



Instituto Universitário de Lisboa

Escola de Tecnologias e Arquitectura (ISTA)

Departamento de Ciências e Tecnologias de Informação

Gerador Automático de Cenários para Jogos Genéricos

Ana Rita Ruivo dos Santos

Dissertação submetida como requisito parcial para obtenção do grau de
Mestre em Engenharia Informática

Orientador:

Prof. Dr. Pedro Santana, Professor Auxiliar,
ISCTE - Instituto Universitário de Lisboa

Co-Orientador:

Prof. Dr. Pedro Mariano, Investigador Auxiliar Convidado,
Universidade de Lisboa

Outubro, 2018

*“Science is what we understand well enough to explain to a computer.
Art is everything else we do.”*

Donald Knuth

Agradecimentos

A presente dissertação resulta de um trabalho que apenas poderia ser realizado com o apoio, encorajamento e motivação transmitidos por parte muitas pessoas às quais quero expressar o maior agradecimento.

Aos meus orientadores, Prof. Dr. Pedro Santana e Prof. Dr. Pedro Mariano, por me terem guiado pelos melhores caminhos, por todo o conhecimento transmitido, por toda a compreensão e disponibilidade prestadas durante a construção desta dissertação.

Aos meus pais, que apesar da distância sempre colocaram a minha educação como prioridade máxima ao acreditarem nas minhas capacidades, e à minha irmã pela compreensão nos piores momentos.

Ao Xavier, expresso aqui a minha gratidão por todo o amor e dedicação transmitidos durante este caminho percorrido, por toda a paciência e motivação intermináveis, por toda a ajuda e disponibilidade, pela pura amizade e pelas horas sem fim.

A toda a minha família e amigos pela preocupação e carinho que sempre demonstraram. E a todos os que contribuíram, direta ou indiretamente, para a conclusão desta dissertação.

Resumo

Jogos Genéricos (do inglês *General Game Playing*) é uma área da Inteligência Artificial que consiste no desenvolvimento de agentes capazes de jogar qualquer jogo, sem conhecimento prévio das regras. Estes agentes são capazes de jogar jogos bastante distintos, como o Xadrez ou o Quatro em Linha, tendo acesso às regras do jogo somente quando o jogo se inicia. À medida que o jogo decorre os agentes adaptam a sua estratégia de forma a obter o melhor resultado possível. Atualmente, as interfaces gráficas destes jogos são bastante pobres, tipicamente em 2D, com uma paleta de cor reduzida e elementos de jogo simplistas, o que contribui para uma fraca experiência de utilizador. Além disso, estas interfaces não são genéricas, ou seja, cada jogo possui uma interface própria, construída especificamente para o jogo, e como tal não pode ser reutilizada para outro jogo sem as devidas modificações necessárias. Esta dissertação tem o objetivo de enriquecer a área dos jogos genéricos ao criar um gerador automático de cenários para jogos genéricos capaz de oferecer uma interface de jogo elaborada e funcionalidades úteis para o utilizador, em que os cenários são os elementos gráficos que compõem o jogo, como as peças e tabuleiro. O gerador de cenários possui uma estrutura genérica, que interpreta a informação relativa ao jogo disponibilizada pelo motor de jogos genéricos usado, e cria uma interface de jogos genéricos. Com JavaScript, a tecnologia utilizada para criar a interface gráfica dos jogos genéricos, foi possível a criação de ambientes 3D com interações intuitivas, podendo ser usada num computador ou dispositivo móvel. As funcionalidades presentes dão um maior controlo quanto à interface gráfica do jogo, em relação a alternativas existentes, como escolher um estilo de jogo ou funcionalidades que têm em conta necessidades do utilizador (e.g., sistema de cores adaptado à condição de daltonismo). Para analisar a experiência de utilização do gerador de cenários, identificar possíveis melhorias e recolher a opinião quanto ao interesse em contribuir na área do jogos genéricos, realizaram-se testes com 30 utilizadores. Os resultados obtidos indicam que o gerador de cenários oferece uma experiência de utilização e interação intuitivas, e que pode ser expandido e personalizado, adaptando-se a um grande número de jogos, devido às suas características genéricas. Através do gerador de cenários a experiência dos jogos genéricos é melhorada, potenciando assim a atração de mais utilizadores para a comunidade.

Palavras-chave: Jogos Genéricos; Gerador de Cenários; Interface Gráfica; Experiência de Utilizador.

Abstract

General Game Playing is an Artificial Intelligence field of study that consists in the development of agents capable of playing any game without prior knowledge of the rules. These agents are able to play quite distinct games, such as Chess or Four in Line, by having access to the rules of the game only when the game starts. As the game progresses the agents adapt their strategy in order to achieve the best possible outcome. Currently, the graphical interface of these games is rather poor, typically 2D, with a reduced color palette and simplistic game elements, which contributes to a poor user experience. Another limiting aspect of current graphical interfaces for general game playing is that the interface is not generic, that is, each game has its own interface, having been built specifically for that game, and it cannot be reused for another game without the proper modifications. The purpose of this dissertation is to enrich the area of generic games by creating an automatic generator of game scenarios for generic games capable of offering an elaborate game interface and useful user functionalities, in which the scenarios are the graphical elements that make the game, such as the game pieces and the game board. The scenarios generator has a generic structure, that interprets the game information available by a used generic game engine, and creates a generic game interface. With JavaScript, the technology used to create the graphical interface of generic games, it is possible to create 3D environments with intuitive interactions, which can be used on a computer or mobile device. The present features give greater control over the graphical interface of the game, comparing with other existing alternatives, such as choosing a game style or functionalities that respect specific user needs (e.g., color mode suitable to the color blindness condition). In order to analyze the experience of using the scenarios generator, identify possible improvement points and gather the opinion about the interest in contributing to the generic games field, tests were carried out with 30 users. The results indicate that the scenarios generator offers an intuitive user experience and interaction, and it can be increased and customized, adapting to a great number of games, thanks to its generic characteristics. With the scenarios generator, the generic games user experience is improved, suggesting the ability to attract more users to the community.

Keywords: General Game Playing; Game Scenarios Generator; Graphical Interface; User Experience.

Conteúdo

Agradecimentos	v
Resumo	vii
Abstract	ix
Lista de Figuras	xv
Lista de Código Fonte	xvii
Abreviaturas	xix
1 Introdução	1
1.1 Contextualização	1
1.2 Motivação	2
1.3 Questões de Investigação	3
1.4 Objetivos	3
1.5 Método de Investigação	5
1.5.1 Ciclo de Relevância	5
1.5.2 Ciclo de Design	6
1.5.3 Ciclo de Rigor	6
1.6 Estrutura do Documento	7
2 Trabalho Relacionado	9
2.1 Jogos de Tabuleiro	9
2.1.1 Jogos de Tabuleiro Digitais	10
2.1.2 Benefícios dos Jogos de Tabuleiro	11
2.1.3 Jogo com Significado	12
2.2 Inteligência Artificial	13
2.3 Jogos Genéricos	14
2.3.1 Abordagens dos Jogos Genéricos	15
2.3.2 Estrutura dos Jogos Genéricos	15
2.3.3 Descrição dos Jogos Genéricos	17
2.3.4 Histórico dos Jogos Genéricos	22
2.3.5 Agentes de Jogos Genéricos	26
2.4 Design de Interface	26
2.5 Sistemas de Jogos Genéricos	28
2.5.1 Agentes de Jogos Genéricos	28

2.5.2	Servidores de Jogos Genéricos	28
2.5.3	Competições de Jogos Genéricos	31
2.6	Modelação de cenários para Jogos Genéricos.....	31
2.6.1	JavaScript.....	32
2.6.2	Three.js.....	32
2.7	Daltonismo.....	33
3	Desenho do Gerador de Cenários	35
3.1	Metodologia.....	35
3.2	Elementos de Design.....	36
3.3	Layout.....	36
3.4	Interações	37
3.5	Funcionalidades.....	38
3.5.1	Carregamento e Criação da Interface Gráfica.....	39
3.5.2	Modo de Cor.....	39
3.5.3	Estilo de Jogo.....	40
4	Implementação do Gerador de Cenários	41
4.1	Ferramentas de Desenvolvimento.....	41
4.2	Estrutura do Gerador de Cenários.....	42
4.3	Interpretação	42
4.4	Interface Gráfica	43
4.5	Modelação de Elementos	44
4.6	Gerador de Cenários.....	45
4.6.1	Modo de Cor.....	49
4.6.2	Estilo de Jogo	52
4.6.3	Restrições.....	54
4.7	Adaptação do Gerador ao Dispositivo.....	55
5	Testes e Análise de Resultados	57
5.1	Testes de Utilizador	57

5.2	Perfil dos Participantes.....	58
5.3	Questionário.....	59
5.3.1	Teste de desempenho	60
5.3.2	Teste de satisfação	63
5.3.3	Interesse em contribuir para a área dos Jogos Genéricos	64
5.4	Análise dos Resultados	64
6	Conclusões	77
6.1	Conclusões Gerais.....	77
6.2	Contribuições.....	78
6.3	Trabalho Futuro.....	79
7	Bibliografia	81

Lista de Figuras

1.1	Design Science Research Cycles, segundo Hevner (2007).....	5
2.1	Jogo <i>The Royal Game of Ur</i> (Sassoon, 1993)	9
2.2	Jogo <i>Settlers of Catan</i> (Branca e Johansson, 2017)	10
2.3	Jogo <i>Minesweeper</i> (Pirillo, 2005).....	11
2.4	Jogo de Xadrez entre Garry Kasparov e Deep Blue (Nadel, 1997).....	13
2.5	Exemplo de um conjunto de estados de um jogo com jogador único (Genesereth e Thielscher, 2014)	16
2.6	Jogo do <i>Tic-tac-toe</i> no Gamemaster da Universidade de Stanford (Gamemaster) .	29
2.7	Jogo <i>Breakthrough</i> no GGP Server da Universidade de Dresden (Dresden GGP Server)	29
2.8	Jogo <i>Two-player free-for-all</i> no Tiltyard do portal GGP.org (Tiltyard)	30
2.9	“The Aviator”, uma animação criada com Three.js de Karim Maaloul (2016)	33
2.10	Edição UNO ColorADD (The Verge, 2017).....	33
3.1	Layout simplificado do gerador de cenários	37
3.2	Gestos: Toque, Swipe e Pinch (da esquerda para a direita)	38
3.3	Paleta de 7 cores para daltonismo de Bang Wong (2011).....	39
4.1	Estrutura do gerador de cenários	42
4.2	Estrutura da composição da imagem no Three.js.....	44
4.3	Modelo 3D no Blender	45
4.4	Estrutura de pastas do gerador de cenários.....	46
4.5	Jogo <i>Hunter</i> no Gerador de Cenários para Jogos Genéricos.....	46
4.6	Logotipo do gerador de cenários	47
4.7	Diferentes jogos genéricos no gerador de cenários com os respectivos nomes	49
4.8	Paletas de cor do gerador de cenários.....	50
4.9	Jogo <i>Breakthrough</i> no modo de cor por omissão ou não daltônico	50
4.10	Jogo <i>Breakthrough</i> no modo de cor Protanopia (à esquerda) e modo de cor Protanopia visto por um utilizador com este tipo de daltonismo (à direita).....	51
4.11	Jogo <i>Breakthrough</i> no modo de cor Deuteranopia (à esquerda) e modo de cor Deuteranopia visto por um utilizador com este tipo de daltonismo (à direita).....	51
4.12	Jogo <i>Breakthrough</i> no modo de cor Tritanopia (à esquerda) e modo de cor Tritanopia visto por um utilizador com este tipo de daltonismo (à direita)	52
4.13	Jogo <i>Breakthrough</i> no modo de cor Monocromacia (à esquerda) e modo de cor Monocromacia visto por um utilizador com este tipo de daltonismo (à direita).....	52
4.14	Jogo <i>Skirmish</i> no estilo Clássico	53

4.15	Diferentes jogos genéricos, no gerador de cenários, com respectivos nomes e estilos de jogo	54
4.16	Comportamento responsivo do gerador em cada dispositivo.....	56
5.1	Resultados da questão 1.....	65
5.2	Resultados da questão 2.....	66
5.3	Resultados da questão 3.....	66
5.4	Resultados da questão 4.....	67
5.5	Resultados da questão 5.....	68
5.6	Resultados da questão 6.....	69
5.7	Resultados da questão 7.....	69
5.8	Resultados da questão 8.....	70
5.9	Resultados da questão 9.....	71
5.10	Resultados da questão 10.....	72
5.11	Resultado da pontuação relativa às afirmações da questão 11	74

Lista de Código Fonte

2.1	Definição de dois jogadores, GDL (Genesereth e Thielscher, 2014).....	18
2.2	Estado inicial de todas as células, GDL (Genesereth e Thielscher, 2014)	19
2.3	Definição dos movimentos legais, GDL (Genesereth e Thielscher, 2014)	19
2.4	Definição da atualização do estado do jogo para o estado, GDL (Genesereth e Thielscher, 2014)	20
2.5	Definição da pontuação do jogo, GDL (Genesereth e Thielscher, 2014).....	21
2.6	Definição dos conceitos de suporte, GDL (Genesereth e Thielscher, 2014).....	21
2.7	Definição da condições para o final do jogo, GDL (Genesereth e Thielscher, 2014).....	22
2.8	Exemplo do início de um jogo <i>Tic-tac-toe</i> , XML	23
2.9	Exemplo do estado do tabuleiro em cada jogada de um jogo <i>Tic-tac-toe</i> , XML	25
2.10	Exemplo de todas as ações realizadas por cada jogador de um jogo <i>Tic-tac-toe</i> , XML.....	26
4.1	Restrições do jogo <i>Alquerque</i> , JavaScript	55

Abreviaturas

AAAI	Association for the A dvancement of A rtificial I ntelligence
API	A pplication P rogramming I nterface
CSS	C ascading S tyle S heets
GDL	G ame D escription L anguage
GGP	G eneral G ame P laying
HTML	H yper T ext M arkup L anguage
IA	I nteligência A rtificial
IBM	I nternational B usiness M achines
JSON	J ava S cript O bject N otation
LISP	L I S t P rocessor
MCTS	M onte C arlo T ree S earch
XML	e X t ensible M arkup L anguage

Capítulo 1

Introdução

1.1 Contextualização

Os jogos de tabuleiro tiveram origem no Egito há 5000 anos atrás representando tradicionalmente um confronto entre dois exércitos e evoluíram ao longo do tempo tornando-se mais complexos e desafiantes (Piccione, 1980). Geralmente, os jogos de tabuleiro, são compostos por uma superfície com um conjunto de desenhos que ilustram as casas por onde peças se movem de acordo com as regras do específico jogo. Estes jogos podem basear-se em sorte, estratégia ou uma mistura de ambos, tendo sempre um objetivo final comum, vencer o oponente.

Com as novas tecnologias, os jogos de tabuleiro acompanharam o seu aparecimento e evolução, tendo sido adaptados aos novos meios digitais, exemplo disso é a presença do jogo Solitaire no sistema operativo Windows desde a sua versão 3.0 (Hunt, 2015). Os jogos digitais acompanham de tal forma esta evolução que podem ser usados para medir a capacidade gráfica de um computador, ao correr um jogo com determinadas definições (Wilde, 2015).

A internet veio possibilitar jogar contra adversários que não se encontram na mesma sala, sendo atualmente possível jogar num smartphone um jogo de xadrez com um adversário que se encontre no outro lado do mundo (Adams, 2013). Jogar contra o computador é possível graças à inteligência artificial, pois uma das suas aplicações é a criação de agentes capazes de jogar contra adversários, humano ou computador. Um dos agentes mais famosos é o Deep Blue, que em 1997 derrotou Garry Kasparov no Xadrez (IBM, 1997).

Jogos Genéricos (em inglês *General Game Playing* - GGP) têm como base jogos de tabuleiro jogados por agentes de inteligência artificial que conseguem jogar vários jogos sem algum conhecimento prévio programado relativamente à lógica do jogo (Sharma et al., 2008).

O Deep Blue não se enquadra nesta categoria, pois é um algoritmo específico para jogar Xadrez, não conseguindo jogar outros jogos, como Breakthrough.

“Os Jogos Genéricos fornecem uma *framework* teórica para modelação de sistemas discretos dinâmicos que pode definir um método racional de resolução de problemas como a falta de informação e recursos. Esta *framework* tem utilização prática de lógica em áreas em que as suas funcionalidades são importantes, como Economia e Direito, levantando questões sobre a natureza da inteligência e servindo de laboratório onde são avaliadas abordagens concorrentes de inteligência artificial” (Genesereth, 2013).

1.2 Motivação

A interface gráfica de um jogo genérico é a representação gráfica de todos os elementos que constituem um jogo (tabuleiro, peças e movimentos das peças) em cada jogada. Existem ferramentas que geram a interface gráfica de um jogo genérico, e que permitem que estes sejam jogados, por humanos ou agentes de inteligência artificial, como o Gamemaster¹. No entanto, estas ferramentas oferecem uma experiência de utilizador pobre e bastante simples, com pouca interatividade, o que advém do facto da comunidade que desenvolve esta área focar-se principalmente no desempenho dos algoritmos de jogo.

O ambiente gráfico dos Jogos Genéricos é tipicamente composto por elementos que permitem somente identificar o jogo em questão e a interatividade é limitada revelando problemas de usabilidade, necessitando de uma intervenção com a criação de uma experiência de jogo mais rica e cativante. O gerador automático de cenários para jogos genéricos proposto nesta dissertação, daqui em diante identificado por *gerador de cenários*, visa melhorar substancialmente os elementos e experiência de jogo com a criação de ambientes gráficos 3D e mecanismos de iluminação em que o utilizador pode interagir de forma intuitiva. Outro aspeto é o facto de se adaptar a características do utilizador, nesta dissertação focou-se no caso de utilizadores com a condição de daltonismo, mas outras limitações podiam ter-se tido em conta. Uma apresentação gráfica, tendo em conta os elementos de jogo e interatividade, coloca este tipo de jogos no patamar atual da evolução gráfica dos videojogos, pois a qualidade gráfica dos jogos tem um papel preponderante na escolha de grande parte dos jogadores (Usher, 2014).

¹ O Gamemaster é um servidor disponibilizado pela Universidade de Stanford. Disponível em <http://ggp.stanford.edu/gamemaster/>

O gerador de cenários pode ser utilizado por programadores de agentes de inteligência artificial, para testarem as suas criações num sistema mais avançado graficamente, e pelo jogador comum interessado na área dos jogos de tabuleiro. Deste modo os jogos genéricos poderão aumentar a sua comunidade e reputação, ao conseguir atrair novos programadores que podem testar os seus próprios agentes de jogo, praticando assim as suas capacidades de programação em inteligência artificial, e também atrair o jogador comum, pois uma vantagem dos jogos de tabuleiro, além de exercitar-se o cérebro, é desenvolver habilidades estratégicas (e.g., resolução de problemas e gestão de recursos) e conhecer novas pessoas.

1.3 Questões de Investigação

O gerador de cenários foi desenvolvido com o objetivo de responder às seguintes questões de investigação:

1. Pode o Gerador de Cenários para Jogos Genéricos ajudar utilizadores a compreenderem e identificarem jogos ao navegar entre as jogadas dos mesmos?
2. Que representações gráficas genéricas (cores e formas) e perspectivas de visualização 3D contribuem para uma experiência de utilizador mais satisfatória?
3. Até que ponto alguém, incluído na área das tecnologias, teria interesse em desenvolver novos jogos de tabuleiro?
4. Será este gerador de cenários uma mais valia em comparação com aplicações semelhantes?

1.4 Objetivos

O projeto desenvolvido nesta dissertação enquadra-se, num âmbito mais alargado, na área de desenvolvimento de algoritmos de inteligência artificial para jogos genéricos, e tem como objetivo dar destaque à componente gráfica dos jogos genéricos, que tem sido posta de parte e sem grande trabalho relevante desenvolvido. Ou seja, o objetivo é melhorar a interface gráfica, a usabilidade e experiência dos jogos genéricos, atraindo também novos utilizadores para o desenvolvimento de cenários de jogo, bem como os agentes que nele participam, pois

tal como na constante evolução dos jogos de vídeo tradicionais, a jogabilidade e gráficos visuais ditam, em grande parte, o sucesso de um jogo.

O gerador de cenários para jogos genéricos desenvolvido no âmbito da dissertação poderá ser utilizado por programadores de agentes de inteligência artificial e por utilizadores em geral. Os programadores de agentes de inteligência artificial poderão utilizar o gerador de cenários para analisarem as suas criações num sistema mais avançado graficamente, e os utilizadores em geral, para terem uma experiência diferente ao interagir com a interface gráfica de jogos genéricos, podendo navegar pelas jogadas de cada jogo, aprender as regras dos diferentes jogos e até criar novos jogos de tabuleiro e navegar entre as suas jogadas de forma alternativa. O gerador de cenários também pode ser utilizado por utilizadores com a condição de daltonismo pois a interface gráfica pode adaptar-se às suas necessidades.

O gerador de cenários desenvolvido é alimentado por um ficheiro XML, numa linguagem gráfica genérica desenvolvida no âmbito desta dissertação, e numa biblioteca Three.js para criar a interface gráfica de jogos genéricos. O ficheiro XML de cada jogo é gerado por ferramentas como o Gamemaster e contém o histórico das jogadas realizadas por agentes de inteligência artificial e/ou humanos do início ao fim desse jogo, como abordado na Secção 2.3.4. Cada termo do histórico do jogo (contido no ficheiro XML) está associado a um conjunto de funções e objetos da linguagem gráfica genérica, logo, possuindo históricos de jogos diferentes, o gerador de cenários consegue interpretar a informação. Isto é possível pois a organização dos termos do histórico do jogo é consistente, o que resulta na habilidade do gerador em interpretar bastantes jogos. Este facto facilita a criação de novos jogos, usando a organização e designação típicas dos termos dos históricos dos jogos. O Three.js é uma biblioteca/API JavaScript que permite gerar ambientes 3D em *browsers* web, e foi escolhida para construir a interface gráfica de jogos genéricos do gerador de cenários. O Three.js tem uma forte presença na web, suporte e comunidade vastos, o que também são fatores relevantes.

Através dos dados do histórico de um jogo em conjunto com a linguagem gráfica genérica desenvolvida e o Three.js, é gerada uma interface gráfica do jogo com todos os elementos que constituem esse jogo (tabuleiro, peças e movimentos das peças). A interface gráfica do jogo oferece uma experiência rica que recorre a um ambiente 3D e mecanismos de iluminação e processamento gráfico provenientes do Three.js. A interface gráfica é gerada no *browser* (i.e., está preparada para ser apresentada num *browser* da internet) dando assim

maior polivalência ao gerador de cenários, podendo ser mais acessível aos utilizadores. Deste modo é possível investigar o impacto desta representação na experiência do jogo em si.

Após o desenvolvimento do Gerador de Cenários para Jogos Genéricos realizou-se uma fase de testes. Os testes de utilizador analisaram, através de inquéritos, a facilidade em identificar os jogos de tabuleiro, o nível de usabilidade, intuição e interação da aplicação, o impacto de diferentes paletas de cores e formas dos elementos de jogo, o impacto das diferentes perspectivas de visualização 3D e a satisfação do utilizador.

1.5 Método de Investigação

Esta dissertação representa um projeto de investigação alargado, pretendendo ser reconhecido como relevante na sua área e podendo servir de base para outros projetos, contribuindo para o enriquecimento da temática dos Jogos Genéricos. Para atingir o objetivo proposto, procurou-se um método eficiente e robusto, tendo-se optado pelo *Design Science Research*. Este método procura melhorar iterativamente através do desenvolvimento de novos artefatos e processos de construção para gerar os mesmos. O *Design Science Research* representa-se como a junção de três ciclos de atividades ilustrados na Figura 1.1 (Hevner, 2007).

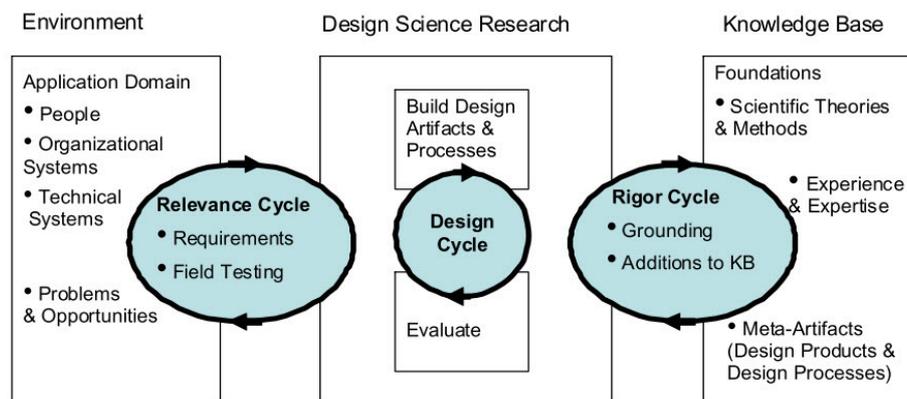


FIGURA 1.1: Design Science Research Cycles, segundo Hevner (2007).

1.5.1 Ciclo de Relevância

O *Relevance Cycle* inicia a investigação num contexto que fornece os requisitos para a pesquisa, identifica possíveis abordagens ao problema e define critérios de aceitação para a avaliação de possíveis soluções. No âmbito do projeto enquadram-se:

- Estudo das melhores tecnologias a utilizar;
- Desenvolvimento da linguagem genérica;
- Estudo e design da arquitetura do gerador de cenários;
- Direção gráfica dos cenários.

1.5.2 Ciclo de Design

O *Design Cycle* é o núcleo de um projeto de *Design Science Research*, que atua entre duas atividades, a construção de artefatos e a sua consequente avaliação. Neste ciclo são descritos as possibilidades identificadas ao longo do projeto e a sua avaliação, tendo em conta os requisitos e objetivos estabelecidos, até que se atinjam resultados satisfatórios. Este ciclo engloba:

- Abordagens possíveis para a concepção do gerador de cenários, que inclui a relações entre a informação necessária para gerar os cenários e os objetos dos cenários;
- Abordagens a nível gráfico que irão ser desenvolvidas para representar os Cenários dos Jogos Genéricos.

1.5.3 Ciclo de Rigor

O *Rigor Cycle* tem em conta o conhecimento anterior ao projeto, de forma a garantir a sua componente de inovação, servindo também de referência e garantia que os resultados obtidos são de facto contribuições de investigação, não se tratando de trabalho e processos já anteriormente desenvolvidos. Este ciclo resulta de conhecimento construído a partir de teorias e métodos científicos que fornecem fundamentos para uma investigação rigorosa. As componentes da sua base são:

- Experiências e conhecimento a partir dos quais se desenvolve a Contextualização Teórica do projeto em causa;
- Artefatos e processos identificados no domínio da investigação, que podem servir de orientação para o problema proposto.

