**Francisco Miguel Moutinho Ferreira**

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

# Utilização de Jogos Digitais para auxílio do Desenvolvimento infantil - Treino da velocidade leitora

Dissertação para obtenção do Grau de Mestre em
[Engenharia Eletrotécnica e de Computadores]

Orientador: Doutor Tiago Cardoso, Professor Auxiliar,
Faculdade de Ciências e Tecnologia

Coorientador: Doutora Filipa Alexandra Moreira Ferrada,
Professora Auxiliar Convidada,
Faculdade de Ciências e Tecnologia

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

**Março,2019**

**[Utilização de Jogos Digitais para auxílio do Desenvolvimento infantil - Treino da velocidade leitora]**

Copyright © Francisco Miguel Moutinho Ferreira, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

*À minha família e amigos.*

# Agradecimentos

Gostaria de agradecer à Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa por todas as oportunidades que tive ao longo dos anos.

Agradecer também ao professor Tiago Cardoso pela oportunidade de explorar este tema e pela sua orientação ao longo do projeto, mas também agradecer à professora Filipa Ferrada pela sua ajuda ao longo da dissertação.

Sem o apoio do Centro Diferenças não seria possível desenvolver esta dissertação e assim agradecer aos terapeutas que me ajudaram a melhorar o protótipo: Dr.ª Fátima Trindade o Dr. Daniel Neves Dias.

Por fim, agradecer a todos os meus amigos que me ajudaram ao longo dos anos, especialmente nesta fase de grande importância que é o fim do curso.

# Resumo

Atualmente, com a sociedade sempre a transmitir-nos grandes quantidades de informação, é necessário que a capacidade de ter uma velocidade de leitura adequada esteja bem desenvolvida. Contudo existem pessoas que sofrem de problemas durante a leitura devido às dificuldades que tiveram durante a sua época escolar. Existe, portanto, uma necessidade de ajudar estas crianças para que consigam atingir as metas de leitura propostas de modo a contribuir para um melhor sucesso profissional.

Os métodos atuais, denominados de métodos tradicionais, são bem-sucedidos, mas é importante utilizar os avanços tecnológicos que existem para melhorar os métodos atuais para novos métodos mais divertidos e eficientes para o treino da velocidade de leitura das crianças.

De modo, a atualizar os métodos atuais surgem os jogos sérios com o objetivo de juntar a componente educativa que é necessária transmitir às crianças com a componente lúdica para aumentar o interesse das crianças durante as suas sessões de jogo. Surge então o "Reino dos Livros" que tem como objetivo ajudar as crianças, dos 6 aos 10, no seu treino de velocidade de leitura.

**Palavras-Chave:** jogos sérios, velocidade de leitura, *Edutainment, Game-Engine*, Unreal Engine.

# Abstract

In Today's society, people are constantly exposed to copious amounts of information on a daily basis, being now more important than ever that people have well-developed reading skills to keep up with the influx of data. However, there are people that suffer reading disabilities developed during their school years, that were never corrected or given proper attention during their development throughout the school. Therefore, an increasing need arises to aid and develop the reading skills of the people affected by this disability, whether they are already in adulthood or more importantly when they are still in their early years of development. The current methods implemented are tested and proven, but still use traditional approaches in an ever more digital world. To increase the reading speed and capabilities of children it is necessary to use new concepts, such as "Serious Games". The term "Serious Games" refers to the use of games that children enjoy as a tool to teach them the much-needed skills to have success in their life. Comparing to their counterpart, the traditional methods, Serious Games are far more engaging, adaptive, and give a higher sense of reward and accomplishment while educating and bettering important skills such as reading. The purpose of this thesis is to develop a prototype for a Serious Game with children aged 6 to 10, that were considered as the main age group with the need to develop their reading skills. **Keywords:** Serious Games, Edutainment, Game-Engines, Reading Skills, Unreal Engine

