura 4.27b e Figura 4.27c, este menu disponibiliza três opções ao jogador, sendo estas as seguintes:

1. Usar poção de vida;
2. Usar poção de inspiração;
3. Atacar/defender.

A poção de vida recupera 25% da vida de cada monstro no esquadrão do jogador, mas só para aqueles que não estão desmaiados. A poção de inspiração adiciona 10% de chance de sucesso às habilidades que utilizam o mecanismo de chance para serem ativadas. As poções só podem ser usadas uma vez por cada fase, de maneira a evitar o clique duplo e não proporcionar 100% de chance de sucesso às habilidades.

Por cada ronda num Duelo o jogador tem que atacar e defender, logo a terceira opção do menu irá mudar de legenda conforme a fase em que o jogador está. Quando o jogador começa a primeira ronda do Duelo, o jogo SquadUp determina de maneira aleatória quem é que ataca primeiro, logo tanto o jogador como o inimigo podem atacar ou defender inicialmente.

Ao longo das fases os painéis com as barras de vida irão mudar em preenchimento e cor conforme a quantidade de vida do monstro que representam. A mudança de cor é aplicada para que o jogador reconheça o estado de saúde dos monstros via a percepção visual.

Após cada fase são sempre apresentados dois painéis de análise de combate, estes painéis têm como objetivo informar o jogador dos vários acontecimentos gerados. Como é possível observar na Figura 4.28 e Figura 4.29, existem dois tipos de painel de análise de dados, cada um com uma finalidade distinta.

ATACANTE (TU)				RONDA 1		DEFENSOR (BOT)			
		ATAQUE BASE	ATAQUE TOTAL			DEFESA BASE	DEFESA TOTAL		
		33.4	51			16.1	8.6		
-5.8	-0	+0	+0	+2.8	+50	+20	+35		
-0	+11.8	-13.7		-6.6	-17.5	-3.7			
		ATAQUE BASE	ATAQUE TOTAL			DEFESA BASE	DEFESA TOTAL		
		0	16.5			0	16.3		
-3.8	+4.4	-3.8	-9	+2.4	-16.3	-0	-1		
+3.6	+3.8			+2.3	-0				
DANO COMBINADO			83.9	CONTINUAR		33	DEFESA COMBINADA		

Figura 4.28: Exemplo do painel de análise do Ataque e Defesa.

O primeiro painel(Figura 4.28) é composto pelos seguintes elementos:

- Identificador numérico de ronda;
- Placas identificadores do atacante e do defensor;
- Um sub painel dedicado a cada monstro de cada esquadrão;
- Placa informativa da quantidade de dano combinado;
- Placa informativa da defesa combinada;
- O botão “Continuar” para o jogador prosseguir para o seguinte painel.

O primeiro painel apresenta ao jogador os valores base de ataque e de defesa assim como os valores na totalidade de ataque e de defesa para cada monstro. Os valores de ataque e de defesa na totalidade são apresentados devido a estes serem complementados pela quantidade extra atribuída por cada nível e também pelas habilidades.

Cada sub painel possui os ícones das habilidades que afetam diretamente o valor de ataque ou de defesa do tipo de monstro referido no sub painel. Estas habilidades são representadas pelos seus ícones. Sendo assim o jogador consegue associar qual é a habilidade e o efeito desta sobre o valor base de ataque ou defesa do monstro.

As habilidades representadas em cada sub painel são as habilidades do jogo SquadUp e como já referido o inimigo também consegue usar estas habilidades. Sendo assim os monstros do jogador também serão afetados pelas habilidades do inimigo. Isto deve-se ao facto de haver habilidades cujos efeitos são negativos para os monstros do esquadrão oponente.

O jogador consegue observar o total de dano combinado pelos monstros atacantes no fundo do painel assim como o total de defesa combinada que os monstros defensores conseguem aplicar ao dano dirigido a eles.

Após o jogador pressionar o botão “Continuar” é lhe apresentado o segundo painel(Figura 4.29), sendo este composto pelos seguintes elementos:

- Um subpainel dedicado à percentagem de dano a atribuir a cada monstro defensor;
- Um subpainel dedicado à quantidade de cada tipo de dano, dada a cada monstro defensor;
- Placa informativa do total de vida do esquadrão atacante;
- Placa informativa do total de vida do esquadrão defensor;
- O botão “Terminar Ronda” que fecha os subpainéis e passa para a próxima fase da ronda.

Este segundo painel informa o jogador da percentagem de dano que cada monstro defensor irá receber, através do primeiro sub painel. Esta informação é vital devido a certas habilidades afetarem a atribuição de percentagem de dano para cada monstro. Esta mecânica das habilidades é exemplificada no capítulo “Mecânica de Habilidades”.

O segundo sub painel apresenta o valor total de cada tipo de dano e o valor de cada tipo de dano aplicado a cada monstro defensor. Apresenta-se o estado de saúde do monstro, que irá mudar conforme se o monstro perdeu toda a vida e reviveu ou simplesmente perdeu toda a vida.

A habilidade de reviver é identificada no sub painel através do seu ícone e informa se esta foi usada ou não para cada um dos monstros do esquadrão defensor. O mesmo acontece com a habilidade de curar. Esta habilidade tem como origem a linha de habilidades de defesa e proporciona aos três monstros do esquadrão restaurar uma percentagem de vida.

Estas habilidades são apresentadas para que o jogador tenha conhecimento se os seus monstros ou os monstros oponentes ainda tem a possibilidade de reviver ou se conseguem curar pontos de vida. Esta mecânica é mais aprofundada no capítulo “Mecânica de Duelo”.

RONDA I						
PERCENTAGEM DE DANOS						
		0			35	
					15	
		0				
TOTAL	26.7	TOTAL	46.6	TOTAL	26.7	
RESULTADO						
TOTAL DANO NORMAL	50.9	TOTAL DANO CRÍTICO	0	TOTAL DANO PASSIVO	0	
	DANO	DANO	DANO	ESTADO	REVIVER	CURA
	13.6	0	0	VIVO	POR USAR	8.6
	23.7	0	0	VIVO	POR USAR	20.8
	13.6	0	0	VIVO	POR USAR	8.6
TOTAL ESQUADRÃO HP	79.8	TERMINAR RONDA	38	TOTAL ESQUADRÃO HP		

Figura 4.29: Exemplo do painel de análise de Resultado de Dano.

O jogador consegue verificar a quantidade de vida de cada esquadrão com as duas placas informativas no fundo do painel e passar para a próxima fase da ronda clicando no botão “Terminar Ronda”.

O duelo só termina quando um dos esquadrões possuir todos os monstros em estado desmaiado. Sendo assim o Duelo tem como base gerar um ciclo de ataques e de defesas até que um dos esquadrões pereça, ou seja, até que um dos esquadrões não possua nenhum monstro com vida.

Quando o Duelo chega a este estado é dado como concluído. O GUI Duelo muda para a do menu final, exemplificado pela Figura 4.30, que demonstra as variações do menu final para o caso de vitória e de derrota.

O menu final é composto pelos seguintes elementos:

- Uma placa informativa do sucesso que o jogador teve no Duelo;
- Uma placa informativa dedica aos pontos de experiência;
- Uma placa informativa dedicada à gestão de dano;
- Caso evolua, uma placa informativa dedica à evolução de nível;
- O botão “Terminar Duelo” que transporta o jogador para o cenário Mapa-mundo.



(a) Vitória.



(b) Derrota e Evolução.

Figura 4.30: GUI dos Menus Resultado do Cenário Duelo.

O menu final do Duelo tem como propósito informar o jogador da experiência que este ganhou e a quantidade que ele possui no momento. Informa o jogador da quantidade de dano que este recebeu, deu e defendeu durante o Duelo todo.

Caso o jogador evolua com a experiência ganha pelo Duelo que fez, é lhe apresentado uma mensagem a informar da sua evolução de nível e para qual o nível que evolui. Seguida da quantidade de pontos de habilidades que ganhou e a nova quantidade de pontos máxima que ele possui com a evolução.



Mecânicas do Jogo SquadUp

O jogo SquadUp é composto por vários cenários, em que cada cenário para que seja jogável implica ao jogador ter conhecimento da funcionalidade e limitação de cada elemento que compõe o cenário em si.

À medida que adquire mais experiência de jogo, o jogador vai criando um modelo mental sobre os conceitos implementados no jogo que lhe permite interagir com o jogo de uma maneira automatizada. Este tipo de comportamento é o comportamento padrão que o jogador desenvolveu através das mecânicas de jogo do jogo SquadUp.

Um jogo é composto por várias regras, limites, estruturas e sinergias entre as várias estruturas implementadas que guiam o jogador ao longo do jogo. Para que um jogo consiga cativar um jogador é necessário introduzir elementos que captem a atenção deste. Para que este objetivo fosse cumprido foram implementadas várias mecânicas de jogo para que o jogador tenha uma ampla variedade de escolha.

As mecânicas de jogo implementadas estão atribuídas a vários elementos do jogo, algumas tem sinergia entre elas assim como dependência mútua. Esta interligação deve-se ao facto de que uma mecânica que possua tal dependência só demonstra a sua funcionalidade, objetividade, significado na totalidade, através da existência da outra mecânica que é dependente e vice-versa.

As mecânicas do jogo SquadUp serão descritas em seguida, com o objetivo de aprofundar a descrição de elementos pilares da estrutura do jogo SquadUp.

5.1 Mecânicas de Monstro

Os monstros no jogo SquadUp são compostos por várias características que os tornam distintos uns dos outros. Destas características existem três que ao estarem interligadas criam uma sinergia entre elas e consequentemente também providenciam uma mecânica de jogo. Esta mecânica de jogo tem como estrutura a dependência entre os poderes de um monstro e a sua tipologia associada.

As características referidas são as seguintes:

1. Tipo;
2. Subtipo;
3. Raridade.

5.1.1 Tipo

Um monstro pode ser do tipo “Ataque”, “Defesa” ou “Sabotagem”. A tipologia associada ao monstro determina o seu poder principal, o seu poder secundário assim como a que posição pertence no esquadrão:

1. Se um monstro for do tipo “Ataque”,
 - (a) O poder principal é ataque,
 - (b) O poder secundário é defesa.
2. Se um monstro for do tipo “Defesa”,
 - (a) O poder principal é defesa,
 - (b) O poder secundário é ataque.
3. Se um monstro for do tipo “Sabotagem”,
 - (a) O poder principal é sabotagem,
 - (b) O poder secundário pode ser:
 - i. ataque;
 - ii. defesa.

Apesar dos monstros só poderem ter um poder principal e um secundário o monstro de “Sabotagem” consegue ter ataque e defesa como poder secundário. Isto deve-se ao facto de que o poder principal de “Sabotagem” ser repartido pelo ataque e defesa deste. A repartição é feita para que o monstro de “Sabotagem” consiga contribuir para a fase de “Ataque” e “Defesa” do Duelo, pois o poder “Sabotagem” não contribuí para o dano nem para a defesa.

Esta é uma das mecânicas de jogo, que torna os monstros de “Sabotagem” extremamente versáteis e valiosos em termos de sinergia com os outros monstros do esquadrão e com as habilidades em que o jogador aplica os seus pontos.

Um exemplo do que esta mecânica introduz ao jogador é poder tornar o seu esquadrão mais resistente ao dano, quando o jogador ainda está em níveis baixos.

Para isto o monstro de “Sabotagem” tem que ter como poder secundário o de defesa. Ao entrar em Duelo o poder principal de Sabotagem irá ser repartido e acumular com o poder secundário. Logo neste caso o monstro de “Sabotagem” irá ter a sua defesa complementada com metade do poder de “Sabotagem”, enquanto o seu ataque irá apenas ter a outra metade do poder de “Sabotagem”.

O jogador acumula o valor de defesa do monstro de “Sabotagem” ao monstro de “Defesa” e consegue uma defesa bastante forte. Também é possível aplicar a mesma estratégia para o caso de tornar o esquadrão mais poderoso em ataque.

5.1.2 Subtipo

Um monstro possui um poder principal e um poder secundário como já foi descrito acima. Mas estes poderes têm valores associados que são pré-definidos dentro do jogo através das limitações aplicadas pela mecânica de subtipo de monstro.

O subtipo de um monstro é uma característica que dita os valores base dos poderes e os valores de multiplicadores usados para aumentar os valores dos poderes por cada nível.

Os subtipos existentes são os seguintes:

1. **Subtipo Tank:** O monstro tem o seu valor de vida aumentado em comparação aos outros monstros com subtipos diferentes. Mas para que haja equilíbrio em termos de valorização de características e no frente-a-frente dos Duelos, os seus valores base e de multiplicadores dos poderes principal e secundário serão abaixo da norma;
2. **Subtipo Glasscannon:** Para este monstro acontece o oposto do subtipo “Tank”, o monstro irá possuir valores base e de multiplicadores dos poderes principal e secundário acima da norma. Já o valor de vida irá ser muito abaixo da média;
3. **Subtipo Wellrounded:** Os monstros de subtipo “Wellrounded”, são monstros que possuem um equilíbrio em todas as partes, ou seja, o valor de vida e os valores base e de multiplicadores dos poderes principal e secundário serão medianos. Os monstros com este subtipo são o que representam a média, distinguindo assim os outros subtipos.

5.1.3 Raridade

Um monstro também possui uma terceira tipologia associada a ele, sendo esta a raridade. Existem quatro tipos de raridade:

1. Comum;
2. Incomum;
3. Raro;
4. Mítico.

No contexto do jogo SquadUp a raridade é usada para ordenar a atribuição dos valores base e de multiplicadores pré-definidos aos monstros conforme a sua raridade.

Os valores base e de multiplicadores referidos foram criados externamente para associação aos monstros, sendo repartidos pelo tipo e subtipo em diferentes intervalos. Mas dentro destes intervalos há valores mais vantajosos que outros valores, para que se pudesse justificar a razão de um monstro do mesmo tipo/subtipo possuir valores mais vantajosos que outro do mesmo tipo/subtipo foi criado este sistema de distinção.

Esta tipificação de monstro não significa que os monstros de raridade mítica sejam melhores que os monstros de raridade incomum, pois cada monstro tem sempre uma desvantagem em comparação aos outros através dos valores que lhe foram atribuídos às características. Assim, o jogador tem que estudar melhor as combinações de monstros que deseja fazer para poder conseguir mais sucessos nos Duelos.

5.2 Mecânica de Habilidades

A linha de habilidades, também conhecida por árvore de habilidades ou *TechTree* é um elemento usado em vários jogos, para representar efeitos a aplicar (ver Figura 5.2) , evolução tecnológica de uma civilização (ver Figura 5.1) ou características únicas que uma personagem encarna.

A linha de habilidades é baseada numa mecânica de jogo usada há já muito tempo para complementação da evolução do jogador durante o contexto do jogo. Este tipo de elemento proporciona ao jogador:

1. **Tática de jogo**, em que o jogador irá mudar a sua jogabilidade conforme as várias habilidades que ativa. O jogador no início pode deparar-se com esquadrões num Duelo superiores às capacidades do seu, mas ao evoluir/desbloquear mais habilidades, estes esquadrões superiores tornam-se rudimentares;
2. **Estratégia**, neste caso o jogador pode mudar a sua estratégia, ou seja, o jogador pode tentar completar o “Duelo” de imediato, em vez de tentar melhorar as características dos seus monstros. Isto tudo devido à influência das habilidades ativas.



Figura 5.1: Árvore de habilidades de Age of Empires I.

As habilidades tentam melhorar as características bases dos poderes principais e secundários dos monstros do esquadrão do jogador. Também adicionam novos efeitos aos monstros do esquadrão, alguns destes efeitos são penderes de chance para a sua ativação enquanto outros são permanentes. Estas modificações só se aplicam em Duelos, ou seja, as habilidades terão efeito quando o esquadrão do jogador estiver a enfrentar outro esquadrão no cenário Duelo. As habilidades também tentam piorar as características e efeitos referidos acima, mas do esquadrão oponente.

Logo o que podemos verificar aqui é que as habilidades apenas se aplicam quando o jogador passa para o cenário Duelo, mas as suas mecânicas de jogo estão diretamente interligadas com as dos monstros. Isto deve-se ao facto de que certas habilidades tem uma sinergia muito superior aquando combinadas com determinados monstros do que com outros. Assim, quando o jogador vai escolher os seus monstros para adicionar ao esquadrão deve ter em consciência se os pontos que aplicou nas habilidades fazem uma boa combinação com estes ou não.

Através das várias sinergias originadas pela combinação de certos monstros com certas habilidades, são criados vários padrões aplicáveis. O jogador irá aperceber-se à medida que vai jogando que há vários tipos de monstros com determinadas características que combinam melhor com certas habilidades.

Isto vai permitir ao jogador guiar-se melhor nas suas abordagens aos Duelos, através das limitações originadas pelos padrões que este adquiriu com a experiência de jogo. Esta mecânica de jogo é importante para cativar o jogador, ao despertar a sua curiosidade através de um leque de combinações que só conseguirá descobrir ao ganhar experiência de jogo.



Figura 5.2: Árvore de habilidades de League of Legends.

Ao criar no jogador uma curiosidade infindável o jogo ganha longevidade por interação/foco do jogador. Assim cria-se um ciclo entre o jogador e o jogo, partindo do princípio que o jogador sentir-se-á implicado a descobrir a maioria, senão tudo, o que há por descobrir referente às combinações.

Para demonstração deste padrão de sinergia entre as mecânicas de monstro e de habilidades é apresentado de seguida um exemplo.

Na Figura 5.3 podemos observar a habilidade “Atropelar” que tem como efeito reduzir a percentagem de dano atribuída ao monstro de defesa do esquadrão inimigo.

Ao reduzir a percentagem atribuída ao monstro de defesa irá aumentar a percentagem de dano dos monstros restantes vivos, ou seja, os monstros de ataque e de sabotagem.

Este efeito só é alcançável por vias de chance, ou seja, é preciso que o jogo tenha uma chance menor que a da habilidade, para que a habilidade consiga ser ativada.

O jogador consegue gerar sinergia entre esta habilidade (Atropelar) e os monstros que escolhe para o seu esquadrão. Se escolher os monstros de ataque e de sabotagem que sejam do subtipo Glasscannon e que o seu monstro de sabotagem tenha como poder secundário o poder de Ataque, o jogador consegue formar um esquadrão com um valor base de ataque alto.



Figura 5.3: Árvore de habilidades de League of Legends.

Para compensar a falta de vida dos monstros de ataque e de sabotagem, o jogador tem que escolher um monstro de defesa do subtipo Tank. O esquadrão fica com um valor equilibrado em vida, um valor baixo em defesa, mas um valor alto em ataque.

O valor de ataque alto é bastante eficaz se a habilidade ativar, pois os monstros de ataque e de sabotagem oponentes irão ter muito mais dano recebido que o monstro de defesa.

Aqui também entra em ação a experiência de jogo do jogador ao reconhecer que tanto os monstros de ataque como os monstros de sabotagem seja qual for o subtipo não possuem tanta vida como os monstros de defesa.

Esta é uma estratégia de alto risco, mas com alta compensação quando é concretizada e uma das muitas possíveis de se criar, através da sinergia entre as habilidades e as tipificações dos monstros.

5.3 Mecânicas de Evolução

Ao contrário de certos jogos que possuem campanhas, níveis por missão, sessões, este jogo não possui transição de níveis ou qualquer coisa relacionada com tal.

As progressões possíveis de fazer no jogo SquadUp são através dos níveis de jogador e dos níveis dos monstros. Em seguida é descrito como pode evoluir o jogador e os monstros de nível e o que acontece quando evoluem.

Para que um jogador possa evoluir de nível é preciso que este adquira um valor definido de pontos de experiência. O jogador só consegue ganhar pontos de experiência ao fazer Duelos. Ao evoluir de nível o jogador ganha três pontos de habilidade e o valor para evoluir de nível é reiniciado, em certos casos aumenta de valor,

Ao ganhar pontos de habilidades também aumenta o seu número máximo de pontos que pode aplicar nas linhas de habilidades. O máximo de pontos de habilidades que um jogador pode ter é 120 pontos, que é o equivalente ao total de pontos a aplicar pelas três linhas de habilidades.

Os monstros também conseguem evoluir de nível. Para que um monstro consiga evoluir é necessário que o jogador capture um número de monstros iguais. O número é apresentado ao jogador no menu “Monstro” e irá aumentar à medida que o monstro evolui de nível. Ao evoluir de nível o monstro irá aumentar os seus valores de poderes principal, secundário e de vida.

Nem o jogador nem nenhum monstro tem nível máximo de evolução.

5.4 Mecânicas de Captura de Monstro

As mecânicas referentes ao cenário de captura tratam de mecânicas de priorização e adaptação. A mecânica de priorização refere-se ao facto de que quando o jogador toca no monstro este pode optar por fugir. O jogador com a experiência de jogo fica a perceber que o monstro à qual à captura deste fugiu, irá continuar no mapa de jogo.

Quando o jogador estiver a capturar monstros, ele pode dar prioridade aos monstros que estão há mais tempo no mapa de jogo, vendo pela ordem de aparecimento. Caso o jogador se engane e toque no monstro errado para capturar, o jogador aplica a opção de fugir e tenta a captura do monstro com a maior prioridade no momento, sabendo que o monstro errado continuará no mapa de jogo.

A segunda mecânica refere-se ao movimento gestual que o jogador tem que aplicar para capturar um monstro. O jogador terá que adaptar a sensibilidade da órbita de captura à força que aplica no movimento gestual e como aplica o movimento.

Esta é uma mecânica física de jogabilidade em que o jogador precisa de aplicar o método de tentativa erro até conseguir adquirir a memória muscular ideal para o movimento. Devido a esta mecânica depender do estatuto e habilidade física do jogador o movimento de captura irá ser aplicado de maneira diferente de jogador para jogador.

5.5 Mecânica de Duelo

É no cenário Duelo que o jogador consegue ganhar experiência de jogo para conseguir formar e aplicar as várias mecânicas de “Monstro” e de “Habilidades”. Quando o jogador faz um Duelo pela primeira vez ele fica a perceber mais sobre o efeito das habilidades e a vantagem que estas dão à sua hipótese de vitória. Com o decorrer das rondas também fica a conhecer o potencial dos monstros que escolheu para o esquadrão.

É através da experiência de jogo adquirida ao jogar um elevado número de Duelos que o jogador consegue aperceber-se dos vários padrões que consegue aplicar na criação do seu esquadrão em complementação com os pontos a aplicar nas habilidades. Após a sua estratégia estar criada, o jogador pode aplicá-la no Duelo e verificar se funciona ou se precisa de alterações.

Ao não proporcionar consequências negativas ao jogador no caso de derrota, o cenário Duelo serve como mapa livre para o jogador poder aplicar as suas estratégias. Quando o jogador já adquiriu muita experiência de jogo, ele também consegue perceber que o esquadrão que vai enfrentar é superior ao seu, por reconhecer os monstros e a tipificação destes.

O Duelo também proporciona ao jogador certas mecânicas, sendo uma destas a utilização das poções em conjunção com a combinação de esquadrão/habilidades feita pelo jogador. Ao tomar conhecimento

que as poções de vida recuperam 25% da vida de cada monstro do seu esquadrão e as poções de inspiração adicionam 10% de chance às habilidades de chance que o jogador tem ativas, o jogador consegue controlar melhor o ritmo do Duelo.

Isto quer dizer que o jogador pode formar o seu esquadrão e contar com a utilização das poções de vida para manter os seus monstros vivos por mais fases.

O jogador também pode contar com as poções de inspiração para retirar mais sucesso das habilidades que utilizam o mecanismo de chance.

A mecânica da análise de dados é a mecânica de jogabilidade mais importante que o cenário Duelo consegue fornecer ao jogador. Ao observar os dois painéis apresentados sempre em cada fase, o jogador consegue descobrir mais combinações, melhorar a sua estratégia de batalha, aperceber-se dos seus erros e descobrir novos padrões.

Em seguida é exemplificado um caso bastante provável de acontecer. Um o jogador tem os seus monstros a metade da vida e as habilidades de reviver respetivas por usar. Se o inimigo já tem as habilidades de reviver dos seus monstros utilizadas. Isto quer dizer que os monstros do esquadrão inimigo estarão com um valor de vida baixo e podem ser derrotados facilmente na próxima fase de ataque.

Sendo assim jogador pode poupar a utilização de poções de vida e simplesmente aplicar a poção de inspiração para que as habilidades de reviver tenham mais sucesso. Este caso serve para quando o jogador não possui muitas poções para gastar.