1.6 Estrutura do Documento

A dissertação é organizada em sete capítulos. Este, o primeiro capítulo, apresenta a introdução global da temática e a sua contextualização identificando o problema, a sua motivação, objetivos e investigação a realizar. O segundo capítulo engloba o Estado da Arte, traçando uma narrativa que apresenta as várias temáticas intervenientes no projeto, onde se destacam os Jogos de Tabuleiro, a Inteligência Artificial, os Jogos Genéricos e os Projetos de Referência.

No terceiro capítulo é apresentado a metodologia, layout, interações e funcionalidades a desenvolver. O quarto capítulo contém todo o processo de desenvolvimento do gerador de cenários, incluindo as ferramentas utilizadas, estrutura e design. No quinto capítulo são apresentados os testes de utilizador realizados, e a consequente análise de resultados. O sexto capítulo apresenta as conclusões gerais do trabalho realizado, as suas contribuições e trabalho futuro a realizar.

Capítulo 2

Trabalho Relacionado

2.1 Jogos de Tabuleiro

A criação dos jogos de tabuleiro remonta a tempos pré-históricos da civilização humana, 5000 A.C., ainda antes de se estabelecer a primeira linguagem escrita (Woods, 2010). O jogo Dados é o primeiro de que há registo e jogava-se com uma forma primitiva de um dado que poderia ser esculpido em pedra, osso ou até madeira (Attia, 2016).

No Antigo Egito os jogos de tabuleiro marcam uma forte presença e são conhecidos como um passatempo da realeza, como é o caso do jogo Senet presente num hieróglifo de 3100 A.C. Senet, que significa “jogo da passagem”, é um dos primeiros jogos que incorpora elementos de sorte e estratégia, e possuía uma forte ligação espiritual às crenças da época, tendo alguns exemplares do jogo sido encontrados em túmulos. Isto deve-se ao facto de o jogo ser considerado um talismã que seria útil para a vida após a morte (Piccione, 1980).

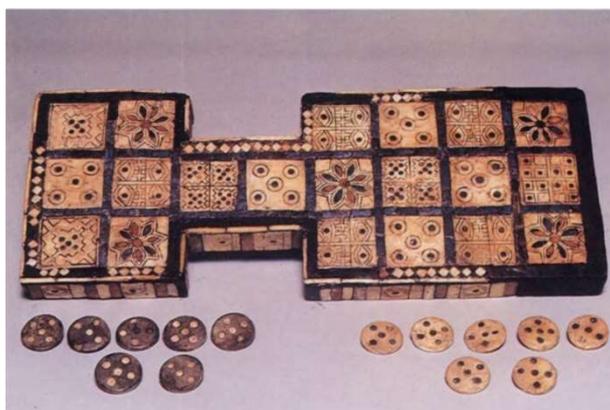


FIGURA 2.1: Jogo *The Royal Game of Ur* (Sassoon, 1993).

O Jogo Real de Ur, ilustrado na Figura 2.1, foi criado em 2600 A.C. na Mesopotâmia, e é o jogo com maior longevidade da civilização, tendo as suas regras evoluído ao longo do

tempo, sendo ainda jogado no dias de hoje (Edwards, 2017). Considera-se que juntamente com o jogo Senet, foram os precursores para outros jogos importantes, como o Gamão (Udo Becker, 2003). Estes jogos enquadram-se na categoria de Jogos de Corrida, em que o objetivo do jogador é mover todas as suas peças para uma determinada posição e os seus movimentos são definidos por um dado (Parlett, 1999).

Atualmente os jogos de tabuleiro estão bastante presentes na sociedade sendo usados em ambiente de convívio ou competição. Desde 1990 que o seu mercado tem vindo a aumentar e, mais recentemente, considera-se mesmo que estamos na era dourada dos jogos de tabuleiro (Greenwood, 2006). Isto deve-se à evolução de jogos já existentes bem como o aparecimento de novos jogos como *Settlers of Catan*, ilustrado na Figura 2.2, e *Carcassonne*, que possuem uma forte componente estratégica e mecânicas de jogo nunca antes vistas (Freeman, 2012). O mercado revela tendência a crescer, prova disso é a criação de novos jogos ser uma das categorias mais financiadas no Kickstarter (Roeder, 2015).



FIGURA 2.2: Jogo *Settlers of Catan* (Branca e Johansson, 2017).

2.1.1 Jogos de Tabuleiro Digitais

O aparecimento do computador pessoal possibilitou que alguns jogos de tabuleiro fossem recriados digitalmente, como o Solitário, Copas, Freecell e Minesweeper (ilustrado na Figura 2.3), presentes no Windows 3. Estes jogos foram e continuam a ser um sucesso, sendo incluídos nas novas versões de Windows, tipicamente com novas melhorias visuais. Estes jogos além de serem uma ótima forma de passar o tempo, inicialmente a sua inclusão foi estratégica tendo o objetivo de ensinar o utilizador um conjunto de interações que é usado em

outros programas do computador, como o movimento do rato, os cliques esquerdo e direito e o *drag and drop*. Deste modo o utilizador aprendia a interagir melhor com o computador enquanto jogava (Hunt, 2015). Pode-se considerar que o mesmo é válido para os novos utilizadores de dispositivos móveis como tablet e smartphone.



FIGURA 2.3: Jogo *Minesweeper* (Pirillo, 2005).

A opinião relativamente ao jogos de tabuleiro versus a sua versão digital pode não ser unânime pois ambos possuem vantagens e desvantagens (Kain, 2012). Os jogos de tabuleiro digitais podem ser jogados contra o computador ou outros jogadores que podem estar na mesma sala ou no outro lado do globo, graças à internet. Tipicamente o computador controla todos os elementos do jogo, como o dado e as peças, certificando que o jogo ocorre sem batota (Walker, 2014). No entanto, dado o estado atual da sociedade, onde a comunicação é bastante menos presente fisicamente, os jogos de tabuleiro são vistos como uma forma de juntar as pessoas.

2.1.2 Benefícios dos Jogos de Tabuleiro

As crianças desde cedo praticam jogos de tabuleiro típicos na sua faixa etária, como o jogo Cobras e Escadas (Siegler e Ramani, 2009). Jogar com a família ou amigos desperta a sua competitividade e vontade de melhorar habilidades como: o reconhecimento de números e letras bem como a sua contagem e leitura; a percepção visual e reconhecimento de cor; e a coordenação e destreza manual (Parent & Child Magazine, 2017). O Monopoly é um jogo com mais de uma centena de anos e continua atual, transmitindo importantes lições de

economia, como guardar dinheiro de parte, ser paciente e que por vezes o mais dispendioso não é o mais valioso (Gad, 2017).

Para os adultos e idosos a prática destes jogos reduz o risco de doenças como Alzheimer e demência, e a nível geral reduz o stress e relaxa, dada a qualidade do tempo e convívio de que usufruem. Em família este convívio é útil para fortalecer laços, estando comprovado que as crianças que o fazem conseguem ter melhores notas e menor probabilidade de demonstrar comportamentos de risco, como ingerir bebidas alcoólicas ou consumir drogas (Schmid, 2009).

2.1.3 Jogo com Significado

Existem jogos de tabuleiro que perduram ao longo do tempo, sendo jogados há várias gerações. Para que isto aconteça quais são os fatores necessários para poder tornar um jogo num fenómeno intemporal?

Um exemplo é o Jogo do Galo, que remonta ao Antigo Egipto, é extremamente simples podendo ser jogado com uma folha de papel e um lápis (Zaslavsky, 1982). O tabuleiro é construído com quatro linhas e é jogado por dois jogadores que desenham as suas peças a cada jogada. As regras são simples, o primeiro jogador a fazer uma linha de peças suas na vertical, horizontal ou diagonal vence. Apesar da aparente simplicidade existem 362.880 maneiras de poder dispor as peças no tabuleiro.

O sucesso de um jogo deve-se ao conceito de jogo com significado (do inglês *meaningful play*) (Zimmerman e Salen, 2003). Trata-se um conceito de design de jogos (do inglês *game design*) que se centra na experiência do jogo e pode ser descrito de duas formas, descritiva e avaliativa. Na primeira assume-se que o significado resulta da relação da ação do jogador e do seu resultado no jogo, ou seja, o significado que a ação representa no jogo. Na forma avaliativa o significado ocorre quando as relações entre as ações e os seus resultados são discerníveis e integrados no contexto geral do jogo. Discernível significa que o jogador entende o resultado imediato da ação e a Integração significa que o resultado da ação faz parte do sistema do jogo como um todo. A compreensão da definição de Jogo com Significado através destas duas formas é necessária para o design de um jogo com sucesso.

2.2 Inteligência Artificial

O Deep Blue é um famoso exemplo de inteligência artificial, tratando-se de um agente desenvolvido para um jogo específico, o Xadrez. Em 1997, como ilustrado na Figura 2.4, o Deep Blue venceu Garry Kasparov, o campeão mundial de xadrez da época, reforçando o paradigma homem versus máquina, onde um algoritmo desenvolvido por humanos, consegue ser superior a um humano graças à sua capacidade de computação.



FIGURA 2.4: Jogo de Xadrez entre Garry Kasparov e Deep Blue (Nadel, 1997).

A resolução de problemas é um dos campos onde a Inteligência Artificial (IA) tem um papel preponderante no universo dos jogos, de tabuleiro ou outros, e que servem de base para a teoria de existência de IA Fraca e IA Forte. A IA Fraca aborda problemas não determinísticos e um dos seus maiores estudiosos foi Alan Turing que questionou a existência de computadores inteligentes, através de um teste desenvolvido por si, o Teste de Turing (Costa e Simões, 2008). O teste consiste em colocar questões a um computador e um humano, não sendo possível saber de quem são as respostas. O computador passa no teste caso não seja possível distinguir as suas respostas das do humano. A IA fraca centra-se no desenvolvimento de inteligência artificial que não é capaz de raciocínio e resolução de problemas verdadeiramente. Uma máquina com esta inteligência age como tal, mas não possui noção da sua existência e consciência.

A IA Forte investiga o desenvolvimento de máquinas capazes de raciocínio e resolução de problemas, sendo denominadas de auto-conscientes. Esta teoria é controversa pois evoca problemas éticos sobre como se deve lidar com uma entidade não humana mas que

é cognitivamente semelhante. O filme “Eu, Robot” realizado por Alex Proyas aborda esta teoria, usando as três Leis da Robótica criadas pelo escritor Isaac Asimov (1942). Leis que visam o controlo comportamental dos robôs tendo em conta o bem-estar dos humanos.

Agentes como o Deep Blue, desenvolvido por humanos com a finalidade de jogar Xadrez enquadram-se na teoria da IA Fraca. Caso se coloque o *Deep Blue* a jogar Damas, que apesar de ser jogado num tabuleiro semelhante ao Xadrez, este agente não tem capacidade de resposta pois não possui “inteligência” para jogar outro jogo que não Xadrez, e é neste campo que se distinguem os Agentes de Jogos Genéricos.

2.3 Jogos Genéricos

Jogos Genéricos é uma área de estudo de jogos na Inteligência Artificial. Consiste numa framework de modelação de sistemas dinâmicos discretos e tem em conta complexidades como problemas de representação, limite de recursos e falta de informação. Os Jogos Genéricos também levantam questões sobre a natureza da inteligência, servindo de base de estudo em abordagens concorrentes de inteligência artificial (Genesereth e Thielscher, 2014).

John McCarthy (1959) criou o conceito de “Conselheiro”, que se tornou um clássico da Inteligência Artificial ao pretender desenvolver uma máquina que pudesse ser programada através de descrições, sem necessidade de programação tradicional. Fornecendo os objetivos e o ambiente à máquina, esta deveria retornar o comportamento adequado. Segundo McCarthy, esta máquina deveria melhorar iterativamente o seu comportamento, através de várias descrições.

Os agentes de Jogos Genéricos identificam-se com o conceito de McCarthy, e para estes agirem, têm de assumir que os jogos partilham características semelhantes. Jesper Juul (2005) define os jogos de acordo com as seguintes características: são baseados em regras; os resultados são variáveis e quantificáveis; os resultados podem ser positivos ou negativos; o esforço do jogador influencia o resultado; as emoções do jogador dependem do resultado; e um jogo pode ser jogado com ou sem consequências na vida real. Tendo em conta esta definição, a IA dos agentes pode desenvolver-se quantificando as ações do jogador, os resultados do jogo e como se aplicam as regras, que através dos seus algoritmos encontram o melhor caminho (Levine et al., 1998).

2.3.1 Abordagens dos Jogos Genéricos

Existe um conjunto de abordagens à temática dos Jogos Genéricos, sendo que a do Grupo de Lógica da Universidade de Stanford (2017) é mais relevante, recorrendo à Computação Lógica, uma área da Ciência da Computação especializada na representação e processamento de informação em declarações lógicas. Um dos seus objetivos é o estabelecimento de normas que são a base a seguir pela comunidade que faz investigação nesta área, o que facilita a comunicação e troca de ideias.

Existem outras duas abordagens desenvolvidas sobre a temática dos Jogos Genéricos, nomeadamente o *MetaGame* e o *Zillion of Games*. *MetaGame* foi criado por Barney Pell (1993), tendo sido o primeiro programa a gerar regras automaticamente para jogos semelhantes ao Xadrez, e consequentemente desenvolveu o sistema *Metagamer*, capaz de jogar sem necessitar de interação humana, usando somente as regras que lhe eram fornecidas na linguagem de descrição de jogo (GDL, do inglês *Game Description Language*).

Zillion of Games foi criado por Mark Lefler e Jeff Mallet em 1998, e esta abordagem cria a sua função de avaliação a partir das regras do jogo baseadas no movimento das peças, objetivos do jogo e estrutura do tabuleiro (Lefler e Mallett, 2017). Os algoritmos usados são semelhantes aos utilizados em jogos de Xadrez no computador e as regras são definidas usando uma linguagem semelhante a LISP (List Processor).

Existe uma grande variedade de jogos usados na área de Jogos Genéricos, sendo os Jogos de Tabuleiro os mais utilizados. No entanto podem também ser usados Jogos de Vídeo (Mnih et al, 2015) e Jogos de Mesa, nos quais os Jogos de Tabuleiro se inserem.

2.3.2 Estrutura dos Jogos Genéricos

Os jogadores de Jogos Genéricos, para participarem num jogo, devem ligar-se a um servidor que monitoriza as jogadas, indicando se estas são legais, ou seja, se as jogadas são válidas de acordo com as regras, e que informa os jogadores quando ocorrem mudanças de estado, ou seja, cada vez que uma jogada é realizada. Apesar da variedade de jogos, todos partilham uma estrutura abstrata comum. Cada jogo desenrola-se num ambiente com um conjunto de estados finitos, existindo um estado inicial e um ou mais estados finais. Além disso cada jogo possui

um número fixo de jogadores e cada jogador possui um conjunto de ações finito em cada estado, e cada estado representa um valor associado a cada jogador.

Os Jogos Genéricos usam um modelo dinâmico sincronamente atualizado, onde todos os jogadores movem-se em todas as jogadas, podendo o “não mover-se” ser um movimento, ou seja, as peças ficam na mesma posição, e o ambiente só se altera em resposta a ações realizadas pelos jogadores. Os estados, ou jogadas, dos jogos podem ser representados por grafos como ilustrado na Figura 2.5, em que é demonstrado um exemplo de um jogo, a ser jogado apenas por um jogador, com oito estados (nomeados de s_1, \dots, s_8), em que o estado inicial é s_1 e o estado final pode ser s_4 ou s_8 , consoante as jogadas feitas pelo jogador. Cada estado contém um valor (neste caso 0, 25, 50 ou 100), e as setas entre os estados representam a transição entre estados e contêm as funções de transição do jogo. Assumindo que o jogo se encontra no primeiro estado e o jogador realiza a ação a o jogo passará para o estado s_2 , mas se o jogador realizar a ação b , o jogo passa para o estado s_5 (Michael Genesereth e Michael Thielscher, 2014).

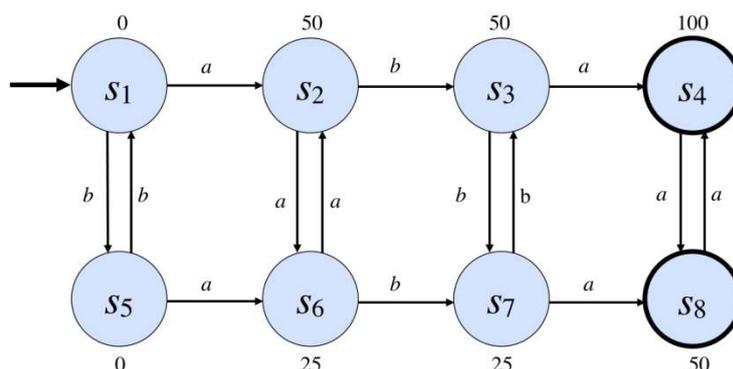


FIGURA 2.5: Exemplo de um conjunto de estados de um jogo com jogador único (Genesereth e Thielscher, 2014).

A representação baseada em estados, apresentada no grafo da Figura 2.5, é uma alternativa à forma extensiva de definição de jogos em teoria de jogos. Apesar da forma extensiva normal poder ser mais apropriada para certos tipos de análise, a representação baseada em estados apresenta vantagens nos jogos genéricos, como a codificação de jogos numa forma mais compacta que uma representação direta. GDL suporta isto através de uma conceptualização dos estados dos jogos enquanto base de dados e baseando-se na lógica para definir as noções de legalidade e afins. Na forma extensiva, o jogo é modelado como uma árvore de jogo, e não como um grafo. Numa árvore de jogo, cada nó representa um estado, e é

ligado aos seus nós sucessores através de setas que indicam as ações possíveis do jogador a cada estado. Apesar de nós diferentes corresponderem a estados diferentes, é possível que nós diferentes correspondam ao mesmo estado. A descrição de jogos baseada num grafo de estados torna-se mais compacta, oferecendo uma jogabilidade mais eficiente (Genesereth e Thielscher, 2014).

2.3.3 Descrição dos Jogos Genéricos

Os Jogos Genéricos podem ser definidos através de regras descritas em GDL (*Game Description Language*), uma linguagem criada no Projeto de Jogos Genéricos da Universidade de Stanford. GDL descreve o estado de um jogo como uma série de factos e as mecânicas do jogo, como regras lógicas. Segundo Nathaniel Love (2008), um ficheiro GDL contém a descrição e regras dos jogos e está dividido pelas seguintes propriedades e respectivas definições:

- *role*: define os jogadores que irão jogar e o seu nome, como ilustrado no Código Fonte 2.1;
- *init*: indica o estado inicial do jogo, como ilustrado no Código Fonte 2.2;
- *true*: testa factos do jogo para os identificar como verdadeiros ou não, como ilustrado no Código Fonte 2.3;
- *does*: verifica se um jogador realiza uma certa jogada, como ilustrado no Código Fonte 2.4;
- *next*: atualiza o estado do jogo para o estado seguinte, como ilustrado no Código Fonte 2.4;
- *legal*: expressão que testa se os movimentos do jogador são permitidos, como ilustrado no Código Fonte 2.3;
- *goal*: expressão que define a pontuação de cada jogador consoante o seu desempenho no jogo, como ilustrado no Código Fonte 2.5;
- *terminal*: expressão que define as condições para o final do jogo, como ilustrado no Código Fonte 2.7.

Como referido anteriormente, considerando que todos os jogos são finitos, é possível descrevê-los através de um grafo de estados (Genesereth e Thielscher, 2014). No caso de jogos mais complexos a representação pode ser bastante extensa, como a do jogo *skirmish* (equivalente ao Xadrez), onde existem centenas de movimentos e mais de 10^{30} estados. No entanto, os seus estados não são monolíticos, podendo estes ser conceptualizados com todos os seus elementos, como as peças, células, linhas, colunas e diagonais do tabuleiro. Usando a estrutura de grafo de estados é possível definir jogos numa forma mais compacta do que usando uma representação mais direta, pois esta não possui mecanismos de simplificação na definição de jogo. Este tipo de conceptualização pode ser reproduzido usando GDL.

Nos Código Fonte 2.1, 2.2, 2.3, 2.4, 2.5, 2.6 e 2.7 é demonstrado um exemplo da descrição de um jogo *Tic-tac-toe* (equivalente ao Jogo do Galo), usando GDL, de Genesereth e Thielscher (2014). No Código Fonte 2.1 são definidos dois jogadores, um com o nome “branco” (`role(white)`) e outro com o nome “preto” (`role(black)`) (Genesereth e Thielscher, 2014).

```
1:   role(white)
2:   role(black)
```

CÓDIGO FONTE 2.1: Definição de dois jogadores, GDL (Genesereth e Thielscher, 2014).

No Código Fonte 2.2 é demonstrado que num estado do jogo, ou seja, numa das jogadas, em que o jogador “branco” (`white`) irá jogar (`init(control(white))`) todas as células (`cell`) encontram-se vazias (`b`). Na linha 3 encontra-se um exemplo em que a célula, com posição X e Y igual a 1, está vazia (Genesereth e Thielscher, 2014).

```
3:   init(cell(1,1,b))
4:   init(cell(1,2,b))
5:   init(cell(1,3,b))
6:   init(cell(2,1,b))
7:   init(cell(2,2,b))
8:   init(cell(2,3,b))
9:   init(cell(3,1,b))
10:  init(cell(3,2,b))
11:  init(cell(3,3,b))
```

```
12:  init(control(white))
```

CÓDIGO FONTE 2.2: Estado inicial de todas as células, GDL (Genesereth e Thielscher, 2014).

Os movimentos legais são definidos como demonstrado no Código Fonte 2.3. Um jogador (`white` ou `black`) pode marcar (`mark`), ou seja, pode inserir uma peça numa célula (`cell`) se esta estiver vazia (`X`, `Y`, `b`) e for a sua vez de jogar, ou seja, se estiver com o controlo da jogada (`control`). O operador *if* designa-se por ‘:-’, e a expressão à sua esquerda é realizada, ou `legal` (`legal`), caso as expressões que se seguem sejam verdadeiras (`true`). Em GDL a definição de variáveis começa por letra maiúscula (e.g., `X`), enquanto que as constantes começam por minúsculas (e.g., `legal`). Visto isto, caso o jogador não tenha o controlo da jogada e haja pelo menos uma célula vazia, então o movimento legal é a não ação, designada por `noop`. Nas linhas 16, 17 e 18 encontram-se um exemplo em que a ação do jogador “branco” terá de ser `noop` pois o jogador “preto” tem o controlo da jogada e há células vazias (Genesereth e Thielscher, 2014).

```
13:  legal(W,mark(X,Y)) :-
14:    true(cell(X,Y,b)) &
15:    true(control(W))
16:  legal(white,noop) :-
17:    true(cell(X,Y,b)) &
18:    true(control(black))
19:  legal(black,noop) :-
20:    true(cell(X,Y,b)) &
21:    true(control(white))
```

CÓDIGO FONTE 2.3: Definição dos movimentos legais, GDL (Genesereth e Thielscher, 2014).

No Código Fonte 2.4 é definida a atualização do estado do jogo para o estado seguinte (`next`). Uma célula (`cell`) é marcada (`mark`) com um de dois valores possíveis (`x` ou `o`), dependendo do valor atribuído ao jogador correspondente a jogar. Caso a célula já se encontre marcada, mantém a mesma marca anteriormente realizada. Se uma célula estiver vazia e não for marcada nessa jogada, esta mantém-se vazia. No final o controlo (`control`) do jogo é alternado a cada jogada, como no exemplo das linhas 39 e 40, em que caso o jogador “branco” tenha o controlo da jogada, o mesmo passa a ser do jogador “preto” (Genesereth e Thielscher, 2014).

```
22: next (cell (M,N,x) ) :-
23:     does (white,mark (M,N) ) &
24:     true (cell (M,N,b) )
25: next (cell (M,N,o) ) :-
26:     does (black,mark (M,N) ) &
27:     true (cell (M,N,b) )
28: next (cell (M,N,W) ) :-
29:     true (cell (M,N,W) ) &
30:     distinct (W,b)
31: next (cell (M,N,b) ) :-
32:     does (W,mark (J,K) ) &
33:     true (cell (M,N,b) ) &
34:     distinct (M,J)
35: next (cell (M,N,b) ) :-
36:     does (W,mark (J,K) ) &
37:     true (cell (M,N,b) ) &
38:     distinct (N,K)
39: next (control (white) ) :-
40:     true (control (black) )
41: next (control (black) ) :-
42:     true (control (white) )
```

CÓDIGO FONTE 2.4: Definição da atualização do estado do jogo para o estado seguinte, GDL (Genesereth e Thielscher, 2014).

No Código Fonte 2.5 define-se a pontuação do jogo. O jogador “branco” recebe 100 pontos caso haja uma linha de x (o valor associado ao jogador “branco”) e nenhuma linha de o (o valor associado ao jogador “preto”), tal como no exemplo da linha 43. No caso de existir uma linha de o e nenhuma de x, o jogador “branco” recebe 0 pontos. No caso de nenhum dos jogadores ter feito uma linha, cada um recebe 50 pontos. O sistema de pontuação é repetido para o jogador “preto”, mas invertendo os valores x e o (Genesereth e Thielscher, 2014).

```
43: goal (white,100) :- line (x) & ~line (o)
44: goal (white,50) :- ~line (x) & ~line (o)
45: goal (white,0) :- ~line (x) & line (o)
46: goal (black,100) :- ~line (x) & line (o)
```

```
47: goal(black,50) :- ~line(x) & ~line(o)
48: goal(black,0) :- line(x) & ~line(o)
```

CÓDIGO FONTE 2.5: Definição da pontuação do jogo, GDL (Genesereth e Thielscher, 2014).

Os conceitos de suporte são demonstrados no Código Fonte 2.6 em que uma linha (*line*) define-se como um conjunto de marcas (*mark*) com o mesmo valor (*Z*), que neste jogo pode conter *x* ou *o*, na mesma linha (*row*), coluna (*column*) ou diagonal (*diagonal*). Uma linha significa que existe três valores iguais em que a primeira coordenada dos valores é a mesma. As linhas 52 a 55 apresentam um exemplo de uma linha, em que *M* poderia conter o valor 3 e *Z* conter o valor *x*. O mesmo é válido analogamente para a coluna e a diagonal (Genesereth e Thielscher, 2014).

```
49: line(Z) :- row(M,Z)
50: line(Z) :- column(M,Z)
51: line(Z) :- diagonal(Z)
52: row(M,Z) :-
53:   true(cell(M,1,Z)) &
54:   true(cell(M,2,Z)) &
55:   true(cell(M,3,Z))
56: column(N,Z) :-
57:   true(cell(1,N,Z)) &
58:   true(cell(2,N,Z)) &
59:   true(cell(3,N,Z))
60: diagonal(Z) :-
61:   true(cell(1,1,Z)) &
62:   true(cell(2,2,Z)) &
63:   true(cell(3,3,Z))
64: diagonal(Z) :-
65:   true(cell(1,3,Z)) &
66:   true(cell(2,2,Z)) &
67:   true(cell(3,1,Z))
```

CÓDIGO FONTE 2.6: Definição dos conceitos de suporte, GDL (Genesereth e Thielscher, 2014).