# Conteúdo


**1. INTRODUÇÃO ..... 1**

1.1. ENQUADRAMENTO E MOTIVAÇÃO ..... 2

1.2. OBJETIVOS DA DISSERTAÇÃO ..... 3

1.3. ORGANIZAÇÃO DA DISSERTAÇÃO ..... 4

**2. ESTADO DA ARTE ..... 7**

2.1. ESTUDO DOS JOGOS DIGITAIS ..... 7

2.1.1. O QUE É UM JOGO DIGITAL? ..... 7

2.1.2. JOGOS DIGITAIS *VERSUS* JOGOS TRADICIONAIS ..... 8

2.1.3. O QUE TORNA UM JOGO DIGITAL NUM JOGO SÉRIO? ..... 10

2.1.4. APLICAÇÃO DE JOGOS SÉRIOS ..... 11

2.1.5. BENEFÍCIOS DA UTILIZAÇÃO DE JOGOS SÉRIOS NA ÁREA DA LEITURA. ..... 12

*2.2. GAME-ENGINES* ..... 16

2.2.1. O QUE SÃO MOTORES DE JOGO (GAME-ENGINE)? ..... 16

2.2.2. *GAME-ENGINES* DISPONÍVEIS NO MERCADO ..... 18

2.3. ANÁLISE DO ESTADO DA ARTE ..... 20

**3. REINO DOS LIVROS ..... 21**

3.1. REQUISITOS DO JOGO ..... 21

3.2. MODELAÇÃO DO SISTEMA ..... 23

3.2.1. UTILIZADOR ..... 24

3.2.2. PROPOSTA DE IMPLEMENTAÇÃO ..... 25

3.2.2.1. PROPOSTA PARA O NÍVEL 1 ..... 28

3.2.2.2. PROPOSTA PARA O NÍVEL 2 ..... 32

3.2.3. SISTEMA DE DADOS ..... 35

**4. IMPLEMENTAÇÃO DA FERRAMENTA ..... 39**

4.1. *GAME-ENGINE*, UNREAL ENGINE ..... 39

4.2. ANÁLISE DO *GAME-ENGINE* ..... 40

4.2.1. *LAYOUT* INICIAL DA FERRAMENTA ..... 41

4.2.2. *INTERFACE* DE PROGRAMAÇÃO ..... 43

4.2.3. VAREST - PLUGIN PARA O UNREAL ENGINE ..... 46

4.3. IMPLEMENTAÇÃO DO PROTÓTIPO ..... 48

*4.3.1. GAME-INSTANCE* ..... 48

4.3.2. SISTEMA DE LOGIN ..... 48

4.3.3. SISTEMA DE REGISTO ..... 50

4.3.4. MENU PRINCIPAL ..... 53

4.3.5. IMPLEMENTAÇÃO DO NÍVEL 1 ..... 56

4.3.6. IMPLEMENTAÇÃO DO NÍVEL 2 ..... 61

4.3.7. MENU DE PERFIL ..... 65

4.3.8. MENU DE CRÉDITOS ..... 67

**5. ANALISE DE RESULTADOS ..... 69**

5.1. RECOLHA DE DADOS ..... 69

5.2. ANALISE DE RESULTADOS ..... 71

5.2.1. RESULTADOS DO NÍVEL 1 ..... 71

5.2.2. RESULTADOS DO NÍVEL 2 ..... 73

**6. CONCLUSÕES E TRABALHOS FUTUROS ..... 77**

**REFERÊNCIAS ..... 80**

# Lista de Figuras

FIGURA 1.1 - DADOS FORNECIDOS PELO CENTRO DIFERENÇAS-CENTRO DE DESENVOLVIMENTO INFANTIL ..... 4
FIGURA 2.1 - OS TRÊS PONTOS PRINCIPAIS PARA O DESENVOLVIMENTO DE UM JOGO SERIO.[14] ..... 11
FIGURA 2.2-DADOS CORRESPONDENTES ÀS DIFERENTES ÁREAS DA LEITURA UTILIZANDO AVG E NAVG. ..... 14
FIGURA 3.1-DIVISÃO DO SISTEMA EM TRÊS MÓDULOS ..... 23
FIGURA 3.2-CASOS DE USO DO SISTEMA ..... 24
FIGURA 3.3- FLUXOGRAMA REFERENTE AO SISTEMA DE REGISTO ..... 27
FIGURA 3.4-MENU PRINCIPAL DO PROTÓTIPO ..... 28
FIGURA 3.5-DIAGRAMA DE MODELAÇÃO DO NÍVEL 1 ..... 29
FIGURA 3.6-DIAGRAMA DE MODELAÇÃO DO NÍVEL 2. ..... 32
FIGURA 3.7-DIAGRAMA DE SEQUÊNCIA DAS COMUNICAÇÕES ENTRE OS MÓDULOS DO SISTEMA ..... 35
FIGURA 3.8-DIAGRAMA DE ENTIDADES E RELACIONAMENTOS. ..... 37
FIGURA 4.1- INTERFACE INICIAL DO UNREAL ENGINE ..... 41
FIGURA 4.2-INTERFACE DE PROGRAMAÇÃO. ..... 43
FIGURA 4.3- PROGRAMAÇÃO DE BLUEPRINTS ..... 44
FIGURA 4.4-PROCEDIMENTO PARA CRIAR UMA MENSAGEM JSON NO JOGO. ..... 46
FIGURA 4.5 - LOAD DE UM SAVEGAME. ..... 49
FIGURA 4.6-MENU INTRODUTÓRIO PARA O REGISTO DO UTILIZADOR. ..... 51
FIGURA 4.7-MENU DE REGISTO DO UTILIZADOR. ..... 52
FIGURA 4.8-MENU PRINCIPAL DO JOGO. ..... 54
FIGURA 4.9-MENU DE EXEMPLO CORRESPONDENTE AO NÍVEL. ..... 54
FIGURA 4.10-MENU PARA ADICIONAR UMA NOVA PALAVRA ..... 55
FIGURA 4.11-MODELO *3D* PRESENTE NO NÍVEL 1. ..... 57
FIGURA 4.12 - MECANISMO INICIAL PARA MOSTRAR AS PALAVRAS. ..... 57
FIGURA 4.13 - PROCEDIMENTO DAS *TIMELINES* PARA A SUBIDA DA PALAVRA ..... 58
FIGURA 4.14 - DRIVER DA TIMELINE QUE CORRESPONDE A UMA RAMPA. ..... 58
FIGURA 4.15 - ESCOLHA DA PALAVRA NO NÍVEL 1 ..... 59

FIGURA 4.16-MENU DE RESULTADOS DO PRIMEIRO NÍVEL. ..... 60
FIGURA 4.17 - EXEMPLO DE UM TEXTO ESCOLHIDO PARA O UTILIZADOR LER. ..... 62
FIGURA 4.18-ESCOLHA DA TEMÁTICA CORRETA RELATIVA AO TEXTO ANTERIORMENTE LIDO ..... 63
FIGURA 4.19-MENU DE RESULTADOS DO SEGUNDO NÍVEL. ..... 63
FIGURA 4.20 - EXEMPLO DE UM GRÁFICO CORRESPONDENTE AO TEXTO 9. ..... 65
FIGURA 4.21-MENU DE PERFIL. ..... 65
FIGURA 4.22-RESULTADOS *ONLINE* DO UTILIZADOR. ..... 66
FIGURA 5.1-GRÁFICO CIRCULAR DA TAXA DE SUCESSO PARA A PRIMEIRA DIFICULDADE. ..... 71
FIGURA 5.2-GRÁFICO CIRCULAR DA TAXA DE SUCESSO PARA A SEGUNDA DIFICULDADE ..... 72
FIGURA 5.3-GRÁFICO CIRCULAR DA TAXA DE SUCESSO PARA A TERCEIRA DIFICULDADE. ..... 72
FIGURA 5.4-GRÁFICO CIRCULAR DA TAXA DE SUCESSO PARA A QUARTA DIFICULDADE. ..... 73
FIGURA 5.5-GRÁFICO DE VELOCIDADES DE LEITURA PARA A PRIMEIRA DIFICULDADE. ..... 74
FIGURA 5.6-GRÁFICO DE VELOCIDADES DE LEITURA PARA A SEGUNDA DIFICULDADE. ..... 75
FIGURA 5.7-GRÁFICO DE VELOCIDADES DE LEITURA PARA A TERCEIRA DIFICULDADE. ..... 75
FIGURA 5.8-GRÁFICO DE VELOCIDADES DE LEITURA PARA A QUARTA DIFICULDADE. ..... 76
FIGURA 6.1-PROTÓTIPO DE UMA PERSONAGEM PARA O JOGO. ..... 78