6

Módulos Técnicos

Com o objetivo de estruturar o jogo, foram implementados diversos cenários do jogo SquadUp de forma modular. Cada cenário é constituído por vários elementos que gerem o componente base de dados e gráfico.

Nas secções seguintes são descritos vários elementos chave responsáveis por processar a informação a passar da base de dados aos painéis de conteúdo gráfico e textual.

6.1 Tutoriais

6.1.1 Funcionamento Programático dos Tutoriais

Os tutoriais implementados no SquadUp tem um processo de iniciação, apresentação e encerramento.

O processo de iniciação compõem-se em associar as funções necessárias aos botões que representam as opções “Fechar”, “Anterior” e “Próximo”. Após a associação de referências das opções é iniciado o gestor do rodapé que atribui as legendas necessárias a cada botão referido acima e são preenchidos os vários painéis do tutorial e os seus campos com os seus contextos no idioma selecionado pelo jogador, completando o processo de iniciação.

O processo de apresentação de um tutorial é gerido pelo seu valor de estado, sendo este um valor inteiro. Este valor aumenta ou diminui através do evento associado aos botões “Anterior” e “Próximo”. O valor mínimo do estado é 0 e o máximo é referente à quantidade de painéis que o tutorial possui para apresentar. Os valores mínimo e máximo são valores limitadores com o objetivo de evitar que o jogador não retroceda ou avance para um valor de estado inexistente.

Quando o jogador avança para o último painel do tutorial o botão “Próximo” muda a sua legenda para “Fechar”, anunciando ao jogador que o tutorial irá terminar. O jogador também pode terminar o tutorial

carregando o botão “Fechar”, localizado no canto inferior esquerdo, sendo este unicamente dedicado à função de terminar o tutorial.

Caso o jogador pressione um destes botões acima referidos é dado início ao processo de encerramento, sendo introduzida uma condição booleana na *cache* do jogo SquadUp referente à opção de “Desligar Ajuda”. Esta opção possibilita ao jogador bloquear a apresentação automática dos tutoriais. Assim quando o jogador entra em um menu ou cenário com um tutorial dedicado, a apresentação deste irá ser ditada pelo valor da condição booleana armazenada na *cache* referente ao tutorial. Esta condição booleana é representada através de valores inteiros com a seguinte estrutura:

- 1, estado positivo, indica à condição para não apresentar o tutorial ao jogador;
- 0, estado negativo, indica à condição para apresentar o tutorial ao jogador.

O jogo verifica se o jogador já visualizou o tutorial referente ao local do jogo onde este se situa ou não, caso o jogador nunca tenha visualizado é lhe apresentado o tutorial automaticamente, sendo esta a única exceção de apresentação automática sem verificação da condição mencionada acima. Caso o jogador queira aceder ao tutorial é lhe disponibilizado o botão “Ajuda” para acesso automático ao tutorial referente à sua localização no jogo.

6.2 Mapa Mundo

No segundo cenário o utilizador irá visualizar um mapa mundo em formato 3D e um avatar que o representa, também este em 3D. Este avatar estará situado na localização do jogador através das coordenadas geográficas obtidas através do GPS do dispositivo móvel. O início da sessão do jogo é constituído por vários passos:

1. Leitura/Escritura dos dados do jogador:

- (a) A aplicação começa por verificar e ler o ficheiro de dados do jogador;
- (b) Após a leitura/extração de dados estes são transpostos para a tipificação dedicada ao jogador, sendo o tipo designado de **Player**.

2. Iniciação das Habilidades:

- (a) O GameManager inicia o seu campo do tipo AbilitiesContainer para a criação das habilidades;
- (b) A iniciação de cada habilidade conta com os dados do jogador para atribuição do valor de pontos atribuídos da sessão prévia;
- (c) Caso seja a primeira sessão são iniciadas de maneira *standard*.

3. Leitura/Escritura dos dados dos monstros para instanciar:

- (a) O componente MainCamera tem agregado a si o *script* MonsterFactory que é responsável por gerar monstros para o utilizador captura-los.
- (b) Após gerar os monstros, o utilizador irá ter ao seu dispor várias opções, sendo estas as seguintes:

- i. Aceder ao menu;
 - ii. Capturar um monstro dentro do cenário Captura;
 - iii. Entrar no modo de batalha dentro do cenário Duelo caso tenha o esquadrão formado.
4. Para o utilizador aceder ao menu este basta pressionar no botão que diz Menu, aqui o utilizador irá poder aceder as seguintes opções:
 - (a) Sair – Serve para sair do jogo diretamente;
 - (b) Definições – Mudar as definições de linguagem utilizadas no jogo;
 - (c) Bestiário – O utilizador pode visualizar os monstros do jogo que já capturou ou viu em jogo;
 - (d) Jogador – O utilizador consegue visualizar os seus dados pessoais em relação ao jogo;
 - (e) Esquadrão – Esta opção serve para o utilizador gerir o seu esquadrão de monstros;
 - (f) Habilidades – Esta opção serve para o utilizador gerir as suas linhas de habilidades.

6.3 Módulo de Habilidades

O módulo Habilidades é um dos mais importantes devido a abranger vários setores do jogo, desde o menu de habilidades ao decorrer de uma batalha no cenário “Duelo”. Foi procurado obter o maior nível de redundância e afetação de valores com segurança por motivos de evitar qualquer corrupção em tempo de execução dos valores dos campos/corrupção dos valores através de vias externas. Os vários *scripts* e as suas tarefas serão apresentadas em seguida para melhor contextualização do funcionamento do módulo.

6.3.1 Estrutura

A classe abstrata Ability é o componente chave para criar uma habilidade, é através desta classe que se determina as características únicas de uma habilidade e é com esta classe que gere todas as habilidades do jogo SquadUp. A partir desta abordagem é possível criar habilidades de maneira simples e eficaz. Para gerar uma nova habilidade é necessário derivar da classe abstrata Ability, reescrever os campos abstratos e atribuir-lhes os valores às propriedades respetivas.

A Figura 6.1 demonstra a derivação da Habilidade AttackI e a implementação das múltiplas propriedades e métodos que são necessários para uma habilidade possuir contexto. Uma habilidade no jogo SquadUp é um efeito a ser aplicado sobre um monstro de um esquadrão alvo, sendo esta ativada por uma condição ou simplesmente automática em termos de afetação.

Para implementar uma habilidade no jogo SquadUp foi necessário dar-lhe uma estrutura, em que as suas características são representadas por propriedades. Cada propriedade indica uma parte identificadora da habilidade através da sua tipologia e conteúdo, as propriedades identificadoras são as seguintes:

1. AbilityName, representa o nome da habilidade;
2. Description, representa a descrição de efeito da habilidade;
3. AbilityIndex, representa o identificador numérico da habilidade;

4. AbilityType, representa o tipo de poder da habilidade;
5. ActionType, representa o tipo de efeito da habilidade.

Para que o jogador consiga uma melhor contextualização da descrição da habilidade e da sua afetação ao monstro ou monstros do esquadrão alvo são usados valores no formato de percentagem. Estes valores pertencem às características efeito, combo e condição.

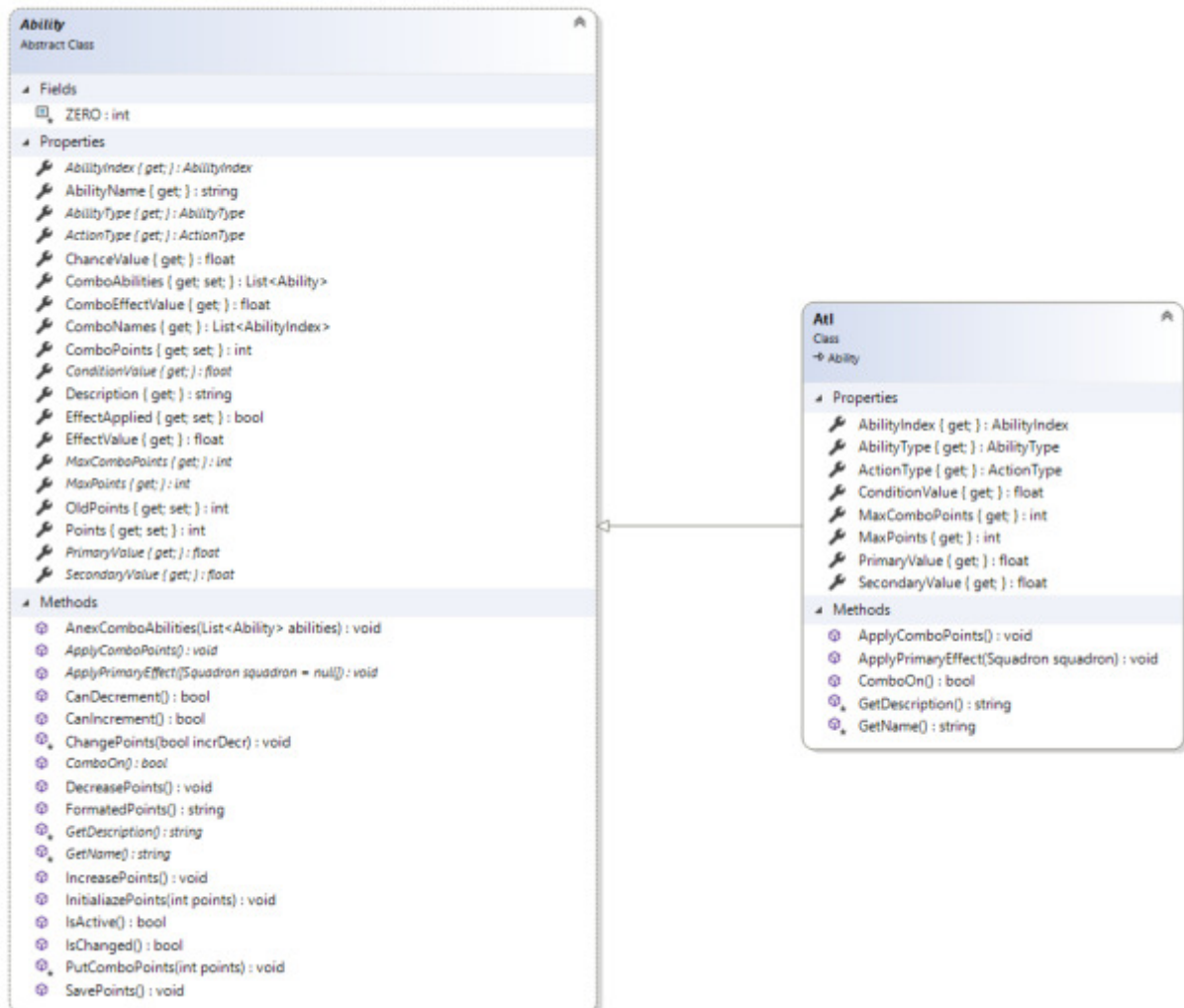


Figura 6.1: Estrutura de uma Habilidade.

Apesar de estas características guardarem o seu valor base é necessário guardar também o valor modificado. O valor modificado é gerado através da multiplicação entre o valor base e a quantidade de pontos aplicados na habilidade. São os valores modificados que são apresentados na descrição da habilidade no jogo SquadUp, para que o jogador consiga visualizar o incremento/decremento conforme aplica os seus pontos na habilidade.

Todas estas características assumem um papel na estrutura de uma habilidade sendo aplicadas através de um único método. Este método é reescrito pela habilidade que derivou da classe Ability, possibilitando a singularidade de como aplicar e a quem aplicar estes valores.

Estes campos são representados pelas seguintes propriedades e métodos:

1. `PrimaryValue`, representa o valor base do efeito;
2. `SecondaryValue`, representa o valor base do combo a adicionar ao efeito ou condição;
3. `ConditionValue`, representa o valor base da condição do efeito;
4. `EffectValue`, representa o valor base do efeito multiplicado pelos pontos aplicados na habilidade;
5. `ComboEffectValue`, representa o valor base do combo multiplicado pelos pontos aplicados na habilidade;
6. `ChanceValue`, representa o valor base da condição multiplicado pelos pontos aplicados na habilidade;
7. `ApplyPrimaryEffect`, método responsável por aplicar o efeito da habilidade. Aplica o efeito ao monstro ou monstros do esquadrão passado por parâmetro. Este efeito utilizará os valores dos itens 4 e 5 acima descritos para modificar uma das características do monstro alvo, caso a habilidade tenha de verificar se consegue aplicar o seu efeito por mecanismo de chance é usado o item 6 e comparado com um valor gerado no momento para o prosseguimento da aplicação;
8. `EffectApplied`, método responsável por limitar a aplicação do efeito da habilidade a uma única vez. No jogo SquadUp existem várias habilidades cujos efeitos são aplicados uma única vez por Duelo.

Cada habilidade no jogo SquadUp tem um número máximo de pontos que o jogador pode aplicar e como a quantidade de pontos aplicados em uma habilidade estão diretamente interligados com a ativação do seu combo é necessário guardar os dados respetivos aos pontos.

Devido à importância dos pontos para uma habilidade foi necessário criar vários vários campos e métodos para gerir a quantidade e modificação destes. Estes são os seguintes:

1. `MaxPoints`, representa a quantidade máxima de pontos que se pode aplicar na habilidade;
2. `MaxComboPoints`, representa a quantidade máxima de pontos combo que se podem gerar;
3. `Points`, representa a quantidade de pontos aplicados que a habilidade possui no momento;
4. `OldPoints`, representa a quantidade de pontos aplicados à habilidade, guardada em memória;
5. `ComboPoints`, representa a quantidade de pontos combo que a habilidade conseguiu gerar;
6. `SavePoints`, verifica se houve alteração de pontos comparando os campos `Points` e `OldPoints`. Caso haja alteração guarda o novo valor de `Points`;
7. `IncrementPoints`, aumenta por um, o valor de pontos da habilidade e diminui o valor de pontos que o jogador pode aplicar. Se o valor de pontos for igual à quantidade máxima, não aplica nenhuma modificação;
8. `DecrementPoints`, diminui por um, o valor de pontos da habilidade e aumenta o valor de pontos que o jogador pode aplicar por um. Se o valor de pontos for igual a zero, não aplica nenhuma modificação;

9. ApplyComboPoints, verifica se a condição de ativação de combo da habilidade com as outras respetivas habilidades está ativa. Caso esteja ativa aplica a respetiva quantidade pontos combo ao campo ComboPoints, verificando sempre os limites máximos (MaxComboPoints) e mínimos (zero).

A classe abstrata Ability e as habilidades derivadas desta tem uma implementação fechada ao exterior limitando a modificação dos seus campos ao mínimo possível. Com este tipo de estrutura é possível proteger as propriedades das habilidades ao evitar a modificação com intenção prejudicial de terceiras partes.

Os métodos estendidos pelas habilidades verificam sempre as várias afetações aos campos antes de as aplicarem, só os métodos implementados têm a liberdade de implementação. Isto deve-se ao facto de os métodos a serem implementados terem verificações e afetações únicas, logo é necessário que a habilidade tenha margem de manobra para este tipo de eventos.

O objetivo deste tipo de implementação é a limitação de manipulação/transposição de dados às classes derivadas designando-lhes a única responsabilidade de executarem o seu efeito, para qual foram concebidas.

O jogo SquadUp têm no total trinta habilidades, estando repartidas pelas três linhas de habilidades (ataque, defesa, sabotagem). A classe abstrata Ability e as habilidades derivadas desta podem ser observadas na Figura 6.2.

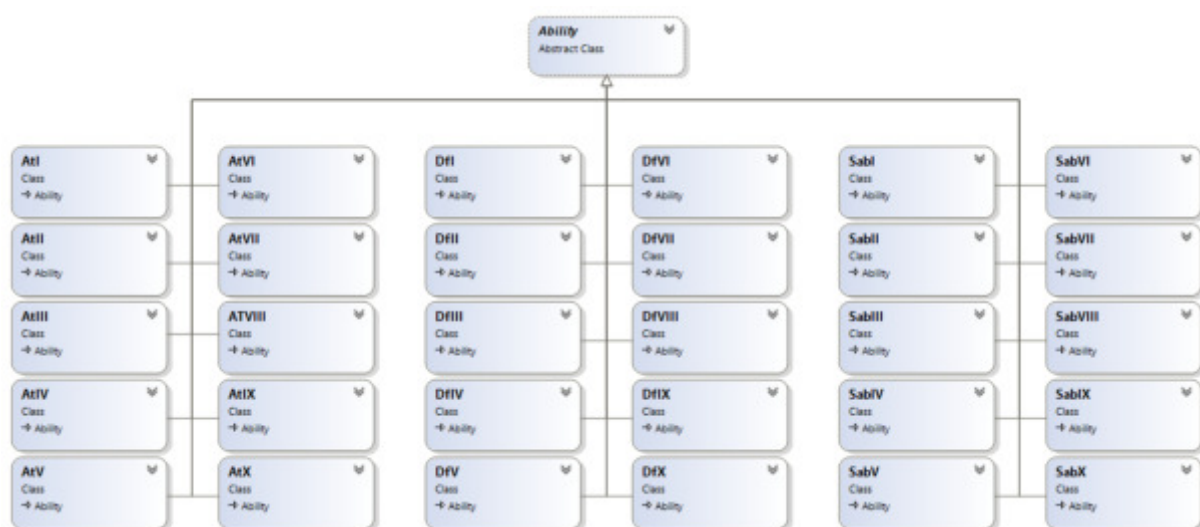


Figura 6.2: Derivação da classe Ability pelas várias Habilidades.

6.3.1.1 Componentes da Habilidade

Dos campos identificadores da Habilidade, três utilizam um tipo específico como valor. Esta integração de tipologia é feita para que estes campos só tenham um lote de valores específicos, evitando possíveis erros. Esta abordagem permitiu utilizar os vários campos identificadores para vários métodos de pesquisa e recolha das habilidades conforme o tipo de campo. Estes tipos são os seguintes:

1. `AbilitiesNameIndex`, contém todos os nomes internos das habilidades. Esta nomenclatura é usada para distinguir as várias habilidades internamente e para gestão dos combos das habilidades(combo). É usado para preencher o campo 3;
2. `AbilityType`, contém os três tipos de poder (ataque, defesa, sabotagem) e é utilizado para atribuição do tipo de poder da habilidade, preenchendo o campo 4;
3. `ActionType`, contém todos os tipos de efeito que as habilidades podem assumir, sendo usado para preencher a propriedade 5.

Cada habilidade no jogo SquadUp possui uma combinação com outras habilidades gerando assim o combo, este evento é despoletado através da condição que cada habilidade possui para determinar se tem o seu combo ativo ou não. Por exemplo a primeira habilidade de ataque combina com a primeira habilidade de defesa e sabotagem. Mas para o seu combo estar ativo, a habilidade de ataque precisa que o jogador lhe aplique o máximo de pontos assim como nas outras duas habilidades. Enquanto as três habilidades não tiveram o máximo de pontos aplicado, o combo destas não irá estar ativo.

Na criação de habilidades o processo também necessita de indicar quais são as habilidades que combinam com a habilidade a ser gerada, para isto foi implementado a classe `AbilitiesComboIndex` que ao receber o campo `AbilityIndex` da habilidade retorna uma lista com as habilidades que combinam com a própria.

O jogo SquadUp guarda os dados voláteis das habilidades através da classe serializável `AbilityData`. Esta classe recebe como parâmetro de construtor a própria Habilidade e extrai os campos `AbilityIndex` e `Points`. O campo `Points` é o campo volátil sendo necessário guardar a quantidade de pontos que a habilidade possui no momento e o `AbilityIndex` serve para inicialização da habilidade através dos dados guardados. Utilizando o `AbilityIndex` procura-se a habilidade com o valor correspondente e transmite-se o valor de pontos que esta possuía quando foi gravada da última vez.

6.3.1.2 Gestor de Habilidades

Todas as Habilidades do jogo SquadUp são criadas, inicializadas, geridas e acedidas através da classe `AbilitiesContainer`. A classe `AbilitiesContainer` é como o nome indica um contentor de habilidades que através dos seus construtores instância todas as habilidades do jogo.

Quando o jogo SquadUp é iniciado pela primeira vez e o jogador acede ao cenário Mapa-mundo, a classe `AbilitiesContainer` irá ser instanciada e criar as habilidades de raiz. Após a primeira vez, irá sempre utilizar os valores armazenados das habilidades para a criação destas.

A mesma classe também é usada para fornecer habilidades ao esquadrão inimigo quando o jogador acede a um Duelo. A inicialização das habilidades é conseguida através da passagem da quantidade de pontos aplicadas às habilidades mas por tipo de poder. Ou seja, a classe `AbilitiesContainer` irá receber um valor equivalente à quantidade de pontos que o jogador aplicou nas suas habilidades de Ataque. Este valor irá ser usado para atribuir pontos às habilidades do tipo ataque, em que a escolha das habilidades é aleatória. O mesmo acontece com as habilidades do tipo poder Defesa e Sabotagem.

A AbilitiesContainer também disponibiliza acesso às habilidades através de vários filtros, retornando uma habilidade ou uma lista de habilidades que contenham um valor de campo identificador específico. Também consegue retornar todas as habilidades do jogo.

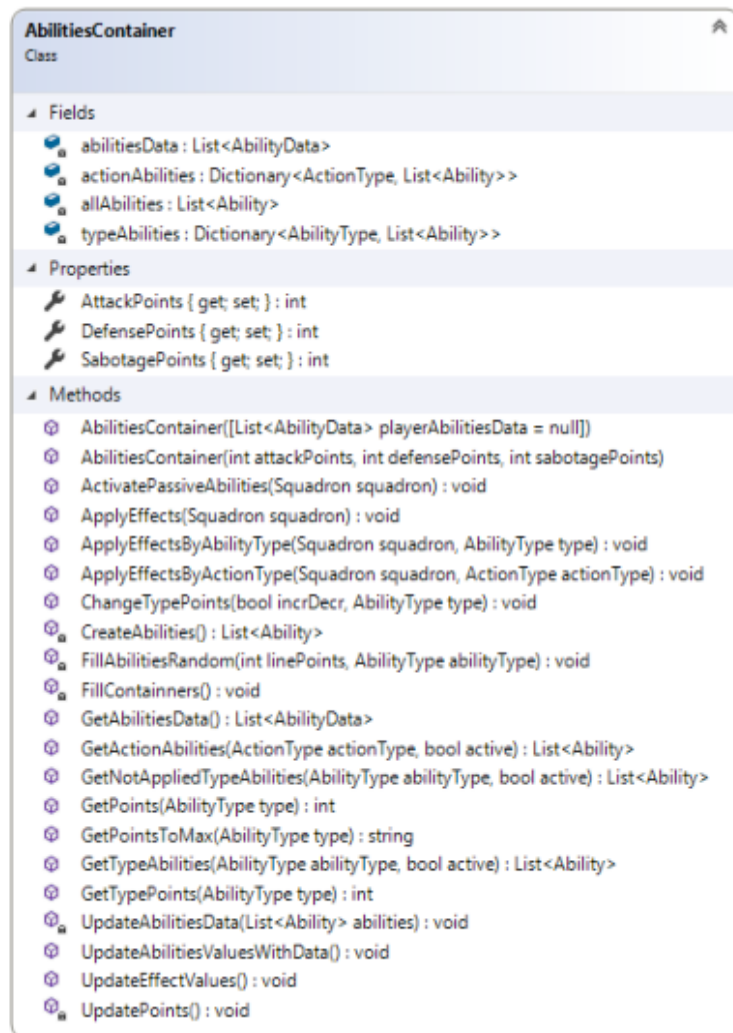


Figura 6.3: Estrutura de AbilitiesContainer.

Sendo responsável pelas Habilidades também conta e guarda a quantidade de pontos aplicadas nas habilidades por tipo, disponibilizando o total de pontos aplicados aos três tipos de poder. Estas contagens são usadas para o contador de pontos, incluído no cabeçalho do GUI do menu Linha de Habilidades.

6.4 Módulo Monstro

Os monstros são um dos componentes principais do jogo SquadUp, sem os monstros o jogo não teria nenhum contexto. Cada monstro no jogo SquadUp possui características pessoais e características de combate. Estas são possíveis de se observar no menu Monstro que demonstra os valores destas características do monstro escolhido na lista de monstros no menu Bestiário.