No Código Fonte 2.7 são apresentadas as condições para que o jogo termine. O jogo termina: (1) quando existe uma linha de valores do mesmo tipo, *x* ou *o*, exemplo deste caso

seria a linha 68; (2) quando deixa de estar aberto (um dos jogadores ter saído do jogo), exemplo deste caso seria a linha 70; (3) ou quando não existem células (`cell`) vazias, exemplo deste caso seria a linha 71, em que `M` e `N` são as coordenadas da célula vazia (`b`) (Genesereth e Thielscher, 2014).

```
68: terminal :- line(x)
69: terminal :- line(o)
70: terminal :- ~open
71: open :- true(cell(M,N,b))
```

CÓDIGO FONTE 2.7: Definição das condições para o final do jogo, GDL

(Genesereth e Thielscher, 2014).

Tendo em conta o acesso a todas as hipóteses, as relações descritas podem assumir-se como falsas caso não tenham sido provadas verdadeiras. Logo, devem existir definições completas das relações `legal`, `next`, `goal`, e `terminal`, usando os termos `true` e `does`. A relação de verdadeiro é iniciado no `init`, e a cada jogada é alterada para corresponder à extensão da próxima relação (`next`) nessa jogada (Genesereth e Thielscher, 2014).

2.3.4 Histórico dos Jogos Genéricos

O histórico de um Jogo Genérico é tipicamente criado e guardado durante e após uma partida, quer seja entre agentes, humanos ou ambos. A análise do histórico permite analisar o desempenho dos agentes e conseqüentemente, processá-los para os tornar mais eficientes. O gerador de cenários irá utilizar o histórico da partida de um jogo para recriar graficamente a partida, permitindo a análise jogada a jogada, respondendo desta forma a um dos desafios propostos nesta dissertação, que passa por tornar a interface gráfica dos jogos genéricos numa experiência mais rica e satisfatória. No Código Fonte 2.8, 2.9 e 2.10 é demonstrado um exemplo do histórico de um jogo *Tic-tac-toe*, em XML, realizado no âmbito desta dissertação no Gamemaster. No Código Fonte 2.8 é iniciado um jogo (`match`) e indicada toda a informação relativa ao jogo como o nome do jogo (`game`) em que o valor é `Tic-tac-toe`, os jogadores do jogo (`role`) em que neste caso existem dois jogadores `white` e `black`, o tipo de jogador (`player`) em que neste caso são ambos humanos, entre outros dados como `scrambled`, `timed`, `startclock`, `playclock`, `tournament` e `owner`.

```
1:  <match id='m01'>
2:    <game>tictactoe</game>
3:    <role>white</role>
4:    <role>black</role>
5:    <player>human</player>
6:    <player>human</player>
7:    <scrambled>no</scrambled>
8:    <timed>no</timed>
9:    <startclock></startclock>
10:   <playclock></playclock>
11:   <tournament></tournament>
12:   <owner>anonymous</owner>
```

CÓDIGO FONTE 2.8: Exemplo do início de um jogo *Tic-tac-toe*, XML.

O Código Fonte 2.9 contém o estado do tabuleiro em cada jogada. Esta informação está contida no elemento `herstory`, sendo diferente do elemento `history` presente no Código Fonte 2.10. No elemento `herstory` cada estado (`state`) contém vários factos (`fact`), sendo que cada facto indica o conteúdo de uma célula (`cell`) do tabuleiro, contendo também três argumentos (`argument`) em que os dois primeiros indicam as coordenadas (`x,y`) das células e o terceiro indica se a célula em questão está vazia (`b`) ou o nome da peça contida na célula (`x` ou `o`). O último facto indica o jogador em controlo (`control`). No exemplo dado, é indicado que na primeira jogada (linhas 14 a 37), as células estão todas vazias (linhas 19, 25 e 31) e o jogador em controlo é o “branco” (linhas 34 e 35) que fará a jogada ilustrada no estado seguinte. No segundo estado (linhas 38 a 61) podemos concluir que o jogador “branco” jogou com a peça `x` a célula com as coordenadas (1,2) (linhas 45 a 50).

```
13:  <herstory>
14:    <state>
15:      <fact>
16:        <relation>cell</relation>
17:        <argument>1</argument>
18:        <argument>1</argument>
19:        <argument>b</argument>
20:      </fact>
```

```
21:      <fact>
22:          <relation>cell</relation>
23:          <argument>1</argument>
24:          <argument>2</argument>
25:          <argument>b</argument>
26:      </fact>
27:      <fact>
28:          <relation>cell</relation>
29:          <argument>1</argument>
30:          <argument>3</argument>
31:          <argument>b</argument>
32:      </fact>
33:      <fact>
34:          <relation>control</relation>
35:          <argument>white</argument>
36:      </fact>
37:  </state>
38:  <state>
39:      <fact>
40:          <relation>cell</relation>
41:          <argument>1</argument>
42:          <argument>1</argument>
43:          <argument>b</argument>
44:      </fact>
45:      <fact>
46:          <relation>cell</relation>
47:          <argument>1</argument>
48:          <argument>2</argument>
49:          <argument>x</argument>
50:      </fact>
51:      <fact>
52:          <relation>cell</relation>
53:          <argument>1</argument>
54:          <argument>3</argument>
55:          <argument>b</argument>
56:      </fact>
57:      <fact>
```

```
58:         <relation>control</relation>
59:         <argument>black</argument>
60:     </fact>
61: </state>
62: </herstory>
```

CÓDIGO FONTE 2.9: Exemplo do estado do tabuleiro em cada jogada
de um jogo *Tic-tac-toe*, XML.

O Código Fonte 2.10 contém as ações realizadas por cada jogador descritas no elemento `history`, em que cada movimento (`move`) contém duas ações, `noop`, que está associado ao jogador que não está em controlo nesta jogada e `action`, que contém o tipo de ação (`mark`) e as coordenadas (`x,y`) da célula em que ocorreu a ação. Neste caso o primeiro jogador marcou a célula com as coordenadas (1,2) e o segundo jogador não estava no controlo da jogada, logo o primeiro jogador tem a sua ação representada como `action` e o segundo jogador representada como `noop` (linhas 64 a 71). Existem outros tipos de ação possíveis em outros jogos como `move` e `jump`. Depois do elemento `history` temos a informação relativa à pontuação do jogo no elemento `rewards`, mas neste caso, como apenas temos dois estados e uma ação, ainda não foi realizada uma linha, coluna ou diagonal e as células também não estão todas preenchidas, logo, ainda não existe informação relativa à pontuação do jogo. Finalmente é fechado o elemento `match` e a informação relativa ao histórico do jogo com a primeira jogada está completa. Se existirem mais jogadas, então cada uma é representada por um elemento XML cujo elemento pai é `move`.

```
63:     <history>
64:         <move>
65:             <action>
66:                 <relation>mark</relation>
67:                 <argument>1</argument>
68:                 <argument>2</argument>
69:             </action>
70:             noop
71:         </move>
72:     </history>
73:     <rewards>
```

74: </rewards>

75: </match>

CÓDIGO FONTE 2.10: Exemplo de todas ações realizadas por cada jogador de um jogo *Tic-tac-toe*, XML.

2.3.5 Agentes de Jogos Genéricos

Os agentes de jogos genéricos são capazes de jogar qualquer jogo de tabuleiro sem possuir o conhecimento prévio do mesmo, tendo apenas o conhecimento somente quando se inicia o jogo (Genesereth e Thielscher, 2014). A sua técnica de jogo evolui ao longo da partida, ao contrário de agentes especializados, como o Deep Blue, que possui algoritmos previamente desenvolvidos. No caso dos agentes de Jogos Genéricos, os algoritmos são criados jogando partidas contra outros agentes ou humanos.

Os agentes atuam em ambientes dinâmicos ou estáticos, podendo não ter acesso a toda a informação. Podem também participar em jogos simples, como o *Tic-tac-toe*, ou em jogos mais complexos, como o *Skirmish*. O desenvolvimento de um agente de Jogos Genéricos requer conceitos de várias áreas, como o raciocínio, a decisão racional e a representação de conhecimento, unindo-os de forma sinérgica, e a sua performance depende desta união.

A IA dos agentes é desenhada com o objetivo de jogar vários tipos de jogos, logo a sua arquitetura não pode ser definida para jogar somente um conjunto específico de jogos, usando algoritmos com métodos aplicáveis a um vasto número de jogos. Existem diversos métodos que podem ser utilizados no desenvolvimento da IA dos agentes. Tipicamente são métodos de procura em árvore, como o algoritmo Monte Carlo Tree Search (MCTS) um dos mais usados (Méhat e Cazenave, 2012). Variações deste método têm sido desenvolvidas para jogar jogos de vídeo como o Pacman (Frydenberg, 2015).

2.4 Design de Interface

O gerador automático possui uma forte componente visual, que se reflete na conclusão dos objetivos traçados, dado o resultado gráfico ser o principal meio de analisar e alcançar a referida conclusão. Este resultado gráfico depende em grande parte do design de interface. O design de interface é o mecanismo que permite tornar visível a funcionalidade de um produto

que inicialmente é invisível (Saffer, 2010). Com o objetivo de criar uma interface eficiente, Ben Shneiderman (2011) criou as “oito regras de ouro do design de interface”:

- A aplicação deve ser consistente e coerente, de forma a que, quando o utilizador navega, saiba que o que interage relaciona-se com o que já foi interagido e com o que espera interagir;
- Presença de atalhos visuais para o utilizador poder navegar mais rapidamente na aplicação;
- Dar ao utilizador sentimentos de início, meio e fim do desempenho de uma ação, de modo a que esta transmita que foi devidamente cumprida;
- O utilizador deve poder “desfazer” as suas ações, de modo a que o risco de perder dados seja minimizado;
- A aplicação deve ser desenhada e implementada de forma a que seja o utilizador a iniciar uma ação e não a aplicação;
- Evitar transmitir elevadas quantidades de informação em simultâneo ao utilizador, de modo a não sobrecarregar a sua memória a curto prazo;
- Objetos interativos devem ter tamanhos razoáveis, pois facilita a interação;
- As mensagens de erro nunca devem ser centradas no utilizador, mas sim na máquina, de forma a que o utilizador não sinta culpa por algo errado que tenha feito.

É através dos componentes que constituem o interface que o utilizador interage com a aplicação, permitindo-lhe o seu controlo. A identificação destes componentes é um trabalho que ocorreu de forma mais exaustiva durante a fase de desenho da aplicação, apresentada no Capítulo 3.

Design responsivo é uma abordagem de web design que resulta numa correta adaptação de um website ou aplicação web numa variedade de dispositivos com diferentes tamanhos e tipos de ecrã (Subić et al., 2014). O design responsivo deve ter em conta os conteúdos a apresentar, design e performance do produto final. Dado que o gerador de cenários está assente em tecnologia web, tem-se como objetivo o disponibilizar no maior número possível de dispositivos.

2.5 Sistemas de Jogos Genéricos

2.5.1 Agentes de Jogos Genéricos

O desenvolvimento de algoritmos de jogo é onde existe mais trabalho desenvolvido. Neste âmbito identificam-se duas áreas específicas, o estudo de heurísticas mais eficientes a utilizar e a apresentação e estudo de agentes. No estudo de heurísticas são exemplo a Construção Automática de Heurísticas (Kuhlmann et al., 2006), que revela um conjunto de métodos que podem ser usados para a construção de agentes de jogo na sua totalidade, e a Avaliação de Funções Heurísticas (Clune, 2007), que descreve um método de interpretação e criação de funções que representam os valores simplificados de um jogo.

Na área do desenvolvimento de agentes salientam-se o CadiaPlayer, em que o seu núcleo é o controlo de procura informado, usado para realizar simulações (Finnsson e Bjornsson, 2011). No entanto, esta abordagem falha ao não conseguir tirar partido de conceitos de jogo de alto-nível, que são importantes para jogadas de habilidade em vários jogos. O agente Gamer, testa uma abordagem diferente, onde primeiro se instancia o jogo e se procura resolvê-lo, só depois o GDL é interpretado, aumentando assim a velocidade do agente, usando uma solução óptima (Kissmann e Edelkamp, 2011).

2.5.2 Servidores de Jogos Genéricos

Existem em funcionamento alguns projetos notáveis, como o Gamemaster, desenvolvido pela Universidade de Stanford. O Gamemaster trata-se de um website de gestão da informação oficial de competições Jogos Genéricos, como os jogos, partidas jogadas, torneio, jogadores e pessoas, sendo também usado no lecionamento da disciplina que aborda os Jogos Genéricos. Além desta informação também são apresentadas a descrição, regras, histórico e folhas de estilo para os jogos registados baseados em ficheiros GDL, XML e JavaScript (Mohapatra e Genesereth, 2015). Na Figura 2.6 é ilustrado o jogo *Tic-tac-toe* no Gamemaster em que o jogador *White* ganhou a partida pois concluiu uma diagonal com o valor *x* e assim obteve uma pontuação de 100, na sua última jogada, em que marcou (*mark*) com o valor *x*, associado ao jogador *White*, a posição, ou célula, (3,1) do tabuleiro. Tem-se acesso no menu de topo à informação relativa a todas as jogadas, em XML, realizadas durante a partida, e à informação relativa às regras do jogo, em GDL, respectivamente no botão *History* e no botão *Game*.

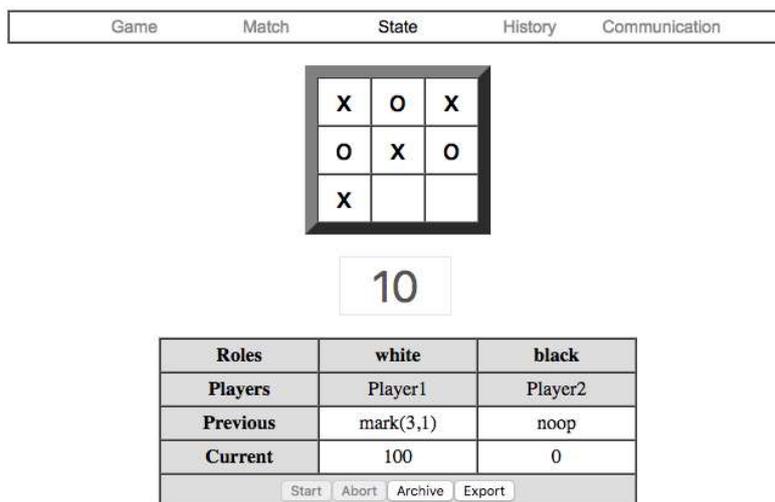


FIGURA 2.6: Jogo Tic-tac-toe no Gamemaster da Universidade de Stanford (Gamemaster).

O GGP Server desenvolvido pelo grupo de Computação Lógica da Universidade de Dresden, é semelhante ao Gamemaster, dando a possibilidade de testar agentes numa grande variedade de jogos contra outros jogadores, podendo registrar-se um agente e o servidor coloca-o automaticamente a jogar (Michulke e Thielscher, 2009).

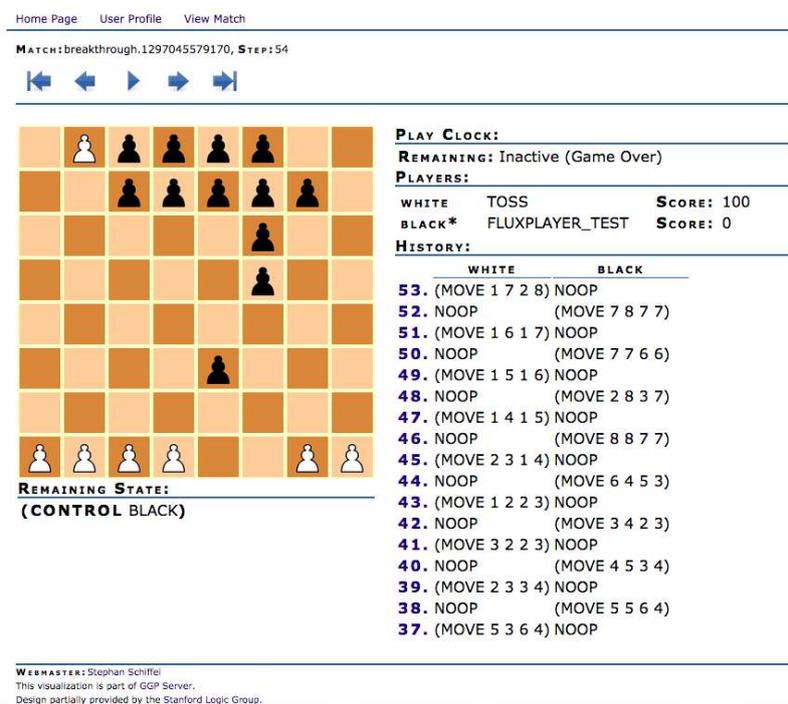


FIGURA 2.7: Jogo Breakthrough no GGP Server da Universidade de Dresden (Dresden GGP Server).

A Figura 2.7 ilustra o jogo Breakthrough, no GGP Server, na última jogada (número 54) em que o jogador *Black*, com o nome *FLUXPLAYER_TEST*, está em controlo da jogada

atual mas o jogo já terminou pois o jogador *White*, com o nome *TOSS*, obteve 100 de pontuação ao realizar a sua última jogada. Também podemos ver que na jogada número 50 o jogador *White*, como não estava em controlo da jogada, não fez nenhum movimento (*NOOP*), e o jogador *Black* fez um movimento (*MOVE*) da peça que estava na posição do tabuleiro (7, 7) para a posição (6, 6). Tem-se acesso no menu de topo à informação relativa a todas as jogadas, em XML, realizadas durante a partida, no botão *View Match*, e à informação relativa às regras do jogo, em GDL, na página do respetivo jogo no website do GGP Server.

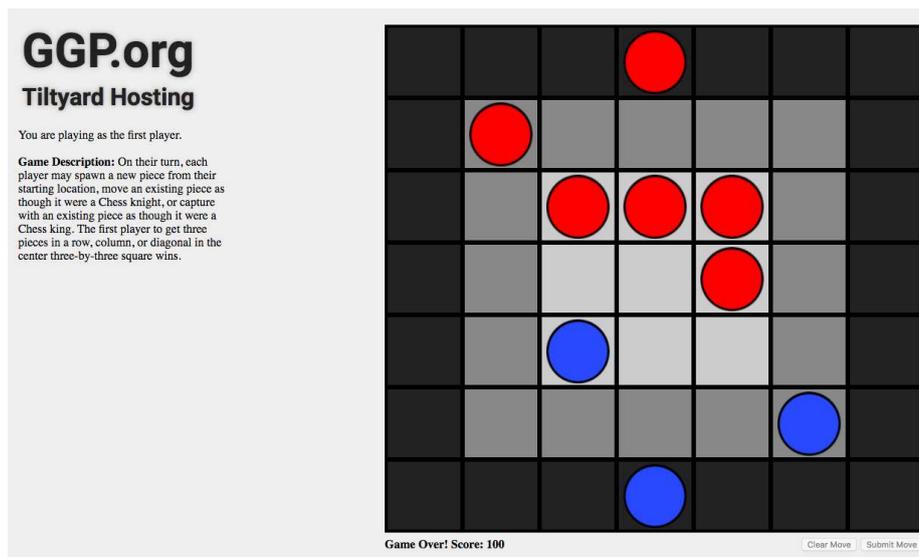


FIGURA 2.8: Jogo *Two-player free-for-all* no Tiltyard do portal GGP.org (Tiltyard).

O Tiltyard é um servidor semelhante aos apresentados anteriormente. É de código aberto e faz parte do portal GGP.org, que oferece serviços a programadores e investigadores para a criação de jogos e serviço de alojamento de sistemas. Este servidor corre continuamente, criando partidas de Jogos Genéricos, facilitando o teste de agentes (Trutman e Schiffel, 2016). Na Figura 2.8 é ilustrado o jogo *Two-player free-for-all* no Tiltyard em que dois jogadores completaram um jogo em que a condição para o jogo terminar era que um dos jogadores criasse uma linha, coluna ou diagonal, com as peças a que está associado, dentro do quadrado central (3x3). O jogador que ganhou foi o primeiro, o que está associado às peças vermelhas, com 100 de pontuação. Neste servidor apenas se tem acesso a uma ilustração em 2D do estado atual do jogo e uma breve descrição sobre as regras do jogo, não tendo acesso a informação detalhada como todas as jogadas já realizadas, o nome do jogo e os nomes dos jogadores. A informação relativa às regras do jogo, em GDL, está disponível na página de lista de jogos do portal GGP.org.

Os projetos anteriormente indicados, Gamemaster, GGP Server e Tiltyard, são bastante semelhantes nas suas funcionalidades e objetivo: o teste de agentes de Jogos Genéricos. Apenas no Tiltyard não se tem acesso à informação detalhada quanto a todas as jogadas realizadas pelos jogadores. A interface gráfica é bastante simples e pouco convidativa quanto à sua utilização, um dos aspectos que esta dissertação se propõe a melhorar. Portanto escolheu-se o Gamemaster para alimentar o gerador de cenários com informação relativa a todas as jogadas realizadas ao longo dos jogos.

2.5.3 Competições de Jogos Genéricos

Desde 2005 que ocorrem competições de Jogos Genéricos na conferência anual da AAAI (Association for the Advancement of Artificial Intelligence). Esta competição foca-se nos jogos de tabuleiro e trata-se de um bom método de avaliação do desempenho dos agentes. Na competição, numa primeira fase é analisada o desempenho dos agentes em diferentes jogos, consoante a sua habilidade para desempenhar movimentos legais e tempo necessário para completar o jogo. Numa segunda fase os agentes competem entre si em jogos mais complexos. O agente vencedor é o que conseguir ganhar mais jogos. Até 2013 o prémio era de 10 mil dólares. O Cadia-Player é o agente que mais competições venceu, contando três até ao momento (Genesereth e Björnsson, 2013).

2.6 Modelação de Cenários para Jogos Genéricos

A modelação de cenários para jogos genéricos é um dos pontos essenciais desta dissertação, fazendo parte do processo do funcionamento do gerador de cenários desenvolvido. As soluções atuais, Gamemaster, Dresden GGP Server e Tiltyard, oferecem uma experiência de jogo pobre relativamente à sua componente gráfica, e de controlo. Os cenários destas soluções são construídos em 2D, e não representam transições do ambiente do jogo entre cada jogada, em que o utilizador tem somente acesso a uma imagem estática de cada jogada. Os controlos existentes permitem somente a navegação entre jogadas.

Foi escolhido o *browser* como suporte para apresentar o cenário dos jogos genéricos pois é uma aplicação presente nativamente em qualquer sistema operativo, chegando a considerar-se que o *browser* pode ser o próximo Sistema Operativo (Wayne, 2013). Tendo em conta este facto e as linguagens atualmente usadas nos Jogos Genéricos, a escolha da

tecnologia para o desenvolvimento do gerador de cenários recaiu sobre Javascript, pois é a linguagem de programação mais indicada para o desenvolvimento de aplicações web (Mozilla, 2017).

2.6.1 JavaScript

De acordo com a plataforma Stack Overflow (2018), JavaScript é em 2018, a linguagem principal de programação do lado do cliente em navegadores web, tendo também uma presença notável do lado do servidor, graças a ambientes como o Node.js. Este facto representa uma mais-valia, pois permite aumentar a abrangência de possíveis contribuições futuras para o projeto. Juntamente com HTML e CSS, JavaScript é uma das três tecnologias principais da Internet, possibilitando páginas web interativas, uma das componentes principais de aplicações web, também conhecidas por webapps (Flanagan, 2011).

JavaScript foi concebida com o objetivo de ser uma linguagem script orientada a objetos baseada em protótipos, tipagem fraca e dinâmica e funções de primeira classe. Também possui suporte à programação funcional e outros recursos tipicamente indisponíveis em linguagens conhecidas como Java e C++.

2.6.2 Three.js

Para o desenvolvimento do gerador de cenários, decidiu-se usar uma framework que facilite a construção dos cenários, e que visualmente permita alcançar bons resultados. Construir uma framework de WebGL a partir do zero, não é o objetivo do projeto, pois a sua construção representaria por si só um projeto de investigação. Optou-se pela framework Three.js, dada a sua documentação bem elaborada, o extenso portfólio com diversos exemplos bastante ricos visualmente e a sua vasta comunidade, que contribui para o seu crescimento. Three.js é de código aberto, tendo sido publicada em 2010, segundo o primeiro *commit* do projeto no Github (Dirksen, 2013). O Three.js é uma biblioteca / API Javascript, baseada em WebGL, que permite a criação gráfica de modelos 3D no *browser*. O processamento de modelos e animações complexas é possível sem necessitar de uma aplicação ou *plugins* específicos, como a apresentada na Figura 2.9.



FIGURA 2.9: “The Aviator”, uma animação criada com Three.js de Karim Maaloul (2016).

2.7 Daltonismo

Faz parte do gerador de cenários uma funcionalidade que o adapta ao utilizador, neste caso usou-se o daltonismo como exemplo de adaptação com o objetivo de garantir a melhor experiência de jogo possível para utilizadores com a condição de daltonismo. O Daltonismo caracteriza-se pela incapacidade de diferenciar cores, sendo uma perturbação da percepção visual tipicamente com origem genética. Um dos exemplos mais frequentes é a dificuldade em distinguir entre verde e vermelho. Identifica-se pelos grupos Monocromacia, Dicromacias e Tricromacia. Estes grupos representam tipos específicos, sendo os mais comuns: Protanopia, Deuteranopia, Tritanopia e Monocromacia (National Eye Institute, 2015).



FIGURA 2.10: Edição UNO ColorADD (The Verge, 2017)

No mundo dos jogos de tabuleiro e videojogos tem vindo a crescer a consciencialização na matéria do daltonismo, onde os designers dos jogos têm em conta os

jogadores daltónicos oferecendo dessa forma uma experiência de jogo total. Um caso prático interessante nesta temática é o jogo UNO, ilustrado na Figura 2.10, sobre o qual a empresa produtora do UNO criou uma versão com o ColorADD, que se trata de um código baseado em formas simples que traduzem as cores (The Verge, 2017).