# Lista de Tabelas

TABELA 2.1-DESCRIÇÃO DAS CARACTERÍSTICAS DE UM JOGO. ..... 8
TABELA 2.2-TABELA COM A COMPARAÇÃO ENTRE UNITY3D E UNREAL ENGINE ..... 19

# Lista de Acrónimos

| | |
|---|---|
| SDK | *Software Development Kits* |
| AVG | *Attentional Action Games* |
| NAVG | *Non-Attentional Action Games* |
| HTTP | *HyperText Transfer Protocol* |
| HTTPS | *Hyper Text Transfer Protocol Secure* |
| PHP | *Hypertext Preprocessor* |
| SQL | *Structured Query Language* |
| JSON | *JavaScript Object Notation* |
| HUD | *Head-up display* |

1

# 1. Introdução

Atualmente estima-se que 5% da população sofra de algum tipo de perturbação de aprendizagem durante o seu percurso escolar, o que torna estas crianças sugestíveis a gerar graves problemas durante o seu período académico e posteriormente no seu emprego [1]. No documento da presente dissertação o foco incidirá sobre as perturbações de aprendizagem referentes às capacidades de leitura das crianças como por exemplo: precisão de leitura de palavras, velocidade, fluência da leitura e compreensão da leitura.

Devido a estas perturbações as crianças demonstram um menor rendimento escolar para os patamares normais da sociedade correspondente à sua idade e nível de escolaridade. Para melhorar o aproveitamento escolar, estas crianças são acompanhadas por técnicos qualificados em centros de desenvolvimento infantil, de modo a melhorar o seu percurso escolar nas áreas afetadas, pelas perturbações, mas também promover a integração destas crianças que por vezes sofrem de isolamento devido a serem consideradas diferentes dos outros alunos.

## 1.1. Enquadramento e Motivação

Nestes centros de desenvolvimento infantil, são aplicadas terapias, préviamente estudadas e testadas, para estimular a aprendizagem, normalmente em suporte de papel, através de exercícios escolares, mas também através da utilização de jogos tradicionais. Contudo, as ferramentas utilizadas encontram – se desatualizadas na medida que a utilização de atividades maioritariamente ou exclusivamente em suporte escrito leva, numa perspetiva de longo prazo, ao desinteresse da criança devido à falta de imersividade e criação de interesse em torno dessa atividade, eventualmente podendo conduzir à estagnação do progresso de aprendizagem da criança. Para colmatar este problema, os centros encontram-se à procura de jogos digitais de modo a tornar os exercícios necessários para as terapias mais divertidos para as crianças. Aqui inserem se os jogos sérios numa perspetiva de *Edutainment*, jogos estes divertidos para tornar as sessões de acompanhamento mais apelativas para as crianças, mas sempre com o objetivo de passar uma mensagem ou comportamento que estimule as crianças a acompanhar as terapias, como por exemplo um jogo didático que ensine as crianças a melhorar a sua leitura.

No entanto, é necessário estudar as melhores implementações durante o desenvolvimento deste tipo de jogos digitais, de modo a não contrariar as terapias já implementadas pelos profissionais destas áreas. Para isso, são formadas parcerias entre os centros infantis e a universidade para que o jogo desenvolvido, durante a dissertação, se encontre adequado para as crianças.

## 1.2. Objetivos da Dissertação

O objetivo desta dissertação é complementar as terapias tradicionais já implementadas com de um jogo sério focado na velocidade leitora, mas também na compreensão de textos escolares por parte das crianças tentando promover um melhor aproveitamento das sessões de terapia e aproveitamento escolar. Resumindo o objetivo será responder à seguinte pergunta de investigação:

| |
|---|
| -Será possível analisar o desempenho de leitura de uma criança num jogo sério e melhorar o seu desempenho? |

O jogo e consequente investigação necessária para a implementação deste terão com objetivo a implementação de uma ferramenta para crianças que se encontram no primeiro ciclo de escolaridade, na faixa etária dos 6 aos 10 anos. As etapas, correspondentes aos exercícios, implementadas no jogo serão aprovadas pelo Centro Diferenças, Centro de Desenvolvimento Infantil, de modo a garantir que o jogo se enquadra nas sessões. O jogo será acompanhado por uma recolha de dados correspondentes aos resultados de desempenho durante as sessões de modo a disponibilizar as estatísticas de evolução das crianças aos técnicos e de igual modo como método de validação para avaliar o sucesso do jogo.

Os terapeutas do Centro Diferenças que irão acompanhar todo o decorrer do desenvolvimento e implementação do protótipo são: a Dr.ª Fátima Trindade e o Dr. Daniel Dias Neves.