6.4.1 Estrutura

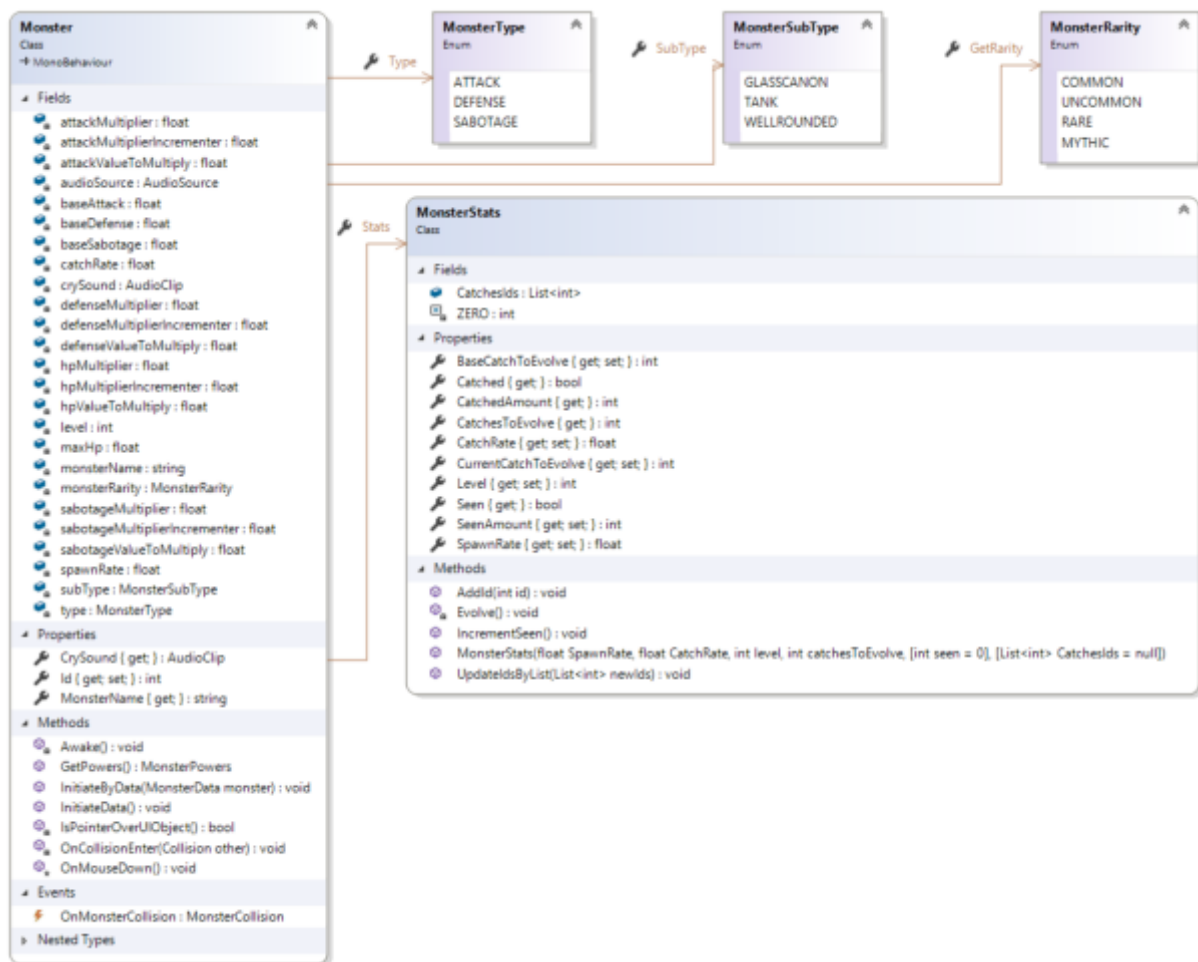


Figura 6.4: Estrutura da classe Monster.

A classe Monster é responsável por guardar todos as características e valores base dos poderes dos monstros. Todos estes valores são incluídos diretamente no *script* via a opção `SerializeField`. Esta opção permite que inserir/escolher os valores dos vários campos através do motor de jogo Unity. Assim, cada monstro no jogo SquadUp possui um *script* Monster associado, com os valores dos campos serializáveis introduzidos.

O identificador de um monstro é um valor com o objetivo de contabilizar a quantidade de vezes que um monstro já foi capturado. Quando um monstro é criado para o mapa de jogo as suas características tem que ser inicializadas. Existem dois procedimentos:

1. Se o monstro nunca foi criado, o motor do Unity invoca o método `Awake` do *script* Monster associado ao `GameObject` do monstro para inicialização dos seus valores;
2. Se o monstro ja foi criado, os seus valores são atribuídos através do tipo `MonsterData`. Este tipo é responsável por serializar os campos do monstro referentes ao progresso deste.

No primeiro procedimento o identificador do monstro é gerado de maneira aleatória pela classe `MonsterRng`. Esta classe é responsável por gerar todos os identificadores dos monstros que são criados de

maneira aleatória.

Para evitar a duplicação de identificadores, todos os identificadores guardados nas listas de identificadores dos monstros são adicionados a uma lista de identificadores que a classe `MonsterRng` contém como campo privado. Estes identificadores são adicionados através do segundo procedimento acima mencionado.

Como já mencionado em outros capítulos, um monstro no jogo `SquadUp` possui um tipo, um subtipo e uma raridade. Estes campos são limitados através da tipificação como valor que podem adquirir. Para isto foram implementados três *scripts*:

1. `MonsterType`, responsável pelos tipos de poder a associar a um monstro,
 - (a) `Attack`,
 - (b) `Defense`,
 - (c) `Sabotage`;
2. `MonsterSubType`, responsável pelos subtipos a associar a um monstro,
 - (a) `Glasscanon`,
 - (b) `Tank`,
 - (c) `Wellrounded`;
3. `MonsterRarity`, responsável pelo nível de raridade a associar a um monstro,
 - (a) `Common`,
 - (b) `Uncommon`,
 - (c) `Rare`,
 - (d) `Mythic`;

Cada monstro tem os seus valores de características pessoais assim como a quantidade de vezes que foi visto, capturado e o seu nível geridos pela classe `MonsterStats`. Esta classe é responsável por gerir todos estes campos assim como de evoluir o monstro caso este atinja a condição necessária para a sua evolução.

Todas as características de combate de um monstro são geridas e calculadas através do *script* `MonsterPowers`. Este *script* só é gerado quando é necessário ter acesso às características de combate de um monstro e é o método `GetPowers` do *script* `Monster` que está encarregue de retornar uma instância de `MonsterPowers`.

A gestão das características de combate do monstro foi implementada desta maneira, para evitar a criação e gestão de instâncias sem qualquer finalidade para vários cenários. Se o jogador estiver no cenário `Captura` não é necessário que o jogo tenha uma instância de `MonsterPowers` do monstro que o jogador habilitou-se a capturar. Ou seja, este *script* é indicado para quando o jogador acede aos menus `Monstro` e `Esquadrão` ou para quando o jogador executa um `Duelo`.

Como é possível observar na Figura 6.5, o *script* `MonsterPowers` tem como propriedades os vários *scripts* dedicados a cada tipo de poder. Cada monstro no jogo `SquadUp` tem um poder principal e um secundário. Por exemplo um monstro de ataque, tem como poder principal o poder de ataque e como poder secundário o poder de defesa.

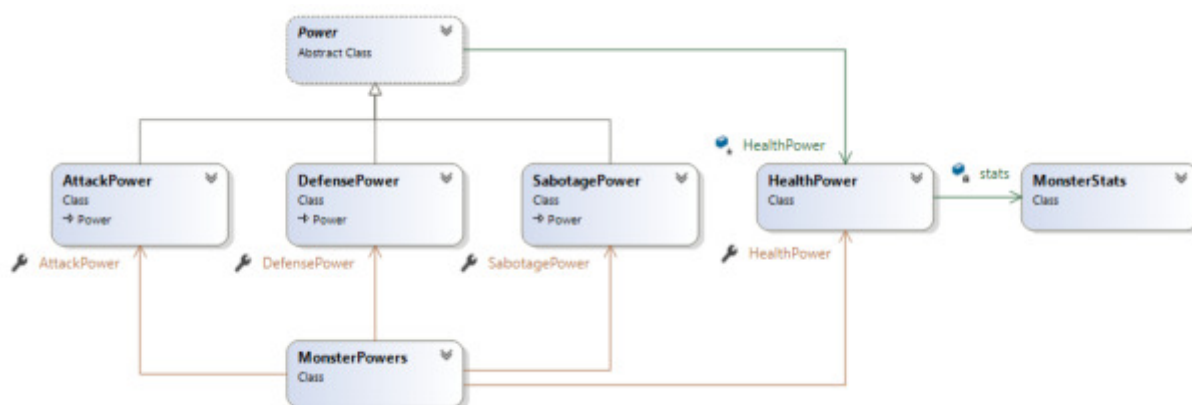


Figura 6.5: Estrutura de MonsterPowers.

Para que estas regras fossem respeitadas e ao mesmo tempo fosse simples o acesso aos valores das características de combate de qualquer monstro, foi utilizada esta abordagem. Assim, o *script* `MonsterPowers` serve como o componente principal de acesso e gestão a estas determinadas características.

Através da figura acima referida, os *scripts* `AttackPower`, `DefensePower` e `SabotagePower` derivam da classe abstrata `Power`. Esta classe fornece métodos utilitários aos seus derivantes e utiliza o *script* `HealthPower` para confirmação do estado de saúde do respetivo monstro.

A maior parte das propriedades dos *scripts* `AttackPower`, `DefensePower` e `SabotagePower` verificam sempre o estado de saúde do monstro quando é requisitado o seu valor. Esta verificação feita na origem do valor permite evitar código repetitivo nos *scripts* dedicados ao Duelo. Os *scripts* `AttackPower`, `DefensePower`, `SabotagePower` e `HealthPower` são os recetáculos dos efeitos das habilidades, guardando os valores destes em cada propriedade dedicada a cada habilidade.

6.4.1.1 Gestor

Para gerir os monstros do jogo `SquadUp` foi implementada a classe `MonsterManager`. Os monstros são contidos em uma lista preenchida diretamente via o Unity, esta abordagem é conseguida através da utilização do atributo `SerializeField` que permite o *script* demonstrar/afetar o campo através da interface gráfica do editor do Unity. O `MonsterManager` pode fornecer monstros através da lista inteira, de maneira aleatória ou através dos seguintes parâmetros específicos:

1. Nome;
2. `MonsterType`;
3. `MonsterRarity`;

Devido ao funcionamento do jogo `SquadUp`, todos os monstros que são criados são também destruídos pelo próprio. Para guardar a informação necessária para contabilização do progresso do jogador em referência aos monstros, o `MonsterManager` gera a contabilização ligada aos *ids* e quantidade de vezes que um monstro foi visto/capturado.

O `MonsterManager` utiliza o *script* `Catching` que fornece o nome, id e verificação de captura do monstro. A partir destes campos o `MonsterManager` acede à sua lista de monstros, encontra o monstro equivalente através do campo nome e adiciona o id do monstro capturado ao campo representante da lista de ids. Todos os monstros contém uma lista de ids que é mantida pela classe `MonsterStats`.

Sempre que um monstro é visto pelo jogador, é utilizado o mesmo processo de procura e acesso ao monstro da lista de monstros e incrementado o campo referente à quantidade de vezes que o monstro foi visualizado pelo jogador em jogo.

O `MonsterManager` também contabiliza o total de vezes que todos os monstros, da sua lista de monstros, foram vistos, capturados, não capturados e o rácio de sucesso de captura de monstros do jogador. Estes valores são utilizados na GUI do menu do jogador.

6.4.1.2 Gerador

A classe `MonsterFactory` tem como função popular o mapa mundo em redor da posição do utilizador com monstros. A escolha de monstros a gerar é feita de maneira aleatória assim como o posicionamento destes. Apesar dos valores de posicionamento serem gerados aleatoriamente são verificados com os valores de posição dos monstros já gerados para evitar monstros sobrepostos. Os identificadores dos monstros e as suas posições são guardadas em uma lista do tipo `MonsterLocationData` para que a classe `MonsterFactory` consiga gerar estes outra vez caso o jogo avance para um cenário de captura ou de torre. Foi escolhida esta opção para que o utilizador tenha mais hipóteses de captura.

A classe `MonsterLocationData` serve como contentor de informação geográfico de cada monstro gerado, para tal foi necessário inserir os campos de coordenadas assim como a assinatura temporal para as verificações necessárias feitas pela classe `MonsterFactory`. Foi lhe atribuído a definição `Serializable` para que os seus dados possam ser escritos e lidos em ficheiro.

6.5 Módulo Player

O módulo `Player` é responsável por conter as classes respetivas aos dados do jogador no jogo `SquadUp`. Em qualquer ação que o jogador realize no jogo `SquadUp`, irá haver modificações ou acesso aos dados que este módulo representa.

6.5.1 Estrutura

Todos os dados de identificação do jogador estão contidos na classe `Player` assim como os dados de outros módulos. Esta classe é responsável por disponibilizar o acesso a estes dados e geri-los. Apesar de haver acesso público aos campos a maior parte destes campos adquire os seus valores através da leitura dos dados armazenados pelo jogo, por motivos de proteção.

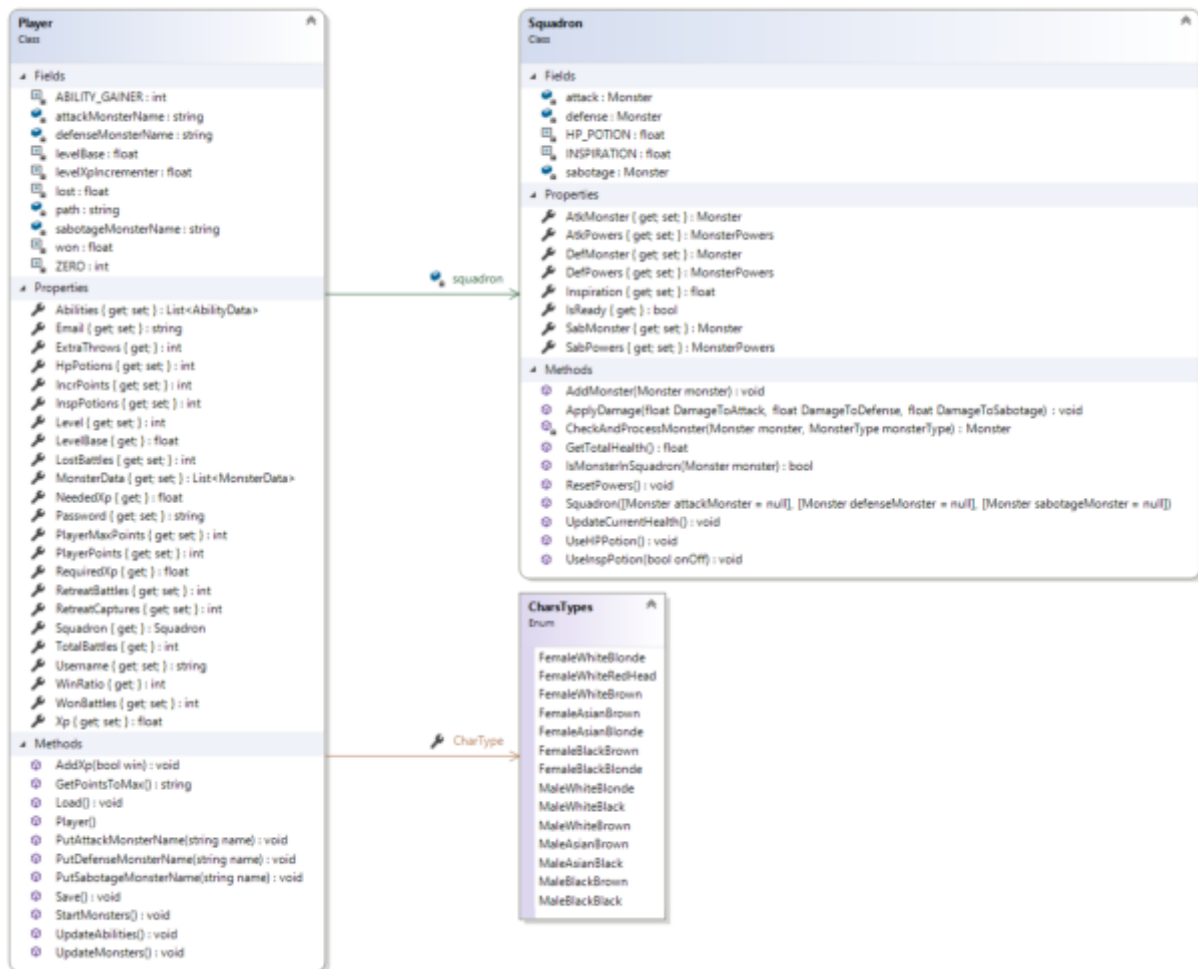


Figura 6.6: Estrutura de Player.

A gestão de experiência e nível do jogador são feitos através desta classe, sempre que o jogador adquire experiência ou evolui os dados são guardados. Este comportamento também se aplica aos valores das Habilidades e dos Monstros. A classe Player guarda e gere todos os valores diretamente interligados ao jogador.

A classe Player contém todos os dados relativos à experiência do jogador, todos os dados relativos aos Duelos e Capturas que o jogador fez, os pontos que este tem, as poções que tem e dados de identificação. Esta classe é utilizada extensivamente pelos vários módulos do jogo SquadUp devido aos dados que contém e fornece. A afetação dos monstros e das habilidades também precisam de ser guardadas e para isto a classe Player é invocada para que os dados sejam guardados assim que haja alguma modificação.

Todos os dados da classe Player são guardados através da classe PlayerData que é uma classe serializável. Ou seja, a classe PlayerData utiliza o atributo Serializable que permite a escrita da sua estrutura de dados em ficheiro. Isto é importante devido ao facto da classe Player possuir uma vasta coleção de propriedades necessárias a guardar assim como listas de outras classes com o atributo Serializable.

Quando o jogador começa o jogo pela primeira vez a classe PlayerData é encarregada de fornecer os dados fornecidos pelo jogador no processo de registo, assim como inicializar as várias propriedades e listas de dados com um valor padrão.

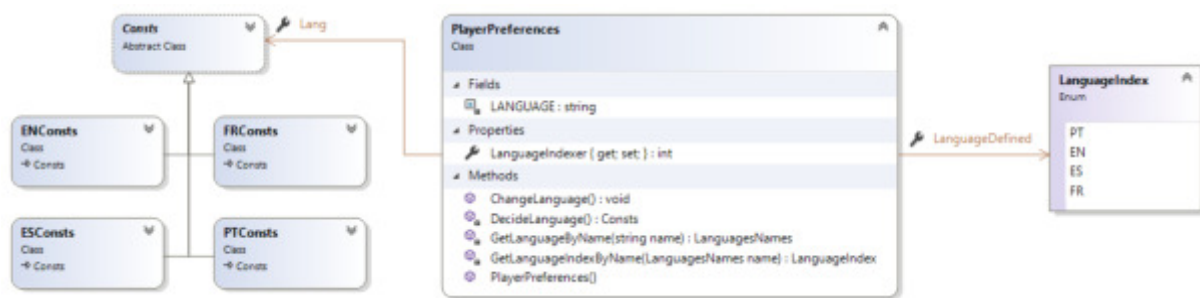


Figura 6.7: Estrutura de PlayerPreferences.

Apesar da classe Player servir de contendor dos dados do jogador, a preferência de idioma por parte do jogador é gerida pela classe PlayerPreferences. O idioma escolhido pelo jogador é guardado em *cache* não volátil. Esta classe também é responsável por fornecer acesso ao módulo de mensagens do jogo, que contém todas as mensagens e textos utilizados nas várias GUI dos seus menus, tutoriais e cenários.

6.5.1.1 Esquadrão

Uma das principais características do jogo SquadUp é a possibilidade de o jogador poder utilizar os seus monstros para combater em Duelos contra outros monstros inimigos. Para isto o jogador precisa de formar o seu esquadrão. A classe Squadron é a classe que representa o esquadrão dentro do jogo SquadUp, não só para o do jogador mas também para o dos monstros inimigos quando o jogo se situa no cenário Duelo.

A classe Squadron serve para guardar os monstros que o jogador designou como monstros do seu esquadrão. Esta classe é utilizada extensivamente no cenário Duelo devido a fornecer os monstros dos esquadrões para as classes que necessitam de aceder às características de combate de cada um. Ainda no cenário Duelo, a classe Duelo é invocada no processo de atribuir o dano ao esquadrão defensivo.

6.6 Módulo Capture

O módulo Capture é responsável por criar e gerir o cenário "Captura". É composto por várias classes responsáveis pela gestão e verificação dos vários dados passados pelo cenário Mapa Mundo para o cenário Captura. Todos os processos principais de cada componente do módulo irão ser introduzidos em seguida.

6.6.1 Introdutor

Quando o jogo SquadUp passa do cenário Mapa-mundo para o cenário Captura é necessário verificar se a demonstração do tutorial é necessária e caso não seja é necessário apresentar ao jogador as opções iniciais em termos de procedimento. Esta cadeia de painéis gráficos são geridos e mantidos pela classe CaptureUIManager. A inicialização desta classe trata por demonstrar o tutorial referente ao cenário caso seja necessário e inicializa os botões do painel de opções iniciais. Para a inicialização dos botões é necessário embutir-lhes métodos, para que estes sirvam de eventos. Estes eventos são despoletados quando o botão é pressionado pelo jogador. Os procedimentos da inicialização da classe CaptureUIManager são os seguintes:

1. Verificação e demonstração do tutorial caso a opção de "Desligar Ajuda" esteja desativada;
2. Inicializa o botão de "Ajuda" com o método ToggleTutorial, que é responsável por apresentar o tutorial referente ao cenário Captura;
3. Inicializa o botão "Fugir" com o método ActivateConfirmationSubPanel. Este método trata de apresentar um painel de confirmação da opção fugir ao jogador. A este painel existem duas opções representadas por dois botões:
 - (a) O botão com a opção "Sim" recebe um evento criado em tempo de execução que trata de incrementar a quantidade de vezes que o jogador fugiu de capturas e em seguida invocar a transição para o cenário Mapa-mundo;
 - (b) O botão com a opção "Não" desativa o painel de confirmação, voltando ao painel com as duas opções "Fugir" e "Continuar".
4. Inicializa o botão "Continuar" com o método ContinueToCapture que desativa o painel das opções iniciais e ativa a órbita de captura.

6.6.2 Gestor

Após o processo inicial de opções o processo da tentativa de captura é gerido pela classe CaptureScene-Manager. Esta tem como responsabilidade instanciar o monstro que o jogador selecionou para captura, as órbitas que o jogador irá poder atirar para capturar o monstro e a invocação do menu referente ao resultado de captura. Quando o jogo passa para o cenário de Captura, este *script* é invocado automaticamente devido à sua derivação da classe MonoBehaviour que o motor Unity disponibiliza. Este comportamento é possível, caso o *script* esteja associado a um GameObject presente e ativo no cenário, em que neste caso o GameObject ao qual o *script* CaptureSceneManager está associado é a câmara de jogo.

A inicialização da classe CaptureSceneManager trata de instanciar e posicionar o monstro no mapa de jogo do cenário, em seguida calcula e escreve na GUI do cenário a quantidade de tentativas que o jogador possui para atirar as órbitas ao monstro.

Os restantes métodos pertencentes a esta classe gerem o cenário através de invocação externa. Para o caso do jogador falhar a sua tentativa de captura de órbita, é invocado o método OrbDestroyed, que verifica e contabiliza a quantidade de tentativas que o jogador ainda têm. Caso o jogador ainda tenha órbitas é criada uma nova e posicionada para utilização do jogador, caso não tenha mais tentativas é parada a criação de órbitas e invocado o menu de resultado da captura. O método OrbDestroyed é invocado pela órbita que foi criada e que não entrou em colisão com o monstro.

Quando a órbita entra em colisão com o monstro é invocado o método MonsterCollision, este método trata de invocar o MonsterManager através da classe GameManager e adicionar o id do monstro capturado. São atualizados os dados dos monstros na lista de MonsterData do jogador e para a criação de órbitas para o mapa de jogo. Em seguida é invocado o menu de resultado de captura.