Color Blind Pal é outro caso prático interessante, tratando-se de uma aplicação desenvolvida por Vincent Fiorentini, para as plataformas Android e iOS. A aplicação permite aos utilizadores usar imagens captadas em tempo real, por exemplo através da câmara do telemóvel, e saber a designação das cores que estão presentes na imagem. Existe também a possibilidade de aplicar filtros nas imagens que alteram a sua paleta de cor para uma paleta mais perceptível, consoante o tipo de daltonismo (MacAlpine e David, 2016). No caso dos videojogos usualmente a paleta de cor está fortemente ligada à narrativa criando a atmosfera ideal, podendo a paleta de cores ser confusa para jogadores daltónicos. Videojogos como o Overwatch, Call of Duty e entre outros, corrigem esta situação oferecendo a possibilidade de o jogador seleccionar um ou mais filtros de cor (Hardin, 2016). Inspirado nestas aplicações de sucesso o gerador de cenários também utiliza um sistema em que as cores a apresentar são escolhidas de forma a adaptarem-se a utilizadores com a condição de daltonismo, sendo este sistema descrito na Secção 4.6.1.

Capítulo 3

Desenho do Gerador de Cenários

3.1 Metodologia

A metodologia de desenho do gerador de cenários enquadra-se nas disciplinas de design de interface e usabilidade, na qual é necessário identificar os diferentes elementos que compõem o gerador de cenários e organizá-los seguindo um conjunto de detalhes, como a sua hierarquia, relacionamento lógico e boas práticas no âmbito da experiência do utilizador. O resultado final esperado deve possuir características de um website e de uma aplicação móvel, com o objetivo de poder ser usável em qualquer dispositivo. Esta orientação reforça a necessidade de consistência externa. Cada elemento constituinte encontra-se num local expectável e a sua interação devolve um resultado previsível (Nikolov, 2017). Assim garante-se uma facilidade de aprendizagem e memorização mais eficiente, duas das preocupações principais da usabilidade.

Jakob Nielsen (1999) escreveu que a usabilidade é uma disciplina que oferece um conjunto de conceitos teóricos e práticos que permite estabelecer linhas de desenvolvimento e critérios de avaliação. O desenvolvimento do gerador de cenários baseia-se neste conhecimento, que auxilia na tomada de decisões a nível de design, garantindo a criação de um produto usável para o utilizador. Tendo em conta as características expectáveis de design do gerador de cenários, nas suas componentes de apresentação e interação, investigou-se conjuntos de padrões e guias de design que sirvam de orientação. Identificaram-se os guias Google Material, desenvolvido pela Google, e o Guia iOS de Interface Humano, desenvolvido pela Apple (Dutra, 2015). Estes guias foram criados com o objetivo de apoiar a comunidade interessada em desenvolver aplicações para Android ou iOS, dois sistemas operativos que representam a quase totalidade do mercado de dispositivos móveis a nível mundial (Statista, 2017).

Com o levantamento dos conhecimentos base efetuado, simplifica-se a seleção da metodologia, pois esta tem de permitir uma articulação eficiente do desenvolvimento baseado nestes conhecimentos. O processo de desenho do gerador de cenários, dividiu-se nas seguintes etapas: identificação de elementos, layout, interações e funcionalidades. Estas etapas intersectam-se, sendo possível aplicar uma metodologia de “forma de pensar” (do inglês *design thinking*). Design thinking é um processo iterativo e não-linear, que permite que os resultados vão sendo avaliados e possam ser identificados pontos de melhoria, que originem resultados mais satisfatórios. Com esta metodologia é possível que a fase de design decorra em paralelo com a fase de implementação, permitindo a melhoria de detalhes caso necessário, como a apresentação ou interação de elementos (Interaction Design Foundation, 2017).

3.2 Elementos de Design

Como elementos de design base identificam-se: Tipografia, Controlos, Menu e Cor. Na Tipografia existem vários aspetos a definir como o tipo de fonte, tamanho e cor, que devem ajustar-se de acordo com a natureza do conteúdo que acompanham, privilegiando a sua legibilidade. Os Controlos dão a possibilidade de o utilizador navegar pelo histórico da partida, alterar a posição da câmara virtual, e outras ações interessantes que venham a ser identificadas. O Menu apresenta as funcionalidades principais do gerador de cenários. Cor é essencial, pois define a apresentação dos conteúdos nos ecrã. Usando combinações de cor é possível criar um esquema de cor eficiente e que não apresente problemas, como por exemplo o uso de cores demasiado saturadas que cansam a visão. No sentido de se tomar a melhor decisão possível em relação a estes elementos, pretende-se seguir as melhores práticas presentes nos guias iOS de Interface Humano e Google Material (Dutra, 2015).

3.3 Layout

Realizou-se um primeiro estudo com o objetivo de identificar e definir elementos a apresentar no ecrã e de que forma podem ser agrupados, criando assim as principais secções do layout, que tem como objetivo definir a organização dos elementos que constituem o gerador de cenários. Identificaram-se, como ilustrado na Figura 3.1, três grupos de elementos principais: cabeçalho, conteúdo e rodapé. O cabeçalho alberga as opções de menu, que estão numa camada anterior ao jogo de tabuleiro a processar, como a seleção de um jogo e a sua

apresentação visual. O conteúdo refere-se ao processamento gráfico do jogo do tabuleiro, este deve ocupar a maior área possível do ecrã de modo eficiente, pois tem grande influência nos objetivos do gerador de cenários. O rodapé contém os botões de navegação entre jogadas do jogo, a navegação tem impacto direto no jogo de tabuleiro atual em contido no conteúdo, sendo que as ações identificadas são as de navegação entre as jogadas.



FIGURA 3.1: Layout simplificado do gerador de cenários.

A organização estabelecida permite que o gerador de cenários tenha um bom comportamento ao nível da experiência de utilizador, através do enquadramento lógico das secções, e ao nível da responsividade, adaptando-se facilmente ao ecrã de qualquer dispositivo. Com a abordagem indicada em conjunto com a metodologia proposta, é possível dar resposta a melhorias necessárias que possam ser identificadas no decorrer de fases seguintes.

3.4 Interações

Um dos objetivos do gerador de cenários é a sua apresentação em diferentes dispositivos, como o smartphone, tablet e computador pessoal. Esta variedade de dispositivos significa um conjunto de interações distintas, sendo que nos dispositivos móveis esta é efetuada através de gestos no ecrã e no computador recorrendo ao rato.

O Three.js oferece a tradução das interações entre diferentes dispositivos, facilitando esta tarefa, mas em todo o caso é necessária a definição das interações a realizar. De forma a garantir-se a obtenção de bons resultados na interação, ou seja, não criando interações pouco naturais e de difícil execução, estabeleceu-se com base nos guias iOS de Interface Humano e

Google Material, que as interações possíveis para os dispositivos móveis são através dos gestos ilustrados na Figura 3.2, ou seja, Toque, Swipe e Pinch (Dutra, 2015).

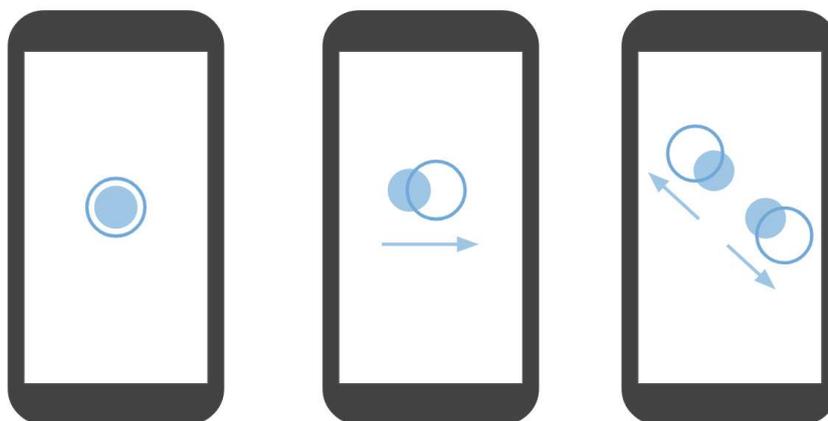


FIGURA 3.2: Gestos: Toque, Swipe e Pinch (da esquerda para a direita).

No computador os gestos são os definidos por omissão no Three.js, sendo realizados da seguinte forma através do rato: Toque - clicar no botão esquerdo; Swipe - manter clicado o botão esquerdo e mover o rato; Pinch - mover o scroll. Estas interações são naturais para o tipo de resultado expectável e consistentes com as apresentadas em aplicações móveis e jogos de computador. A consistência indicada, designa-se de consistência externa, que como referido anteriormente, proporciona uma melhor usabilidade do produto final.

3.5 Funcionalidades

As funcionalidades implementadas no gerador automático foram escolhidas baseando-se nas funcionalidades inerentes aos jogos genéricos e na sua interface gráfica tendo em conta necessidades particulares do público-alvo. Nomeadamente:

- Carregar jogo: permite ao utilizador escolher um jogo a visualizar;
- Visualizar jogo: pressupõe a apresentação do tabuleiro de jogo e a navegação entre todas as jogadas que compõem o jogo. O utilizador pode movimentar a câmara virtual livremente através das interações identificadas;
- Adaptar jogo genérico para diferentes tipos de Daltonismo: permite trocar a paleta de cores do jogo com base num tipo de Daltonismo escolhido,

oferecendo uma experiência de utilização mais apropriada para utilizadores que sofrem desta perturbação;

- Escolher estilo de apresentação do jogo: permite alterar os modelos e o ambiente de jogo, tornando a própria apresentação genérica, possibilitando ao utilizador escolher a apresentação que mais lhe agrade.

3.5.1 Carregamento e Criação da Interface Gráfica

O carregamento e a criação da interface gráfica do jogo genérico são as funcionalidades base do gerador de cenários e estão interligadas, pois dependem uma da outra. Ao efetuar-se o carregamento de um ficheiro com o histórico de um jogo genérico é criada a sua interface gráfica. Não existe interface gráfica sem anteriormente existir o carregamento de um histórico de jogo. Estas funcionalidades são uma única, Carregar Jogo, na qual o utilizador poderá escolher um jogo de um conjunto de jogos genéricos.

3.5.2 Modo de Cor

A funcionalidade de adaptar jogo genérico para diferentes tipos de daltonismo traduz-se no Modo de Cor, onde é possível alterar a paleta de cor a ser usada pelos elementos apresentados no ecrã. Foram criados modos de cor para os tipos de daltonismo mais típicos e que englobam a quase totalidade da população daltónica: Protanopia, Deuteranopia, Tritanopia e Monocromacia (Judd, 1979).

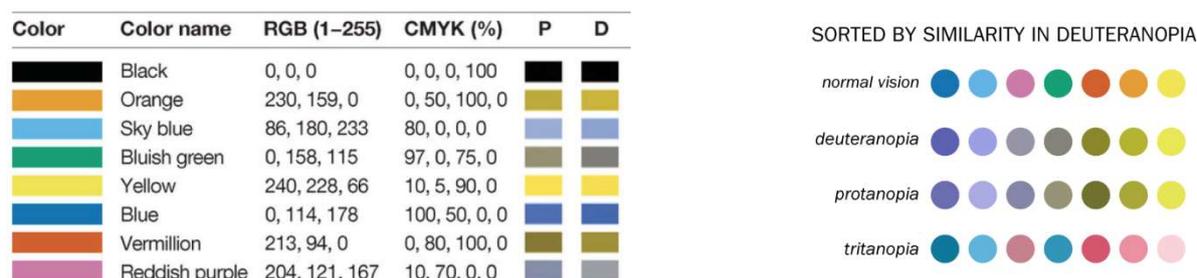


FIGURA 3.3: Paleta de 7 cores para daltonismo de Bang Wong (2011).

O desenvolvimento das paletas de cor para cada tipo de daltonismo é baseado no trabalho “Points of View: Color blindness” de Bang Wong (2011), no qual é abordado o tema do daltonismo e são apresentadas paletas de cor específicas para cada tipo de daltonismo, tal

como ilustrado na Figura 3.3 em que dada uma paleta original de cores em visão normal (do inglês *normal vision*) é apresentada essa mesma paleta mas com as cores modificadas para corresponder à cor real que os três tipos de daltonismo (protanopia, deuteranopia e tritanopia) conseguem ver.

3.5.3 Estilo de Jogo

O Estilo de Jogo é a designação simplificada da funcionalidade escolher estilo de apresentação do jogo. Por apresentação do jogo entende-se a apresentação de todos os elementos que constituem o mesmo, identificando-se as peças, o tabuleiro e o ambiente. Individualmente as peças permitem identificar grande parte dos jogos em questão, sendo o elemento mais representativo de um jogo. Com o tabuleiro respectivo obtemos o jogo na sua totalidade. Um jogo também pode ser identificado ao visualizar os movimentos das peças de cada jogada, ou seja a representação das regras do jogo. O tabuleiro tipicamente possui um formato quadrado, com as suas células no mesmo formato. No entanto, existem outras apresentações tanto ao nível do formato do tabuleiro, bem como o das suas células. Após análise aos servidores de jogos genéricos anteriormente identificados, percebeu-se que o formato de tabuleiro quadrado representa a quase totalidade de jogos apresentados. Desta forma, no âmbito desta dissertação, os jogos com maior foco são os de formato quadrado.

Por ambiente do jogo considera-se a cor de fundo do jogo, sendo suficiente para definir o contraste e atmosfera dos elementos do jogo. Com os três elementos identificados é possível criar vários tipos de apresentação de jogo, que se traduzem em estilos ou temas. Neste sentido identificou-se um conjunto de estilos de forma a demonstrar a capacidade do gerador criar ambientes distintos: Clássico, Floresta, Cidade, Campo e Selva (ilustrados em detalhe na Secção 4.6.2). Cada um destes estilos possui um conjunto de características próprias, relativo às peças e ambiente a mostrar. O estilo Clássico apresenta uma versão clássica do jogo, com as peças e cores tipicamente usadas na representação do jogo e um ambiente neutro. O estilo Floresta apresenta peças que representam vários tipos de árvores e um ambiente de cor verde. Os temas Cidade, Campo e Selva são ilustrados por peças e ambientes que transmitem as sensações tipicamente associadas a estes temas, como por exemplo os modelos de edificios tradicionais da cidade no tema Cidade.

Capítulo 4

Implementação do Gerador de Cenários

4.1 Ferramentas de Desenvolvimento

Analisaram-se as ferramentas necessárias para a criação do gerador de cenários proposto, o gerador de cenários, tendo-se identificado que é necessário um programa para gerar código HTML, CSS, JavaScript e XML, um *browser* para correr o gerador de cenários e dispositivos que correm o *browser*.

Existe uma grande variedade de ambientes de desenvolvimento de aplicações gratuitos no mercado bastante semelhantes, podendo a escolha recair no gosto pessoal, sendo este o caso. Optou-se pelo Visual Studio Code, um programa com atualizações regulares que inclui uma vasta biblioteca de extensões que auxiliam no desenvolvimento.

Na questão do *browser* não é possível escolher somente um, pois tendo em conta o tipo de dispositivos alvo do gerador de cenários, não existe no mercado um líder transversal em todos os dispositivos e também se corria o risco de não incluir um número considerável de utilizadores. Tendo em conta o estudo de utilização de browsers partilhado pelo StatCounter (2017), identificaram-se os *browsers* Chrome, Firefox, Internet Explorer e Safari.

O gerador de cenários é capaz de gerar páginas web nos seguintes dispositivos escolhidos: computador, tablet e smartphone. As suas características permitem que o gerador de cenários se adapte aos diferentes tamanhos de ecrã através do design responsivo do mesmo. O objetivo desta abordagem é que os utilizadores possam aceder ao gerador de cenários em qualquer local, não existindo limitações devido ao dispositivo.

4.2 Estrutura do Gerador de Cenários

A estrutura define os módulos que compõem o gerador de cenários, sendo estes a interpretação e a interface gráfica, como representado na Figura 4.1. O módulo de interpretação é responsável pela interpretação da leitura do carregamento de ficheiros de histórico de jogos GGP, exportados do servidor Gamemaster. Neste módulo são identificados todos os aspetos que dizem respeito ao jogo, e são criadas as instruções necessárias para o seu aspeto visual que são a informação do jogo usada no módulo de interface gráfica. O módulo de interface gráfica usa a informação do jogo e constrói uma representação visual completa do jogo, sendo também responsável por reagir às interações do utilizador através dos seus mecanismos de interação definidos. Estes mecanismos podem comunicar com o módulo de interpretação para navegar entre as jogadas.

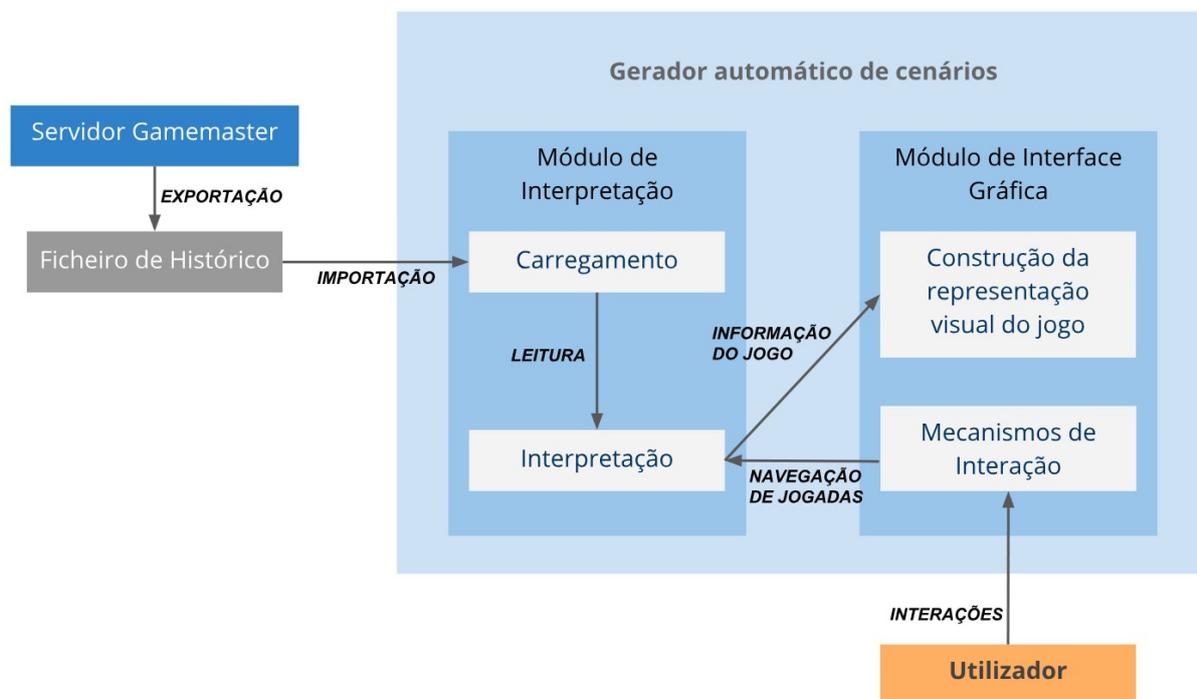


FIGURA 4.1: Estrutura do gerador de cenários.

4.3 Interpretação

A tarefa fulcral do gerador de cenários é desempenhada pelo módulo de interpretação. Através da componente genérica desenvolvida é possível, do ficheiro de histórico, recolher e interpretar um conjunto de características únicas que definem o jogo de tabuleiro em questão:

- Nome do jogo: esta informação permite escolher modelos mais indicados para o jogo em questão, no entanto mesmo que o gerador de cenários não possua informação quanto aos modelos associados a esse jogo, é sempre criada uma interface gráfica do jogo;
- Formato do tabuleiro: que tipicamente é quadrado como o *Tic-tac-toe* (3x3), e as *Breakthrough* e *Skirmish* (8x8), ou retangular como o Quatro em Linha (6x8);
- Número de jogadores: dos jogos identificados ao longo do desenvolvimento do gerador de cenários, que serviram como base de estudo, foram escolhidos os mais conhecidos pelo público geral, sendo que nestes o número pode variar entre 1 a 4 jogadores;
- Peças: esta informação indica o número de peças de cada jogador, cor, tipo e posição. Por exemplo num jogo de Xadrez, um dos jogadores joga com a cor preta e numa determinada jogada possui quatro peças, que são dois peões, um cavalo e o rei;
- Jogadas: esta informação indica o estado do tabuleiro a cada jogada do jogo, ou seja toda a informação do tabuleiro, jogadores e suas peças.

4.4 Interface Gráfica

Three.js, responsável pela criação da interface gráfica do gerador de cenários, trata-se de uma biblioteca/API JavaScript bastante leve e que funciona em browsers com suporte WebGL, ou seja *browsers* posteriores a 2011 (Khronos, 2011). A interface gráfica, em conjunto com os mecanismos de interação definidos no gerador automático, é o resultado da composição da imagem, o processo pelo qual se obtém uma imagem final a partir de uma cena virtual 3D. A composição da imagem é construída do ponto de vista da câmara virtual, que por sua vez é direcionada para a cena que contém malhas poligonais e luz, como representado na Figura 4.2. Na cena estabelece-se uma representação gráfica dos dados interpretados a partir do histórico de um qualquer jogo genérico, como a geometria e a cor (material) da malha poligonal de cada peça.

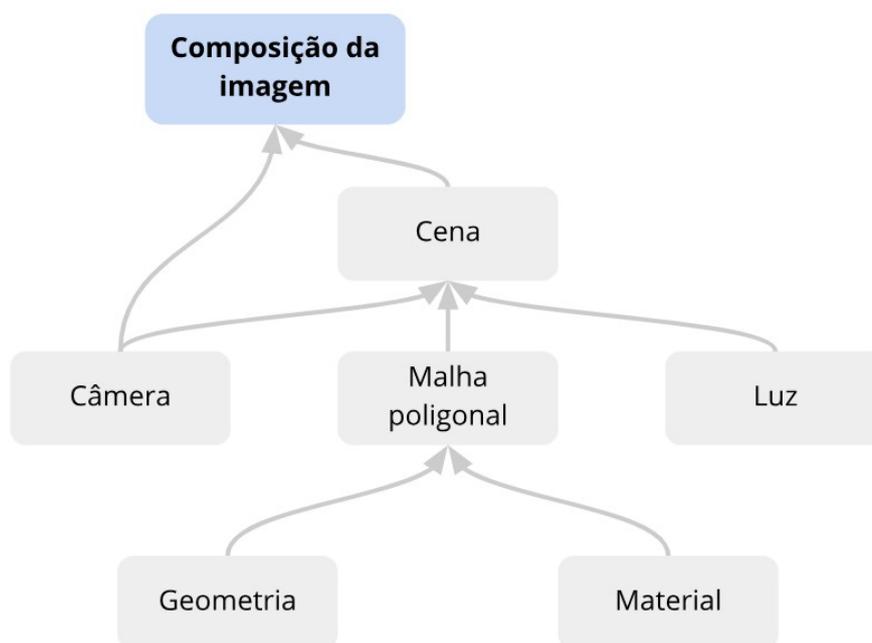


FIGURA 4.2: Estrutura da composição da imagem no Three.js.

4.5 Modelação de Elementos

Os modelos 3D dos elementos apresentados no gerador de cenários desempenham um papel importante, transmitindo vários tipos de informação através do seu aspeto visual. Isoladamente podem indicar o jogo de tabuleiro em questão, bem como o tipo de peça. De forma a garantir que os modelos são consistentes ao longo dos vários jogos, estes foram modelados de raiz ou retirados da plataforma Blend Swap, que possui direitos de utilização livre CC0 1.0 Universal (2018). Usou-se o Blender, um programa de modelação e animação 3D open-source, que permite uma integração direta com o Three.js, exportando os modelos no formato JSON, tornando mais eficiente a sintetização do modelo, segundo a documentação do Three.js (2017). Na Figura 4.3 é apresentado um modelo 3D, no Blender, que é apresentado no gerador de cenários para ilustrar peças de jogos quando o utilizador escolhe a selva como estilo de jogo (explicação sobre o estilo de jogo com mais detalhe na Secção 4.6.2).

Um aspecto importante neste processo de modelação é o facto dos modelos 3D apresentados no gerador se tratarem somente da malha poligonal, sem texturas ou animações, com o objetivo do carregamento destes modelos ser o mais rápido e eficiente possível.

Assume-se que a aplicação de atributos adicionais, como texturas ou animações, seja realizada no próprio gerador, caso se verifique que possam ser uma mais valia.

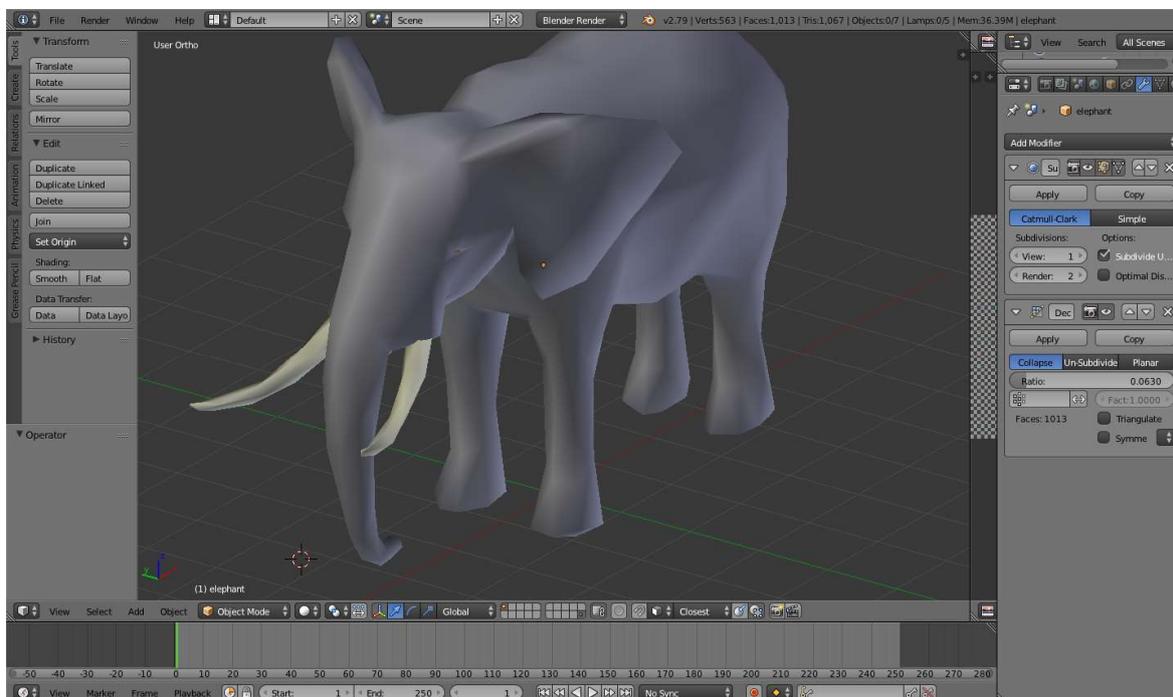


FIGURA 4.3: Modelo 3D no Blender.