As metas curriculares, criadas pela direção geral da educação [2],fornecidas pelo Centro Diferenças, correspondentes à velocidade leitora pretendidas de atingir encontram-se na figura 1.1:

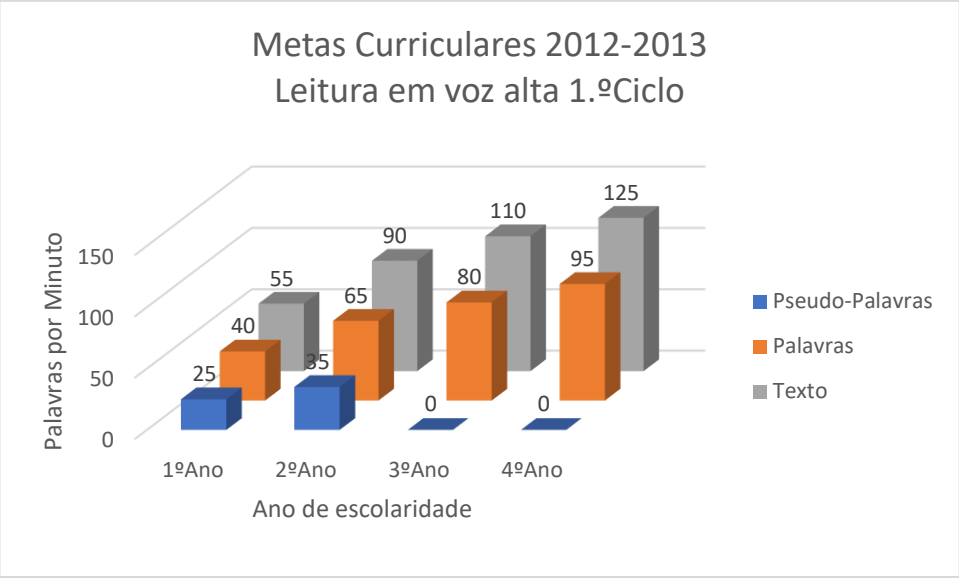


Figura 1.1 - Dados fornecidos pelo Centro Diferenças-Centro de Desenvolvimento Infantil

## 1.3.Organização da Dissertação

O restante deste documento encontra-se dividido nos seguintes capítulos:

- **Capítulo 2** – começa por descrever alguns dos conceitos mais importantes para o desenvolvimento do protótipo assim como uma definição de o que é um jogo digital e o que torna um jogo digital num jogo sério e como é que se enquadra numa perspetiva de educação bem como fornecer uma analise de diferentes *softwares* disponíveis, particularmente os diferentes *Game-Engines* disponíveis no mercado para o desenvolvimento do protótipo.
- **Capítulo 3** – corresponde ao desenho da proposta de desenvolvimento do protótipo através da análise dos requisitos necessários para atingir os objetivos propostos pela dissertação. Esta secção inclui a modelação estática do sistema em função dos casos de uso, mas permite ao leitor consultar as propostas para os diferentes níveis.
- **Capítulo 4** - esta secção é responsável pela análise do *software* escolhido através de exemplos de utilização e explicações detalhadas para ajudar o leitor a compreender melhor as

funcionalidades fornecidas pelo *Game-Engine*. Contudo, inclui também todas as metodologias utilizadas ao longo do desenvolvimento do jogo assim como as alterações que ocorreram face aos problemas encontrados durante o desenvolvimento.

- **Capítulo 5**- após a conclusão do desenvolvimento é necessário analisar os dados. Neste capítulo encontra-se a escolha da metodologia, para obtenção dos dados, proposta e os dados obtidos. Por fim é apresentado uma análise dos dados obtidos para retirar conclusões sobre o funcionamento e implementação do jogo.
- **Capítulo 6**- inclui as conclusões finais sobre o jogo quando analisados os dados assim como uma lista de alterações futuras a aplicar ao jogo para o melhoramento do mesmo, de modo a melhorar a experiência e desempenho das crianças durante a utilização do jogo.

2

# 2. Estado da Arte

Para o desenvolvimento da presente dissertação é necessário primeiro efetuar uma investigação sobre a utilização de jogos digitais, o que os define e quais as metodologias utilizadas para o desenvolvimento dos mesmos. Neste capítulo serão abordados os conceitos do que são jogos digitais, o que os define, os motivos que levam os jogos digitais a serem considerados jogos sérios e a utilização dos mesmos. Durante a implementação também é necessário proceder-se à escolha de um motor de jogo (*Game-Engine)*, *framework* essencial para a construção de um jogo assim como um breve estudo sobre os *Game-Engines* disponíveis no mercado mais aquedados para a implementação desta dissertação.

## 2.1. Estudo dos Jogos Digitais

### 2.1.1. O que é um Jogo Digital?

Existem diferentes tipos de definições sobre o que torna um jogo num jogo digital, ao invés de ser considerado um jogo tradicional, conhecidos normalmente por jogos de mesa. Para a definição de um jogo digital utilizaremos a definição proposta por Frasca. Frasca define um jogo digital como: *"[O jogo digital ] é uma forma de entretimento baseada num software de computador, seja este textual ou de imagens, usando uma plataforma eletrónica como os computadores pessoais ou consolas e que envolvem um ou múltiplos jogadores num meio físico ou em*

*rede*"[3]. Quando Frasca se refere a um software de computador , refere-se à utilização de um motor de jogo, ou seja, a utilização do jogo não se encontra bloqueada apenas a computadores mas sim, disponível para todas as plataformas tecnológicas capazes que processar o motor de jogo. Os dispositivos mais utilizados para este fim, normalmente, são dispositivos móveis como *tablets* ou telemóveis *smartphone*.

Após estabelecida uma definição do que é um jogo digital é interessante fazer uma comparação entre os jogos tradicionais e os jogos digitais referindo também as vantagens da implementação do mesmo num formato digital.

### 2.1.2. Jogos Digitais *versus* Jogos Tradicionais

Na tabela 2.1 podemos ver uma comparação entre os jogos tradicionais e os jogos digitais.