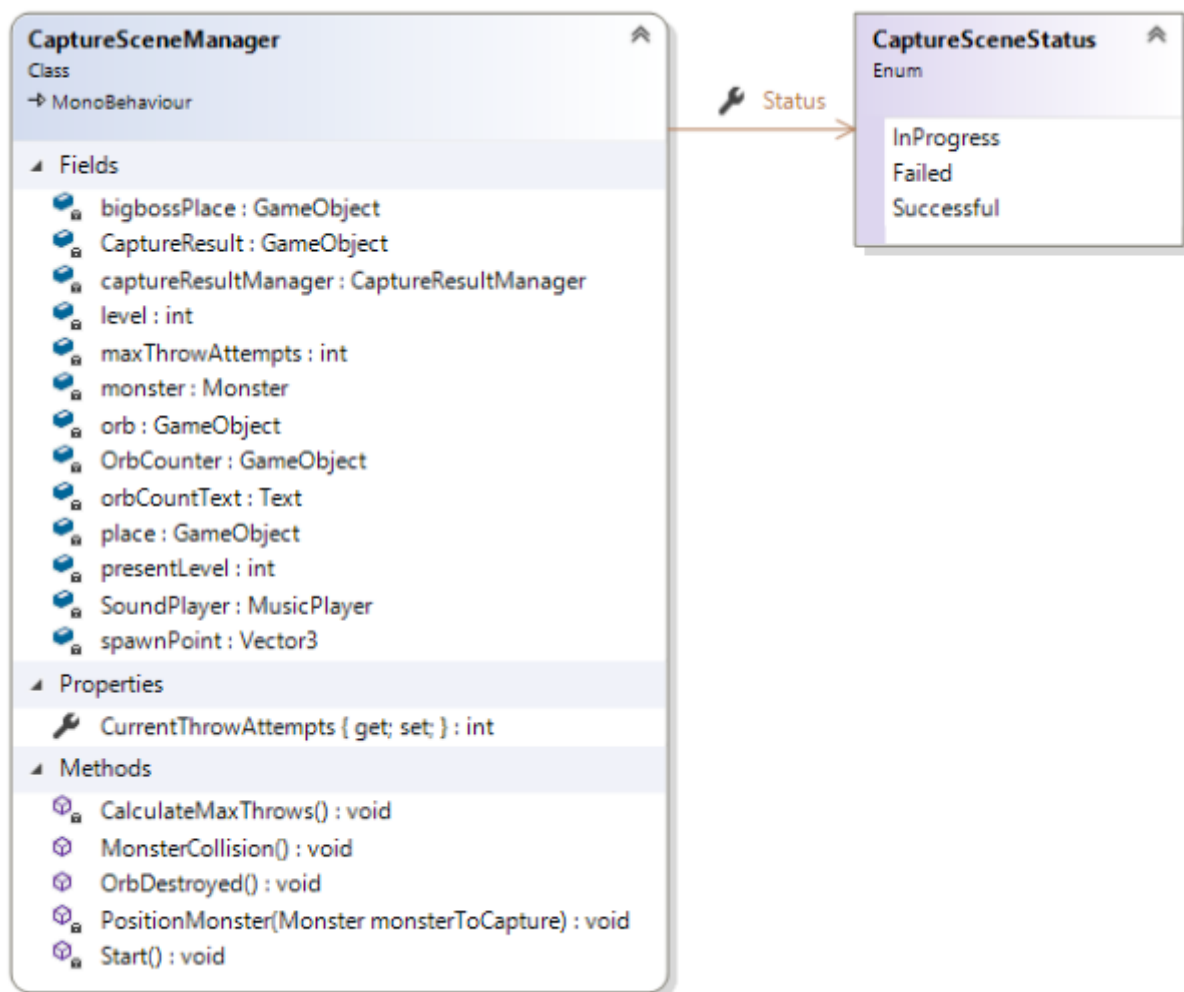


Figura 6.8: Estrutura de `CaptureSceneManager`.

6.6.2.1 Auxiliares de Gestão

Por motivos de simplificação e organização do processo da captura de um monstro foi necessário implementar dois tipos que representam estados para com a ação de captura de um monstro e para com a ação de utilização da órbita de captura. O enumerador `CaptureSceneStatus` é utilizado pela classe `CaptureSceneManager`, este representa os vários estados de situação possível na captura de um monstro. Estes estados são utilizados para a continuação da criação de órbitas de captura no método `OrbDestroyed`, em que o método verifica se a quantidade de tentativas de captura é maior que zero e se o estado não passou de `InProgress` para o estado de Sucesso. Os estados e os seus significados são descritos em seguida:

1. `InProgress` é o estado em que o jogador ainda está a tentar capturar o monstro;
2. `Failed` é o estado em que o jogador não conseguiu capturar o monstro e não possui tentativas;
3. `Successful` é o estado em que o jogador lançou com sucesso a orbita ao monstro e obteve colisão entre os dois objetos.

O enumerador `InputStatus` representa os estados de interação entre o utilizador e a orbita de captura. O significado de cada um destes estados é descrito em seguida:

1. Grabbing é o estado em que o jogador pressiona e mantém pressionada a órbita;
2. Holding é o estado em que o jogador arrasta a órbita pelo painel de jogo;
3. Releasing é o estado em que o jogador retira o toque sobre a órbita;
4. None é o estado em que o jogador não está a cometer nenhuma das ações referidas acima.

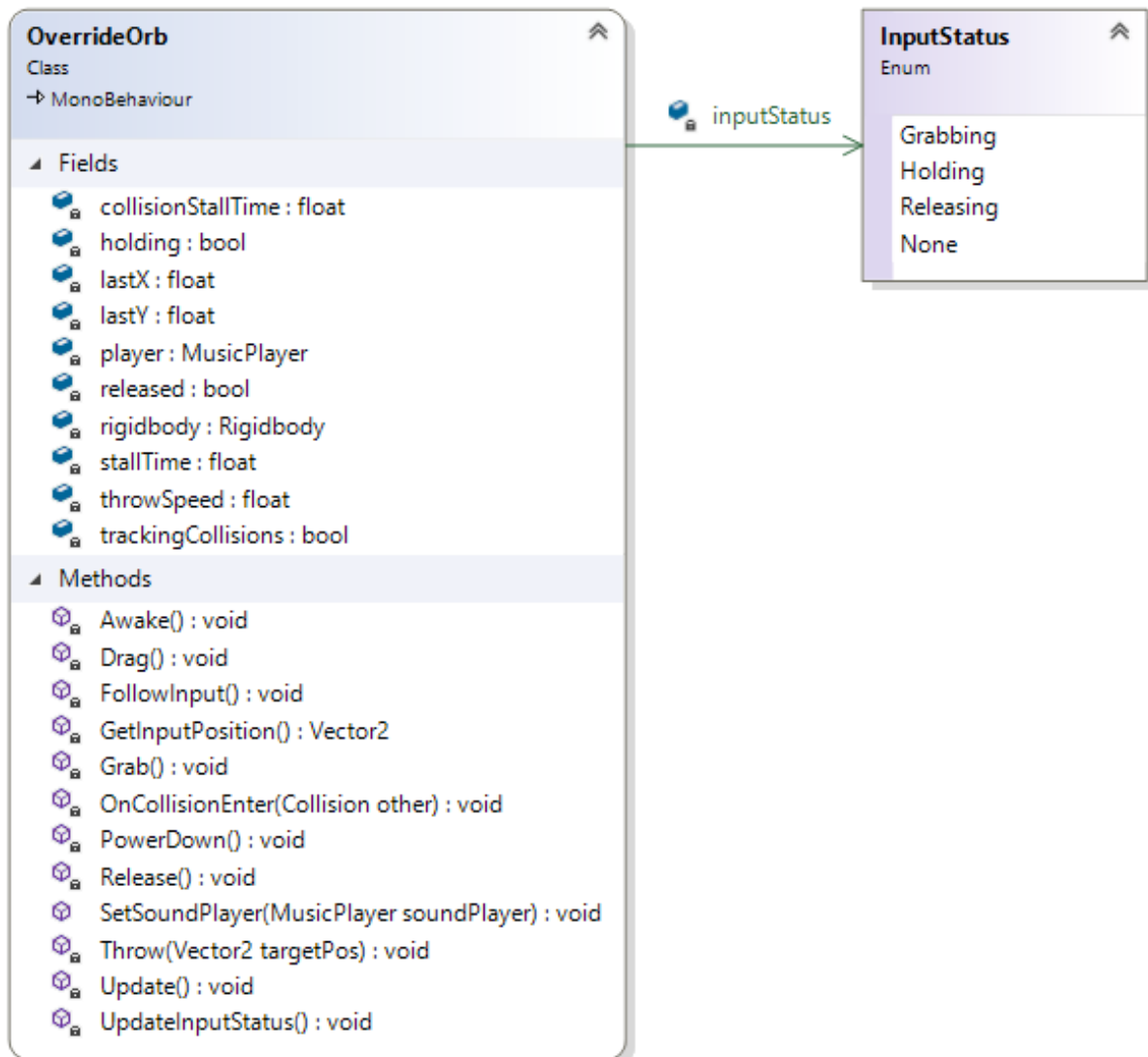


Figura 6.9: Estrutura de OverrideOrb.

Estes estados são utilizados pela classe `OverrideOrb` para gestão das várias ações da órbita em interação com o jogador. Estas ações são as seguintes:

1. Do estado Grabbing é invocado o método `Grab` que determina que a órbita está a ser pressionada;
2. Do estado Holding é invocado o método `Drag` que calcula as coordenadas da órbita em referência ao mapa de jogo;

3. Do estado Releasing é invocado o método Release que verifica se o jogador deixou de entrar em contacto com a órbita e caso seja verdade invoca o método Throw. Este método é responsável calcular a velocidade, força e direção da órbita.

A deteção da interação do jogador com a órbita é feita através do método Update que é executado a cada frame. Este método é responsável por atualizar o estado de interação até que o jogador atire a órbita ao monstro. Quando a órbita é solta pelo jogador a sua colisão irá invocar o método OnCollisionEnter que é responsável por verificar se a órbita entrou em colisão com um monstro ou não para a reprodução de efeitos sonoros respetivos a cada resultado. Se a órbita não colidir com um monstro é invocada a destruição desta através do método PowerDown. Este método também invoca o método OrbDestroyed da classe CaptureSceneManager que é responsável pela atualização da quantidade de tentativas de captura do jogador e o procedimento em caso de estas não serem superiores a zero.

6.6.3 Encerrador

O painel de resultado do cenário de Captura é gerido e preenchido pela classe CaptureResultManager. Esta classe é responsável por preencher todos os dados necessários para que o jogador obtenha a informação respetiva ao encerramento do momento de captura. Os dados que a classe preenche são referentes às características pessoais do monstro, as quantidades de cada tipo de poção que o jogador ganhou e se teve sucesso ou não. Os dados são os seguintes:

1. Nome;
2. Quantidade de vezes que foi visto;
3. Quantidade de vezes que foi capturado;
4. O seu nível e se evoluiu;
5. Quantidade de poções de vida;
6. Quantidade de poções de inspiração.

Todo o preenchimento de dados é feito na inicialização da classe e também é incluído no botão "Voltar" um evento para transacionar para o cenário Mapa-mundo para que o jogador possa voltar ao cenário principal.

6.7 Módulo Duelo

O módulo Duelo é um dos módulos mais complexos do jogo SquadUp devido à sua extensa utilização de *scripts* para cálculos matemáticos e a interligação entre estes. Para respeitar as regras das habilidades foram implementadas classes para cada tipo de poder de maneira a gerir os efeitos destas aos respetivos monstros. Foi focado a automatização do processo de luta entre os dois esquadrões (o do jogador e o do inimigo) para um processo simples e acessível de modificações futuras.

6.7.1 Gestor

O Duelo é gerido pela classe `TowerBattleManager`, esta é responsável pela inicialização de vários componentes:

1. Verificação e apresentação do tutorial;
2. Cria e inicializa o esquadrão inimigo;
3. Inicializa o gestor dos combatentes;
4. Cria os monstros dos esquadrões para o mapa de jogo;
5. Inicializa o gestor de combate;
6. Inicializa as barras de vida dos monstros de ambos os esquadrões;
7. Adiciona aos botões pertencentes ao painel de luta os respetivos eventos;
8. Inicializa o painel de luta.

O item 2 da lista acima é processado através da classe `EnemySquadronGenerator`. Esta classe como o nome indica é responsável por inicializar o esquadrão inimigo e atribuir-lhe os monstros assim como as suas habilidades. Para alcançar este objetivo o `EnemySquadronGenerator` recebe como parâmetro no seu construtor a classe `Player` de maneira a ter acesso aos dados necessários para o seu processo de criação. O processo de criação passa por três fases:

1. O cálculo dos pontos das habilidades que é executado pelo método `DistributePointsToAbilitiesLines`. Este método recebe a quantidade máxima de pontos de habilidades que o jogador tem e distribui os pontos de maneira uniforme por três valores inteiros representantes dos tipos de poder (ataque, defesa, sabotagem). Quando a quantidade de pontos não é divisível pelas três categorias, esta é reduzida até ser alcançado um valor divisível. Os restantes pontos reduzidos da quantidade máxima são distribuídos de forma aleatória pelas três categorias;
2. A criação das habilidades é executada através da classe `AbilitiesContainer`. Utilizando o segundo construtor desta classe são passados os três valores, previamente calculados, para que sejam atribuídos estes pontos pelas várias habilidades de forma aleatória;
3. A atribuição dos monstros ao esquadrão é executada pelo método `GenerateEnemySquadron`. Este método utiliza a classe `MonsterManager` para receber monstros de forma aleatória para cada tipo de poder. Ao possuir os três monstros é gerado o esquadrão e o nível de cada um destes monstros é modificado para ser equivalente ao nível dos monstros do esquadrão do jogador.

O item 5 na lista refere-se à classe `CombatManager` que é responsável por gerir o combate entre os dois esquadrões. Para isto, é passado por parâmetro para o seu construtor a classe `FightersManager` que irá disponibilizar os dois esquadrões com as suas capacidades de combate geridas pela classe `FightStages`. Através da extensa utilização de tipificação para os vários processos de processamento de dados das características de combate dos esquadrões foi possível automatizar o processo de luta entre os dois.

Este processo é representado no método `Fight`, que utiliza o `FightersManager` para receber a entidade ofensiva e defensiva sem verificar qual destes é o jogador e qual é o inimigo. O método foca-se nas várias etapas necessárias para executar o processo de luta entre os dois esquadrões, sendo estas as seguintes:

1. Receber a entidade ofensiva do `FightersManager`;
2. Recebe a entidade defensiva do `FightersManager`;
3. Invocar o `ManageRounds` do `FightersManager` que aplica ao esquadrão ofensivo a propriedade booleana de ter atacado a verdadeiro. Esta condição é utilizada para verificação e procedimento para a seguinte ronda quando já ambos os esquadrões atacaram;
4. Processar as características ofensivas e defensivas das entidades respetivas;
5. Processar os poderes de ataque e defensivo das entidades respetivas;
6. Calcular o total de dano a atribuir;
7. Atribuir o dano ao esquadrão defensivo;
8. Verificar se existe condição de vitória;
9. Restaurar saúde aos monstros do esquadrão defensivo (dependente das habilidades com os respetivos efeitos estarem ativadas).

Esta classe é executada sempre que o jogador pressiona o botão "Atacar" ou "Defender" no painel de combate. Estes eventos são embutidos nos botões do painel de combate, como referido no item 7 da lista. Os botões que aplicam uma poção de saúde ou inspiração aos monstros do esquadrão utilizam os métodos `HealthPotion` e `InspirationPotion`. Estes métodos acedem ao esquadrão do jogador para aplicação dos métodos implementados na classe `Squadron` para este propósito. Cada uma das poções não pode ser utilizada mais que uma vez por cada fase da ronda, sendo limitadas por condições booleanas. O botão de "Atacar" ou "Defender" tem o seu evento aplicado através do método `Fight`. Este método separa o evento por três fases:

1. Fase de processamento de dados. É invocada a classe `CombatManager` para processar a luta entre os esquadrões, em seguida são atualizadas as barras de saúde dos monstros. Após o processamento dos dados estar executado é ativado o painel Ataque e Defesa;
2. Fase dos painéis de dados. O painel de Ataque e Defesa é apresentado ao jogador mas é embutido outro evento no botão "Continuar" deste painel. Este evento serve para a ativação do painel Resultado de Dano e desativação do painel Ataque e Defesa. As condições booleanas que impedem o jogador de utilizar as poções mais que uma vez são desativadas para que este possa as utilizar outra vez na próxima luta;
3. Fase de encerramento. Esta fase aplica ao botão "Terminar", do painel Resultado de Dano, um evento é responsável por desativar o painel e por ativar o menu de resultado de Duelo caso o combate esteja terminado.

O painel de combate Ataque e Defesa é preenchido pelos *scripts* `AttackerManager` e `DefensorManager`. Devido ao painel apresentar nas duas secções (ataque e defesa) os valores dos efeitos das habilidades assim como as características ofensivas e defensivas dos monstros dos esquadrões, foi necessário implementar uma classe dedicada a cada tipo de poder para cada estilo. Como se pode observar pela Figura 6.10, tanto o `AttackerManager` como o `DefensorManager` possuem três classes, cada uma dedicada ao tipo de poder do monstro respetivo. As classes `AttackerAttackManager`, `AttackerDefenseManager`,

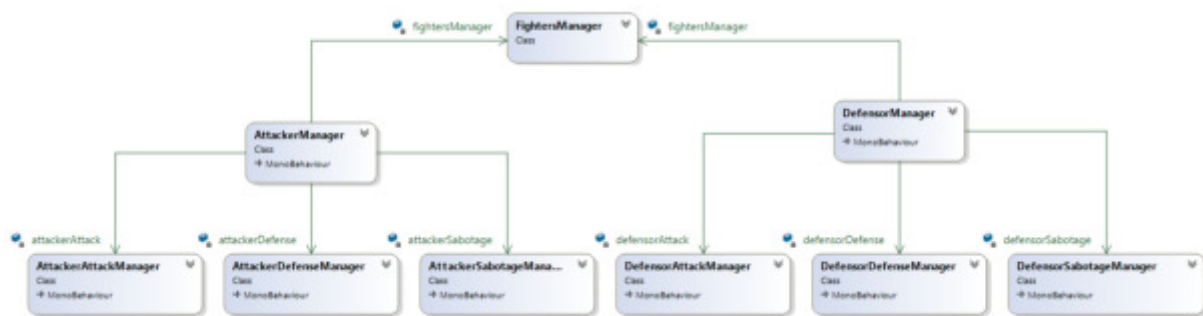


Figura 6.10: Classes responsáveis pelo preenchimento do painel Ataque e Defesa.

AttackerSabotageManager, DefensorAttackManager, DefensorDefenseManager, DefensorSabotageManager tem como único objetivo preencher os campos das habilidades e dos poderes do monstro que representam. O preenchimento destas é feito a partir da sua inicialização que é invocada a partir das classes AttackerManager e DefensorManager.

É aplicado o mesmo tipo de procedimento no preenchimento do painel Resultado de Dano, através da classe DamagePanelManager. Esta classe utiliza o FightersManager para aceder aos valores calculados e armazenados durante a ronda. Assim, o único cargo que a classe tem a executar é passar para os campos que representam as várias caixas de texto a preencher os valores que o FightersManager disponibiliza dos esquadrões ofensivo e defensivo. Um exemplo prático é a demonstração da quantidade de dano normal, dano crítico e passivo dado.

Quando o Duelo termina o jogador passa para o menu resultado, este menu é gerido e preenchido pela classe EndDuelPanel. Esta classe aplica o mesmo procedimento que as várias classes acima aplicam aos seus painéis. Utilizando o FightersManager consegue determinar se o jogador ganhou, também consegue determinar se o jogador evoluiu de nível e assim apresentar-lhe a quantidade de pontos de habilidade que ganhou ao evoluir. Também apresenta certas estatísticas do combate, como a quantidade de dano dado, recebido e defendido.

6.7.1.1 Gestor de Combatentes

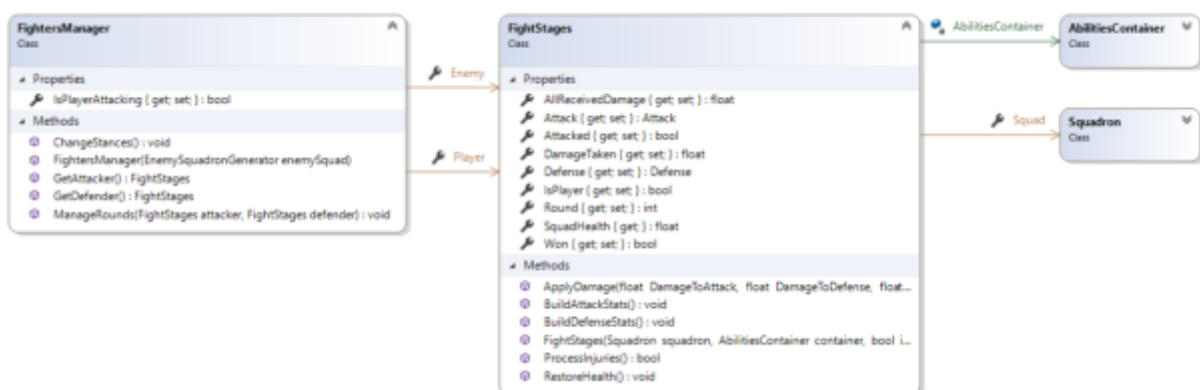


Figura 6.11: Estrutura de FightersManager e ligação com FightStages.

Após a criação do esquadrão inimigo é dada a inicialização da classe responsável por gerir ambos os esquadrões. Este passo referido pelo item 3 na lista, é executado pela classe FightersManager. Ao ser

inicializada, a classe `FightersManager` cria duas instâncias `FightStages` que representam o jogador e o inimigo. Estas instâncias são usadas para o processo de automatização do combate, pois através de uma condição booleana é possível atribuir o jogador à posição de atacante ou defensor e a classe que utiliza as instâncias `FightStages` para gerir o combate não precisa de verificar qual é o jogador qual é o inimigo.

6.7.1.2 Gestor de Esquadrão em Combate

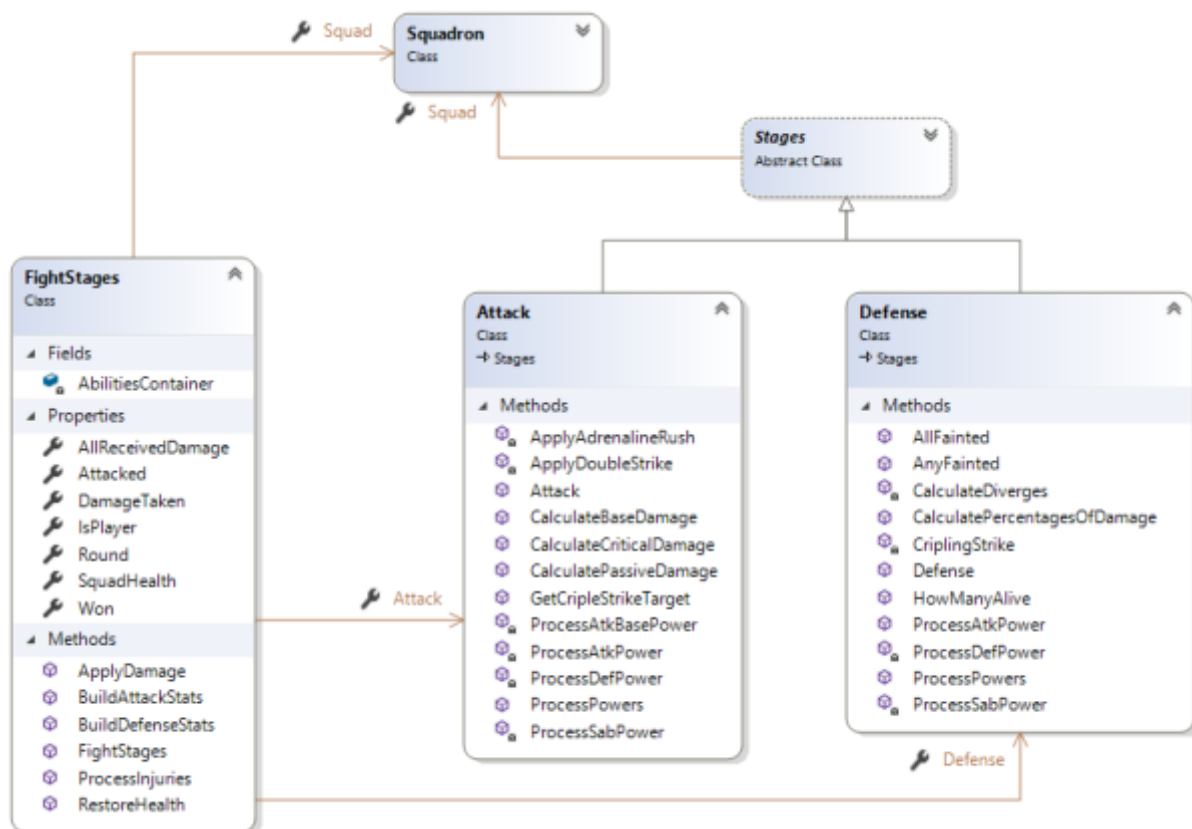


Figura 6.12: Estrutura de `FightStages`.