4.6 Gerador de Cenários

Como indicado anteriormente, o gerador de cenários cria a sua interface gráfica a partir da interpretação do histórico do jogo. O pressuposto do projeto é que qualquer jogo genérico pode ser interpretado pelo gerador de cenários sendo que num vasto conjunto de casos a interface gráfica tem uma relação direta com o jogo em questão através de modelos realistas apresentados. No entanto, o utilizador pode alterar o Estilo de Jogo e optar por uma representação mais abstrata. O gerador de cenários possui informação relativa a um conjunto de jogos, designada por restrições, que contém informação específica que não seja indicada previamente no histórico do jogo, como por exemplo o jogo *Kono*, que apenas contém as cores das peças como nome das peças. Cada jogo pode conter as suas restrições guardadas num ficheiro individual JavaScript, estando este tema mais detalhado na Secção 4.6.3.

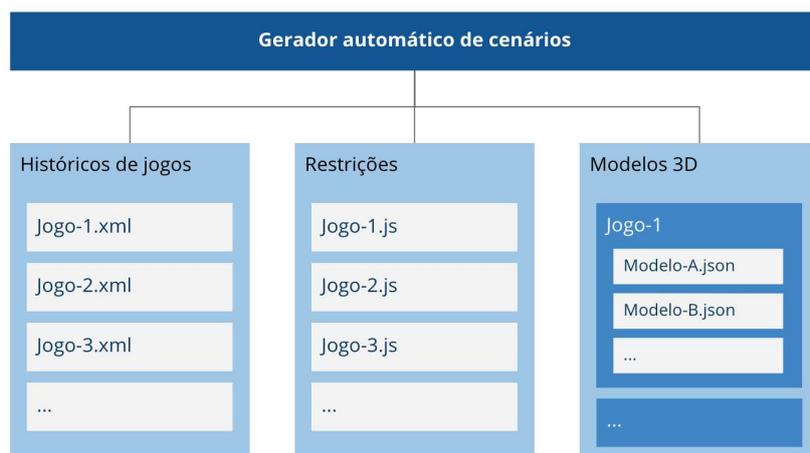
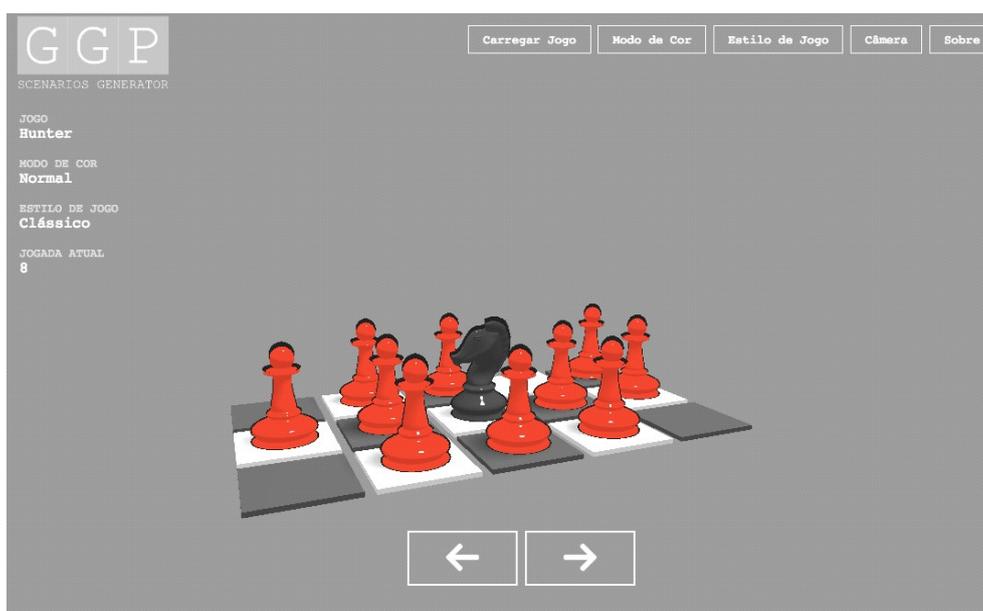


FIGURA 4.4: Estrutura de pastas do gerador de cenários.

Como demonstrado na Figura 4.4, no gerador de cenários o carregamento de toda a informação necessária, como os históricos de jogos, as restrições e os modelos 3D, é realizado dinamicamente lendo o conteúdo de pastas específicas onde estes ficheiros devem encontrar-se apenas quando um jogo é carregado, significando que o gerador de cenários não tem conhecimento prévio dos ficheiros a usar. Desta forma facilita-se a atualização do gerador de cenários em todos os elementos que o compõem, não sendo necessário alterar o código fonte do mesmo. O gerador automático contém uma página de ajuda que apresenta o contexto do gerador e dois guias. O primeiro guia é dirigido à utilização comum do gerador de cenários, explicando as funcionalidades do menu, por exemplo como trocar o modo de cor. O segundo guia é dirigido à administração do gerador, indicando como adicionar mais modelos 3D, mais jogos e respetivas restrições.

FIGURA 4.5: Jogo *Hunter* no Gerador de Cenários para Jogos Genéricos.

A Figura 4.5 ilustra o resultado final do gerador de cenários desenvolvido. A área de informação de jogo no topo esquerdo contém o logotipo do gerador de cenários (versão branca), o nome do jogo, o modo de cor e estilo de jogo ativos, e a jogada atual.

O logotipo criado para o gerador de cenários, como ilustrado na Figura 4.6, fornece uma presença visual enquanto aplicação, transmitindo a sua identidade mais facilmente aos seus utilizadores. No logotipo lê-se *GGP Scenarios Generator* (em português gerador de cenários para jogos genéricos). A tipografia usada no logotipo e em todo o gerador de cenários é a fonte Courier, uma fonte mono-espaciada gratuita, tipicamente associada à programação dada a sua legibilidade.



FIGURA 4.6: Logotipo do gerador de cenários.

Como ilustrado na Figura 4.5, o menu no topo direito apresenta as funcionalidades de Carregamento de jogo, Modo de cor, Estilo de Jogo, Câmera e Sobre, em que esta última abre a página de ajuda. A navegação entre as jogadas está presente no rodapé do ecrã. A área central do ecrã apresenta a interface gráfica do jogo escolhido, tendo disponível toda a área do ecrã, dando uma grande liberdade de visualização ao utilizador, graças também à forma como as outras áreas se encontram organizadas.

No âmbito desta dissertação testou-se um conjunto de jogos genéricos com diferentes características, mais detalhadas na Tabela 4.1, como o nome de cada jogo, o número de jogadores por jogo, o formato do tabuleiro (colunas e linhas), o formato das células (circular ou quadrada), o tipo de tabuleiro, em que tabuleiros com células de apenas uma cor são do tipo único, e tabuleiros com células de duas cores (como tabuleiros axadrezados) são do tipo dual, e o número de tipo de peças diferentes em jogo.

Jogo	Nº jogadores	Tabuleiro			Nº de tipo de peças
		Formato	Célula	Tipo	
<i>Alquerque</i>	2	5x5	Circular	Único	1
<i>Breakthrough</i>	2	8x8	Quadrada	Dual	1
<i>Breakthrough small</i>	2	6x6	Quadrada	Dual	1
<i>Checkers on a barrel no kings</i>	2	8x8	Quadrada	Dual	1
<i>Connect four</i>	2	6x8	Circular	Único	1
<i>Connect four 3p</i>	3	6x8	Circular	Único	1
<i>Connect four team</i>	4	6x8	Circular	Único	1
<i>Dont touch</i>	2	4x4	Quadrada	Único	2
<i>Duikoshi</i>	2	4x4	Quadrada	Único	4
<i>Eight puzzle</i>	1	3x3	Quadrada	Único	8
<i>Free for all</i>	2	7x7	Quadrada	Dual	1
<i>Hunter</i>	1	5x3	Quadrada	Dual	2
<i>Knights tour</i>	1	5x6	Quadrada	Dual	2
<i>Kono</i>	2	4x4	Circular	Único	1
<i>Madness</i>	2	8x8	Quadrada	Dual	5
<i>Skirmish</i>	2	8x8	Quadrada	Dual	6
<i>Skirmish bad</i>	2	8x8	Quadrada	Dual	6
<i>Sukoshi</i>	2	3x3	Quadrada	Único	3
<i>Three puzzle</i>	1	2x2	Quadrada	Único	3
<i>Tic tac toe</i>	2	3x3	Quadrada	Único	2
<i>Trifecta</i>	2	3x3	Quadrada	Único	2
<i>Ttc 4</i>	2	7x7	Quadrada	Dual	4

TABELA 4.1: Variedade de características de jogos genéricos disponíveis no gerador de cenários.

Os dados apresentados na Tabela 4.1 validam a capacidade do gerador de cenários criar interfaces gráficas para vários jogos genéricos com diferentes características, sendo um dos objetivos desta dissertação, e alguns destes jogos genéricos estão ilustrados na Figura 4.7, nomeadamente o jogo (a) com o nome *Alquerque*, (b) com o nome *Duikoshi*, (c) com o nome *Skirmish* e (d) com o nome *Connect Four Team*. Os jogos (a), (b) e (c) têm 2 jogadores e o (d) tem 4 jogadores, os jogos (a), (b) e (c) têm o número de colunas igual ao número de linhas e o (d) tem um número de colunas superior ao de linhas (6x8), os jogos (a) e (d) têm células circulares enquanto que os jogos (b) e (c) têm as suas células quadradas, os jogos (a), (b) e (d) têm o tabuleiro do tipo único pois apenas têm uma cor para as células e o jogo (c) tem o tabuleiro do tipo dual pois tem duas cores para as suas células e, finalmente, os jogos (a) e (d) têm apenas um tipo de peça no jogo enquanto que os jogos (b) e (c) têm, respectivamente, quatro e seis tipos de peças no jogo.

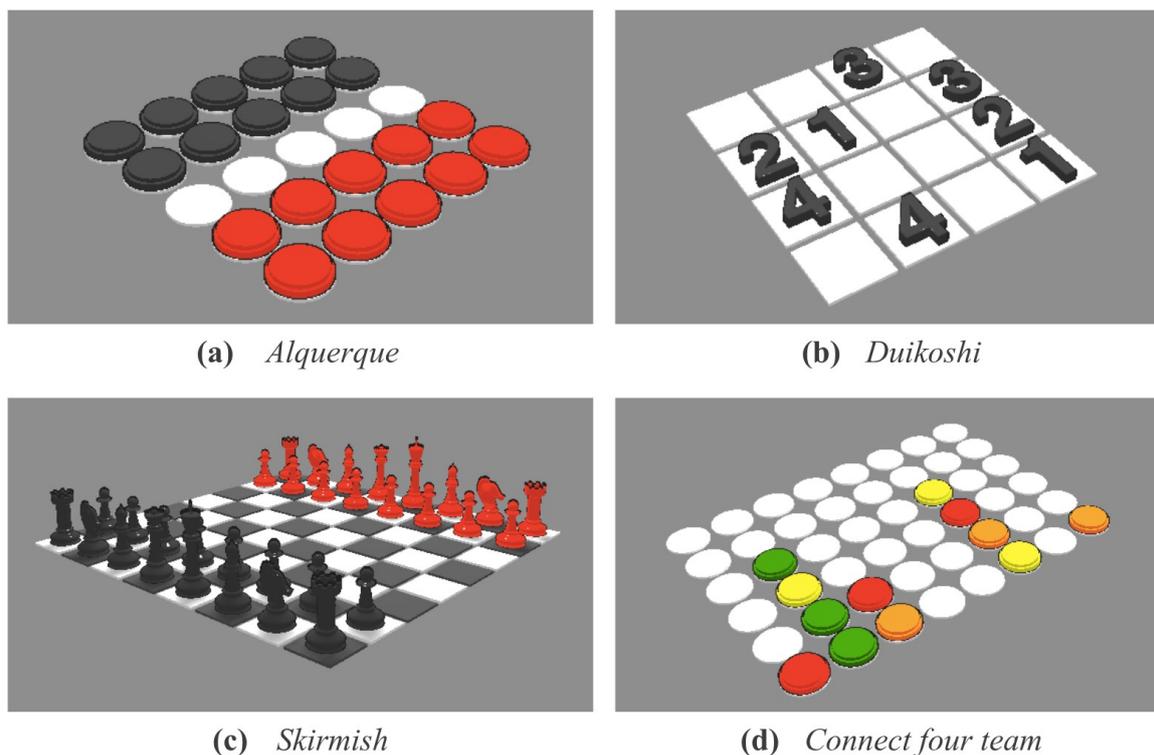


FIGURA 4.7: Diferentes jogos genéricos no gerador de cenários com os respectivos nomes.

4.6.1 Modo de Cor

O objetivo do modo de cor visa oferecer a utilizadores com daltonismo uma experiência mais adaptada às suas necessidades visuais. No gerador de cenários são disponibilizados cinco modos de cor: Protanopia, Deuteranopia, Tritanopia, Monocromacia e Não daltónico. Para aplicar aos modelos 3D das peças de jogo e ambiente de jogo foi criada uma paleta de cor, e em seguida foi feito um estudo ilustrado na Figura 4.8, em que é demonstrado como cada utilizador com o respectivo tipo de daltonismo (ilustrado nas últimas quatro linhas de cores) vê as cores da paleta original (na primeira linha de cores) através da ferramenta online Toptal Color Blind Filter (2017). Desta forma podemos ver que um utilizador com o tipo de daltonismo deuteranopia, ao visualizar a cor vermelha da paleta original (terceira cor, da esquerda para a direita, da primeira linha de cores), irá ver na realidade a cor acastanhada da paleta de deuteranopia (terceira cor, da esquerda para a direita, da terceira linha de cores) e que um utilizador com o tipo de daltonismo tritanopia, ao visualizar as cores azul e verde da paleta original (sétima e nona cor, da esquerda para a direita, da primeira linha de cores), irá ver na realidade duas cores azuis bastante semelhantes da paleta de tritanopia (sétima e nona cor, da esquerda para a direita, da quarta linha de cores).

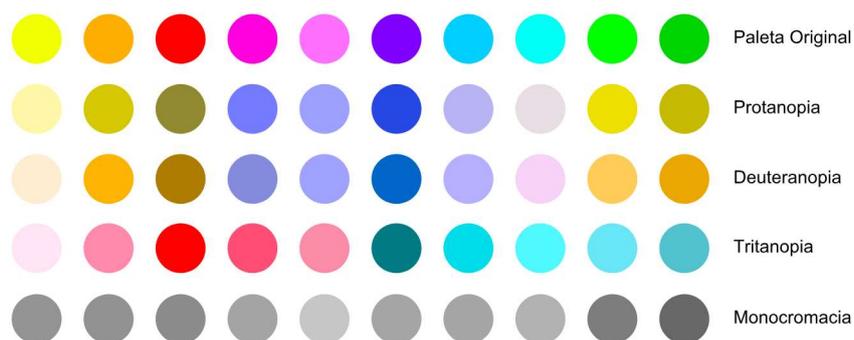
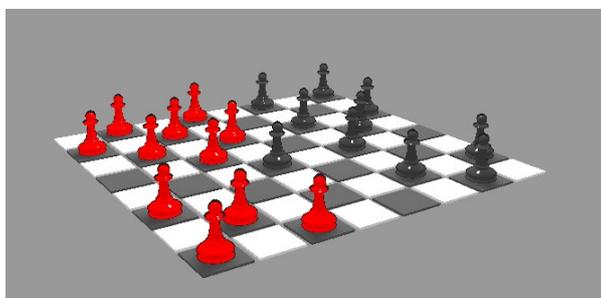


FIGURA 4.8: Paletas de cor do gerador de cenários.

No desenvolvimento do gerador de cenários as cores das cinco paletas ilustradas na Figura 4.8 foram organizadas em combinações de cor com o objetivo de ser fácil distinguir os elementos de jogo para cada tipo de daltonismo. Por exemplo, não foram criadas combinações das cores cor-de-laranja e verde da paleta original para os tipos de daltonismo protanopia e deuteranopia, pois os utilizadores com estes dois tipos de daltonismo não as iriam conseguir distinguir. Esta função evita que sejam combinadas cores demasiado semelhantes em certos tipos de daltonismo. As Figuras 4.9, 4.10, 4.11, 4.12 e 4.13 apresentam exemplos da utilização das paletas de cor, no jogo genérico *Breakthrough* (equivalente ao Jogo das Damas), nos modos de cor desenvolvidos para o gerador de cenários em que todas as figuras, excepto a Figura 4.9 pois corresponde ao modo de cor não daltónico, foram duplicadas (à direita de cada imagem), através da ferramenta online Toptal Color Blind Filter (2017), com as cores que utilizadores com o respectivo tipo de daltonismo realmente vê. Esta funcionalidade de modo de cor torna o gerador de cenários mais inclusivo, sendo assim uma mais valia. Na fase de testes foram realizados testes com utilizadores que têm daltonismo de modo a verificar o desempenho desta funcionalidade.

FIGURA 4.9: Jogo *Breakthrough* no modo de cor por omissão ou não daltónico.

Na Figura 4.10 é ilustrado um exemplo do jogo *Breakthrough* no modo de cor Protanopia. A imagem da esquerda (imagem original) da Figura 4.10 corresponde à realidade

com as cores produzidas pelo gerador de cenários. Estas cores, cor-de-laranja e lilás, são o resultado de uma das combinações de cores disponíveis na paleta original da Figura 4.8 para o tipo protanopia, pois como ilustrado na imagem da direita da Figura 4.10, estas cores são distinguíveis para um utilizador com protanopia que vê a cor cor-de-laranja como amarelo escuro e vê a cor lilás como azul.

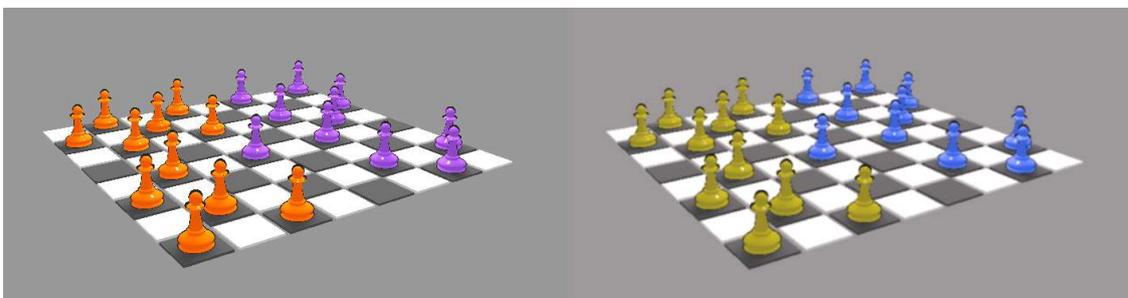


FIGURA 4.10: Jogo *Breakthrough* no modo de cor Protanopia (à esquerda) e modo de cor Protanopia visto por um utilizador com este tipo de daltonismo (à direita).

Na Figura 4.11 é ilustrado um exemplo do jogo *Breakthrough* no modo de cor Deuteranopia. A imagem da esquerda (imagem original) da Figura 4.11 corresponde à realidade com as cores produzidas pelo gerador de cenários. Estas cores, azul e vermelho, são o resultado de uma das combinações de cores disponíveis na paleta original da Figura 4.8 para o tipo deuteranopia, pois como ilustrado na imagem da direita da Figura 4.11, estas cores são distinguíveis para um utilizador com deuteranopia que vê a cor azul como azul claro e vê a cor vermelho como castanho.

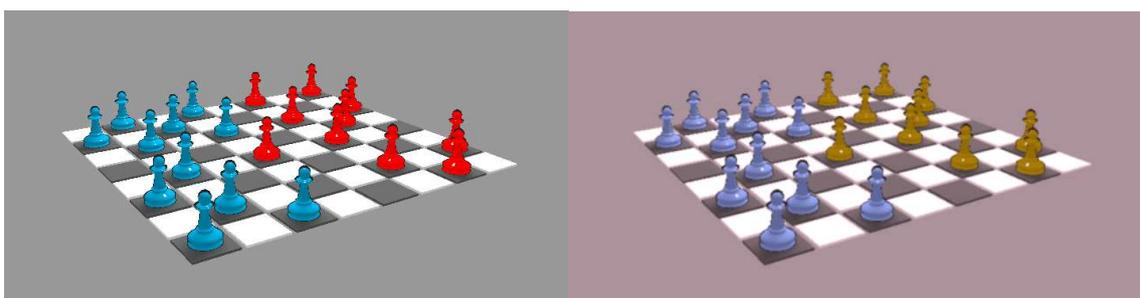


FIGURA 4.11: Jogo *Breakthrough* no modo de cor Deuteranopia (à esquerda) e modo de cor Deuteranopia visto por um utilizador com este tipo de daltonismo (à direita).

Na Figura 4.12 é ilustrado um exemplo do jogo *Breakthrough* no modo de cor Tritanopia. A imagem da esquerda (imagem original) da Figura 4.12 corresponde à realidade com as cores produzidas pelo gerador de cenários. Estas cores, verde e cor-de-laranja, são o resultado de uma das combinações de cores disponíveis na paleta original da Figura 4.8 para o

tipo tritanopia, pois como ilustrado na imagem da direita da Figura 4.12, estas cores são distinguíveis para um utilizador com tritanopia que vê a cor verde como azul claro e vê a cor cor-de-laranja como cor-de-rosa.

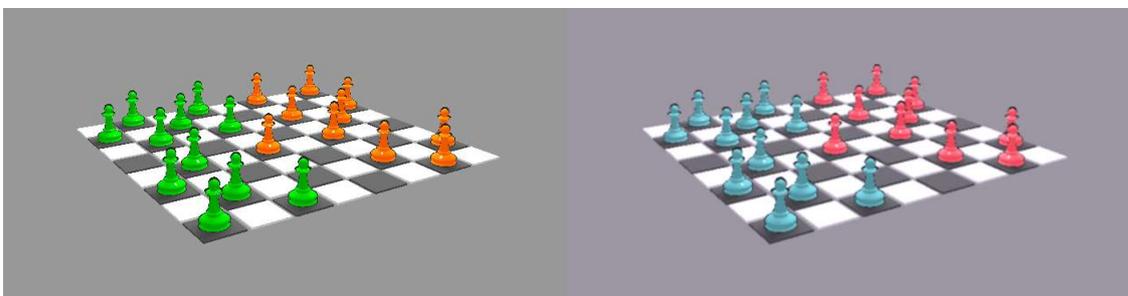


FIGURA 4.12: Jogo *Breakthrough* no modo de cor Tritanopia (à esquerda) e modo de cor Tritanopia visto por um utilizador com este tipo de daltonismo (à direita).

Na Figura 4.13 é ilustrado um exemplo do jogo *Breakthrough* no modo de cor Monocromacia. A imagem da esquerda (imagem original) da Figura 4.13 corresponde à realidade com as cores produzidas pelo gerador de cenários. Estas cores, verde e lilás, são o resultado de uma das combinações de cores disponíveis na paleta original da Figura 4.8 para o tipo monocromacia, pois como ilustrado na imagem da direita da Figura 4.13, estas cores são distinguíveis para um utilizador com monocromacia que vê a cor verde como preto e vê a cor lilás como cinzento claro.

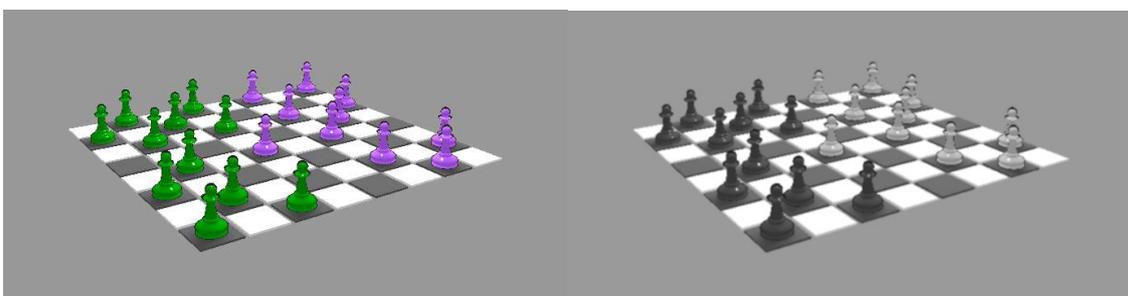


FIGURA 4.13: Jogo *Breakthrough* no modo de cor Monocromacia (à esquerda) e modo de cor Monocromacia visto por um utilizador com este tipo de daltonismo (à direita).

4.6.2 Estilo de Jogo

A funcionalidade Estilo de Jogo tem como objetivo oferecer um conjunto de diferentes tipos de interface gráfica, dando ao jogo uma opção alternativa, ao representar as peças com modelos mais abstratos, que se enquadram num determinado tema. São disponibilizados no

gerador de cenários os seguintes estilos de jogo: Clássico, Floresta, Cidade, Campo e Selva. O estilo Clássico implica que tenham sido criados modelos específicos para um determinado jogo, por exemplo, para o jogo genérico *Skirmish*, ilustrado na Figura 4.14, são disponibilizados modelos do cavalo, torre, rei, entre outros.

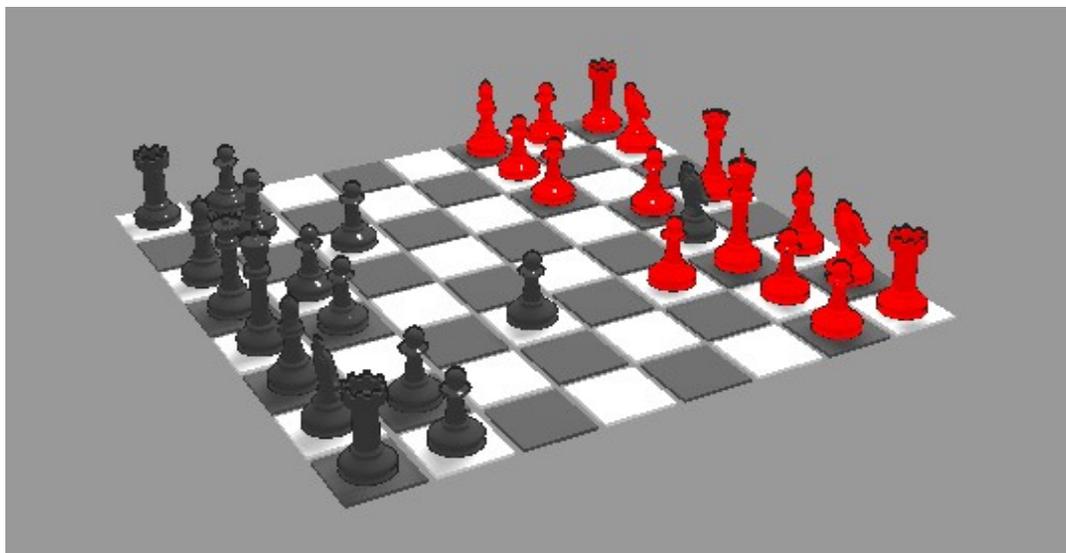


FIGURA 4.14: Jogo *Skirmish* no estilo Clássico.