Tabela 2.1-Descrição das características de um jogo. Adaptado de [3]

| | **Jogos Tradicionais** | **Jogos Digitais** |
|---|---|---|
| **Som** | Não tem | Diversos |
| **Grafismo** | Baixo | Alto |
| **Duração do Jogo** | Baixo | Alto |
| **Grau de Envolvimento** | Baixo | Alto |
| **Dificuldade** | Média | Média |
| **História** | Não tem | Diversas |
| **Dinâmica de Jogo** | Média | Alta |

Analisando a tabela anterior, podemos rapidamente tirar conclusões sobre as vantagens da implementação digital. As vantagens,dos jogos digitais, são as seguintes em relação aos jogos tradicionais para cada característica.

- **Som** – Os jogos digitais permitem incorporar diversos sons de modo a fornecer *feedback* ao utilizador e criar uma melhor imersão ao longo do jogo.

- **Grafismo** – Os *Game-Engines* incorporam diferentes *Software Development Kits* (*SDK*) gráficos como *OpenGL* [4] e *DirectX* [5] que

permitem aumentar a qualidade dos gráficos de um jogo. Existe, portanto, uma relação na medida que o avanço tecnológico destes kits permite-aumentar o grafismo dos jogos.

- **Duração de Jogo** – Embora alguns jogos tradicionais requeiram algum tempo de jogo, os jogos digitais permitem criar uma duração de jogo muito maior e adaptavél ao jogador, levando alguns jogos a demorar inúmeras horas para completar.

- **Dificuldade**-Existem jogos tradicionais difíceis, normalmente na área da cultura e na área de estratégia. Contudo os jogos digitais possiblitam incorporar técnicas, (e.g: redes neuronais) [6], que permitem que os jogos se adaptem ao jogador criando assim uma dificuldade mais adequada para cada jogador.

- **História**- Normalmente os jogos tradicionais não incluem história. Embora existam jogos digitais sem história, normalmente é uma área que se investe na criação de um jogo digital porque permite que o utilizador crie uma experiência imersiva do jogo que está a jogar.

- **Dinâmica de Jogo** – Os jogos tradicionais encontram-se muito restritos ao mundo físico, normalmente utilizando tabuleiros e dados para a gestão de regras do jogo. Isto é fácilmente implementado num jogo digital e permite a criação de novas mecânicas de jogo uma vez que a única limitação é apenas a capacidade do *Game-Engine* ao invés da normal limitação física que os jogos tradicionais enfrentam.

Por fim, uma característica importante de todos os jogos digitais é que permitem atualizações dos criadores ao longo do tempo, impendido assim que o jogo se torne estático, ou seja, o jogo não se mantem inalterado no tempo permitindo, continuar a adicionar novos desafios, contribuído para uma maior inovação e melhoramento do mesmo. As atualizações têm como foco permitir uma

maior adaptabilidade, escalabilidade e mutabilidade do jogo de acordo com as necessidades do público alvo. Estas características permitem manter um nível de motivação constante devido ao novo conteúdo adicionado.

### 2.1.3. O que torna um jogo digital num jogo sério?

Uma vez explicado o conceito de um jogo digital, podemos passar à definição de um jogo sério. Os jogos sérios (*Serious-Games*) são jogos desenvolvidos em que a fundação continua a ser o entretenimento do jogador, mas contendo uma componente educacional associada [7]. A criação deste novo género de jogos enquadra-se dentro do *Edutainment*[1], ou seja, permitir uma *gamification*[2] de conhecimento ou competências de uma maneira divertida para os utilizadores.[8] [7] [9] [10]

No entanto, o termo *Serious-Games* é visto como contraditório devido ao fato que os jogos são apenas considerados como ferramenta de puro entretenimento, onde se depreende que da utilização dos mesmos não é possível obter progressão ou qualquer tipo de aprendizagem/conhecimento relevante. Contudo existem estudos [11] [12] publicados que demonstram que a utilização de jogos provoca um impacto positivo na aprendizagem das crianças.

Para um jogo digital ser considerado "*Serious-Games*" tem que respeitar os três seguintes pontos [13]:

---

[1] *Edutainment* - Transmitir conhecimento e entretenimento aos utilizadores.
[2] *Gamification* - Tornar as atividades mais divertidas através de jogos.

• ***Gameplay*** - A jogabilidade do jogo("*Gameplay*") tem que ser inerentemente motivante. Para o jogo ser motivante implica que o foco da atenção do utilizador seja o jogo, mas também promove a persistência com que é utilizado, com uma elevada probabilidade de repetição por parte do jogador.

• ***Feedback*** - O jogo tem que fornecer *feedback* imediato ao jogador. Este *feedback*, no caso de um jogo de leitura, seria um aviso se o jogador conseguiu ler corretamente ou se conseguiu acertar corretamente a uma pergunta sobre os textos.

• **Conteúdo** - O conteúdo do jogo tem que ser complexo o suficiente para promover a aprendizagem do jogador e ao mesmo tempo simples para não repelir o dito jogador. A aprendizagem pode demonstrar-se na forma de novo conhecimento, mas também, no aumento de uma componente prática como a resolução de problemas.

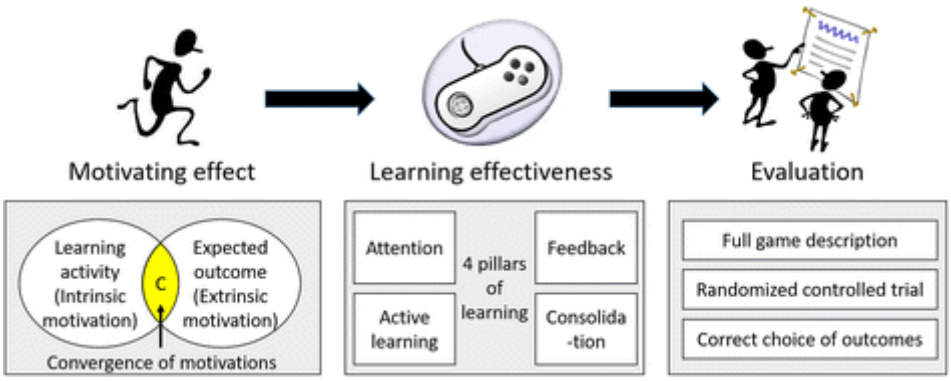


**Figura 2.1 - os três pontos principais para o desenvolvimento de um jogo serio.**[14]

### 2.1.4. Aplicação de jogos sérios

Nesta secção iremos fazer o enquadramento dos jogos sérios em diferentes áreas de aplicação assim como um exemplo de aplicação em cada área[13].