Um combate entre esquadrões pode levar várias rondas a concluir e cada ronda reparte-se pela fase de ataque e de defesa de cada esquadrão. Foi implementado um *script* para cada fase com o objetivo de simplificar o processo de cálculos e afetações a executar. A class `FightStages` é responsável por aplicar várias afetações ao esquadrão que representa, sendo estas as seguintes:

1. Aplicar os efeitos das habilidades aos monstros do esquadrão;
2. Aplicar o dano recebido do esquadrão atacante;
3. Verificar e calcular as condições de saúde de cada monstro;
4. Aplicar o restauro de saúde aos monstros do esquadrão (dependente das habilidades dedicadas a este efeito).

As propriedades desta classe também determinam se pertence ao jogador, se ganhou o combate, a quantidade de dano que o esquadrão recebeu na ronda ou o total de dano recebido ao longo do combate. Para

conseguir gerir o combate esta classe contém uma instância dos *scripts* Attack e Defense. Estas classes são responsáveis pelos cálculos matemáticos necessários para a gestão do combate e atribuição de dano ao esquadrão defensivo.

O *script* Attack é responsável por:

1. Reiniciar os campos que contém os valores calculados das várias habilidades da ronda anterior;
2. Calcular os valores de poder ofensivo de cada monstro do esquadrão, resultantes da aplicação dos efeitos das habilidades às características dos monstros do esquadrão;
3. Calcular as quantidades para cada tipo de dano (normal, crítico e passivo) a causa aos monstros do esquadrão defensivo.

O *script* Defense é responsável por:

1. Reiniciar os campos que contém os valores calculados das várias habilidades da ronda;
2. Calcular os valores de poder defensivo de cada monstro do esquadrão, resultantes da aplicação dos efeitos das habilidades às características dos monstros do esquadrão;
3. Calcular a quantidade máxima de valor defensivo para o dano do tipo normal do esquadrão ofensivo.



Design Funcional

A ergonomia de um comando de consola pode ditar se a experiência de jogo/usabilidade do jogador é positiva ou não. Isto implica que é necessário haver foco na estrutura física do comando para transmitir uma experiência positiva.

Sendo o SquadUp um jogo para *smartphones*, a sua plataforma não tem nem uma estrutura física definida nem um comando externo proprietário. Assim foram testados vários *designs* na procura do melhor design funcional para todos os controlos existentes. Esta abordagem é aplicada desde as coisas mais comuns como um botão de seleção até aos comandos de ação. Este capítulo apresenta as decisões de design realizadas durante o processo de desenvolvimento do jogo.

7.1 Painel de Habilidades

O painel de habilidade é o elemento gráfico que representa uma “Habilidade” em formato visual e lírico, sendo composto por vários elementos cada um com um propósito. Este elemento foi criado para representar as trinta habilidades que o jogo SquadUp possui, dando dez para cada linha de habilidades que existe, ou seja, a linha de habilidades de ataque, a de defesa e a de sabotagem possuem dez habilidades associadas a elas.

Na Figura C.2 e Figura 7.1 é possível observar os vários protótipos do painel de habilidade. O foco da ideia é que o utilizador consiga incrementar/decrementar os pontos na habilidade que deseja e passar para outra habilidade sem qualquer dificuldade de seleção.

Para que estes requisitos sejam respeitados e ao mesmo tempo o utilizador obtenha a melhor experiência visual/física em relação à posição das suas mãos, foi aplicado o layout apresentado pela Figura 4.23.

O layout da habilidade posiciona os únicos elementos interativos em cada ponta. Isto permite que o jogador possa percorrer a lista de habilidades sem a preocupação de pressionar em algum botão.



Figura 7.1: Evolução do design do painel de habilidade.

Existe também outra razão para que tenham sido aplicados os botões em cada ponta e isto deve-se ao posicionamento de mãos no dispositivo móvel.

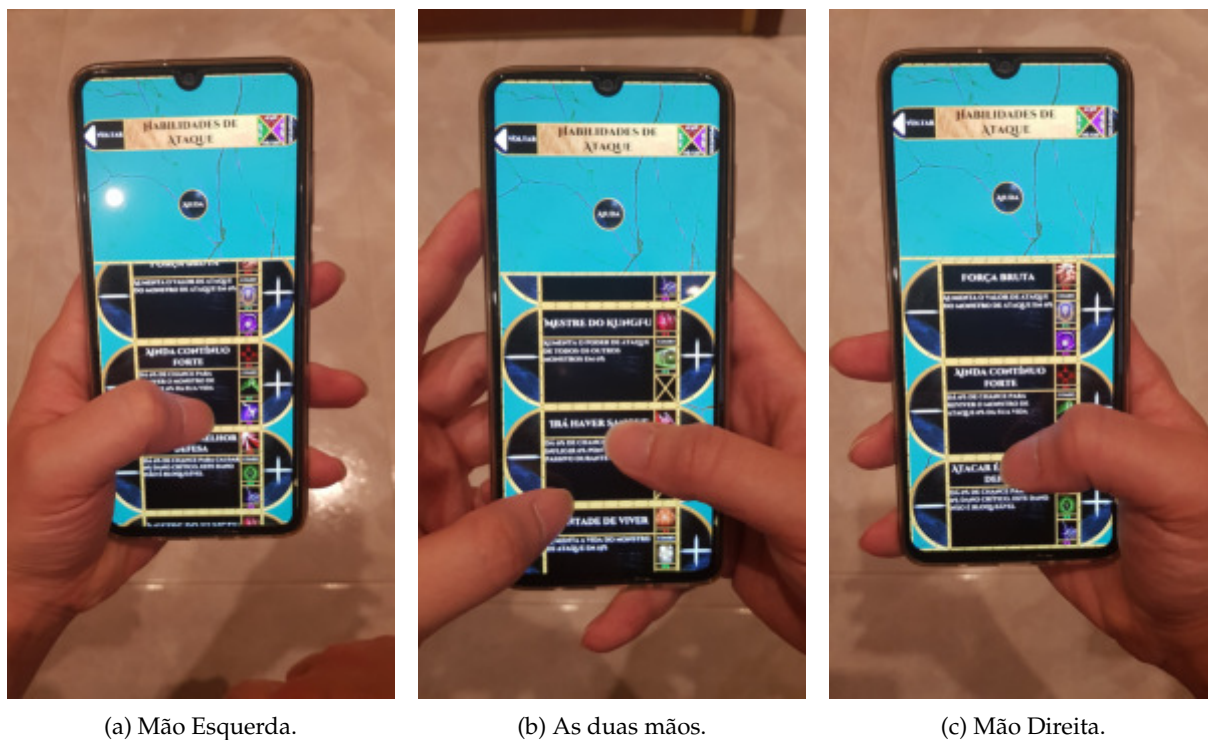


Figura 7.2: Exemplo de ergonomia das mãos.

Podemos observar na Figura 7.2 as variações da ergonomia das mãos do jogador ao utilizar o dispositivo móvel para jogar o jogo SquadUp. O instinto do jogador será agarrar no dispositivo móvel com as duas mãos como se fosse uma consola portátil, como por exemplo o *GameBoy*. Mesmo que o jogador queira utilizar o dispositivo móvel com uma mão, o movimento de dedos para percorrer a lista partirá do polegar. Sendo assim o centro do layout da habilidade revela-se um ótimo local para o jogador posar o seu dedo e aplicar o movimento de arrasto quando deseja percorrer a lista. Este tipo de ergonomia num dispositivo móvel é bastante importante pois o utilizador tem pouco espaço para posicionamento das mãos, especialmente quando o ecrã está em modo vertical de visualização.

7.2 Painel de Monstro

O menu Bestiário disponibiliza uma lista que contém todos os monstros do jogo. O jogador percorre a lista através de movimentos de *swipe up/down*, mas o jogador também pode interagir com a placa de monstro para conseguir aceder ao menu Monstro do monstro que deseja visualizar.

Estes painéis de monstro apresentam bastante conteúdo descritivo, que o jogador pode apreciar à medida que ganha experiência de jogo ao ajudá-lo na procura e seleção de monstros.

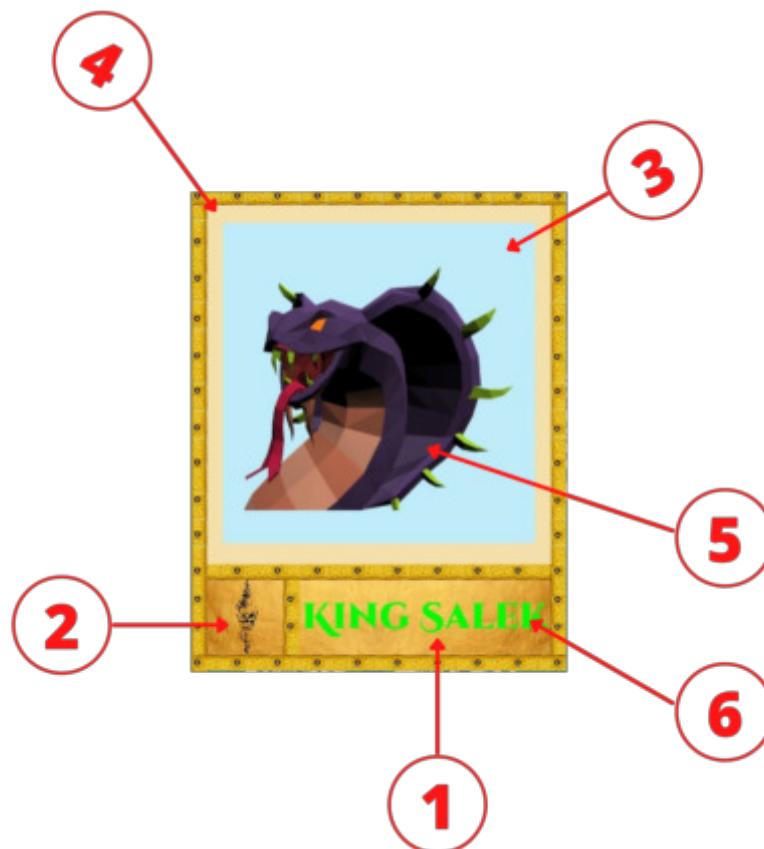


Figura 7.3: Identificação dos elementos do Painel de Monstro.

Os elementos que a painel de monstro apresenta são os seguintes:

1. Nome do monstro;
2. Tipo do monstro;
3. Subtipo do monstro;
4. Raridade do monstro;
5. Busto do monstro;
6. Estado de captura.

Através da Figura 7.3 podemos observar que o painel de monstro é composto por vários elementos, estes elementos estão listados acima numericamente de maneira a corresponder aos números usados na figura. O elemento 1 é o nome do monstro como descrito enquanto o elemento 6 é a cor do nome do monstro. Se o nome for colorido de verde significa que o monstro está capturado, se o nome for colorido de amarelo significa que foi apenas visto e não capturado. Para o caso em que o monstro não foi visto o nome é substituído por “???”.

Em seguida temos o elemento 2 que apresenta o tipo do monstro utilizando os três símbolos associados aos tipos de poder, sendo estes o poder ataque, defesa e sabotagem, como é possível observar pela Figura 7.4.

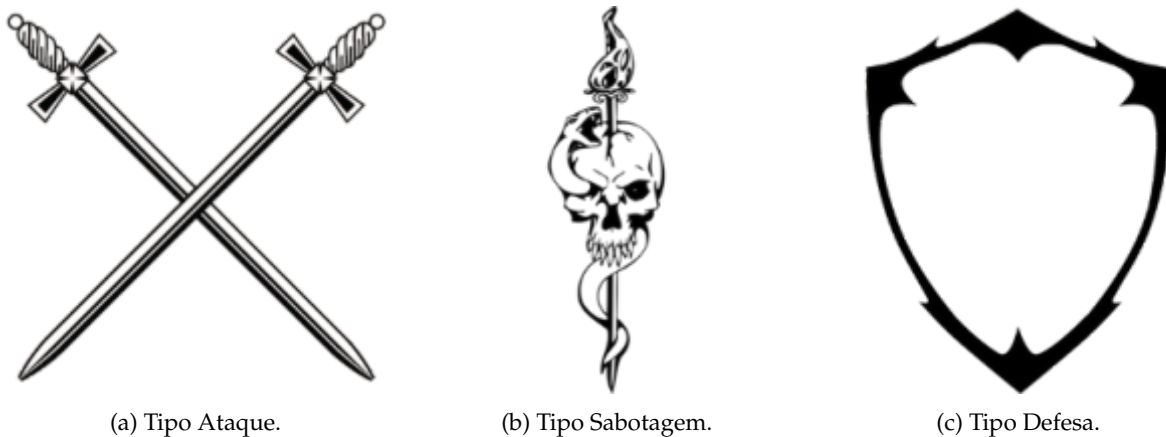


Figura 7.4: Símbolos dos poderes do jogo SquadUp.

O elemento 3 apresenta o subtipo do monstro preenchendo o fundo do painel com uma textura. Ao haver três subtipos são utilizadas três texturas pelos vários painéis de monstros. Na Figura 7.5 podemos observar as várias texturas e os respectivos subtipos que estas representam.

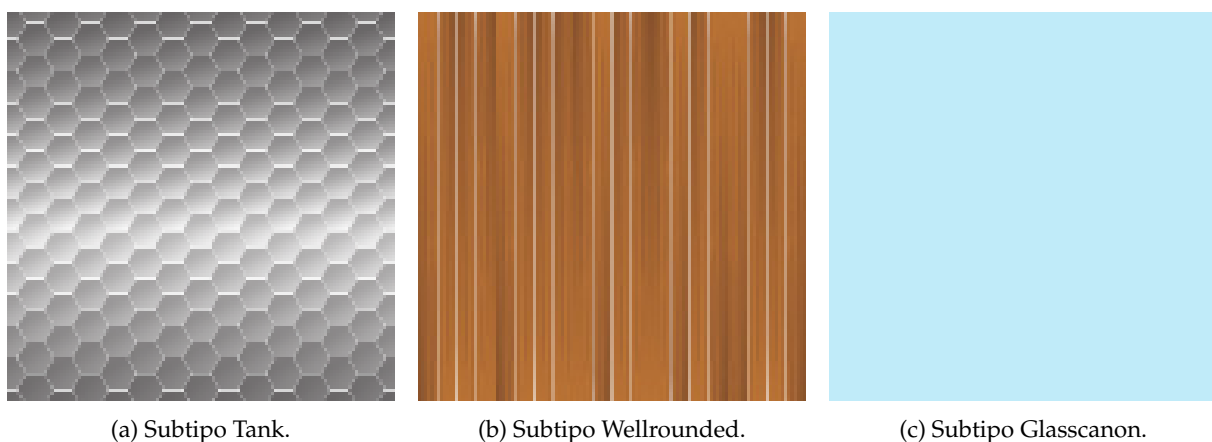


Figura 7.5: Texturas para identificação do subtipo de Monstro.

O elemento 4 apresenta a raridade do monstro através de cores. Este elemento é representado por um quadro em redor do elemento 3 e 5 como se tratasse de uma moldura. A escolha das cores não foi aleatória, devido à raridade que cada cor representa, usou-se as cores de certos elementos do mundo natural que também fazem parte de uma estrutura de raridade. Para a raridade comum foi usado a cor preta como associação à cor do carvão, para a raridade incomum foi usada a cor cinzenta como

associação à prata, para a raridade rara foi usada a cor ouro como associação ao ouro.

Para a raridade mítica usou-se a cor bege como associação à pureza dando ênfase à unicidade que a palavra mítica transmite. As cores são exemplificadas pela Figura 7.6.

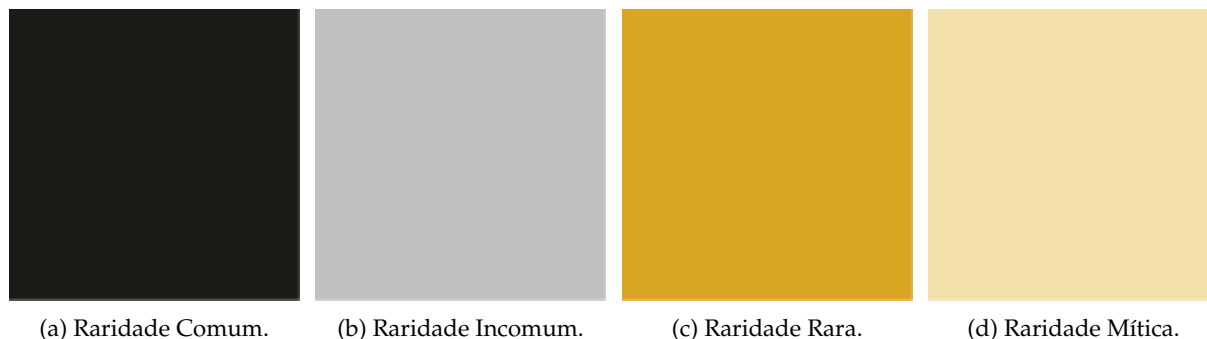


Figura 7.6: Cores para identificação da raridade de Monstro.

Foi aplicado este design no painel de monstro com o objetivo de usar o máximo de espaço disponível, para transmitir informação ao jogador sobre o monstro. À medida que o jogador adquire a experiência de jogo também adquire memória muscular e visual, permitindo ao jogador filtrar a informação apresentada pelos vários painéis na lista do “Bestiário” com mais eficiência e rapidez. Assim, através dos elementos visuais o jogador não precisa de entrar no menu “Monstro” de todos os monstros que tem na sua lista do “Bestiário”, para procurar o monstro que deseja adicionar ao esquadrão.

7.3 Cabeçalho

O elemento cabeçalho foi construído com o propósito de disponibilizar ao jogador informação de localização e ao mesmo tempo outras funcionalidades. Este elemento é referido por cabeçalho devido a ser posicionado sempre no topo dos menus.

Foi posicionado no topo para que os menus tivessem o restante ecrã disponível para a sua interface gráfica e também por seguir a implementação gráfica tradicional da opção “Voltar” ser no topo superior esquerdo. Um exemplo bastante conhecido da opção “Voltar” são os *Webbrowsers* que aplicam o botão no canto superior esquerdo da página, outro é a aplicação *Instagram* que utiliza um símbolo de uma seta, com a ponta direcionada para a esquerda para representar a opção “Voltar Atrás”.

Tendo em conta essas várias regras de implementação gráfica foi feito o cabeçalho de maneira a ocupar o restante espaço à direita do botão “Voltar” de maneira eficaz e funcional.

Como é possível observar na Figura 7.7 existem três variações do cabeçalho, cada uma diferenciada em termos de funcionalidades disponíveis.

Cada cabeçalho tem uma nomeação dedicada, sendo listado os nomes de cada, pela ordem topo para base:

- Cabeçalho Padrão;
- Cabeçalho Incapacitado;
- Cabeçalho Contador.



(a) Variação Padrão.



(b) Variação Incapacitado.



(c) Variação Contador.

Figura 7.7: Exemplos das variações do cabeçalho de menu.

Ao ser usado na maioria dos menus, o cabeçalho apresenta-se ao jogador como um elemento pilar da estrutura gráfica do jogo SquadUp. A aplicação extensiva do cabeçalho ajuda a criar um padrão gráfico no jogo SquadUp, gerando a possibilidade de reutilização caso seja necessário estabelecer novos menus. Também contribuí para uma apresentação simples de menu assim como para o reconhecimento gráfico da parte do jogador, estabelecendo a ligação de tipos de menus aos variados cabeçalhos.

O desenho do cabeçalho foi baseado na ideia de o jogador reconhecer os botões através do formato destes. Sendo o formato dos botões baseado no formato do dedo indicador.

Ao utilizar o formato do dedo indicador é possível estabelecer uma ponta arredondada, estabelecendo a ligação entre o jogador e o formato visual tradicional dos botões associados na interação de menus. Em seguida ao botão é possível reutilizar o espaço para atribuir a legendagem da função do botão.

7.3.1 Cabeçalho Padrão

O cabeçalho padrão é o cabeçalho que se pode observar em todos os menus com exceção do menu “Definições” e de “Linha de Habilidades”. Este cabeçalho disponibiliza ao jogador o nome do menu onde se situa assim como a funcionalidade de voltar para o menu anterior e aceder ao tutorial referente ao menu.

7.3.2 Cabeçalho Incapacitado

O cabeçalho “Incapacitado” é o cabeçalho do menu “Definições”, este cabeçalho disponibiliza ao jogador o nome do menu e a opção de voltar para o menu principal se o jogador estiver no cenário Mapa-mundo, ou para a página de autenticação, caso o jogador esteja no cenário Autenticação.

A opção de aceder ao tutorial foi removida e a sua apresentação gráfica substituída por um “X” no local onde se situa a legenda “Ajuda”. Esta substituição foi aplicada para que o jogador consiga reconhecer que nesta variação de cabeçalho não existe nenhuma opção de aceder ao tutorial. Ao não ser disponibilizado um botão de “Ajuda” externo como em outros menus o jogador consegue deduzir que não existe nenhum tutorial.

Foi escolhido um “X” como símbolo de inexistência/bloqueado devido à associação do símbolo “X” ser atribuída a este tipo de efeitos em diversos conteúdos temáticos e interativos.

7.3.3 Cabeçalho Contador

O cabeçalho contador é utilizado no menu “Linha de Habilidades” com o objetivo de disponibilizar ao jogador uma contagem de pontos de habilidades aplicados pelas diversas linhas de habilidades assim como a contagem de pontos do jogador. Isto é conseguido através da substituição do botão “Ajuda”. Este botão passa a ser um botão externo, com a funcionalidade de acesso do tutorial referente ao menu Linha de Habilidades.



Figura 7.8: Exemplificação do Botão externo Ajuda.

Esta informação é disponibilizada para que o jogador tenha um sítio no menu, onde pode tomar conhecimento dos pontos de habilidades que possui e conseqüentemente decidir a aplicação dos pontos.

O contador do cabeçalho também combina com os contadores disponibilizados no recipiente da Habilidade. Ao ter acesso à quantidade de pontos livres por aplicar e à quantidade de pontos que estão aplicados tanto na habilidade como nas habilidades que fazem combo com a determinada habilidade, o jogador consegue ter um processo de decisão de aplicação de pontos simplificado.

7.4 Menu Principal

O Menu Principal apresenta uma interface gráfica relativamente abstrata devido ao seu conteúdo, mas mesmo assim foi aplicado uma funcionalidade em termos de design. Esta funcionalidade é referente à ordenação dos botões, como se pode ver pela Figura 4.9 no capítulo “Menu Principal” as opções disponibilizadas ao jogador estão ordenadas da seguinte maneira:

1. Sair Da Sessão;
2. Definições;
3. Bestiário;
4. Jogador;
5. Esquadrão;
6. Habilidades.

Na ordenação das opções foram analisados três critérios: (1) importância do conteúdo informativo, (2) utilidade para o jogador e (3) distância dos dedos do jogador conforme o posicionamento do dispositivo móvel na mão deste. A ordenação das opções é da mais importante à menos importante começando pela base ao topo, logo a opção mais importante será a primeira a contar da base. Contudo esta ordenação terá uma exceção para impedir escolhas de opção erradas.

Devido a estes fatores foi escolhido que a opção com menor importância é a primeira, ou seja, a opção “Sair Da Sessão” devido a ser uma opção que irá ser muito menos utilizada em comparação às restantes.

Já a opção mais usada, a opção “Habilidades” é a opção que se situa mais perto da base do ecrã, isto deve-se ao facto do jogador conseguir selecionar esta opção sem muito esforço utilizando apenas os polegares e não tendo que arquear os dedos ou até pôr o dispositivo móvel na palma de uma mão enquanto usa a outra para selecionar esta opção.

A exceção mencionada acima é a opção “Bestiário”, esta opção é a segunda opção mais importante em termos de utilidade ao jogador, mas de maneira a evitar a seleção do menu errado foi separada da opção “Habilidades”. O foco de preocupação em evitar este tipo de acontecimento deve-se a diversos fatores, sendo estes os seguintes:

- A precipitação do jogador em aceder ao menu desejado;
- Falta de espaçamento entre as opções;
- Falta de espaçamento físico devido ao tamanho físico do dispositivo móvel;
- Forma física dos dedos do jogador ser acima da norma.

Sendo assim a opção “Bestiário” foi posta em quarto lugar a contar da base, para que dar um espaçamento de dois botões entre ela e a opção Habilidades.