Na Figura 4.15 (a) é ilustrado o jogo *Tic-tac-toe* no estilo de jogo Floresta em que as peças representam diferentes árvores e tem um ambiente de cor verde. As peças são da mesma cor pois neste jogo é importante que estas sejam diferenciadas pelo seu aspeto visual. E na Figura 4.15 (b) é ilustrado o jogo *Hunter* no estilo de jogo Campo em que as peças representam diferentes árvores e casas típicas de campo e tem um ambiente de cor amarela. As peças são de cores e formas diferentes pois é importante distinguir as peças do único jogador das restantes peças que o jogador interage ao longo do jogo.

Na Figura 4.15 (c) é ilustrado o jogo *Connect Four Team* no estilo de jogo Cidade em que as peças representam diferentes árvores, casas citadinas e prédios e tem um ambiente de cor azul escuro. As peças são de cores e formas diferentes pois é importante distinguir as peças dos quatro jogadores. Finalmente na Figura 4.15 (d) é ilustrado o jogo *Sukoshi* no estilo de jogo Selva em que as peças representam diferentes árvores e animais selvagens e tem um ambiente de cor cor-de-rosa. As peças são de formas diferentes pois é importante distinguir as diferentes peças do único jogador.

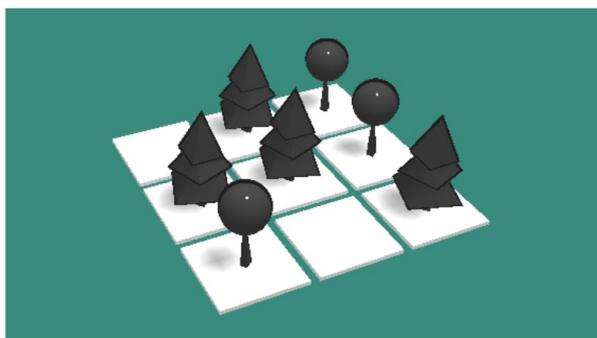
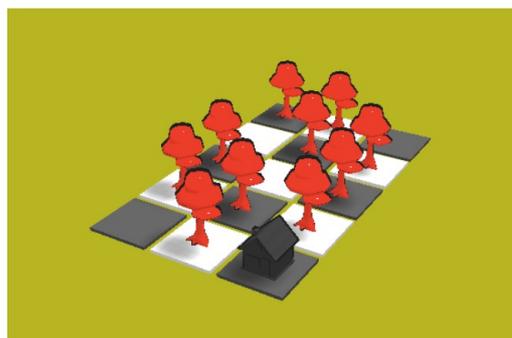
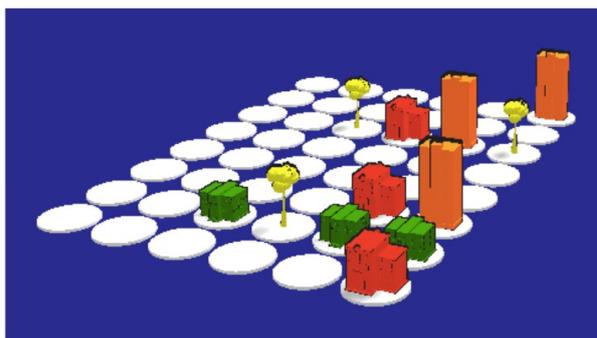
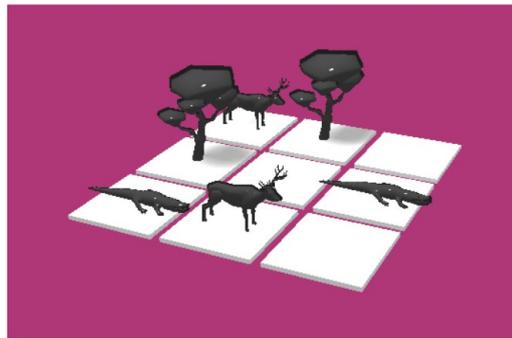
(a) *Tic-tac-toe* no estilo de jogo Floresta(b) *Hunter* no estilo de jogo Campo(c) *Connect four team* no estilo de jogo Cidade(d) *Sukoshi* no estilo de jogo Selva

FIGURA 4.15: Diferentes jogos genéricos, no gerador de cenários, com respectivos nomes e estilos de jogo.

4.6.3 Restrições

As restrições permitem oferecer uma experiência de utilizador mais rica e dar ao gerador de cenários uma maior polivalência na sua oferta visual e pontos de ajuste por parte de potenciais interessados em contribuir para o desenvolvimento e utilização do gerador de cenários. As restrições são um conjunto de informação específica para um determinado jogo, e como ilustrado em JavaScript no Código Fonte 4.1, pode ser definida informação identificada na Tabela 4.1 como: o nome de cada jogo, em que neste caso o nome do jogo é alquerque (name: "alquerque"); o formato do tabuleiro em que neste caso é (5x5) (lines: 5, columns: 5); o formato das células em que neste caso é circular (format: "circle"); e o tipo de tabuleiro em que neste caso é único pois as células são todas da mesma cor (tab: "single-color"). Outras informações também podem ser definidas como: o alinhamento entre as células, se estão alinhadas ou desalinhadas entre si, em que neste caso são alinhadas (alignment: "align"); o código utilizado para identificar células vazias, em que neste caso é blank (blank: "blank") mas também poderá ser b ou nenhum código; a

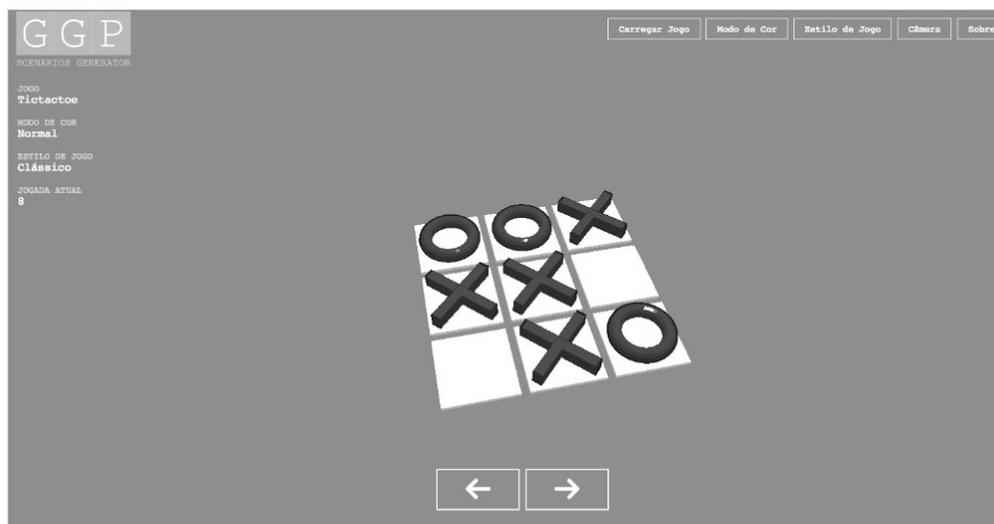
identificação dos modelos 3D das peças a usar no tema Clássico, em que neste caso têm os nomes `black` e `red` (`pieceModel: ["black", "red"]`); e as cores das peças para cada jogador no Modo de Cor Não daltónico, em que neste caso as cores são a variável `black` para a peça com o nome `black` e a variável `red` para a peça com o nome `red`, em que estas variáveis contêm a informação da cor em hexadecimal. Para cada jogo pode ser criado um ficheiro de Restrições no formato JavaScript, que o gerador de cenários reconhece caso esteja a correr o jogo em questão, caso o ficheiro de restrições específico de um determinado jogo não exista, é usado um ficheiro de restrições que tem em conta restrições típicas de jogos genéricos (e.g., tabuleiro com uma só cor e peças de cada jogador com cores distintas).

```
1:   game = {
2:     name: "alquerque",
3:     format: "circle",
4:     alignment: "align",
5:     blank: "blank",
6:     tab: "single-color",
7:     lines: 5,
8:     columns: 5,
9:     pieceModel: ["black", "red"],
10:    pieceColor: ["black", black], ["red", red]
11:  };
```

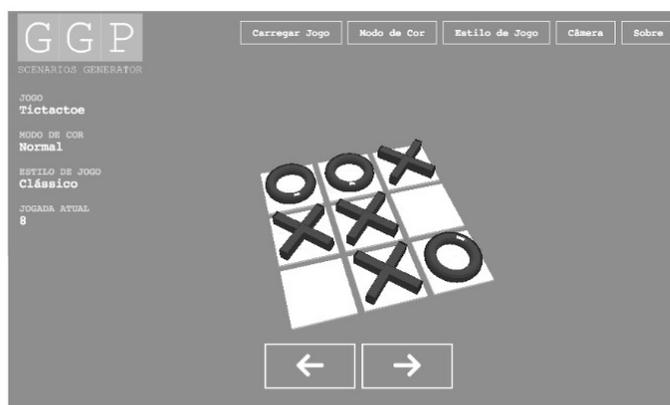
CÓDIGO FONTE 4.1: Restrições do jogo *Alquerque*, JavaScript.

4.7 Adaptação do Gerador ao Dispositivo

Para a aplicação de design responsivo foi necessário implementar a organização para cada tamanho de ecrã identificado, como ilustrado na Figura 4.16, bem como ter em conta as interações inerentes a cada dispositivo: rato e toque (mais detalhado na Secção 3.4). Na implementação do design responsivo foi criado um conjunto de classes no ficheiro HTML principal do gerador de cenários, que posteriormente são estilizadas numa folha de estilos. Na folha de estilos existem regras CSS que são aplicadas consoante a largura do dispositivo final através de *media queries*.



(a) Desktop



(b) Tablet



(c) Smartphone

FIGURA 4.16: Comportamento responsivo do gerador em cada dispositivo.

Exemplo mais notório da implementação do design responsivo é ilustrado na Figura 4.16 em que a apresentação e interação do menu, que em desktop (a) e tablet (b) surge no topo direito com todas as suas opções diretamente acessíveis, e em smartphone (c), por uma questão de hierarquia e aproveitamento de espaço, o menu é apresentado como botão que ao ser clicado abre um painel, com todas as opções de menu. Todos os outros elementos partilham um posicionamento comum, como o logotipo alinhado no topo esquerdo, a informação do jogo surge abaixo do logotipo e os controlos de navegação surgem na base do ecrã, mas graças ao CSS, são aplicadas regras que alteram o tamanho do texto para o tamanho adequado de acordo com o ecrã do dispositivo e através do Three.js os tabuleiros dos jogos genéricos ficam sempre centrados e ajustados ao tamanho do ecrã.

Capítulo 5

Testes e Análise de Resultados

5.1 Testes de Utilizador

O objetivo principal desta fase de testes foi verificar o comportamento de vários utilizadores ao interagirem com o gerador de cenários, permitindo a identificação de possíveis pontos de melhoria. Estes pontos de melhoria podem ser de diversos tipos, como o funcionamento de um botão não ser o esperado ou existir a necessidade de adicionar uma nova funcionalidade.

Para acompanhar os testes foi criado um questionário, onde são indicados pontos específicos que se pretende testar no gerador de cenários, contribuindo para uma análise concreta e técnica dos resultados. Para a sua criação seguiu-se o formato USE (Utilidade, Satisfação e Facilidade de utilização), desenvolvido por Arnold Lund (2008). Este formato mede a usabilidade de software em geral, tendo sido escolhido dado o enquadramento do gerador de cenários e objetivo que se pretende para o questionário. O questionário aborda o interface, a satisfação e desempenho do gerador de cenários. São incluídos dois grupos adicionais, um para participantes com daltonismo e outro para os participantes da área de tecnologias, que visa perceber o interesse em contribuir para a área dos Jogos Genéricos.

Entende-se que para obter *feedback* construtivo nos aspetos do funcionamento e apresentação que definem o gerador, que o público alvo desta fase de testes deve incluir pessoas de diversas áreas, dando relevância a pessoas com conhecimentos de informática ou experiência regular em novas tecnologias. Acredita-se que a sua experiência originou respostas mais construtivas e objetivas. Este perfil de participante também é o que aparenta poder ter mais interesse em contribuir para a área de Jogos Genéricos. Como público alvo definiu-se estudantes e trabalhadores, procurando participantes inseridos na área de tecnologias. Os testes foram realizados num computador portátil com 13 polegadas de ecrã e

um rato com fio, dado ser uma típica ferramenta de trabalho e lazer do público alvo. O gerador de cenários foi apresentado no *browser* Google Chrome.

De acordo com Laura Faulkner (2003), numa fase de testes de usabilidade é possível, com trinta participantes, identificar a maioria dos problemas de usabilidade mais relevantes. Como tal definiu-se a realização de trinta testes de forma a assegurar a maior qualidade possível dos resultados obtidos.

5.2 Perfil dos Participantes

Realizaram-se trinta testes, dos quais três foram com utilizadores com a condição de daltonismo. O perfil dos participantes são categorizados de acordo com a Tabela 5.1, em que dos trinta participantes, onze são do género feminino e dezanove do masculino. Nove dos participantes têm habilitações literárias de ensino secundário, onze têm habilitações literárias de licenciatura e dez têm habilitações literárias de mestrado. Dos 30 participantes, doze são estudantes e dezoito são trabalhadores, havendo dezasseis participantes trabalhadores ou estudantes das áreas das tecnologias de informática. As idades dos participantes estão compreendidas entre os dezoito e quarenta e sete anos.

TABELA 5.1: Perfil dos participantes em que D^* significa participantes com a condição de daltonismo, *Tecnologias* significa trabalhadores ou estudantes das tecnologias de informática, e T^* é relativo ao somatório de cada coluna excepto no caso das idades em que é a média.

Nrº	F	M	Secundário	Licenciatura	Mestrado	Estudante	Trabalhador	Tecnologias	Idade	D^*
1	•		•			•			20	
2	•		•				•		43	
3		•	•				•		35	
4		•		•		•			24	
5		•	•				•		45	
6	•		•				•	•	38	
7		•		•		•		•	23	•
8	•			•		•		•	23	
9		•		•		•			24	
10		•	•			•			19	•
11		•			•		•	•	27	•

Continua na próxima página

TABELA 5.1: Continuação da página anterior

12		•		•		•		•	24	
13		•			•		•	•	31	
14	•				•		•		37	
15	•		•			•			18	
16	•				•		•	•	37	
17		•			•		•	•	41	
18		•		•			•	•	25	
19		•		•			•	•	28	
20		•		•		•		•	23	
21		•		•		•			24	
22	•		•				•		35	
23		•			•		•	•	26	
24		•		•		•		•	24	
25		•			•		•	•	29	
26		•			•		•	•	33	
27	•				•		•		47	
28	•			•			•		35	
29		•			•		•	•	41	
30	•		•			•			19	
T*	11	19	9	11	10	12	18	16	29,9	3

5.3 Questionário

A realização dos testes de utilizador foi individual e sem conhecimento prévio do gerador de cenários, bem como os seus objetivos. O questionário de experiência de utilização foi iniciado com uma breve descrição do gerador de cenários. Todos os participantes foram contextualizados sobre os jogos genéricos disponíveis no gerador de cenários, informados sobre o objetivo do mesmo, e sobre certas funcionalidades que poderiam explorar no gerador de cenários como escolher um jogo de tabuleiro, alterar o estilo do jogo e caso tenha daltonismo, poderiam alterar o modo de cor para o seu tipo de daltonismo. Esta Secção está dividida em três subsecções, a Subsecção 5.3.1 corresponde ao desempenho dos participantes ao realizar tarefas propostas, a Sub-secção 5.3.2 corresponde à satisfação do participante quanto à interface gráfica e componente interativa do gerador de cenários, e a Sub-secção 5.3.3 corresponde ao interesse em contribuir para a área dos jogos genéricos por parte de utilizadores das áreas de tecnologias de informática.

5.3.1 Teste de desempenho

As questões relativas ao teste de desempenho foram colocadas sob a forma de desafios com o intuito de testar se os utilizadores conseguiam desempenhar determinadas tarefas que estão ligadas aos objetivos do gerador de cenários (que nas questões ou afirmações é descrito como aplicação devido à melhor compreensão por parte de utilizadores pouco familiarizados com outros possíveis termos). O tempo de resposta a cada questão é medido numa escala de intervalos de tempo, sendo estes de 0 a 10 segundos, de 10 a 20 segundos, de 20 a 30 segundos, de 30 a 40 segundos, de 40 a 50 segundos, mais do que 50 segundos e não efetuado caso o participante não consiga realizar a tarefa. Existe também um campo de observações para cada questão onde são anotadas informações relevantes na execução da resposta.

As sete questões iniciais têm como objetivo testar o fator intuitivo do gerador de cenários, e identificar o tempo que o participante leva a realizar cada tarefa (caso seja concluída) pela primeira vez, sendo que os participantes apenas tiveram acesso ao gerador de cenários entre 1 a 3 segundos antes de ser colocada a primeira questão. Estas sete questões também têm como objetivo dar a conhecer ao participante todas as funções do gerador de cenários. A oitava, nona e décima questões proporcionam desafios que englobam conjuntos de tarefas já realizadas nas questões anteriores. Desta forma avalia-se se os participantes aprenderam corretamente as funcionalidades testadas nas primeiras sete questões. As questões foram colocadas pela seguinte ordem:

1. Agora que pode explorar a aplicação, consegue indicar qual o nome do jogo?
2. Imagine que gostaria de ver o desenrolar do jogo, qual a ação que escolheria?
3. Nesta fase, consegue indicar-me qual o número da jogada?
4. Suponhamos que quer visualizar o jogo de outro ponto de vista no espaço, o que faria?
5. Imaginando que é um utilizador com daltonismo e tem dificuldades em diferenciar as peças dos diferentes jogadores, o que faria?
6. Sabendo que tem a possibilidade de alterar o estilo do jogo, como o alterava?
7. Agora suponhamos que deseja visualizar outro jogo, o que faria?

8. Tem uma tarefa em mãos, alterar o Estilo do jogo para City, escolher o jogo com o nome “Breakthrough Small”, e visualizar o desenrolar do jogo. Reconhece este jogo?
9. Última tarefa, procurar saber como está o jogo na jogada 22. Quantas peças tem cada jogador?
10. Mude para o tipo de daltonismo que tem, escolha o jogo “Connectfour” e procure a jogada nº 10. Quantas peças tem cada jogador?

A questão número 1 tem como objetivo que o utilizador observe o gerador de cenários e procure a informação relativa ao nome do jogo, em que a resposta correta é *alquerque* (o jogo apresentado por omissão no gerador de cenários dado ser o primeiro alfabeticamente). A questão número 2 tem como objetivo que o utilizador procure botões que desempenham a ação proposta, de ver o desenrolar do jogo, em que a ação correta é clicar nos botões do rodapé que contêm as setas para mudar de jogada do jogo. A questão número 3 tem como objetivo que o utilizador procure a informação relativa ao número da jogada em que se encontra o jogo depois de ter efectuado a ação da questão número 2, em que a resposta correta é o número da jogada em que se encontra o jogo. Nesta fase supõe-se que o participante começa a perceber que a informação do lado superior esquerdo do gerador de cenários é complementar ao jogo em questão. A questão número 4 tem como objetivo que o utilizador observe o gerador de cenários e procure botões que desempenham a ação proposta, de visualizar o jogo de outro ponto de vista no espaço, visto este ser 3D, em que existem duas ações corretas como utilizar o rato e arrastá-lo sobre o ambiente 3D ou clicar no botão câmara, contido no menu do gerador de cenários, que atualiza a posição da câmara para outra posição no espaço, ou seja, o ponto de vista é alterado. A escolha entre uma das duas ações corretas (utilização do rato ou do botão câmara), por parte do participante, é anotada no campo das observações com o objetivo de se estimar qual delas é a mais intuitiva.

A questão número 5 tem como objetivo que o utilizador observe o gerador de cenários e procure o botão que desempenha a ação proposta, mudar o modo de cor do jogo, em que a ação correta é clicar no botão “modo de cor”, contido no menu do gerador de cenários, e escolher uma das opções protanopia, deuteranopia, tritanopia ou monocromacia. A questão número 6 tem como objetivo que o utilizador procure o botão que desempenha a ação proposta, de alterar o estilo do jogo, em que a ação correta é clicar no botão “estilo de jogo”, contido no menu do gerador de cenários, e escolher uma das opções clássico, floresta, cidade,

campo ou selva. A questão número 7 tem como objetivo que o utilizador procure o botão que desempenha a ação proposta, de escolher outro jogo, em que a ação correta é clicar no botão “carregar jogo”, contido no menu do gerador de cenários, e escolher um dos jogos disponíveis.

A questão número 8 é constituída por um conjunto de três questões (nomeadamente a 3, 6 e 7) e tem como objetivo que o utilizador desempenhe as ações propostas nestas questões, em que a resposta final correta é “Jogo das Damas”. Para o utilizador chegar à resposta correta sem ver o jogo no estilo clássico, pois considera-se bastante realista e fácil de identificar o jogo, é proposta a primeira tarefa da questão que é alterar o estilo do jogo para city, desta forma quando realizar a segunda tarefa que é escolher o jogo com o nome *Breakthrough* (jogo genérico equivalente ao jogo das damas) não irá identificar o jogo, apenas com a terceira tarefa, que é ver o desenrolar do jogo, o participante conseguirá, através das posições das peças e da forma como as peças interagem entre si (representação das regras), identificar o jogo e responder corretamente à questão. Nesta questão, para além da contagem dos segundos que o participante leva a completar as três tarefas, também é identificada informação acerca do participante chegar ou não a reconhecer o jogo ou se demonstra reconhecer o jogo através do nome “Breakthrough Small”, que não é comum em linguagem corrente, mas bastante comum na área dos jogos genéricos.

A questão número 9 é uma continuação da questão número 8 e tem como objetivo que o utilizador procure percorrer as jogadas do jogo “Breakthrough Small”, no estilo do jogo para “city”, até encontrar a jogada número 22 e em seguida identificar o número de peças que cada jogador tem. Nesta questão é testada a facilidade de identificar diferentes peças mesmo quando estas não possuem a representação visual clássica. Nesta questão, para além da contagem do tempo que o participante leva a completar a tarefa, também é identificada informação acerca do participante acertar ou não no número de peças de cada jogador e conseguir ou não distinguir as peças.

A questão número 10, que apenas é apresentada a participantes com daltonismo, é semelhante à questão 8 pois também contém um conjunto de três questões (nomeadamente a 3, 5 e 7) e tem como objetivo que o utilizador desempenhe as ações propostas nestas questões. Para o participante chegar à resposta correta é proposta a primeira tarefa da questão, que é alterar o tipo de daltonismo para o que o participante tem. Em seguida é proposta a segunda tarefa da questão que é escolher o jogo “Connect four”. Por fim, a terceira tarefa da questão é

procurar a jogada número 10 deste jogo. O objetivo desta questão é testar se o participante consegue distinguir as peças de cada jogador, pois no final das três tarefas é-lhe questionado o número de peças que cada jogador tem. Nesta questão, para além da contagem dos segundos que o participante leva a completar as três tarefas, também é identificada informação acerca do participante acertar no número das peças de cada jogador ou se não é capaz de distinguir as peças.

5.3.2 Teste de satisfação

Para além do teste de desempenho, procurou-se recolher informação quanto ao grau de satisfação de uma perspectiva de usabilidade, por parte dos participantes, após interagirem com o gerador de cenários. Como tal, foi criada uma questão número 11, com afirmações relativas ao grau de satisfação do participante, em que é pedido aos participantes para avaliarem, sob forma de pontuação, a interface gráfica do gerador de cenários usando a escala de Likert (1932), em que 1 significa que o participante discorda totalmente e 5 significa que o participante concorda totalmente:

1. Consigo utilizar a aplicação sem dificuldades;
2. Aprendi a utilizar a aplicação rapidamente;
3. Considero as funcionalidades disponíveis úteis;
4. As cores do jogo são adequadas;
5. O tamanho dos botões é adequado;
6. O posicionamento dos botões é o ideal;
7. A navegação é fácil de perceber sem instruções (intuitiva);
8. A informação é legível.

Esta questão tem como objetivo medir a facilidade dos participantes aprenderem a utilizar o gerador de cenários de forma intuitiva (como nas afirmações 1, 2 e 7), medir o grau de satisfação quanto às cores, tamanho e posições dos elementos gráficos do gerador de cenários (como nas afirmações 4, 5, 6 e 8), e medir se as funcionalidades disponíveis no gerador de cenários são úteis segundo os participantes (como na afirmação 3). Neste teste de satisfação também foi criada uma questão número 12 de resposta livre: “Que alterações

sugere que fossem feitas na aplicação?” Esta questão tem o intuito de recolher sugestões dos participantes com o objetivo de melhorar o gerador de cenários.

5.3.3 Interesse em contribuir para a área dos Jogos Genéricos

Ao conjunto de participantes com perfil de trabalhador ou estudante nas área das tecnologias informáticas foi colocada a questão número 13 de resposta livre: “Depois de ter interagido com o gerador de cenários para jogos genéricos, e sabendo que: bastará criar um simples conjunto de dados XML com o histórico de um jogo e um curto conjunto de dados JavaScript com as restrições do jogo; o gerador de cenários produz o tabuleiro, o ambiente envolvente e aplica as peças, cores, formas. Teria interesse em criar novos jogos de tabuleiro com as suas próprias regras e características?”. Esta questão tem o objetivo de responder a uma questão de investigação da Secção 1.3 “3. Até que ponto alguém, na área das tecnologias, teria interesse em desenvolver novos jogos de tabuleiro?”.

5.4 Análise dos Resultados

No decorrer da fase de testes notou-se quase imediatamente que o gerador de cenários é bastante intuitivo e no final de cada teste todos os participantes conseguiram desempenhar todas as tarefas propostas durante o teste ainda que com diferentes níveis de prestação. Alguns participantes têm resultados de tempo, a desempenhar tarefas, mais prolongados do que a grande maioria, podendo ser confundidos com participantes que tiveram dificuldades, mas na verdade estes participantes simplesmente não sentiram qualquer pressão durante o teste e a sua maneira de estar era calma. Estes participantes serão identificados ao longo desta secção como participantes que levaram o seu tempo a realizar as tarefas propostas. Considera-se então que os resultados em geral foram bastante positivos, sendo estes descritos ao longo desta secção.

Na questão 1, como ilustrado no gráfico da Figura 5.1 todos os participantes responderam à questão com uma duração máxima de 30 segundos. Vinte e dois dos participantes responderam em 10 segundos, enquanto que seis levaram de 10 a 20 segundos a responder e dois levaram de 20 a 30 segundos a responder. Dos oito participantes que não responderam em 10 segundos um deles pensou que era suposto descobrir o nome do jogo ao olhar para as peças e tabuleiro; três leram o nome do gerador de cenários “GGP Scenarios

Generator” e quatro levaram o seu tempo a analisar o ecrã. Acredita-se que estes resultados devem-se ao facto de os participantes apenas terem visto o gerador de cenários pela primeira vez de um a três segundos antes de ter sido feita esta questão. Conclui-se que os participantes conseguiram desempenhar a tarefa pedida com sucesso, validando que a apresentação criada no gerador de cenários para a informação relativa ao nome do jogo é legível e está localizada apropriadamente.