**Implementação Social** – Estes jogos promovem temas como situações políticas, mas também questões sociais como a reciclagem ou gestão dos

recursos naturais disponíveis. Um exemplo deste tipo de jogo é o Waterbusters, desenvolvido em 2006, que promove uma sensibilização da gestão dos recursos hídricos disponíveis de uma população [15].

**Área Empresarial** – Os jogos na área empresarial são desenvolvidos para que as empresas consigam desenvolver mais rapidamente as competências necessários dos seus empregados para o seu emprego. Um exemplo é o jogo *Objection*, desenvolvido em 2008, que permite instruir aos advogados as competências necessárias para o seu trabalho durante um processo em tribunal.

**Área Militar** – Os jogos desenvolvidos nesta área fornecem conhecimento e competências relacionadas com atividades militares. A implementação destes jogos tende a replicar um ambiente realista para que os militares consigam replicar e realizar os treinos militares sem qualquer perigo. Um exemplo deste tipo de jogo é um jogo chamado *America´s Army,* desenvolvido pelo exército dos Estados Unidos da América, em 2002, que permite aos jogadores receberem um treino básico e como se desenvolveria a sua careira militar[15].

**Área da Saúde** – Os jogos na área da saúde são utilizados para promover melhores hábitos de saúde, redução de situações de risco ou para sensibilizar o jogador sobre a sua doença ou de uma pessoa próxima. Os jogos também permitem desenvolver as capacidades motoras dos seus utilizadores através da repetição de movimentos durante o jogo.

### 2.1.5. Benefícios da utilização de jogos sérios na área da leitura.

Existem muitas crianças que durante o seu percurso escolar demonstram problemas de leitura, seja na sua velocidade leitora ou na compreensão dos textos presentes no programa escolar. Embora existam já implementações tecnológicas para ajudar a aprendizagem destas crianças, muitos destes jogos apenas focam-se num problema em específico como a velocidade ou a compreensão linguística. Uma possível implementação foi uma aplicação desenvolvida pela *Speed Reading Team*, que inclui diferentes tipos de exercícios linguísticos. Isto

permite que a criança consiga praticar as áreas onde tem mais dificuldade. A utilização deste tipo de jogos também se aplica para o tratamento de doenças como a dislexia.

Embora existam diferentes aplicações com o objetivo de ajudar as crianças, não existem jogos em português que disponham de diferentes exercícios para melhorar a velocidade de leitura, de modo a atingir as metas propostas que será o foco principal da implementação proposta.

Quando os jogos sérios foram aplicados em crianças com dislexia, as crianças demonstravam melhores resultados durante as sessões em que os jogos eram utilizados do que nas sessões com métodos alternativos. Uma hipótese para comprovar estes resultados, é o fato que as crianças com dislexia muitas vezes sofrerem de défice de atenção e os jogos promovem o foco e imersão da atenção, das mesmas, no decorrer dos minijogos durante as sessões.[11] Em [11] encontra-se um estudo nas diferentes áreas da leitura entre a utilização de jogos sérios e os métodos tradicionais, representado pela figura 2.2.

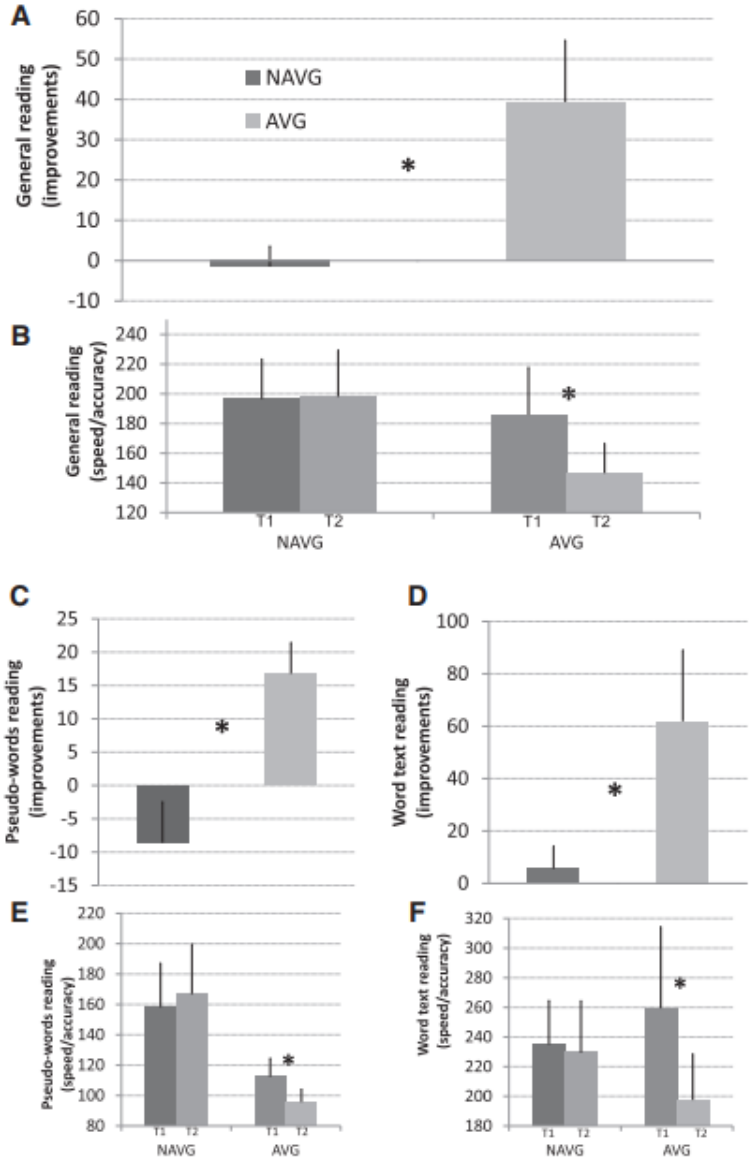


Figura 2.2-Dados correspondentes às diferentes áreas da leitura utilizando AVG e NAVG.