Avaliação

Para avaliar o jogo SquadUp foram realizados testes para avaliar as funcionalidades e a experiência do jogador (jogabilidade) incluindo a usabilidade. Os testes de funcionalidade foram realizados numa primeira fase para não prejudicar a avaliação da jogabilidade, dado que alguns erros de código podem penalizar a experiência do jogador. Este capítulo é focado na apresentação dos resultados obtidos com os testes realizados com utilizadores para avaliação da experiência do jogador.

8.1 Metodologia

Os testes foram realizados individualmente por cada jogador utilizando o seu *smartphone*. O jogo e o questionário foram enviados para os utilizadores por email e redes sociais. Foram dadas três semanas aos utilizadores para experimentarem o jogo e responderem ao questionário.

Para avaliar as questões ligadas à usabilidade, após serem informados sobre os objetivos, os utilizadores usaram um questionário que os orientou ao longo do teste. O questionário, conforme explicado posteriormente neste documento, descreve as tarefas que os utilizadores devem realizar e apresenta questões relacionadas que os utilizadores devem responder após concluir a tarefa correspondente.

Relativamente à avaliação da experiência do jogador, não foi pedido aos utilizadores para seguirem o questionário. Os jogadores experimentaram o jogo e depois responderam a um conjunto de questões relativas à sua experiência enquanto estava a jogar e após jogarem.

O tempo de duração de cada teste foi variado, dependendo da estratégia de cada utilizador para explorar o jogo. As informações obtidas com estes testes têm como objetivo final refinar e melhorar o jogo com o *feedback* dos utilizadores.

8.2 Participantes

Os testes foram realizados por 14 participantes voluntários. As Figuras A.1, A.2 e A.3 no Apêndice A indicam as características principais dos utilizadores que participaram nos testes. A maioria é do género masculino (9 participantes do género masculino), com uma idade entre os 26 e 40 anos de idade. Metade dos utilizadores gasta entre 3 a 5 horas diariamente no *smartphone*. No entanto, na questão referente ao tempo que o utilizador joga no *smartphone* diariamente a resposta mais escolhida foi “Não jogo no *smartphone*”, como é possível de observar na Figura A.4. Contudo só três utilizadores dos participantes responderam que não jogavam em outras plataformas (ver Figura A.5). Com estes dados é possível determinar que a amostra não está habituada a jogar em *smartphones* o que pode levar a alguma divergência nas respostas dadas às questões das seguintes secções.

Uma discrepância elevada nas respostas revela que a maioria dos participantes não concorda com as afirmações feitas pelas questões, o que levanta questões sobre se o jogo pode ou não ter problemas referentes ao conteúdo das questões e secções com elevada variância.

Apesar da maioria dos utilizadores que participaram nos testes não estar habituada a jogar em *smartphones* os valores obtidos pelas várias secções podem revelar várias melhorias a aplicar no jogo SquadUp com o objetivo de atrair jogadores de outras plataformas. Também é possível obter várias críticas, pós questionário, caso os utilizadores queiram disponibilizar quando questionados sobre o que não gostaram, o que melhoravam e o que retiravam se pudessem.

8.3 Questionário

O jogo SquadUp foi avaliado através de um questionário focado em avaliar a usabilidade e experiência de jogo. O questionário foi criado através da ferramenta *Google Forms*. O questionário é composto pelas seguintes secções: (1) caracterização dos jogadores, (2) avaliação da usabilidade e (3) avaliação da experiência do jogador.

A construção da secção de avaliação da usabilidade foi baseada nos questionários, *System Usability Scale* (SUS) [29] e *Software Usability Measurement Inventory* (SUMI) [28]. Esta secção está organizada em tarefas com o objetivo de avaliar componentes específicas da interface utilizador. No fim da secção são colocadas as questões do SUS para fazer uma avaliação da usabilidade geral. A avaliação desta secção é feita através de questões do tipo *Likert*, numa escala composta por 5 pontos, em que o valor mínimo 1 corresponde a “Discorda Completamente” e o valor máximo 5 corresponde a “Concorda Completamente”.

A secção de avaliação da experiência do jogador foi construída com base no questionário, *Game Experience Questionnaire* (GEQ) [10]. Este questionário é composto por vários módulos com o objetivo de avaliar as diversas componentes ligadas à experiência do utilizador. Para avaliar o jogo SquadUp foram utilizados os módulos *Core* e *Post Game*. O primeiro avalia a experiência durante o jogo e o segundo avalia como os jogadores se sentiram depois de pararem de jogar.

O GEQ também inclui um módulo para avaliar a componente social. Não esquecendo o objetivo principal do jogo ser aproximar pessoas, este módulo não foi usado no questionário por motivos de segurança e bem estar causados pela pandemia que decorre.

Os módulos utilizados também são compostos por questões do tipo *Likert* numa escala composta por 5 pontos mas com diferentes significados:

0. *Not at all;*
1. *Slightly;*
2. *Moderately;*
3. *Fairly;*
4. *Extremely.*

Para não correr o risco de alterar os objetivos dos questionários de referência através de traduções que podem não corresponder aos mesmos objetivos na língua portuguesa, o questionário do jogo SquadUp foi escrito na língua Inglesa, tal como nas versões originais do questionários de referência.

8.4 Resultados

<i>Task 1 - Go to the Squad menu. I found the Squad menu without difficulty.</i>							
<i>Task 2 - Change the language of the game to a different one than the current one. I changed the language of the game without difficulty.</i>							
<i>Task 3 - Go to one of the Abilities Line menus and increase the points value of an ability. I increased the points value of an ability without difficulty.</i>							
<i>Task 4 - Access the capture mode of a monster. I entered the capture mode without difficulty.</i>							
<i>Task 5 - Go to the Bestiary menu and change the list of monsters to only those of Defense and with Common rarity. I changed the list of monsters to the specific filters without difficulty.</i>							
<i>Task 6 - Start a duel. I started a duel without difficulty.</i>							
<i>Task 7 - Start a Duel. I understood the panel of values shown during the duel without difficulty.</i>							
Tarefas	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7
Média	4.93	5.00	4.29	4.86	4.36	4.86	3.93
Mediana	5.00	5.00	5.00	5.00	4.50	5.00	4.00
Moda	5.00	5.00	5.00	5.00	5.00	5.00	5.00
Variância	0.07	0.00	0.92	0.27	0.66	0.27	1.21
Desvio Padrão	0.26	0.00	0.96	0.52	0.81	0.52	1.10

Tabela 8.1: Avaliação dos resultados referentes à secção Tarefas do questionário.

A Tabela A.1 no Apêndice A apresenta várias estatísticas calculadas com as respostas dos participantes às questões de usabilidade, relacionadas com as dificuldades dos utilizadores a cumprir as tarefas pedidas. É possível ver os mesmos valores na Tabela 8.1 apresentados em um formato diferente.

Como podemos verificar, a maioria dos participantes do questionário executou as tarefas sem dificuldade, havendo certos casos com alguma disparidade nos valores do desvio padrão. O pior caso é referente à última tarefa revelando que os jogadores tiveram dificuldades em compreender a interface dos painéis de análise de valores (ver Figura 4.28 e Figura 4.29), apesar da média revelar-se acima do valor considerado médio (2.5), o que é positivo, o valor do desvio padrão está acima de 1 o que indica que existe uma disparidade na votação executada. Assim é de concluir que a interface dos painéis deve ser melhorada e/ou refeita com o intuito de proporcionar aos jogadores uma leitura mais simplificada e de percepção quase automática.

A segunda subsecção da avaliação da usabilidade é referente ao SUS, que é um sistema composto por dez questões com o objetivo de avaliar a usabilidade de uma aplicação de um ponto de vista global. Estas afirmações podem ser observadas na Tabela A.2 do Apêndice A, assim como os resultados obtidos. Com as respostas dos utilizadores às questões do SUS é possível calcular uma pontuação que é utilizada para medir a usabilidade da aplicação.

Com base em muitas experiências realizadas foi possível construir a escala [30] apresentada na Figura 8.1. Esta escala permite classificar os problemas de usabilidade com base na pontuação.

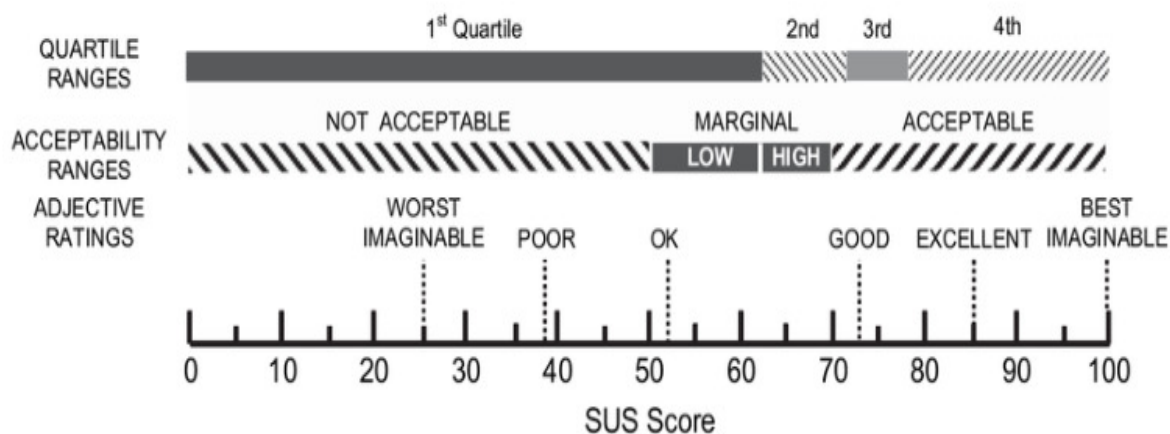


Figura 8.1: Escala de Pontuação do SUS.

A avaliação dos utilizadores resultou numa pontuação de 70.89 (ver Tabela A.3), considerando a tabela de pontuação do SUS este é um resultado bom. Apesar do resultado ser acima do valor médio (68), podemos observar na Tabela A.2 que há muita variância na maior parte das perguntas. O que leva a ser preciso estudar a fundo o que levou a diferença drástica nas votações feitas pelos utilizadores. Por exemplo, a segunda pergunta refere-se ao jogo ser considerado desnecessariamente complexo, à qual a Mediana e a Moda das respostas foi 1. Isto é um valor positivo pois os jogadores discordam da afirmação, mas apesar da Moda indicar que a resposta mais dada foi a opção 1, ainda existe uma variância acima de 1 indicando que existem utilizadores que encontraram o jogo complexo. Isto revela que o jogo necessita de ser melhorado na explicação e apresentação das suas regras/mecânicas aos jogadores.

A experiência do utilizador é avaliada através do módulo *Core*, este divide-se em vários componentes que medem várias partes da experiência. No Apêndice B são representados os vários componentes, cada um por uma tabela específica que apresenta as questões e valores obtidos por parte da avaliação executada pelos jogadores.

O módulo *Core* é composto pelos seguintes componentes:

1. *Competence* (ver Tabela B.1);
2. *Sensory and Imaginative Immersion* (ver Tabela B.2);
3. *Flow* (ver Tabela B.3);
4. *Tension/Annoyance* (ver Tabela B.4);
5. *Challenge* (ver Tabela B.5);
6. *Negative Affect* (ver Tabela B.6);
7. *Positive Affect* (ver Tabela B.7).

Da escala de valores utilizada (0 a 4), podemos determinar que o valor médio é 2. Cada componente tem como objetivo avaliar a experiência de jogo do utilizador em referência ao estado de espírito que indica pelo seu nome. A avaliação dos componentes é conseguida através da média total, ou seja, a média de todas as respostas das perguntas de cada componente. Apesar da avaliação ser feita no total também foi feita a avaliação individual dos valores para cada questão de cada componente. Caso a avaliação total não seja perceptível a avaliação individual pode ajudar a reforçar este resultado e a entender melhor as experiências reportadas.

Na Tabela 8.2 apresenta-se um resumo dos resultados obtidos e reportados na totalidade nas Tabelas B.1 a B.7 do Apêndice B. Todos os componentes foram analisados conforme a média, mediana, moda, variância e desvio padrão.

Componentes	<i>Competence</i>	<i>Sensory And Imaginative Immersion</i>	<i>Flow</i>	<i>Tension Annoyance</i>	<i>Challenge</i>	<i>Negative Affect</i>	<i>Positive Affect</i>
Média	2.71	2.54	1.97	0.83	1.23	0.61	3.21
Mediana	3.00	3.00	2.00	0.00	1.00	0.00	3.00
Moda	3.00	3.00	2.00	0.00	0.00	0.00	3.00
Variância	0.86	1.13	1.40	1.52	1.60	0.99	0.51
Desvio Padrão	0.93	1.06	1.18	1.23	1.27	0.99	0.71

Tabela 8.2: Avaliação dos componentes do módulo *Core*.

A avaliação do módulo *Core* revela que o jogo SquadUp conseguiu atingir mais de metade dos objetivos. Os componentes "*Tension Annoyance*", "*Challenge*" e "*Negative Affect*" têm o seu valor médio na totalidade abaixo de 2. Este resultado é um resultado positivo devido ao que os à consistência dos componentes, ou seja, tendo como exemplo o caso do componente "*Negative Affect*" o seu valor de média total (0.61) é abaixo do valor médio o que indica que a maioria dos jogadores discordou das afirmações. As afirmações deste componente questionam o utilizador em estados de espírito e sensações negativas, ou seja, como o nome do componente indica uma afetação negativa. O desvio padrão total (0.99) tem um valor alto, mas a mediana total (0.00) e o valor médio total indicam que a maioria dos participantes respondeu em conformidade. Este tipo de situações pode acontecer nos casos de utilizadores que podem não entender na totalidade o âmbito das questões ou até mesmo terem-se enganado. Mesmo que possa haver erros de votação é essencial, caso possível verificar com o utilizador se este foram os resultados que ambicionava dar e a razão para tal.

Para os componentes “*Competence*” e “*Sensory and Imaginative Immersion*” podemos verificar que ambos tiveram um valor de média acima do valor médio da escala. Estes componentes são importantes para a avaliação da interligação entre o jogo e os jogadores. O componente “*Competence*” foca-se em questionar o utilizador em relação à sua sensação de realização e integração com o jogo. O componente “*Sensory and Imaginative Immersion*” questiona o utilizador sobre a imersão que este teve com o jogo. A imersão de um jogador para o jogo é extremamente importante para criar uma relação de interesse entre o jogador e o jogo. Se o jogador não tiver atração ao mundo gráfico e à história do jogo, então o jogo não marca os jogadores e não terá muito tempo de vida.

Apesar de ser importante criar uma interligação especial com o jogador, como referido acima, o componente “*Flow*” teve um valor médio abaixo do valor médio, este valor seria considerado negativo, mas não para o caso do jogo SquadUp. Esta conclusão deve-se ao facto do jogo SquadUp tentar interligar o mundo real com a sua versão gráfica(mundo virtual), logo para que o jogador consiga explorar o mundo real e o visualizar de outro modo é necessário que o jogo não abstraia o jogador do mundo externo. Apesar do resultado ser positivo o valor de desvio padrão indica-nos que existem melhorias a aplicar em referência a esta área. Através da Tabela A.6 do Apêndice B podemos verificar o valor de mediana para a questão “*I was deeply concentrated in the game*” ser 3 e com um valor de desvio padrão considerável (0.97) é possível de deduzir que os jogadores estiveram demasiado concentrados no jogo. Isto revela que pode haver falhas na comunicação das regras, funcionamento e mecânicas de jogo aos jogadores. Uma das áreas a melhorar será a área dos tutoriais e a interface destes.

O módulo *Post Game* foca-se em avaliar o estado de espírito do jogador após jogar o jogo SquadUp. Este tipo de avaliação é essencial para determinar se o jogo, cria uma boa ou má memória no jogador. Se o jogo conseguir transmitir uma boa experiência ao jogador, então o jogo conseguirá expandir a sua população de jogadores, através da transmissão de experiência positiva dos jogadores para com os seus respetivos amigos. O módulo *Post Game* é composto pelos seguintes componentes (no Apêndice B são apresentadas várias tabelas com os resultados obtidos):

1. *Positive Experience* (ver Tabela B.8);
2. *Negative Experience* (ver Tabela B.9);
3. *Tiredness* (ver Tabela B.10);
4. *Returning To Reality* (ver Tabela B.11).

Os componentes do módulo *Post Game* têm a mesmo tipo de avaliação que os componentes do módulo *Core*, isto é baseada em questão do tipo *Likert*. Assim, a análise dos seus dados também foi realizada de forma semelhante. Como é possível verificar através da Tabela 8.3, todos os componentes obtiveram um resultado positivo. A Tabela 8.3 sumariza os resultados apresentados no Apêndice B.

Componentes	<i>Positive Experience</i>	<i>Negative Experience</i>	<i>Tiredness</i>	<i>Returning To Reality</i>
Média	2.19	0.24	0.25	1.10
Mediana	2.00	0.00	0.00	0.00
Moda	3.00	0.00	0.00	0.00
Variância	1.51	0.49	0.47	1.66
Desvio Padrão	1.23	0.70	0.69	1.29

Tabela 8.3: Avaliação dos componentes do módulo *Post Game*.

Para o caso dos componentes “*Negative Experience*” e “*Tiredness*” o facto de o valor médio ser abaixo do valor médio da escala é positivo, pelas mesmas razões já constatadas no módulo Core para os componentes negativos. O valor de Desvio Padrão destes componentes foi baixo o que é positivo e indica que a maioria da população votou de uma maneira uniforme. Este valor também reforça o valor Mediana (0) de ambos os componentes, o que indica que a maioria da população discorda completamente do âmbito negativo das afirmações. Este é um resultado positivo, ao indicar que a maioria dos participantes não ficou com uma imagem negativa do jogo.

O componente “*Returning To Reality*” obteve um valor de média abaixo do valor médio de escala, conseguindo um resultado positivo. Apesar do resultado ser positivo, os valores de Variância e Desvio Padrão são os mais altos de todos os outros componentes. O componente “*Flow*” do módulo Core têm o mesmo foco de análise que o componente “*Returning To Reality*”, de uma maneira mais detalhada, mas também obteve valores consideráveis de Variância e Desvio Padrão. Estes resultados podem indicar que os utilizadores possam não ter compreendido totalmente o âmbito das questões de ambos os componentes, devido à sua experiência de jogo em outras plataformas e também pela norma do jogador precisar de se concentrar no jogo para obter resultados positivos. Por outro lado, também pode indicar que o jogo pode estar demasiado complexo em certas partes, como já referido anteriormente. Consequentemente é necessário reformular certas mecânicas ou a demonstração destas.

Os resultados do componente “*Positive Experience*” entram em conflito com os resultados do componente “*Positive Affect*” do módulo Core. Apesar da média ser positiva em ambos os componentes, os seus valores de Média, Variância e Desvio Padrão apresentam uma diferença significativa. Este tipo de diferença pode indicar que os participantes, apesar de terem sentimentos positivos pelo jogo enquanto jogavam, após finalizar a o seu teste ao jogo (deixarem de jogar), a sua experiência ficou ressentida com algumas dificuldades sentidas na sua experiência. Este é um dos casos em que os utilizadores podem ser questionados posteriormente, de maneira a obter um significado mais contextual dos resultados deste componente.

Apesar dos módulos indicarem que o jogo SquadUp precisar de revisões nas suas mecânicas e interfaces, os resultados obtidos são positivos. Esta análise confirma e consolida que a estrutura do jogo foi aplicada corretamente e segue em bom caminho, necessitando de revisões. Este é um dos resultados mais comuns no mundo dos jogos, em que a ideia base apresenta aos utilizadores um conteúdo bastante cativante, mas a sua aplicação irá ser afinada em várias revisões executadas por um conjunto de *testers*.

9

Conclusão

Desde a introdução das redes sociais e *smartphones* a comunicação social evoluiu a um nível nunca antes visto. A possibilidade de as pessoas poderem comunicar globalmente e conseguirem partilhar os seus pensamentos assim como os seus feitos é um avanço extraordinário na comunicação global e individual. Apesar de estes avanços serem positivos para a sociedade, o processo de comunicação simplificado através de um dispositivo móvel/computador fez com que o processo de deslocamento e comunicação cara-a-cara num local físico tenha caído em desuso. Hoje em dia a iniciativa em comunicar com pessoas não conhecidas é cada vez mais baixa, devido ao ser humano não quebrar a falta de prática e apatia social com encontros casuais com outros seres humanos. Através da identificação destes problemas sociais causados pelas novas tecnologias, foi identificada a necessidade de parar a nova norma social e reverter o processo utilizando as mesmas tecnologias.

No decurso do trabalho realizado foi implementado um jogo para utilizar os *smartphones* de maneira a cumprir o objetivo do projeto. Ao utilizar o potencial técnico dos *smartphones* e a sua interação com o ser humano, criaram-se várias mecânicas de jogo para guiar o utilizador a pontos geográficos.

Abstraindo o jogador dos seus bloqueios internos para sair à rua e até mesmo comunicar com outras pessoas, ao cativar o utilizador a explorar o mundo exterior para progredir no jogo. O utilizador encontrar-se á em certos pontos geográficos comuns a outros utilizadores e a partilha de experiência de jogo entre os utilizadores pode simplificar o processo de comunicação e assim atingir o objetivo de aproximar pessoas.

Por razões conhecidas por todos não foi possível verificar se o jogo pode ajudar a aproximar pessoas. Contudo, foi desenvolvido o jogo com essa perspetiva e foram analisadas a usabilidade e a experiência que o jogo proporciona ao jogador. De uma forma geral os resultados foram positivos. Em relação, à usabilidade foi obtida uma pontuação de 70.89 no SUS. Este valor está acima de 68 que é um valor de referência que dá indicações que não há problemas de maior gravidade na usabilidade do jogo.

Relativamente à experiência proporcionada pelo jogo, uma usabilidade sem problemas maiores também é importante para uma experiência positiva. Os restantes componentes também apresentaram valores positivos. O resultado mais crítico e que necessita de ser melhorado, está relacionado com a experiência que fica na memória após jogar o jogo. Este resultado revela que há questões a melhorar porque os jogadores não esqueceram as dificuldades sentidas. Se não esqueceram é porque podem ter relevância na experiência.

9.1 Trabalho Futuro

A avaliação através de questionário aos utilizadores permitiu verificar que existe melhorias a aplicar no projeto. Estas melhorias abrangem várias áreas:

1. Área gráfica:
 - (a) Melhorar interfaces gráficas de utilizador com o objetivo de uma transmissão da informação simplificada;
 - (b) Melhorar a apresentação dos tutoriais e o seu conteúdo.
2. Área programática:
 - (a) Melhorar certas mecânicas de jogo e reavaliar a sua complexidade;
 - (b) Escrita/Leitura na componente *cloud* da aplicação.

É também como propósito no futuro avaliar o objetivo de aproximar pessoas do jogo. Isto é, se a ligação de um espaço virtual a um espaço físico e a necessidade do jogador ter de se deslocar ao espaço físico, pode motivar a aproximação de pessoas que não se conheciam.

9.2 Considerações Finais

A implementação de um jogo para *smartphones* com o objetivo de aproximar pessoas permitiu verificar que os utilizadores podem ser cativados por este conceito. Através das análises de dados do questionário conseguiu-se verificar que os utilizadores apreciam o conceito do jogo e das suas mecânicas. Indicaram também que é preciso melhorias a aplicar através da sua votação, apesar disto não foi possibilitado testar a componente social do jogo. Isto deveu-se à pandemia que decorre não possibilita o contacto entre utilizadores devido às regras de saúde pública.

Apesar de não haver a possibilidade de teste devido aos acontecimentos referidos acima, o facto de os utilizadores apreciarem o jogo e as suas mecânicas de jogo permite concluir que se atingiu em grande parte o objetivo traçado para este jogo.