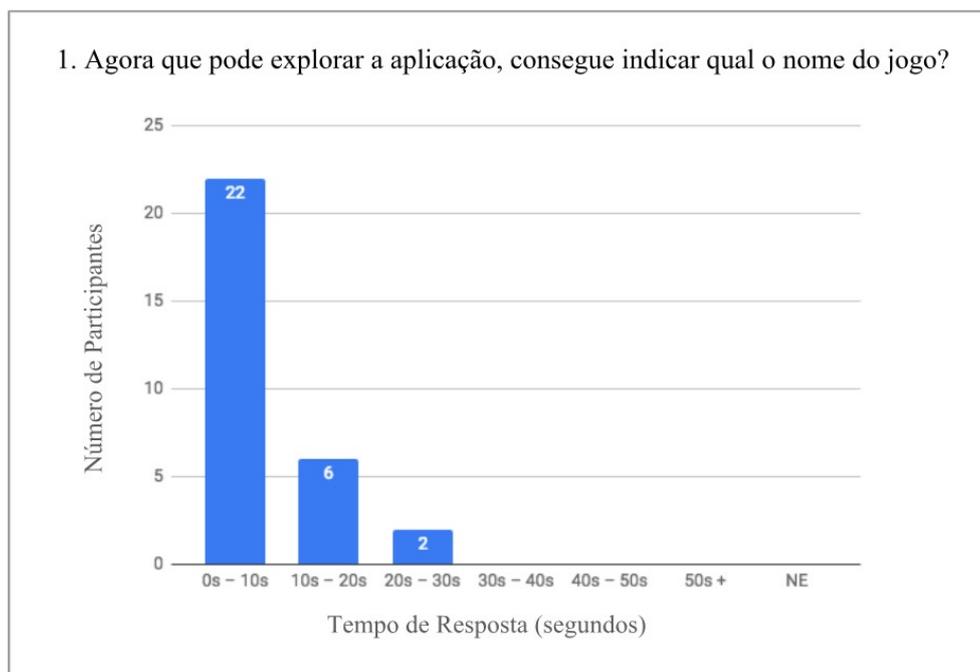


FIGURA 5.1: Resultados da questão 1.

Na questão 2, como ilustrado no gráfico da Figura 5.2, todos os participantes responderam à questão com uma duração máxima de 50 segundos. Vinte e quatro dos participantes responderam em 10 segundos e os seis restantes levaram de 10 a 50 segundos. Dos seis participantes que não responderam em 10 segundos, três não repararam nos botões do rodapé em tempo esperado e três continuaram a ler a área de informação de jogo (a mesma que continha a resposta da questão anterior) pois assume-se que pensaram que a resposta estava lá. Conclui-se que em geral os participantes conseguiram desempenhar a tarefa pedida com sucesso, validando a interação definida para navegar entre jogadas do jogo.

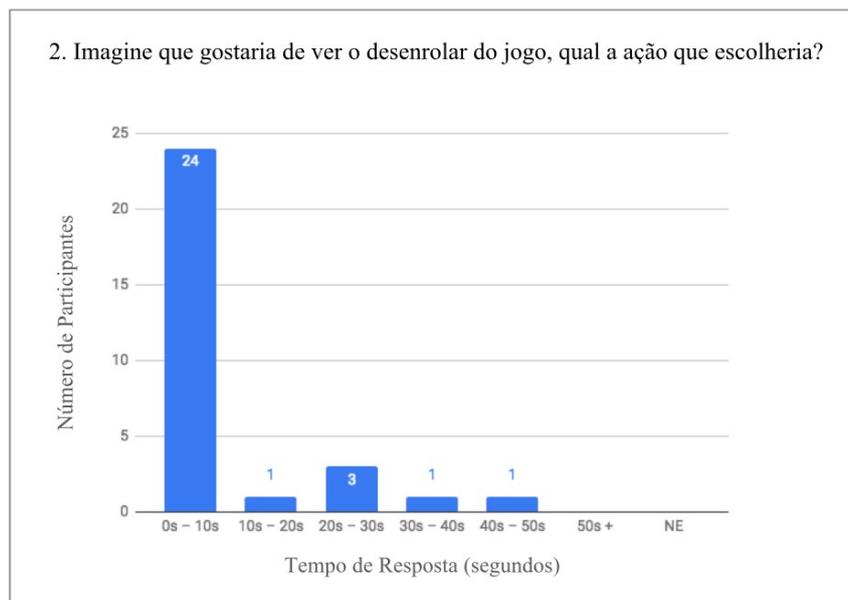


FIGURA 5.2: Resultados da questão 2.

Na questão 3, como ilustrado no gráfico da Figura 5.3, todos os participantes responderam à questão com uma duração máxima de 20 segundos. Vinte e cinco dos participantes responderam em 10 segundos e os cinco restantes levaram de 10 a 20 segundos a responder. Dos cinco participantes que não responderam em 10 segundos, três referiram que não tinham contado as jogadas ao navegar entre elas para responder à questão anterior, mas todos acabaram por perceber que a resposta à questão estaria na área de informação de jogo. Conclui-se que os participantes conseguiram desempenhar a tarefa proposta, validando o bloco de informação definido para apresentar a informação do jogo.

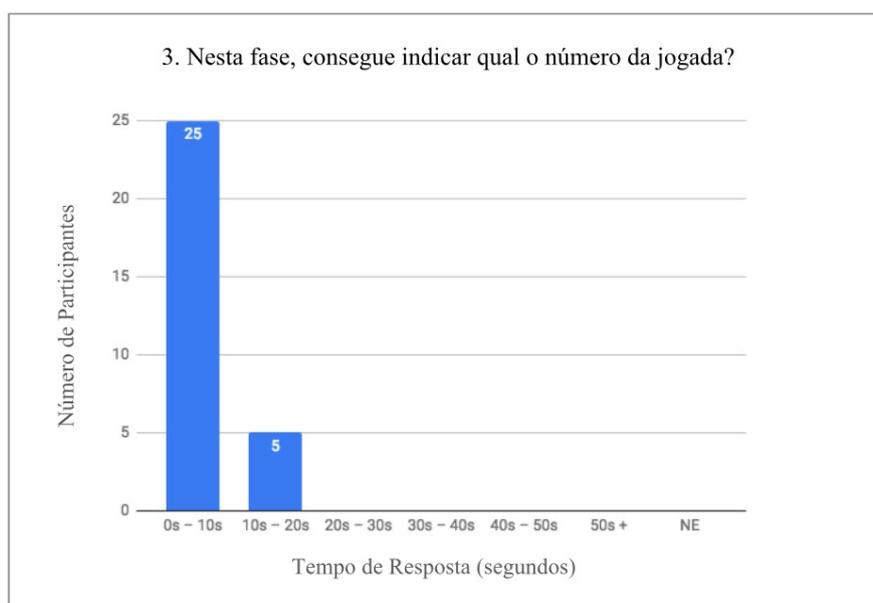


FIGURA 5.3: Resultados da questão 3.

Na questão 4, como ilustrado no gráfico da Figura 5.4, todos os participantes responderam à questão com uma duração máxima de 40 segundos. Vinte e cinco dos participantes responderam em 10 segundos, enquanto os cinco restantes levaram de 10 a 40 segundos a responder. Dezanove dos participantes escolheram o botão “Câmera” para responder à questão, enquanto onze usaram o rato. Dos cinco participantes que não responderam em 10 segundos, quatro, que após terem percebido que as respostas às questões anteriores estavam em botões ou na área informativa, releeram a área informativa e o menu até encontrar o botão “Câmera” que é o último. Um dos participantes clicou com o rato sobre o tabuleiro algumas vezes antes de compreender que teria de arrastar o rato sobre o tabuleiro. Conclui-se que no geral os participantes conseguiram realizar a tarefa proposta com sucesso, validando as ferramentas definidas para interagir com a câmara virtual do jogo.



FIGURA 5.4: Resultados da questão 4.

Na questão 5, como ilustrado no gráfico da Figura 5.5, todos os participantes responderam à questão com uma duração máxima de 20 segundos. Vinte e um dos participantes responderam em 10 segundos, enquanto os nove restantes levaram de 10 a 20 segundos a responder. Os nove participantes que não responderam em 10 segundos releeram o menu até encontrarem o “Modo de cor” e de seguida leram as opções “Protanopia”, “Deuteranopia”, “Tritanopia”, “Monocromacia” e “Não daltónico” antes de escolher uma. O tempo total desta questão inclui a ação de escolher uma das opções. Conclui-se que os

participantes conseguiram desempenhar a tarefa proposta com sucesso, validando a apresentação definida para o menu e a sua lógica de interação e hierarquia, ao apresentar o 2º nível de menu, que contém as opções, quando se coloca o rato sobre o 1º nível respectivo.

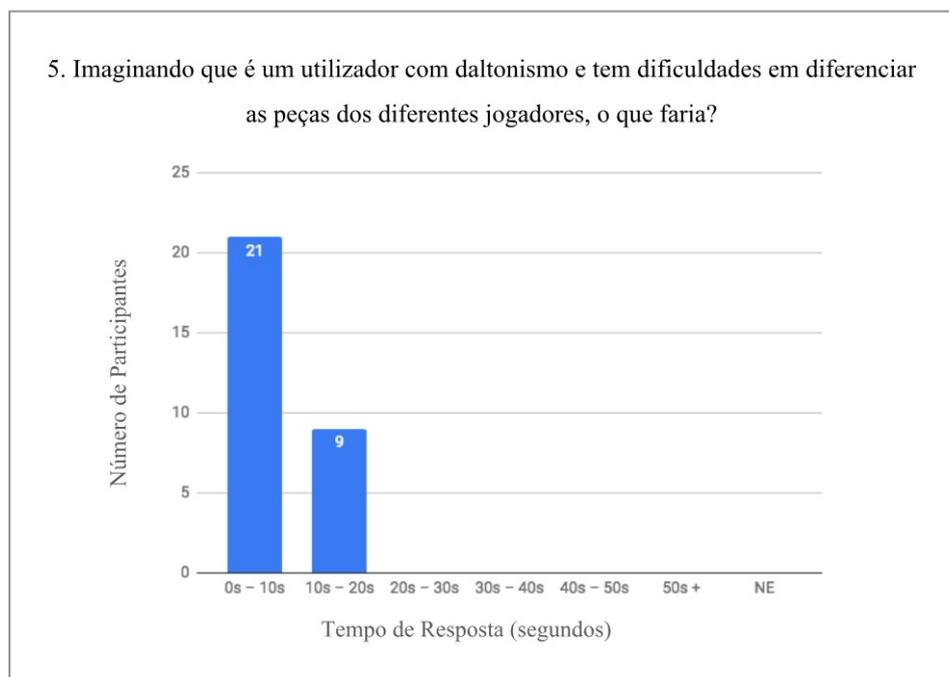


FIGURA 5.5: Resultados da questão 5.

Na questão 6, como ilustrado no gráfico da Figura 5.6, todos os participantes responderam à questão com uma duração máxima de 20 segundos. Vinte e sete dos participantes responderam em 10 segundos, enquanto os três restantes levaram de 10 a 20 segundos a responder. Os três participantes que não responderam em 10 segundos voltaram a ter o mesmo comportamento que nove participantes tiveram na questão anterior, ou seja, releeram o menu até encontrarem o “Estilo de Jogo” e ao clicar neste botão leram as opções “Clássico”, “Floresta”, “Cidade”, “Campo” e “Selva” antes de escolher uma das opções. O tempo total desta questão inclui a ação de escolher uma das opções.

Assume-se que grande parte dos utilizadores foram mais rápidos na questão 6 e não na anterior pois já tiveram uma experiência parecida ao responder à questão anterior. Conclui-se que os participantes conseguiram realizar a tarefa proposta com sucesso, que à semelhança da questão 5, valida a apresentação definida para o menu e a sua lógica de interação e hierarquia.

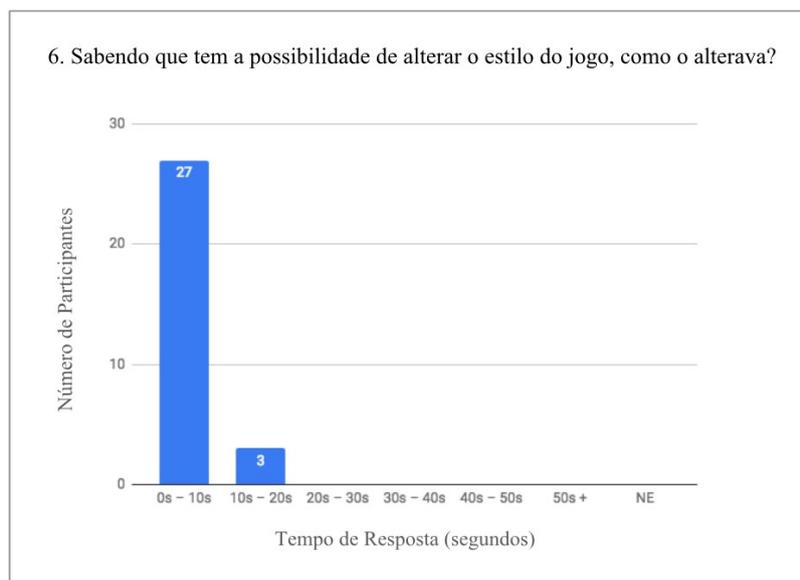


FIGURA 5.6: Resultados da questão 6.

Na questão 7, como ilustrado no gráfico da Figura 5.7, todos os participantes responderam à questão com uma duração máxima de 30 segundos. Vinte e seis dos participantes responderam em 10 segundos, e os quatro restantes levaram de 10 a 30 segundos a responder. Os quatro participantes que não responderam em 10 segundos levaram mais tempo pois não reconheciam o nome da grande maioria dos jogos genéricos e tiveram em momentos de indecisão ao escolher o jogo. O tempo total desta questão inclui a ação de escolher um dos jogos. Conclui-se que os participantes conseguiram realizar a tarefa proposta com sucesso, que à semelhança das questões 5 e 6, permite validar a apresentação definida para o menu e a sua lógica de interação e hierarquia.

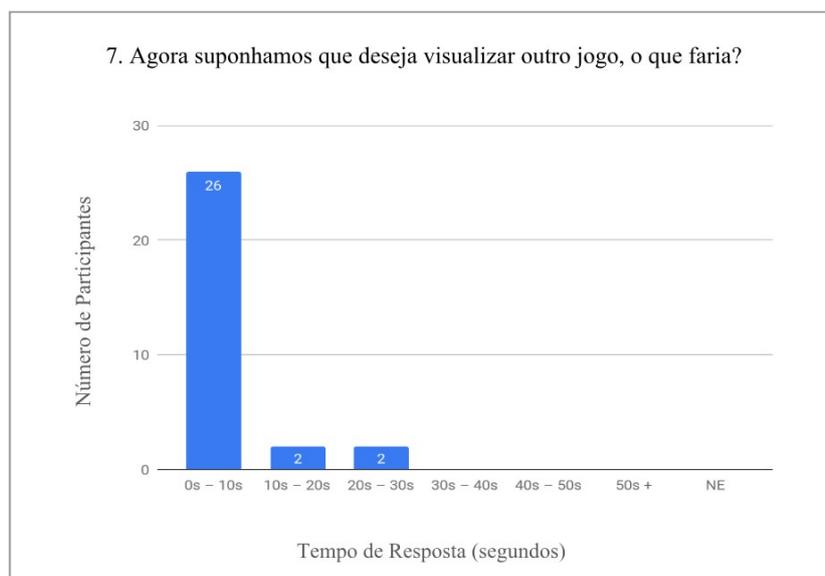


FIGURA 5.7: Resultados da questão 7.

Na questão 8, como ilustrado no gráfico da Figura 5.8, todos os participantes responderam com uma duração entre 10 a 40 segundos. Dezasseis dos participantes responderam entre 10 a 20 segundos, enquanto dez participantes levaram de 20 a 30 segundos a responder e os restantes quatro responderam entre 30 a 40 segundos. Todos os participantes identificaram o jogo ao ver o desenrolar do jogo. Nenhum dos participantes respondeu em 10 segundos pois tinham uma tarefa mais complexa em relação às questões anteriores. Esta questão contém três tarefas numa só e os catorze participantes que levaram entre 20 a 40 segundos a responder não tiveram quaisquer dificuldades, apenas levaram o seu tempo a ler o menu e a lista de jogos para encontrar o jogo “Breakthrough Small”.

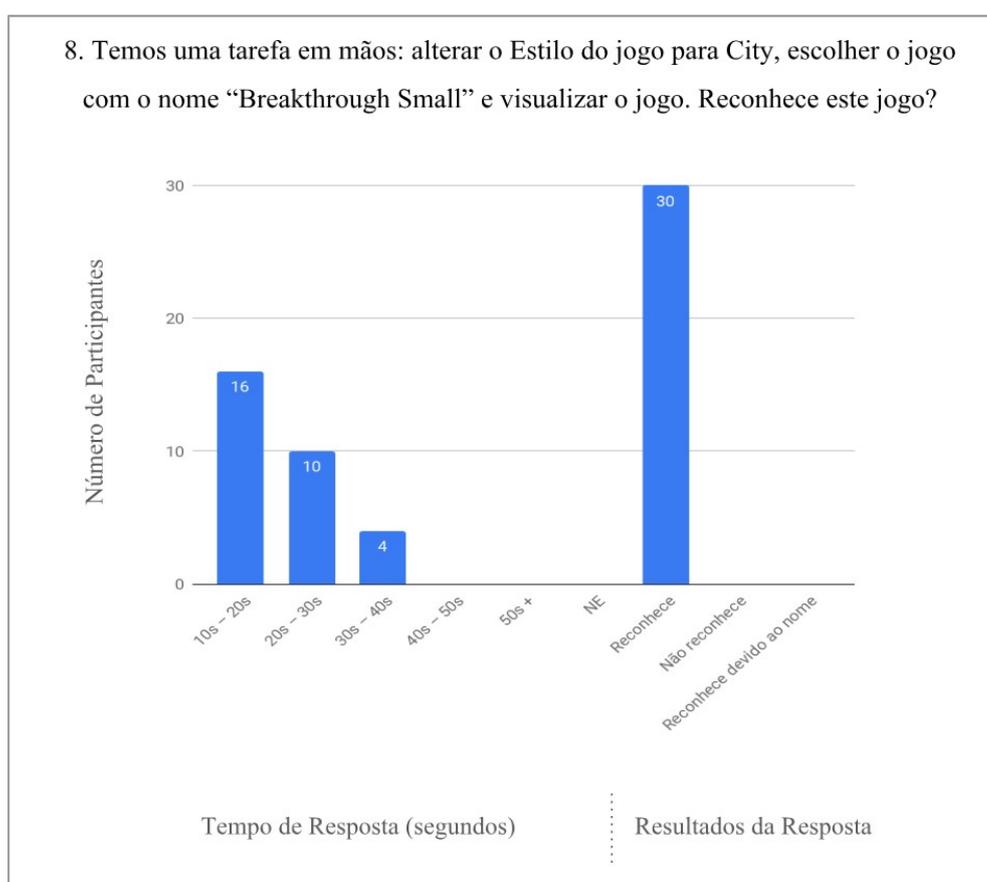


FIGURA 5.8: Resultados da questão 8.

A questão 8 encadeia um conhecimento adquirido através de tarefas anteriormente realizadas nas questões 3, 6 e 7. Os resultados obtidos destas três questões foram de 0 a 10 segundos pela maioria dos participantes, logo assume-se que o somatório do tempo para realizar a questão 8, seria 30 segundos, excluindo o tempo para reconhecer o jogo ao observar os movimentos das peças. Como já referido, dezasseis dos participantes concluíram a questão em 20 segundos, e dez concluíram em 30 segundos, assim concluí-se que os participantes

conseguiram realizar a tarefa proposta com sucesso, demonstrando a facilidade de aprendizagem que o gerador de cenários oferece na sua interação.

Na questão 9, como ilustrado no gráfico da Figura 5.9, todos os participantes responderam com uma duração máxima de 30 segundos. Vinte dos participantes responderam entre 10 a 20 segundos, enquanto seis participantes obtiveram melhores resultados respondendo em 10 segundos, e os quatro restantes levaram de 20 a 30 segundos a responder. Os quatro participantes que não responderam em 20 segundos levaram o seu tempo a contar as peças. O tempo total desta pergunta inclui a ação de contar as 19 peças que estavam em jogo e identificar quantas eram de cada jogador. Todos os participantes reconheceram e diferenciaram as peças de cada jogador. Conclui-se que os participantes conseguiram alcançar com sucesso a tarefa proposta, que requer que estes se foquem num conjunto distinto de elementos da interface gráfica, como aconteceria numa interação natural com o gerador de cenários, verificando-se que a identificação pedida é possível graças à organização e apresentação da interface gráfica.

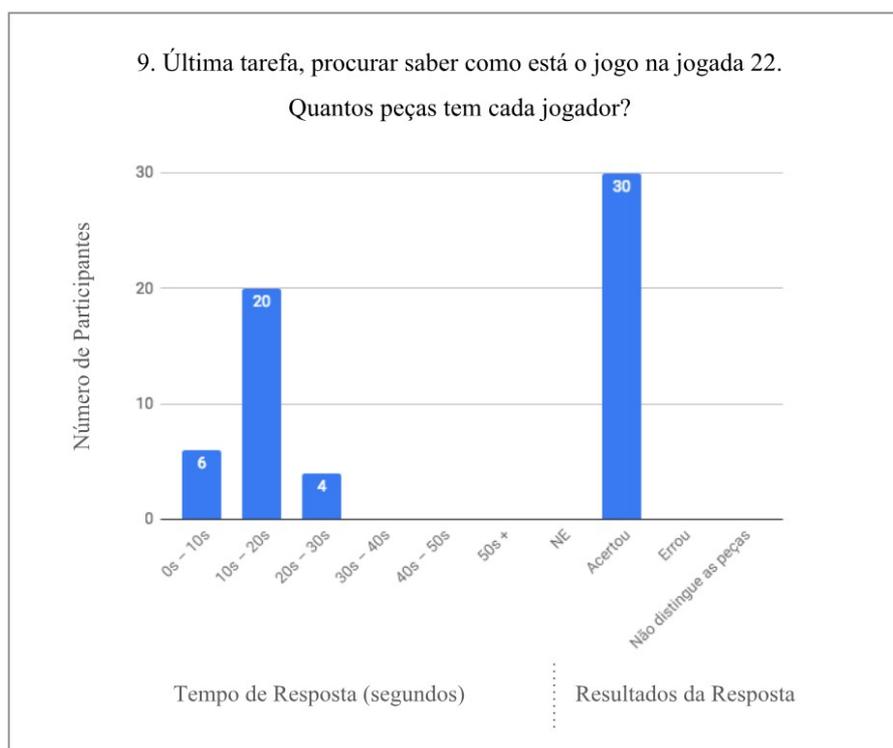


FIGURA 5.9: Resultados da questão 9.

A questão 9 encadeia um conhecimento adquirido através de tarefas anteriormente realizadas nas questões 2 e 3. Os resultados obtidos destas duas questões foram de 0 a 10

segundos pela maioria dos participantes, logo, assume-se que a soma do tempo para realizar a questão 9 seria 20 segundos, excluindo o tempo para contar as peças de cada jogador. Como já referido, vinte dos participantes concluíram a questão em 20 segundos, assim conclui-se que os participantes conseguiram realizar a tarefa proposta com sucesso, demonstrando a facilidade de aprendizagem que o gerador de cenários oferece na sua interação.

Na questão 10, que foi colocada apenas aos participantes com daltonismo, como ilustrado no gráfico da Figura 5.10, todos os participantes responderam com uma duração entre 10 a 30 segundos. Dois dos participantes responderam entre 10 a 20 segundos e um participante levou de 20 a 30 segundos a responder. Os três participantes reconheceram e diferenciaram as peças de cada jogador. Esta pergunta contém quatro tarefas numa só, e um dos participantes levou entre 20 a 30 segundos a responder pois consumiu mais tempo ao avançar 10 jogadas e contar as peças. Todos os participantes conseguiram distinguir as peças com sucesso, conclui-se assim que as paletas de cor utilizadas no gerador de cenários são apropriadas, validando o mecanismo desenhado e implementado no gerador de cenários para utilizadores com daltonismo.

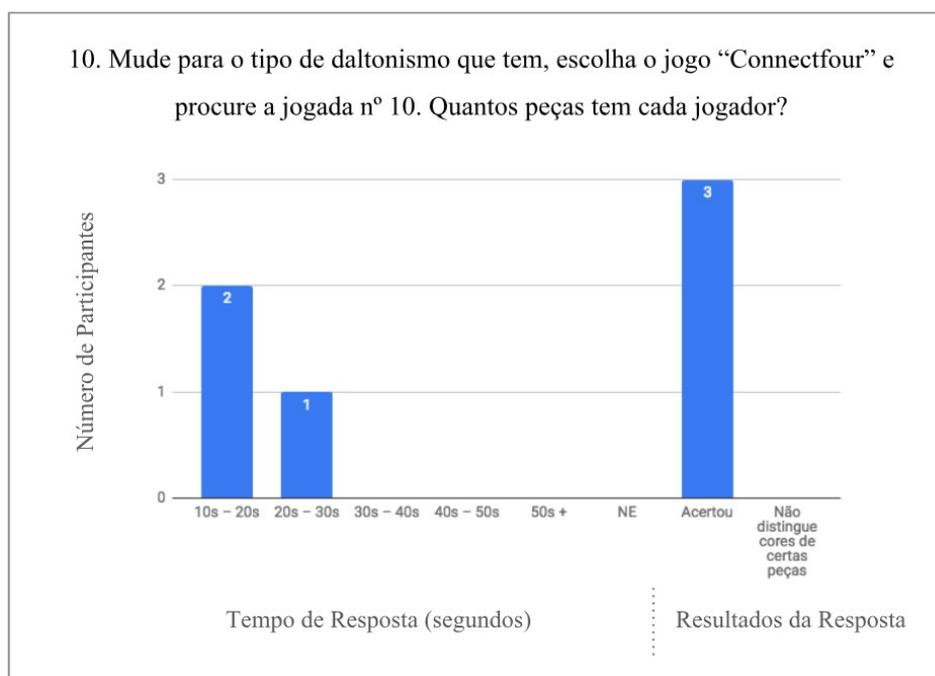


FIGURA 5.10: Resultados da questão 10.