Na figura 2.2 estão representados os diferentes gráficos correspondentes às diferentes áreas de leitura referentes à utilização de jogos digitais durante as sessões de acompanhamento das crianças.

Cada gráfico dos 6 presentes na figura 2.2 utiliza dois conjuntos de dados:

- **T1**: corresponde às capacidades de leitura das crianças sem sessões de jogos.
- **T2**: corresponde às capacidades de leitura após duas sessões utilizando jogos.

A explicação corresponde a cada gráfico encontra-se nos pontos seguintes:

- **Gráfico A**: corresponde ao melhoramento da leitura no geral.
- **Gráfico B**: corresponde à ineficiência da leitura, no geral, das crianças testadas. A ineficiência é calculada através da divisão entre a velocidade de leitura, correspondente ao tempo em segundos necessário para ler uma palavra, e a precisão, rácio entre o número de respostas corretas e o número de perguntas total.
- **Gráfico C**: corresponde à evolução da leitura das crianças referente à leitura de pseudopalavras.
- **Gráfico D**: Analogamente ao gráfico C, este gráfico representa o progresso das crianças referente à leitura de textos.
- **Gráfico E:** Ineficiência da leitura correspondente à leitura de pseudopalavras. Utiliza os mesmos critérios para ineficiência que o gráfico B
- **Gráfico F**: Ineficiência da leitura correspondente à leitura de textos. Utiliza os mesmos critérios para a ineficiência da leitura que o gráfico B

Analisando o gráfico é possível retirar algumas conclusões importantes relativamente à utilização de jogos na área da leitura. A primeira conclusão é que de um modo geral, as crianças que utilizaram os jogos sérios para o treino da velocidade de leitura melhoraram em comparação com o segundo grupo que apenas utilizou os métodos tradicionais. Esta análise correspondente aos gráficos A e B.

As crianças apresentam melhores resultados na leitura de pseudopalavras e leitura de textos quando utilizam os jogos sérios, gráficos C e D, assim como uma menor ineficiência de leitura, calculada através do rácio entre a velocidade de leitura e a precisão de leitura.

Podemos concluir através deste estudo que a utilização de um jogo sério para o melhoramento da velocidade de leitura é uma hipótese válida que merece ser analisada.

## *2.2. Game-Engines*

Nesta secção será discutido o software mais importante para o desenvolvimento de um jogo digital, o *Game-Engine*. O capítulo começa por introduzir o que são motores de jogo e as suas características. Também apresenta alguns motores de jogo já utilizados pelo autor da dissertação assim como a escolha do motor de jogo para a implementação proposta.

### 2.2.1. O que são motores de jogo (Game-Engine)?

Um motor de jogo é uma *framework*, ou seja, uma arquitetura de desenvolvimento de *software*, responsável por fazer a integração de diferentes *Software Development Kits* (*SDK*) de modo a facilitar a reutilização de código genérico durante o desenvolvimento do jogo [16]. O motor de jogo deve ser extensível, de modo a permitir que diferentes programadores utilizem a mesma base para desenvolver jogos diferentes sem grandes modificações [17].Embora os *Game-Engines* sejam desenvolvidos para serem um *software* genérico, existe um *trade-off* entre tornar o *Game-Engine* genérico ou restrito para uma plataforma de *hardware*. Se o *Game-Engine* for genérico a *performance* será idêntica em diferentes plataformas, mas não será igual à *performance* quando comparado com um *Game-Engine* desenvolvido para uma categoria de jogos específica [18].

Um *Game-Engine* é constituído por diferentes subsistemas, correspondentes aos diferentes *SDK* integrados no *Game-Engine*. Os subsistemas mais importantes são os seguintes [17]:

- ***Data Structures and Algorithms (Scripting):*** Subsistema responsável pelas estruturas de dados e os algoritmos responsáveis pela gestão do jogo, tais como transformadas correspondentes às localizações dos objetos, mas também as regras do jogo. Este subsistema inclui toda a programação feita pelo programador do jogo.
- ***Rendering System***: Subsistema responsável pelos *SDK* correspondentes aos gráficos do jogo. Dentro desta categoria de *SDK* os mais utilizados são: o *OpenGL*, desenvolvido pela Khronos Group, e o *DirectX* desenvolvido pela empresa Microsoft.
- ***Collision and Physics System*** – Responsável por fazer uma gestão dos comportamentos físicos do jogo assim como a deteção e gestão das colisões entre os objetos. Existem diferentes *SDK* responsáveis por este subsistema tais como o sistema Havoc, desenvolvido pela empresa Havoc e o PhysX, desenvolvido pela NVidia.
- ***Character Animation System*** – Este subsistema é responsável pelas animações pertencentes ao jogo. Um *SDK* utilizado na indústria é o Havoc Animation.
- ***Artificial Intelligence System*** – Neste subsistema encontra-se a inteligência artificial do jogo. Embora cada jogo pode criar uma inteligência artificial adequada ao seu jogo, foi criado um *SDK*, Kynapse, desenvolvido pela Autodesk de modo a permitir criar um AI de baixo nível de uma forma simples para qualquer jogo.
- ***Audio Engine*** – É responsável por todos os algoritmos necessários para o *Game-Engine* referentes aos sons utlizados no jogo. Um exemplo destes algoritmos é o Steam Audio desenvolvido pela Valve Corporation.