Bibliografia

- [1] *Age of Empires Franchise - Official Web Site*, [Online; Acedido: 30-08-2021]. URL: <https://www.ageofempires.com>.
- [2] *Bumble - Date, Meet, Network Better*, [Online; Acedido: 24-06-2019]. URL: <https://bumble.com>.
- [3] Carlos Martinho, Pedro Santos & Rui Prada, em *Design e desenvolvimento de jogos*, FCA – Editora de Informática, LDA, 2014.
- [4] *Farmville 2*, [Online; Acedido: 23-06-2021]. URL: <https://www.zynga.com/games/farmville-2>.
- [5] *FarmVille Wiki*, [Online; Acedido: 23-06-2021]. URL: https://farmville.fandom.com/wiki/FarmVille_Wiki.
- [6] *Facebook*, [Online; Acedido: 24-06-2019]. URL: <https://www.facebook.com>.
- [7] Mihaly Csikszentmihalyi & Mihaly Csikszentmihalyi, *Flow: The psychology of optimal experience*. Harper & Row New York, 1990, vol. 1990.
- [8] Luis Von Ahn & Laura Dabbish, “Designing games with a purpose”, *Communications of the ACM*, vol. 51, n.º 8, páginas 58–67, 2008.
- [9] João Francisco & Rui Jesus, “Mixing the Game Experience with the Experience of Meeting People”, em *IEEE Games, Entertainment, Media Conference, GEM 2018, Galway, Ireland, August 15-17, 2018*, IEEE, 2018, páginas 1–9. URL: <https://doi.org/10.1109/GEM.2018.8516473>.
- [10] Wijnand A IJsselsteijn, Yvonne AW de Kort & Karolien Poels, “The Game Experience Questionnaire. Eindhoven: Technische Universiteit Eindhoven.(2013)”, *Eindhoven: Technische Universiteit Eindhoven*, vol. 46, n.º 1, 2013.
- [11] *Luís Pinto. Código da dissertação*. [Online; Acedido: 24-07-2021]. URL: <https://gitlab.com/luispintoxxx/dissertacao>.
- [12] *Grindr | The World’s Largest Social Networking App for Gay, Bi, Trans, and Queer People*. [Online; Acedido: 24-06-2019]. URL: <https://www.grindr.com>.
- [13] Yvonne Rogers & Harry Brignull, “Subtle ice-breaking: encouraging socializing and interaction around a large public display”, em *Workshop on Public, Community, and Situated Displays*, Citeseer, vol. 6, 2002.
- [14] *Instagram*, [Online; Acedido: 23-06-2021]. URL: <https://www.instagram.com>.

- [15] Nuno Correia, Luís Alves, Helder Correia, Luis Romero, Carmen Morgado, Luís Soares, José C. Cunha, Teresa Romão, A. Eduardo Dias & Joaquim A. Jorge, “InStory: A System for Mobile Information Access, Storytelling and Gaming Activities in Physical Spaces”, em *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, (Valencia, Spain), sér. ACE '05, New York, NY, USA: Association for Computing Machinery, 2005, 102–109, ISBN: 1595931104. URL: <https://doi.org/10.1145/1178477.1178491>.
- [16] Daniel Alves, Margarida Félix, Rui Jesus & Renato Abreu, “LeucoDiff: Mobile Application for Hematology Students Improve their Skills in Calculating the Leukocyte Formula.*”, em *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG)*, 2019, páginas 1–4. DOI: 10.1109/ENBENG.2019.8692458.
- [17] *League of Legends*, [Online; Acedido: 30-08-2021]. URL: <https://www.leagueoflegends.com/pt-br>.
- [18] *Maps, geocoding, and navigation APIs & SDKs | Mapbox*, [Online; Acedido: 30-08-2021]. URL: <https://www.mapbox.com>.
- [19] Nikolaos M Avouris & Nikoleta Yiannoutsou, “A review of mobile location-based games for learning across physical and virtual spaces.”, *J. Univers. Comput. Sci.*, vol. 18, n.º 15, páginas 2120–2142, 2012.
- [20] Cemile Tokgöz & Burak Polat, “Sociability on location based mobile games: an ethnographic research on pokémon GO and ingress in istanbul”, *European Journal of Social Science Education and Research*, vol. 5, n.º 1, páginas 111–120, 2018.
- [21] The Pokemon Company Niantic, *The Pokemon Company*, “Apanhe Pokémon no mundo real com o Pokémon GO!”, <https://www.pokemongo.com/pt-pt/>, [Online; Acedido: 24-06-2019].
- [22] MattRoot, *Pokémon GO Wiki*, https://pokemongo.fandom.com/wiki/Pok%C3%A9mon_GO_Wiki, [Online; Acedido: 20-06-2019].
- [23] Emily I.M. Collins, Anna L. Cox & Frank Lee, “Say Cheese! Games for Successful Academic and Student Networking”, em *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, (Austin, Texas, USA), sér. CHI PLAY '16, New York, NY, USA: Association for Computing Machinery, 2016, 105–115, ISBN: 9781450344562. URL: <https://doi.org/10.1145/2967934.2968096>.
- [24] Victor Fernandez-Cervantes, Eleni Stroulia, Luis E Oliva, Francisco Gonzalez & Claudio Castillo, “Serious games: Rehabilitation fuzzy grammar for exercise and therapy compliance”, em *2015 IEEE Games Entertainment Media Conference (GEM)*, IEEE, 2015, páginas 1–8.
- [25] Victoria McArthur & Robert J Teather, “Serious mods: A case for modding in serious games pedagogy”, em *2015 IEEE Games Entertainment Media Conference (GEM)*, IEEE, 2015, páginas 1–4.
- [26] Evie Powell, Felesia Stukes, Tiffany Barnes & Heather Richter Lipford, “Snag'em: Creating Community Connections through Games”, em *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 2011, páginas 591–594.
- [27] *Snapchat - The fastest way to share a moment!*, [Online; Acedido: 23-06-2021]. URL: <https://www.snapchat.com>.
- [28] Jurek Kirakowski & Mary Corbett, “SUMI: The software usability measurement inventory”, *British journal of educational technology*, vol. 24, n.º 3, páginas 210–212, 1993.

- [29] John Brooke, PW Jordan, B Thomas, BA Weerdmeester & IL McClelland, "Sus: a "quick and dirty" usability", *Usability evaluation in industry*, vol. 189, 1996.
- [30] John Brooke, "SUS: a retrospective", *Journal of usability studies*, vol. 8, n.º 2, páginas 29–40, 2013.
- [31] Duarte Gonçalves, Rui Jesus, Filipe Grangeiro, Teresa Romão & Nuno Correia, "Tag around: A 3D Gesture Game for Image Annotation", em *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, (Yokohama, Japan), sér. ACE '08, New York, NY, USA: Association for Computing Machinery, 2008, 259–262, ISBN: 9781605583938. URL: <https://doi.org/10.1145/1501750.1501811>.
- [32] *Tinder | Dating, Make Friends & Meet New People*, [Online; Acedido: 24-06-2019]. URL: <https://tinder.com>.
- [33] Ansgar E. Depping, Regan L. Mandryk, Colby Johanson, Jason T. Bowey & Shelby C. Thomson, "Trust Me: Social Games Are Better than Social Icebreakers at Building Trust", em *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, (Austin, Texas, USA), sér. CHI PLAY '16, New York, NY, USA: Association for Computing Machinery, 2016, 116–129, ISBN: 9781450344562. URL: <https://doi.org/10.1145/2967934.2968097>.
- [34] *Twitter. It's what's happening*, [Online; Acedido: 23-06-2021]. URL: <https://twitter.com/?lang=en>.
- [35] Unity Technologies, *Unity*, [Online; Acedido: 30-08-2021]. URL: <https://unity.com>.
- [36] *Visual Studio: IDE and Code Editor for Software Developers and Teams*, [Online; Acedido: 30-08-2021]. URL: <https://visualstudio.microsoft.com>.



Respostas dos Participantes nos Testes de Usabilidade

A.1 Caracterização

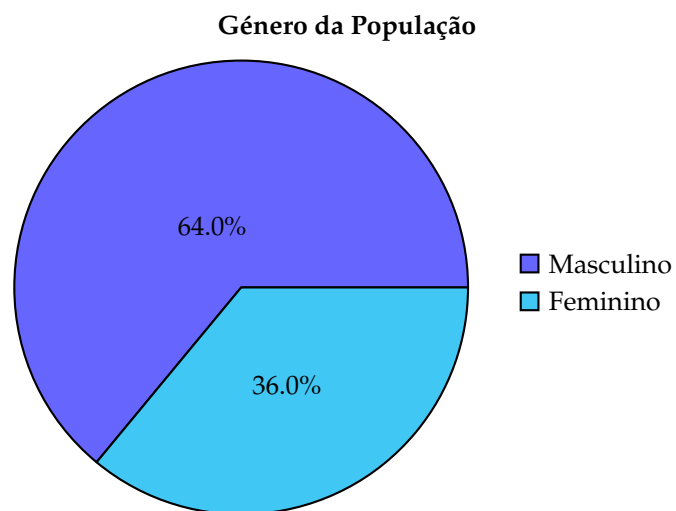


Figura A.1: Análise do género da população do questionário.

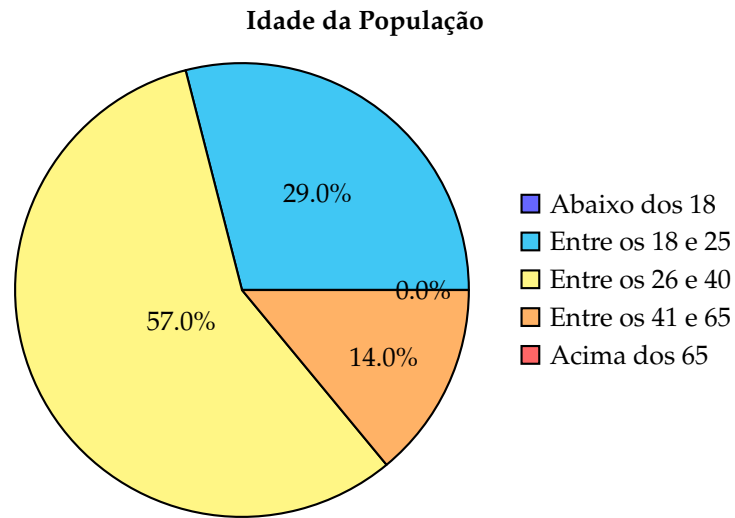


Figura A.2: Análise da idade da população do questionário.

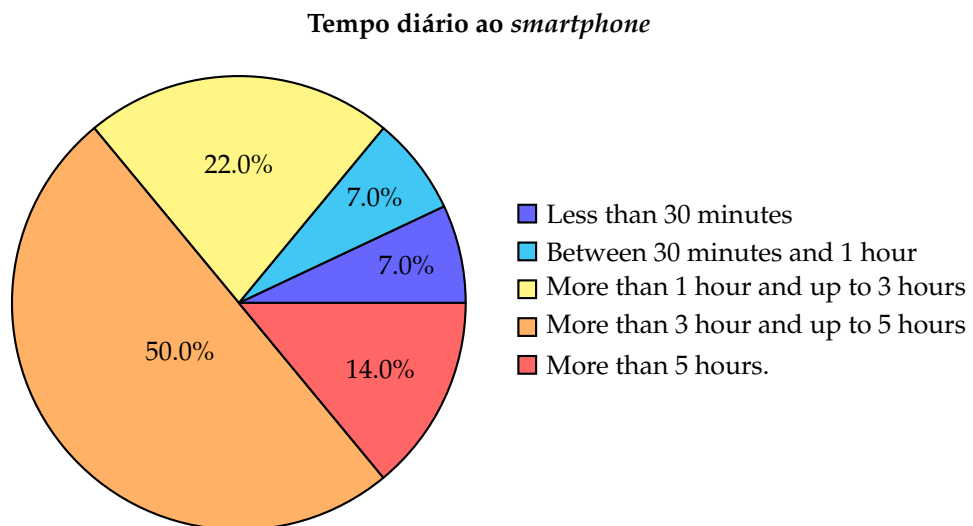


Figura A.3: Análise da quantidade de tempo diário no *smartphone*, da população do questionário.

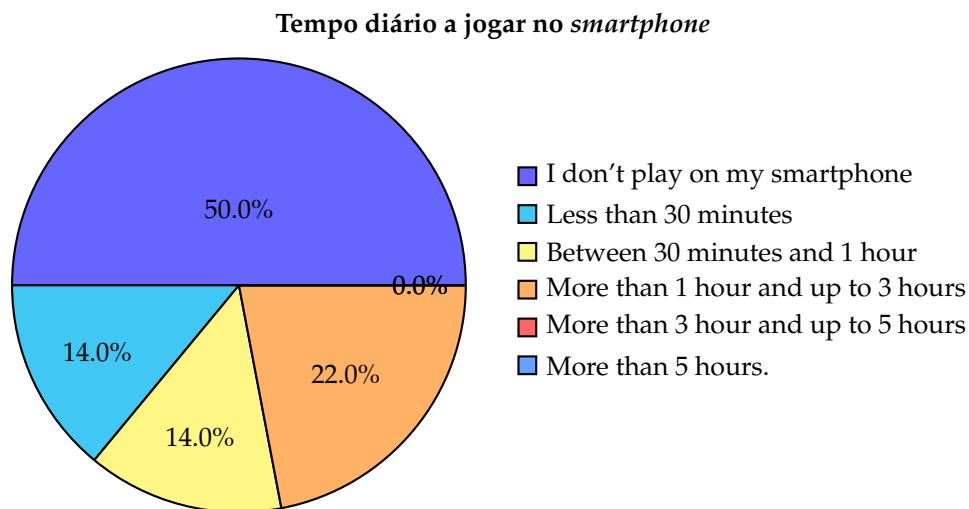


Figura A.4: Análise da quantidade de tempo diário a jogar no *smartphone*, da população do questionário.

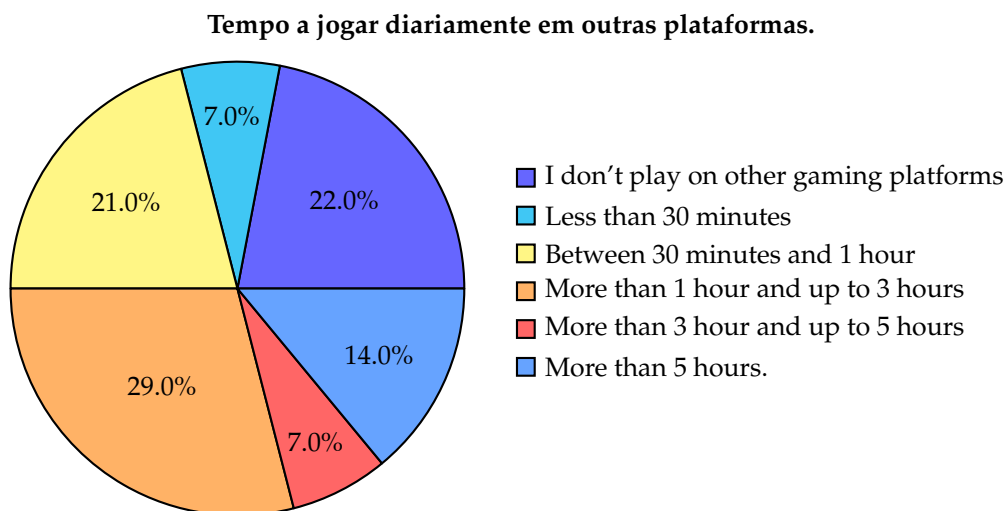


Figura A.5: Análise do tempo diário a jogar em outras plataformas, da população do questionário.

A.2 Tarefas de Usabilidade

Tarefas	Média	Mediana	Moda	Variância	Desvio Padrão
Task 1 - Go to the Squad menu I found the Squad menu without difficulty	4.93	5.00	5.00	0.07	0.26
Task 2 - Change the language of the game to a different one than the current one. I changed the language of the game without difficulty.	5.00	5.00	5.00	0.00	0.00
Task 3 - Go to one of the Abilities Line menus and increase the points value of an ability. I increased the points value of an ability without difficulty.	4.29	5.00	5.00	0.92	0.96
Task 4 - Access the capture mode of a monster. I entered the capture mode without difficulty.	4.86	5.00	5.00	0.27	0.52
Task 5 - Go to the Bestiary menu and change the list of monsters to only those of Defense and with Common rarity. I changed the list of monsters to the specific filters without difficulty.	4.36	4.50	5.00	0.66	0.81
Task 6 - Start a duel. I started a duel without difficulty.	4.86	5.00	5.00	0.27	0.52
Task 7 - Start a Duel. I understood the panel of values shown during the duel without difficulty.	3.93	4.00	5.00	1.21	1.10

Tabela A.1: Avaliação dos resultados referentes à secção Tarefas do questionário.

A.3 SUS

Questões	Média	Mediana	Moda	Variância	Desvio Padrão
I think that I would like to play this game frequently.	3.93	4.5	5	1.49	1.22
I found the game unnecessarily complex.	1.93	1	1	1.78	1.33
I thought the game was easy to play.	4.29	4	4	0.35	0.59
I think that I would need the support of a technical person to be able to play this game.	1.50	1	1	1.11	1.05
I found the various functions in this game were well integrated.	4.21	4	4	0.31	0.56
I thought there was too much inconsistency in this game.	1.64	1	1	1.09	1.04
I would imagine that most people would learn to play this game very quickly.	4.07	4	5	0.92	0.96
I found the system very cumbersome to play.	1.86	1	1	1.12	1.06
I felt very confident playing the game.	3.79	4	4	1.03	1.01
I needed to learn a lot of things before I could get going with this game.	2.14	1.5	1	1.84	1.36

Tabela A.2: Avaliação dos resultados referentes à secção SUS do questionário.

	Utilizadores	Impar	Par	Total	Pontuação
SUS	1	20.00	12.00	80.00	70.89
	2	13.00	11.00	60.00	
	3	11.00	11.00	55.00	
	4	13.00	13.00	65.00	
	5	15.00	14.00	72.50	
	6	13.00	16.00	72.50	
	7	19.00	16.00	87.50	
	8	12.00	11.00	57.50	
	9	11.00	3.00	35.00	
	10	17.00	16.00	82.50	
	11	20.00	16.00	90.00	
	12	16.00	14.00	75.00	
	13	16.00	14.00	75.00	
	14	18.00	16.00	85.00	

Tabela A.3: Análise e pontuação do SUS.



Avaliação da Experiência do Jogador

B.1 Módulo *Core*

Competence					
Questões	I felt skillful	I felt competent	I was good at it	I felt successful	I was fast at reaching the game's targets
Avaliação Singular					
Média	2.71	2.71	2.79	2.57	2.79
Mediana	3.00	3.00	3.00	3.00	3.00
Moda	3.00	3.00	3.00	3.00	3.00
Variância	1.20	0.63	0.74	0.96	0.74
Desvio Padrão	1.10	0.80	0.86	0.98	0.86
Avaliação Total					
Média	2.71				
Mediana	3.00				
Moda	3.00				
Variância	0.86				
Desvio Padrão	0.93				

Tabela B.1: Avaliação do componente Competence do módulo *Core*.

Sensory and Imaginative Immersion						
Questões	I was interested in the game's story	It was aesthetically pleasing	I felt imaginative	I felt that I could explore things	I found it impressive	It felt like a rich experience
Avaliação Singular						
Media	1.79	3.14	2.29	2.64	3.00	2.36
Mediana	2.00	3.00	3.00	3.00	3.00	2.50
Moda	2.00	4.00	3.00	3.00	3.00	3.00
Variância	1.03	0.84	1.20	1.52	0.14	0.80
Desvio Padrão	1.01	0.91	1.10	1.23	0.38	0.89
Avaliação Total						
Media	2.54					
Mediana	3.00					
Moda	3.00					
Variância	1.13					
Desvio Padrão	1.06					

Tabela B.2: Avaliação do componente Sensory do módulo Core.

Flow					
Questões	I was fully occupied with the game	I forgot everything around me	I lost track of time	I was deeply concentrated in the game	I lost connection with the outside world
Avaliação Singular					
Media	2.64	1.36	1.71	2.64	1.50
Mediana	3.00	1.50	2.00	3.00	2.00
Moda	3.00	2.00	2.00	3.00	2.00
Variância	0.80	1.09	1.49	0.94	1.11
Desvio Padrão	0.89	1.04	1.22	0.97	1.05
Avaliação Total					
Media	1.97				
Mediana	2.00				
Moda	2.00				
Variância	1.40				
Desvio Padrão	1.18				

Tabela B.3: Avaliação do componente Flow do módulo Core.

Tension/Annoyance			
Questões	I felt annoyed	I felt irritable	I felt frustrated
Avaliação Singular			
Media	0.79	0.71	1.00
Mediana	0.00	0.00	0.00
Moda	0.00	0.00	0.00
Variância	1.74	1.06	1.71
Desvio Padrão	1.32	1.03	1.31
Avaliação Total			
Media	0.83		
Mediana	0.00		
Moda	0.00		
Variância	1.52		
Desvio Padrão	1.23		

Tabela B.4: Avaliação do componente Tension do módulo Core.

Challenge					
Questões	I thought it was hard	I felt pressured	I felt challenged	I felt time pressure	I had to put a lot of effort into it
Avaliação Singular					
Media	1.29	0.57	2.50	0.43	1.36
Mediana	1.00	0.00	3.00	0.00	1.00
Moda	0.00	0.00	3.00	0.00	0.00
Variância	1.78	0.82	0.82	0.67	1.23
Desvio Padrão	1.33	0.90	0.91	0.82	1.11
Avaliação Total					
Media	1.23				
Mediana	1.00				
Moda	0.00				
Variância	1.60				
Desvio Padrão	1.27				

Tabela B.5: Avaliação do componente Challenge do módulo Core.

Negative Affect				
Questões	It gave me a bad mood	I thought about other things	I found it tiresome	I felt bored
Avaliação Singular				
Media	0.14	1.00	0.71	0.57
Mediana	0.00	0.50	0.00	0.00
Moda	0.00	0.00	0.00	0.00
Variância	0.12	1.29	1.49	0.67
Desvio Padrão	0.35	1.13	1.22	0.82
Avaliação Total				
Media	0.61			
Mediana	0.00			
Moda	0.00			
Variância	0.99			
Desvio Padrão	0.99			

Tabela B.6: Avaliação do componente Negative Affect do módulo Core.

Positive Affect					
Questões	I felt content	I thought it was fun	I felt happy	I felt good	I enjoyed it
Avaliação Singular					
Media	3.21	3.50	3.14	3.07	3.14
Mediana	3.50	3.50	3.00	3.00	3.00
Moda	4.00	4.00	3.00	3.00	3.00
Variância	0.74	0.25	0.55	0.49	0.41
Desvio Padrão	0.86	0.50	0.74	0.70	0.64
Avaliação Total					
Media	3.21				
Mediana	3.00				
Moda	3.00				
Variância	0.51				
Desvio Padrão	0.71				

Tabela B.7: Avaliação do componente Positive Affect do módulo Core.

B.2 Módulo *Post Game*

Positive Experience						
Questões	I felt revived	It felt like a victory	I felt energized	I felt satisfied	I felt powerful	I felt proud
Avaliação Singular						
Media	2.00	2.21	2.07	3.07	1.79	2.00
Mediana	2.50	2.50	2.00	3.00	2.00	2.00
Moda	3.00	3.00	3.00	3.00	3.00	2.00
Variância	2.14	1.31	1.35	0.64	1.60	1.00
Desvio Padrão	1.46	1.15	1.16	0.80	1.26	1.00
Avaliação Total						
Media	2.19					
Mediana	2.00					
Moda	3.00					
Variância	1.51					
Desvio Padrão	1.23					

Tabela B.8: Avaliação do componente Positive Experience do módulo Post Game.

Negative Experience						
Questões	I felt bad	I felt guilty	I found it a waste of time	I felt that I could have done more useful things	I felt regret	I felt ashamed
Avaliação Singular						
Media	0.07	0.00	0.14	1.14	0.07	0.00
Mediana	0.00	0.00	0.00	0.50	0.00	0.00
Moda	0.00	0.00	0.00	0.00	0.00	0.00
Variância	0.07	0.00	0.12	1.69	0.07	0.00
Desvio Padrão	0.26	0.00	0.35	1.30	0.26	0.00
Avaliação Total						
Media	0.24					
Mediana	0.00					
Moda	0.00					
Variância	0.49					
Desvio Padrão	0.70					

Tabela B.9: Avaliação do componente Negative Experience do módulo Post Game.

Tiredness		
Questões	I felt exhausted	I felt weary
Avaliação Singular		
Media	0.29	0.21
Mediana	0.00	0.00
Moda	0.00	0.00
Variância	0.63	0.31
Desvio Padrão	0.80	0.56
Avaliação Total		
Media	0.25	
Mediana	0.00	
Moda	0.00	
Variância	0.47	
Desvio Padrão	0.69	

Tabela B.10: Avaliação do componente Tiredness do módulo Post Game.