Relembrando que esta questão é semelhante à questão 8, excepto o facto de que na questão 8 a funcionalidade de modo de cor é Não daltónico, comparou-se os resultados destes três participantes nas questões 8 e 10. Dois dos participantes obtiveram resultados iguais nas

duas questões, nomeadamente de 10 a 20 segundos e de 20 a 30 segundos, e um dos participantes obteve resultados diferentes nas duas questões, nomeadamente de 20 a 30 segundos na questão 8 e de 10 a 20 segundos na questão 10. Assim, conclui-se que a funcionalidade de modo de cor com o tipo de daltonismo correto é importante para o desempenho de utilizadores com a condição de daltonismo, convém no entanto salientar que o modo de cor Não daltónico também pode ser utilizado por utilizadores com esta condição.

A questão 10 encadeia um conhecimento adquirido através de tarefas anteriormente realizadas nas questões 3, 5 e 7. Os resultados obtidos por parte dos três participantes com a condição de daltonismo nestas três questões foram de 0 a 10 segundos para a questão 3, dois de 0 a 10 segundos e um de 10 a 20 segundos para a questão 5, e um de 0 a 10 segundos, outro de 10 a 20 segundos e ainda outro participante de 20 a 30 segundos para a questão 7. Visto também que os resultados obtidos nestas três questões, por parte dos restantes 27 participantes, foram de 0 a 10 segundos, assume-se que a soma do tempo para realizar a questão 10 seria 30 segundos, excluindo o tempo para contar as peças de cada jogador. Como já referido, dois dos três participantes concluíram a questão em 20 segundos e um concluiu a questão em 30 segundos, assim conclui-se que os participantes conseguiram realizar a tarefa proposta com sucesso, sugerindo a facilidade de aprendizagem que o gerador de cenários oferece na sua interação.

Os resultados obtidos nas questões de 1 a 10, relativas ao teste de desempenho, descritos na Secção 5.3.1, foram no geral bastante positivos dado que as questões apresentadas aos participantes foram efetuadas com sucesso. Estas questões validam os aspectos principais da interação e navegação do gerador de cenários, definidos no desenho e implementação do gerador de cenários.

Na questão 11, como ilustrado no gráfico da Figura 5.11, é pedido a cada participante para indicar na escala de Likert (1932), a sua avaliação de acordo com um conjunto de afirmações, após ter interagido com a interface gráfica do gerador de cenários. Os resultados obtidos nesta questão foram positivos pois todos os participantes responderam positivamente avaliando apenas de 3 a 5 numa escala de 1 a 5. As afirmações com respostas mais positivas foram a 3 e 4, ao verificar-se que obtiveram pontuação 5 por parte de 28 dos participantes. Conclui-se então que, em geral, os participantes consideram que as funcionalidades e cores do gerador de cenários são adequadas. Em seguida seguem-se as afirmações 1 e 7, ao verificar-se que obtiveram pontuação 5 por parte de 27 dos participantes. A afirmação 1 foi a única que

obteve pontuação 3, mas apenas por parte de um participante que sentiu certas dificuldades ao utilizar o gerador de cenários. Conclui-se que, em geral, os participantes não sentem dificuldades a utilizar o gerador de cenários e que a navegação é intuitiva. Depois segue-se a afirmação 8, ao verificar-se que obteve pontuação 5 por parte de 26 dos participantes. Conclui-se então que, em geral, os participantes consideram que a informação é legível. Em seguida seguem-se as afirmações 2 e 5, ao verificar-se que obtiveram pontuação 5 por parte de 24 dos participantes. Conclui-se que, em geral, os participantes consideram que conseguiram utilizar o gerador de cenários rapidamente e que o tamanho dos botões é adequado. Por fim, a afirmação 6, que apesar de ter obtido pontuação 5 por parte de 23 participantes, 7 dos participantes consideram que o posicionamento dos botões é o ideal mas algo pode melhorar, pois pontuaram com 4 esta afirmação. Em geral os resultados da pontuação das afirmações são positivos visto que dos 240 resultados, 207 resultados foram a pontuação 5. Conclui-se então que o grau de satisfação quanto às cores, tamanho e posições dos elementos gráficos do gerador de cenários é elevado, e as funcionalidades disponíveis no gerador de cenários são úteis segundo os participantes.

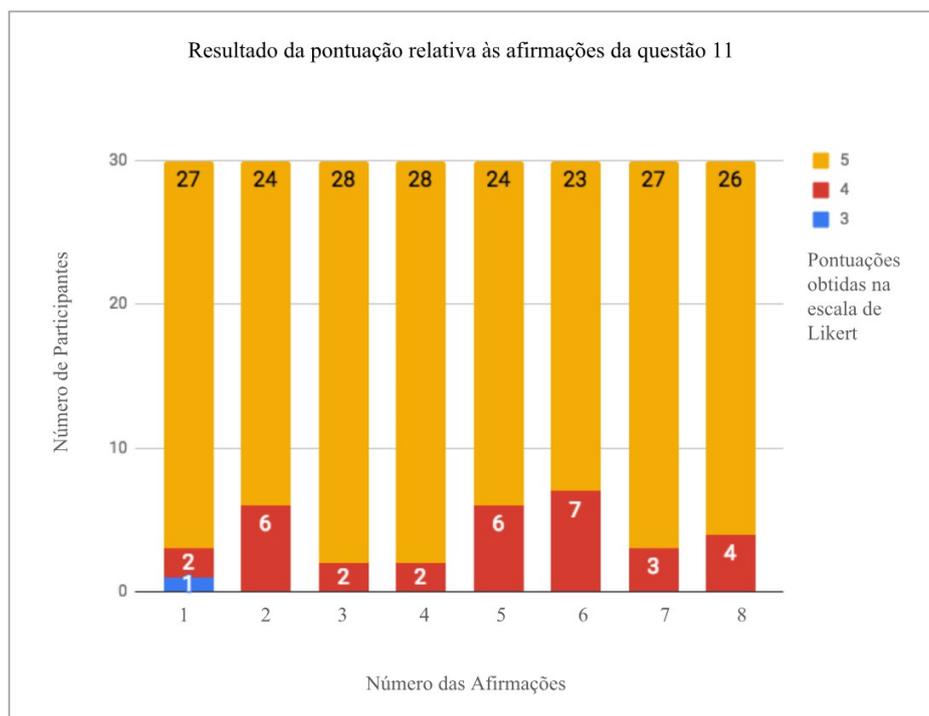


FIGURA 5.11: Resultado da pontuação relativa às afirmações da questão 11.

A questão 12 é de resposta livre e os participantes foram questionados sobre que alterações sugeriam que fossem realizadas no gerador de cenários, e foi sugerido a integração com redes sociais (1 participante), a adição de efeitos sonoros (1 participante) e a

possibilidade de alterar a língua do gerador de cenários (3 participantes). A integração com redes sociais é bastante interessante pois contribui na divulgação do gerador de cenários e os Jogos Genéricos e possibilita a atração de mais pessoas para esta área. A adição de efeitos sonoros é uma nova funcionalidade com potencial pois pode tornar a experiência de utilização do gerador de cenários mais imersiva, por exemplo ao disponibilizar música ambiente e efeitos de som para os diferentes tipos de ação que ocorrem ao longo de cada jogo. A opção para alterar a língua do gerador torna-o mais inclusivo, ao ser mais acessível para pessoas de outras línguas que não o Português. Estas sugestões serão tidas em conta para trabalho futuro.

A questão 13, em que apenas os participantes das áreas das tecnologias de informática foram inquiridos, de resposta livre, tem como objetivo descobrir o interesse em desenvolver novos jogos de tabuleiros por parte deste tipo de participantes. Obteve-se um conjunto de respostas-tipo:

- “Sim, porque assim não teria que desenvolver a parte da interface gráfica do jogo”;
- “Sim, pois gostava de trabalhar com as tecnologias usadas no gerador”;
- “Sim, parece ser um desafio interessante”;
- “Sim, porque gostava de criar os meus próprios jogos de tabuleiro e partilhar com outras pessoas”;
- “Sim, porque gosto do aspeto visual e não teria trabalho nessa parte”;
- “Sim, porque já trabalhei em projetos com inteligência artificial e os jogos genéricos são uma área interessante”.

Seis dos participantes referiram que o facto de o gerador de cenários automaticamente construir a interface gráfica do jogo facilita o esforço em criar novos jogos de tabuleiro. Nove dos participantes baseiam a sua resposta no desafio que esta área e tecnologias representa. E um participante refere que tem interesse em criar novos jogos de tabuleiro para partilhá-los. Desta forma é demonstrado que o gerador de cenários suscita interesse em pessoas das áreas das tecnologias de informática em contribuir para os Jogos Genéricos, criando novos jogos usando o gerador.

Capítulo 6

Conclusões

6.1 Conclusões Gerais

O objetivo principal desta dissertação consistiu no estudo, desenho e desenvolvimento de uma ferramenta genérica denominada Gerador Automático de Cenários para Jogos Genéricos, capaz de oferecer uma interface gráfica 3D de jogos genéricos, com uma experiência de utilizador e interação intuitivas, podendo sempre ser expandida e personalizada, adaptando-se a um grande número de jogos, devido às suas características genéricas. Com isto o utilizador tem à sua disposição ambientes gráficos 3D e mecanismos de iluminação e representação realistas, com os quais pode interagir através de controlos intuitivos.

O público alvo do gerador de cenários pode ser dividido em duas categorias, a comunidade GGP e os utilizadores comuns. No primeiro pretende-se que o gerador de cenários possa ser utilizado como uma das opções de representação dos Jogos Genéricos, suscitando o interesse no desenvolvimento e crescimento do Gerador. Pretende-se deste modo atrair utilizadores, interessados em ver agentes de jogo, num ambiente customizado pelos mesmos (utilizadores), algo que se acredita ser possível tendo em conta a abertura da comunidade no apoio a soluções que melhorem o universo dos Jogos Genéricos e os resultados obtidos na fase de testes com os participantes das áreas das tecnologias de informática. Estes participantes demonstraram interesse na área dos Jogos Genéricos, após tomarem conhecimento da mesma através do gerador de cenários e sabendo que o têm à disposição para criar novos jogos de tabuleiro. Em relação aos utilizadores comuns, foi possível observar através dos resultados obtidos na fase de testes, vários aspectos da usabilidade do gerador de cenários, nomeadamente a facilidade e satisfação oferecidos pela interação, concluindo-se que o gerador de cenários atingiu os objetivos estabelecidos. O Modo de Cor presente no gerador de cenários contribuiu para uma inclusão e acessibilidade

por parte de utilizadores com a condição de daltonismo, tendo este facto sido comprovado com os participantes daltónicos na fase de testes, traçando o caminho para que outros aspectos que melhorem a experiência de utilização possam ser desenvolvidos e adicionados ao gerador.

O gerador de cenários pode ser utilizado num vasto conjunto de dispositivos, como o smartphone, tablet e computador, graças às tecnologias web usadas no seu desenvolvimento, sendo necessário um *browser* para aceder ao gerador de cenários. As tecnologias utilizadas revelaram ser uma escolha adequada, pois possibilitaram o resultado final abordado nas questões de investigação propostas nesta dissertação.

6.2 Contribuições

Considerando as questões de investigação apresentadas, bem como os objetivos propostos por esta dissertação, identificam-se as seguintes contribuições:

1. Os utilizadores podem identificar e compreender melhor os jogos ao interagir com o gerador de cenários de acordo com a análise dos resultados obtidos na fase de testes, em que os participantes realizaram com sucesso as tarefas relacionadas com a identificação e compreensão dos jogos.
2. As representações gráficas genéricas e a liberdade de controlo da perspectiva de visualização são satisfatórias para o utilizador. Este facto baseia-se nos resultados obtidos na fase de testes em questões que pretendiam verificar estes aspectos, nomeadamente a questão 11 que teve como objetivo medir a satisfação de utilização por parte dos participantes, cujas respostas indicam um elevado nível de satisfação.
3. Verificou-se a existência de interesse na contribuição para o desenvolvimento de jogos genéricos. Para verificar esta questão foi criada, na fase de testes, a questão 13 direcionada a participantes das áreas das tecnologias de informática com o objetivo de perceber o interesse em desenvolver novos jogos neste âmbito, sabendo que tinham à disposição o gerador de cenários. As respostas de todos os participantes foram afirmativas, demonstrado interesse em desenvolver novos jogos, complementando a sua resposta com a sua motivação.

4. O gerador de cenários é genérico em comparação com aplicações semelhantes anteriores, pois analisa e constrói o ambiente de jogo e os seus elementos gráficos de forma genérica. Esta afirmação baseia-se na comparação direta com outras aplicações semelhantes já apresentadas na Secção 2.5, que quando comparadas com o gerador de cenários, este demonstra uma implementação e utilização genérica superior a aplicações semelhantes, que requerem ajustes substanciais para poderem apresentar diferentes jogos genéricos.

6.3 Trabalho Futuro

Como trabalho futuro pretende-se apresentar à comunidade GGP o gerador de cenários, com o objetivo de que este seja dado a conhecer aos intervenientes, como professores e alunos, podendo integrá-la nos seus projetos e contribuir para a sua utilização e desenvolvimento. Pretende-se também complementar a documentação do gerador de cenários com conteúdos específicos para orientar pessoas interessadas em contribuir no desenvolvimento do mesmo. Pretende-se ainda disponibilizar o código do gerador de cenários em formato livre (do inglês *open source*) numa plataforma colaborativa, como o GitHub, pois deste modo permite-se o seu crescimento através da contribuição de outros utilizadores, ficando acessível num repositório remoto que controla o seu versionamento e facilita a comunicação entre colaboradores.

A publicação de um artigo científico é também um objetivo de trabalho futuro de forma a dar a conhecer o trabalho realizado, pelo que a sua execução é expectável. Pretende-se ainda realizar uma investigação de forma a explorar novas possibilidades como a realidade aumentada e a realidade virtual, que são áreas em que se acredita ser possível desenvolver e adaptar uma interface gráfica específica para o gerador de cenários pois, deste modo enriquece-se a experiência de utilizador, oferecendo novas possibilidades de navegação e interação. Finalmente pretende-se também incorporar no gerador de cenários as sugestões dos participantes na fase de testes, ou seja, a integração com redes sociais, a adição de efeitos sonoros e a funcionalidade de alterar a língua.

Bibliografia

- Piccione, P. (1980). In Search of the Meaning of Senet. *Archaeology, Volume 33*, 55-58.
- Hunt, J. (2015). *The True Purpose of Microsoft Solitaire, Minesweeper, and FreeCell*.
Acedido a 12 de Dezembro de 2016 de:
mentalfloss.com/uk/technology/32106/the-true-purpose-of-solitaire-minesweeper-hearts-and-freecell
- Wilde, T. (2016). *How do I benchmark my PC*. Acedido a 18 de Outubro de 2016 de:
www.pcgamer.com/how-do-i-benchmark-my-pc
- Adams, E. (2013). *Fundamentals of Game Design*. New Riders.
- IBM (1997). *Deep Blue*. Acedido a 20 de Outubro de 2016 de
www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/
- Sharma, S., Kobti, Z. & Goodwin, S. (2008). Knowledge Generation for Improving Simulations in UCT for General Game Playing. *AI 2008: Advances in Artificial Intelligence*, 49-55.
- Genesereth, M. (2013). *General Game Playing*. Acedido a 24 de Outubro de 2016 de:
logic.stanford.edu/classes/cs227/2013/logistics/information.html
- Stanford University. *Gamemaster*. Acedido a 14 de Outubro de:
gpp.stanford.edu/gamemaster/
- Usher, W. (2014). *75% Of Gamers Say That Graphics Do Matter When Purchasing A Game*.
Acedido a 25 de Outubro de 2016 de: www.cinemablend.com/games/75-Gamers-Say-Graphics-Do-Matter-Purchasing-Game-64659.html
- Hevner, A. (2007). The three cycle view of design science research. *Scandinavian Journal of Information Systems, Volume 19*, 87-92.
- Woods, C. (2010). *Visible Language, Inventions of Writing in the Ancient Middle East and Beyond*. Oriental Institute of the University of Chicago.
- Attia, P. (2016). *The Full History of Board Games*. Acedido a 7 de Fevereiro de 2017 de:
medium.com/swlh/the-full-history-of-board-games-5e622811ce89
- Edwards, J. (2014). Saving Families, One Game at a Time. Acedido a 7 de Fevereiro de 2017 de: www.visionandvalues.org/docs/familymatters/Edwards_Jason.pdf
- Becker, U. (2003). *Enciclopedia de los símbolos*. American Bar Association.
- Parlett, D. (1999). *Oxford History of Board Games*. Oxford University Press.
- Sassoon, J. (1993). *From Sumer to Jerusalem: The Forbidden Hypothesis*. Intellect Ltd.
- Greenwood, D. (2006). *A Brief History of WBC*. Acedido a 3 de Janeiro de 2017 de:
www.boardgamers.org/history.htm

- Freeman, W. (2012). *Why board games are making a comeback*. Acedido a 4 de Janeiro de 2017 de: www.theguardian.com/lifeandstyle/2012/dec/09/board-games-comeback-freeman
- Roeder, O. (2015). *Crowdfunding Is Driving A \$196 Million Board Game Renaissance*. Acedido a 4 de Janeiro de 2017 de: fivethirtyeight.com/features/crowdfunding-is-driving-a-196-million-board-game-renaissance/
- Branca, L. & Johansson, S. (2007). Using Multi-agent System Technologies in Settlers of Catan Bots, *Blekinge Institute of Technology*.
- Pirillo, C. (2005). *Windows Vista - Minesweeper*. Acedido a 5 de Janeiro de 2017 de: www.flickr.com/photos/lockergnome/43132632/in/photolist-4q39M4-cHcN-4P4Q5-4KHhMC-oofNch-cZ8Cpy-5UxCFV-DL7HT-vPqNb-kGVJiV-j6sU
- Kain, E. (2012). *Are Board Games Better Than Video Games?*. Acedido a 6 de Janeiro de 2017 de: www.forbes.com/sites/erikkain/2012/04/19/are-board-games-better-than-video-games/#394a6238333d
- Walker, D. (2014). *A Book of Historic Board Games*. Lulu.com.
- Siegler, R. & Ramani, G. (2009). Playing Linear Number Board Games – But Not Circular Ones – Improves Low-Income Preschoolers’ Numerical Understanding. *Journal of Educational Psychology, Volume 101, No. 3, 545-560*.
- Parent & Child Magazine (2017). *The Benefits of Board Games*. Acedido a 11 de Janeiro de 2017 de: www.scholastic.com/parents/resources/article/creativity-play/benefits-board-games
- Gad, S. (2017). *5 Lessons Monopoly Teaches Us About Finance And Investing*. Acedido a 11 de Janeiro de 2017 de: investopedia.com/articles/basics/12/lessons-monopoly-teaches.asp
- Schmid, J. (2009). *Games for People with Alzheimer’s*. Acedido a 20 de Janeiro de 2017 de: best-alzheimers-products.com/games-for-people-with-alzheimers.html
- Zaslavsky, C. (1982). *Tic-tac-toe: and other three-in-a row games from ancient Egypt to the modern computer*. Harpercollins.
- Zimmerman, E. & Salen, K. (2003). *Rules of Play*. The MIT Press.
- Nadel, A. (1997). *Did a Computer Bug Help Deep Blue Beat Kasparov?*. Acedido a 22 de Janeiro de 2017 de: wired.com/2012/09/deep-blue-computer-bug/
- Costa, E. & Simões, A. (2008). *Inteligência Artificial*. FCA.
- Asimov, I. (1942). *Runaround*. Street & Smith.
- Genesereth, M. & Thielscher, M. (2014). *General Game Playing (Synthesis Lectures on Artificial Intelligence and Machine Learning)*. Morgan & Claypool Publishers.
- McCarthy, J. (1959). Programs With Common Sense. *Semantic Information Processing*, 403-418.
- Juul, J. (2005). *Half-Real: Video Games Between Real Rules and Fictional Worlds*. The MIT Press.

- Levine, J., Congdon, C., Ebner, M., Kendall, G., Lucas, S., Miikkulainen, R., Schaul, T. & Thompson, T. (1998). *General Video Game Playing*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Stanford Logic Group (2017). Acedido a 12 de Janeiro de 2017 de: logic.stanford.edu/
- Pell, B. (1993). Strategy Generation and Evaluation for Meta-Game Playing. *KI - Künstliche Intelligenz, Volume 25*, 71-72.
- Lefler, M. & Mallett, J. (2017). *Custom Game & Puzzle Development by Zillions Development Corporation*. Zillions Development Corporation.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, Volume 518*, 529-533.
- Love, N., Hinrichs, T., Haley, D., Schkufza, E. & Genesereth, M. (2008). *General Game Playing: Game Description Language Specification*. Acedido a 12 de Janeiro de 2017 de: logic.stanford.edu/classes/cs227/2013/readings/gdl_spec.pdf
- Méhat, J. & Cazenave, T. (2012). Monte-Carlo Tree Search for General Game Playing. *Laboratoire d'Informatique Avancée de Saint-Denis*.
- Frydenberg, F., Andersen, K., Risi, S. & Togelius, J. (2015). Investigating MCTS Modifications in General Video Game Playing. *2015 IEEE Conference on Computational Intelligence and Games, Taiwan*, 107-113.
- Saffer, D. (2010). *Designing for Interaction: Creating Innovative Applications and Devices*. New Riders
- Shneiderman, B. (2011). *Shneiderman's Eight Golden Rules of Interface Design*. Acedido a 13 de Janeiro de 2017 de: faculty.washington.edu/jttenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html
- Kuhlmann, G., Dresner, K. & Stone, P. (2006). Automatic Heuristic Construction in a Complete General Game Player. *American Association for Artificial Intelligence*.
- Clune, J. (2007). Heuristic Evaluation Functions for General Game Playing. *American Association for Artificial Intelligence*.
- Finnsson, H. & Bjornsson, Y. (2011). CadiaPlayer: Search-Control Techniques. *KI - Künstliche Intelligenz, Volume 25*, 9-16.
- Kissmann, P. & Edelkamp, S. (2011). Gamer, a General Game Playing Agent. *KI - Künstliche Intelligenz, Volume 25*, 49-52.
- Mohapatra, A. & Genesereth, M. (2015). Automating the Design of Game Visualizations. *Stanford University*.
- Gamemaster. Acedido a 13 de Janeiro de 2017 de: ggp.stanford.edu/gamemaster/
- Michulke, D. & Thielscher, M. (2009). Neural Networks for State Evaluation in General Game Playing. *Machine Learning and Knowledge Discovery in Databases*, 95-110.
- Dresden GGP Server. Acedido a 13 de Janeiro de 2017 de: ggpserver.general-game-playing.de/

- Trutman, M. & Schiffel, S. (2016). Creating Action Heuristics for General Game Playing Agents. *GIGA 2015: Computer Games*, 149-164.
- Tiltyard. Acedido a 13 de Janeiro de 2017 de: <http://tiltyard.ggp.org/>
- Genesereth, M. & Björnsson, Y. (2013). The International General Game Playing Competition. *Stanford University*.
- Wayner, P. (2013). *10 Reasons the Browser is Becoming the Universal OS*. Acedido a 13 de Janeiro de 2017 de: www.infoworld.com/article/2609165/web-browsers/10-reasons-the-browser-is-becoming-the-universal-os.html
- Mozilla (2017). *Javascript*. Acedido a 14 de Janeiro de 2017 de: developer.mozilla.org/en-US/docs/Web/JavaScript
- Stack Overflow (2018). *Most Popular Technologies*. Acedido a 8 de Junho de 2018 de: insights.stackoverflow.com/survey/2018#most-popular-technologies
- Flanagan, D. (2011). *JavaScript - The definitive guide*. O'Reilly Media.
- Dirksen, J. (2013). *Learning Three.js: the JavaScript 3D library for WebGL*. Packt Publishing.
- Maaloul, K. (2016). *The Making of "The Aviator": Animating a Basic 3D Scene with Three.js*. Acedido a 14 de Janeiro de 2017 de: tympanus.net/codrops/2016/04/26/the-aviator-animating-basic-3d-scene-threejs/
- Judd, D. (1979). *Contributions to Color Science*. National Bureau of Standards.
- Nikolov, A. (2017). *Design principle: Consistency*. Acedido a 20 de Novembro de 2017 de: uxdesign.cc/design-principle-consistency-6b0cf7e7339f
- Nielsen, J. (1999). *Designing Web Usability*. New Riders.
- Dutra, J. (2015). *Mobile Media & Cultural Heritage. Bauhaus-Universität Weimar*.
- Statista (2018). *Global market share held by smartphone operating systems 2009-2017, by quarter*. Acedido a 10 de Janeiro de 2018 de: www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/
- Interaction Design Foundation (2017). *What is Design Thinking and Why Is It So Popular?*. Acedido a 21 de Novembro de 2017 de: www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular
- National Eye Institute (2015). *Facts About Color Blindness*. Acedido a 5 de Julho de 2017 de: nei.nih.gov/health/color_blindness/facts_about
- The Verge (2017). *Uno is finally getting a colorblind-friendly edition*. Acedido a 20 de Outubro de 2017 de: www.theverge.com/2017/9/6/16262046/uno-colorblind-accessibility-inclusiveness-46-years
- MacAlpine, R. & Flatla, D. (2016). Real-Time Mobile Personalized Simulations of Impaired Colour Vision. *ASSETS '16 Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, 181-189.

- Hardin, B. (2016). *Colorblind accessibility in video games – is the industry heading in the right direction?*. Acedido a 1 de Fevereiro de 2018 de: www.gamersexperience.com/colorblind-accessibility-in-video-games-is-the-industry-heading-in-the-right-direction/
- StatCounter (2017). *Browser Market Share Worldwide*. Acedido a 17 de Fevereiro de 2018 de: gs.statcounter.com/browser-market-share#monthly-201707-201707-map
- Creative Commons (2018). *CC0 1.0 Universal*. Creative Commons.
- Khronos (2011). *Khronos Releases Final WebGL 1.0 Specification*. Acedido a 8 de Junho de 2017 de: khronos.org/news/press/khronos-releases-final-webgl-1.0-specification
- Three.js Docs (2017). *JSONLoader*. Acedido a 8 de Junho de 2017 de: threejs.org/docs/#api/en/loaders/JSONLoader
- Wong, B. (2011). *Nature Methods*. Nature Publishing Group.
- Toptal Color Blind Filter (2017). Acedido a 21 de Abril de 2018 de: www.toptal.com/designers/colorfilter/
- Subić, N., Krunic, T. & Gemović, B. (2014). Responsive web design – Are we ready for the new age?”. *Online Journal of Applied Knowledge Management, Volume 2*, 93-103.
- EduTechWiki (2008). *Usability and user experience surveys*. Acedido a 12 de Maio de 2018 de: edutechwiki.unige.ch/en/Usability_and_user_experience_surveys
- Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments & Computers August 2003, Volume 35*, 379-383.
- Likert, R. (1932). A Technique for the Measurement of Attitudes. *Archives of Psychology, Volume 140*, 1-55.