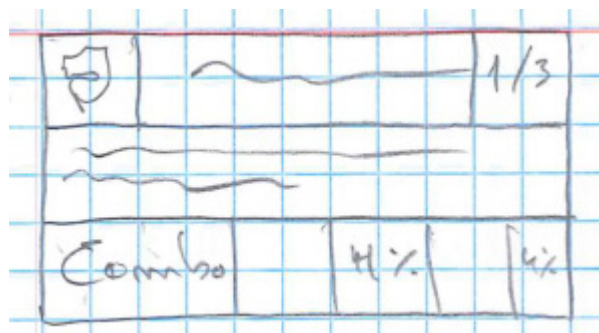
Returning to Reality			
Questões	I found it hard to get back to reality	I felt disoriented	I had a sense that I had returned from a journey
Avaliação Singular			
Media	0.93	0.50	1.86
Mediana	0.50	0.00	2.00
Moda	0.00	0.00	2.00
Variância	1.07	1.54	1.41
Desvio Padrão	1.03	1.24	1.19
Avaliação Total			
Media	1.10		
Mediana	0.00		
Moda	0.00		
Variância	1.66		
Desvio Padrão	1.29		

Tabela B.11: Avaliação do componente Returning to Reality do módulo Post Game.

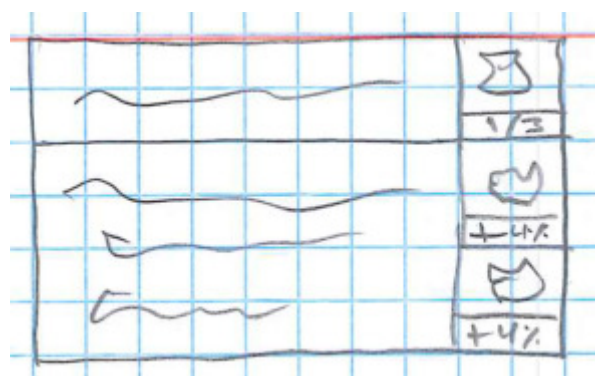


Arquivo de Protótipos

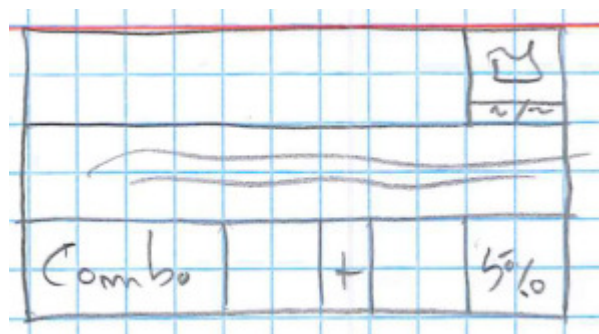
C.1 Protótipos de *layouts* de várias GUIs e HUDs



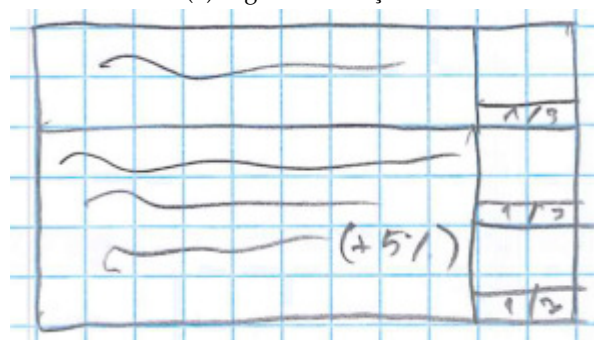
(a) Primeira variação.



(b) Segunda variação.

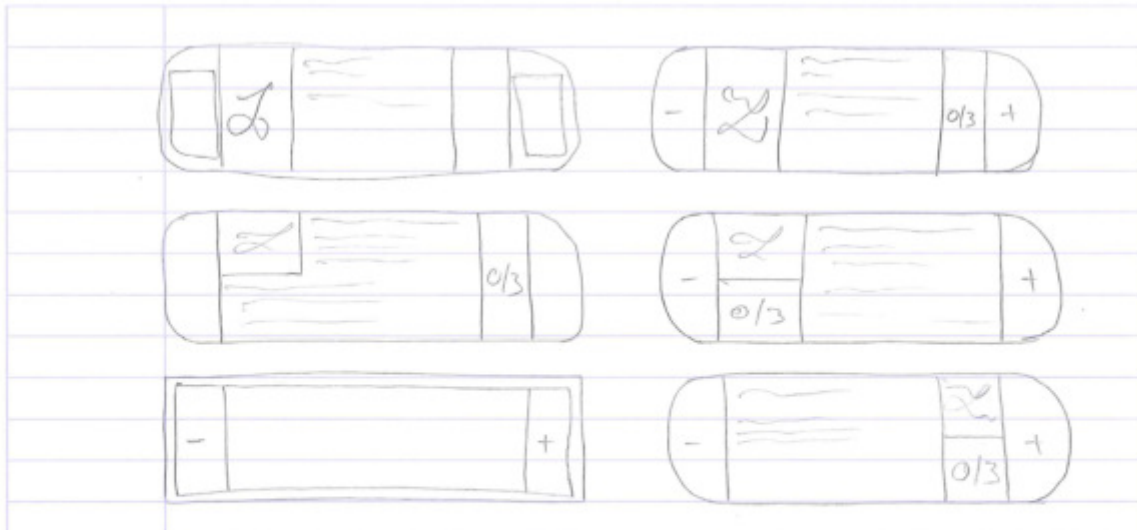


(c) Terceira variação.

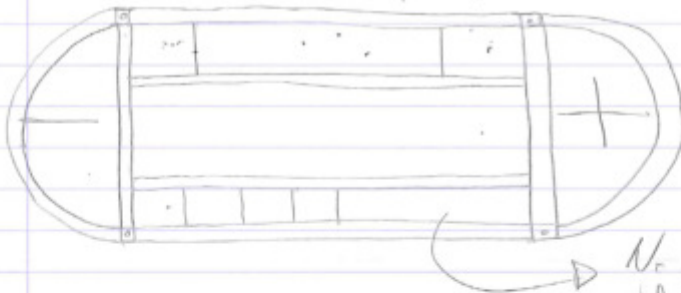


(d) Quarta variação.

Figura C.1: Desenhos dos vários *layouts* de organização do conteúdo do painel de habilidade.



Para tradução do texto poderia - se aplicar referências via Unity de cada placeholder do texto para que este recebesse o conteúdo via programaticamente.



No máximo pode haver dois ícones de 100x100px logo pode - se contabilizar melhor o espaço.

Porquê só dois ícones? Porque só existe mais dois ramos logo não há mais combos. Isto pode fazer o espaço a ser usado nos cantos estão.



Figura C.2: Desenho de protótipos de *layout* do painel de habilidade.

em Ponto
crível: 3



Experiência		1	} 1
Atual	Para evoluir	2	
		3	
Pontos		4	} 2
Por usar	Atual	5	
		6	
Monstros		7	} 3
Apanhados	Vistos	8	
		9	
Batalhas		10	} 4
Fugidas	Perdidas	11	
		12	
Garbas	Total	13	} 4
		14	
Prêmio vitória	*	15	
Pocões		16	} 4
Vida	Intelecto	17	
x	x	18	

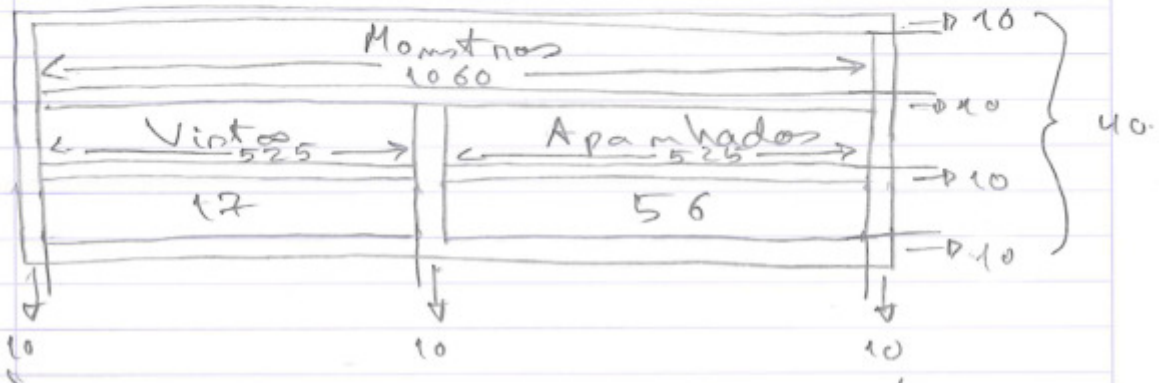


Figura C.3: Desenho de protótipo do menu Jogador.

1			
2	Experiência		1
			2
3			3
4	Pontos		4
5			5
6			6
7	Monstros		7
8			8
9			9
10	Batalhas		10
11			11
12	Fugidos		11
13			12
14	Ganhos		13
15			14
16	Raças		15
17	Pelações		16
18	Vida		17
19			18
↳ 17			

Figura C.4: Desenho de protótipo final do layout do menu Jogador.

Stats					
Type	Attack	Subtype	Camouflage		50 1
Saw		Caught			2
Level		To Evolve			3
Health	Base		Level		4
Attack					5
Defense					6
Subtype					7

Figura C.6: Desenho de protótipo do menu Monstro.

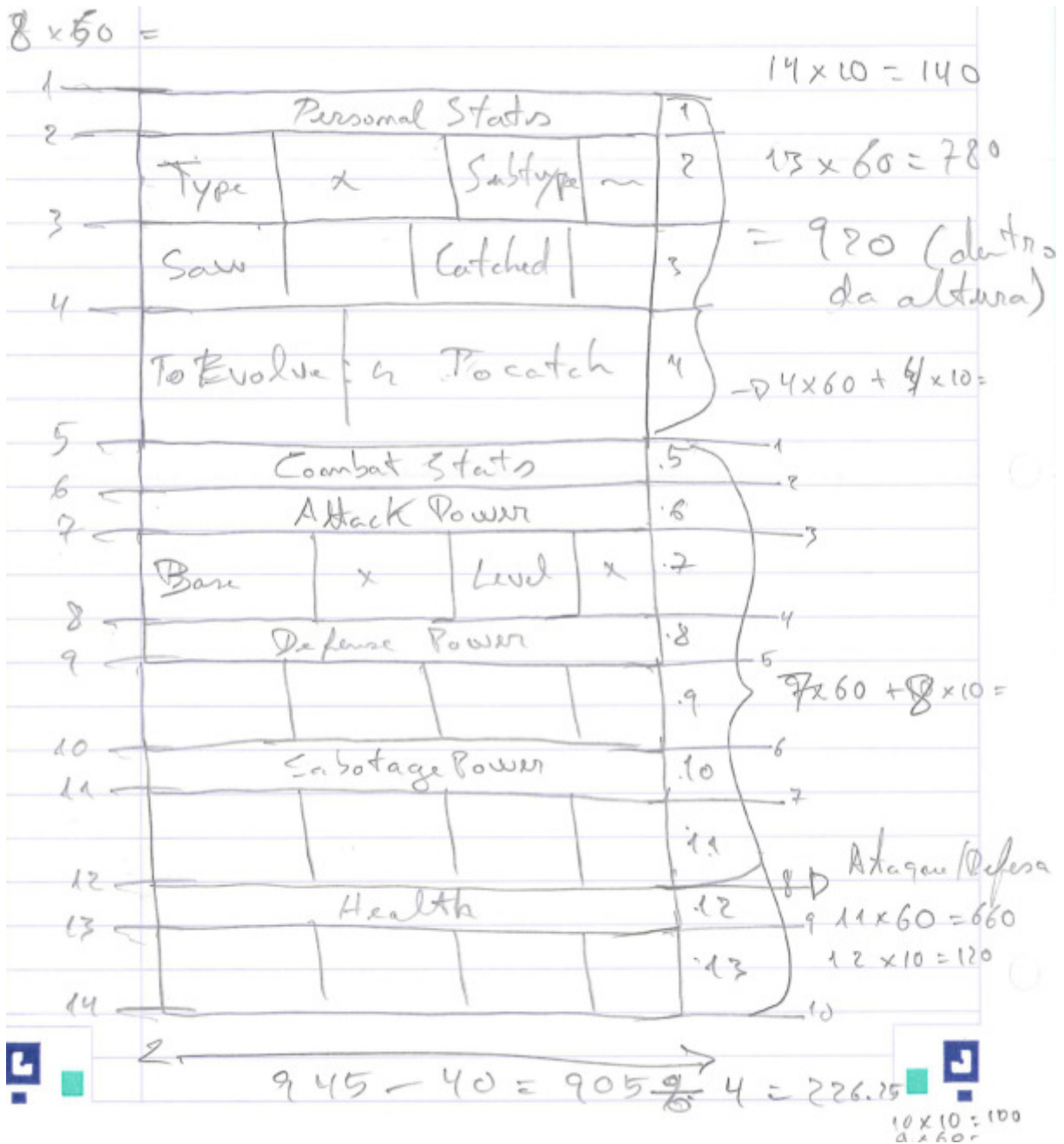


Figura C.7: Desenho de protótipo final do layout do menu Monstro.

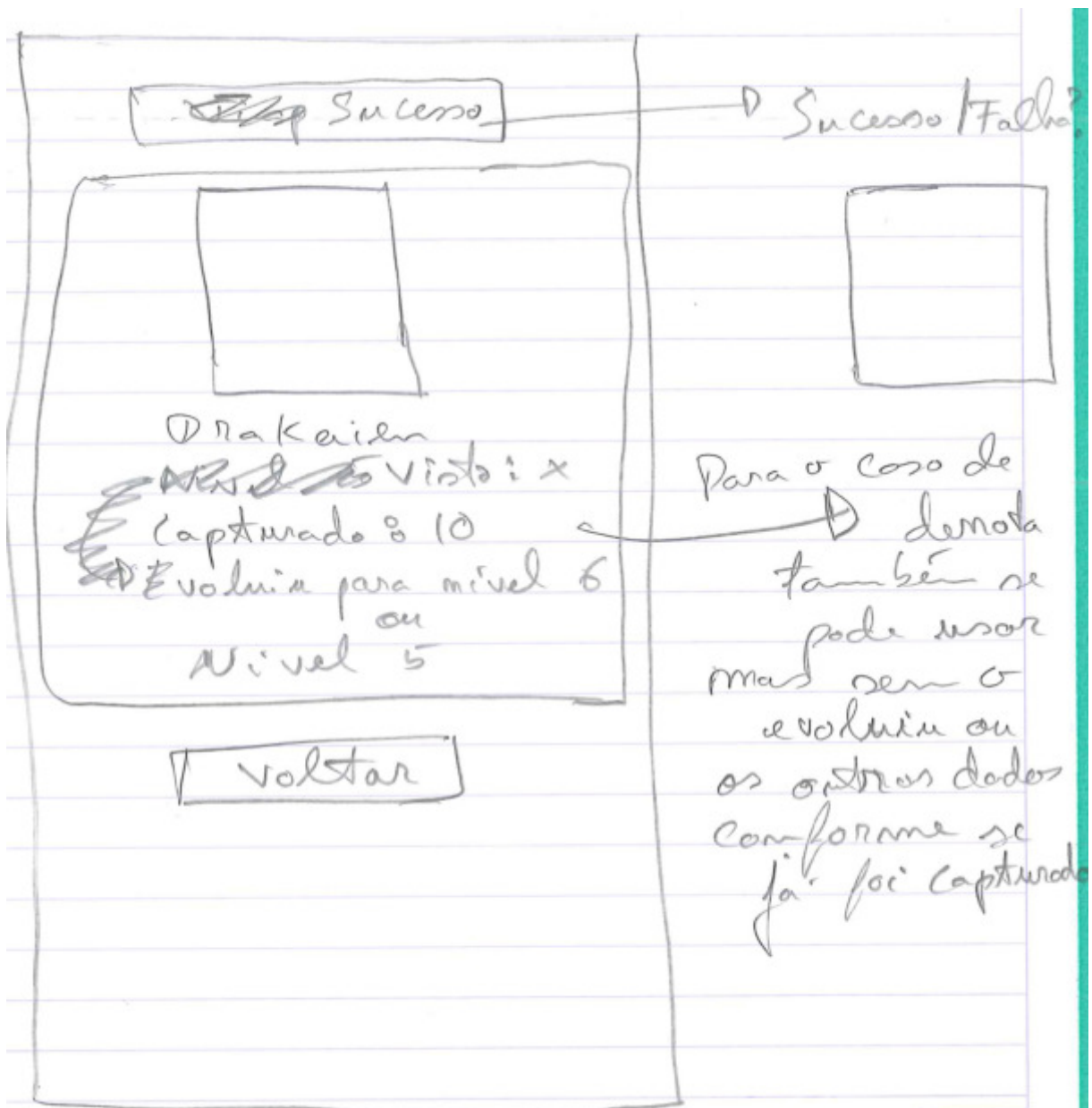


Figura C.8: Desenho de protótipo do menu de resultado do cenário Captura.

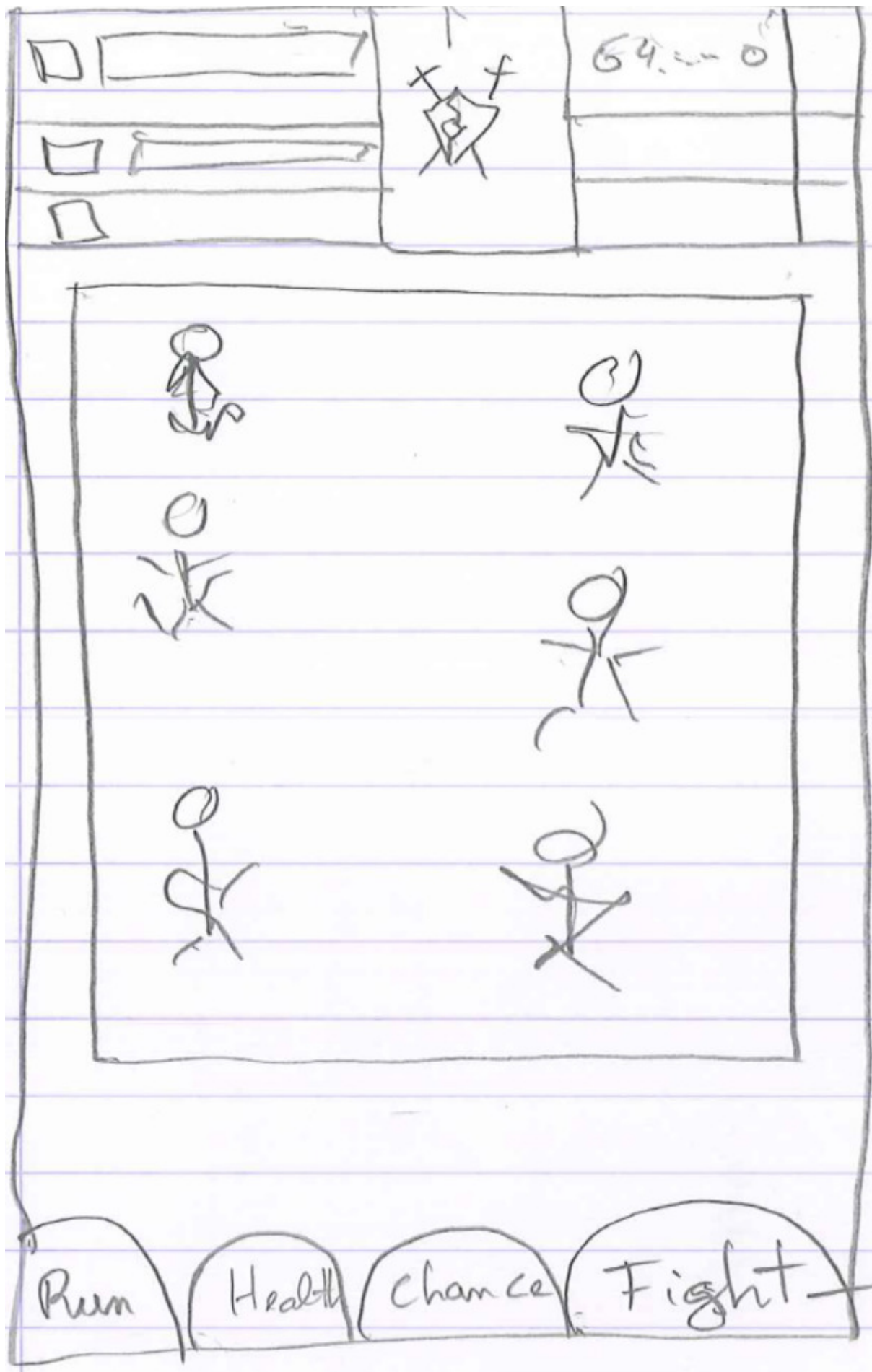


Figura C.9: Desenho de protótipo final de layout para o HUD do cenário Duelo.

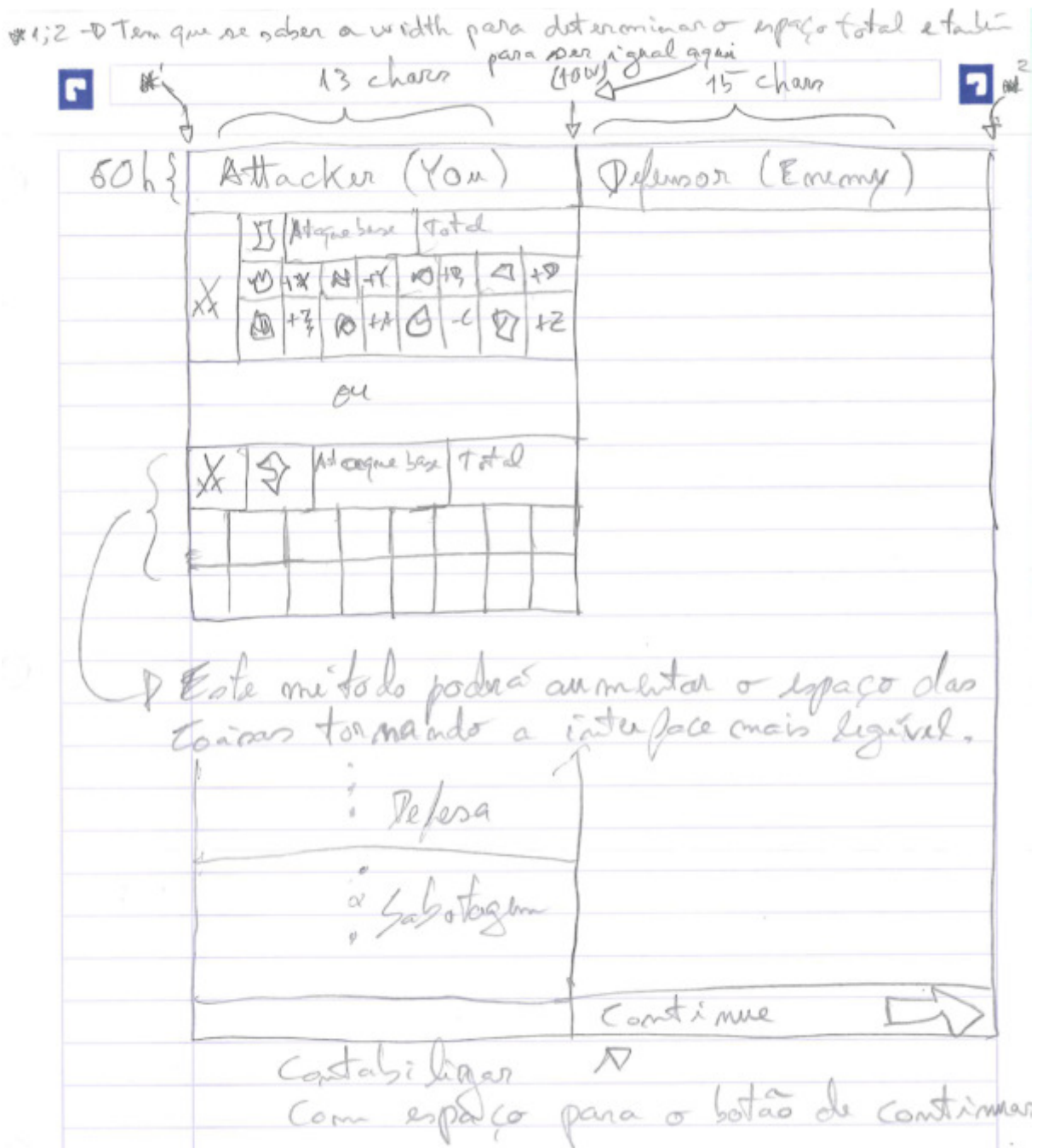


Figura C.10: Desenho de protótipo do menu de ataque e defesa do cenário Duelo.

Vitória	
Experiência	
Ganha	Total
Evolui para nível 3	
Ganhador + 3 pontos	
Total Pontos	43
Dano	
Recebido	
Dado	
Defendido	

Figura C.11: Desenho de protótipo do menu de resultado do cenário Duelo.