### 2.2.2. *Game-Engines* disponíveis no mercado

Atualmente no mercado, existem diferentes motores de jogo como o Unreal Engine, Unity3D, Gamemaker Studio e CryEngine [19]. Na presente dissertação optou-se por focar apenas no Unreal Engine e no Unity3D devido ao autor possuir conhecimentos e experiência prévia e permitirem a maior compatibilidade nas diferentes plataformas previstas para a proposta de implementação.

- **Unity3D** – foi desenvolvido em 2005 pela empresa Unity Tecnologies. Inicialmente a plataforma de desenvolvimento de *software* era apenas compatível com o Mac OS. Posteriormente, o *Game-Engine* foi adaptado para permitir o desenvolvimento de jogos em Windows. O Unity3D permite criar jogos bidimensionais, mas também tridimensionais. O motor de jogo permite que o jogo seja exportado para diferentes plataformas de *hardware* para disponibilizar uma maior adaptabilidade referente às plataformas disponíveis [20]. Atualmente é um *Game-Engine* gratuito, mas com uma versão *premium*. A Unity Tecnologies disponibiliza um mercado de *assets* compatíveis com o *software*, onde alguns são pagos, mas com muitos produtos grátis. Disponibiliza também documentação referentes às funções inerentes ao *Game-Engine* [21]. As linguagens de programação utilizadas, no Unity3D, são: JavaScript e C#

- **Unreal Engine** – desenvolvido em 1998 pela empresa Epic Games. Inicialmente este motor de jogo foi desenvolvido para o desenvolvimento de jogos "*First Person Shooters*", contudo nas iterações seguintes do *software*, a empresa decidiu investir e tornar o seu motor de jogo otimizado para diferentes géneros de jogos permitindo a criação de jogos tridimensionais (*3D*) e bidimensionais (*2D*). Neste momento, o *software* de desenvolvimento é compatível com Windows e Mac OS mas

permite exportar para todas as plataformas de hardware comuns no mercado.

Anteriormente ao lançamento da quarta versão do *Game-Engine*, a programação era feita utilizando uma linguagem de programação chamada UnrealScript. Mais tarde com o lançamento da quarta versão, a empresa Epic Games optou por trocar a linguagem de programação principal, UnrealScript, para C++.

Inicialmente era um *Game-Engine* pago, mas a empresa optou por tornar o *Game-Engine* gratuito[22], com um pagamento de 5% de *royalties* quando o lucro é superior a 3000 euros por produto[23].

A Epic Games disponibiliza também a documentação referente ao *Game-Engine* e as suas funções já incorporadas.[24]

Contudo, a Epic Games disponibiliza o código fonte do seu *software*, Unreal Engine, permitindo que qualquer programador efetue as alterações que considere importantes durante o desenvolvimento de um jogo.

Resumindo podemos exemplificar as diferentes características comuns de ambos os *Game-Engines* através da seguinte tabela 2.2.

Tabela 2.2-Tabela com a comparação entre Unity3D e Unreal Engine

| Características | Unity3D | Unreal Engine |
|---|---|---|
| Plataformas | Windows/Android | Windows/Android |
| Editor | Simples | Simples |
| Facilidade de utilização | Alta | Alta |
| Código Fonte | Indisponível | Disponível |
| Linguagem de programação | JavaScript/C# | C++ |
| Programação Visual | Não incluída | Blueprints |
| Mercado | Grande | Grande |
| *Templates* | Incluídos | Incluídos |

| | | |
|---|---|---|
| **Documentação** | Alta | Alta |
| **Suporte** | Alto | Alto |
| **Preço** | Versão Gratuita com possibilidade de versão premium. | Versão Gratuita com 5% Royalties no caso de um lucro superior a 3000 euros. |

## 2.3. Análise do Estado da Arte

Resumindo o estudo da teoria necessária para o desenvolvimento de um jogo sério é importante fazer uma revisão dos conceitos abordados ao longo do capítulo.

No início, correspondente à secção 2.1, optou-se por começar por estudar o conceito mais importante, para esta dissertação, que é o que é um jogo digital. Contudo é necessário exemplificar uma comparação entre os jogos tradicionais, como no caso dos jogos clássicos de tabuleiro ou jogos físicos, e os jogos digitais para ser possível tirar conclusões da escolha de utilizar um jogo digital e não um jogo tradicional. Como o objetivo é utilizar o jogo na área da educação, para ajudar as crianças, estudou-se o que são jogos sérios e como é que um jogo digital se enquadra nos jogos sérios assim como a apresentação de diferentes áreas onde a utilização de jogos sérios foi benéfica para os utilizadores.

Por fim, é feito um estudo do *software* utilizado para o desenvolvimento de jogos atualmente. Começa por uma caracterização do que é um motor de jogo e qual é o seu papel e funcionalidade para este projeto assim como uma apresentação de diferentes motores de jogo disponíveis para utilizar para o desenvolvimento do jogo.

3

# 3. Reino dos Livros

Neste capítulo será apresentado uma proposta de resolução, que tem como objetivo, melhorar as capacidades de leitura das crianças pertencentes ao primeiro ciclo de escolaridade, de modo a que cumpram as metas propostas no capítulo 1.2.

Para garantir que a ferramenta cumpre as características necessárias para a aprendizagem por parte dos utilizadores é necessário definir os requisitos de funcionamento, descritos na secção 3.1, assim como o método de modelação do sistema escolhido na secção 3.2.

## 3.1. Requisitos do Jogo

De modo a garantir que o jogo se enquadrará na aprendizagem dos utilizadores, definiu-se uma lista de características que deve incluir. Os requisitos foram escolhidos, tendo como base, a orientação do Centro Diferenças, de modo a tornar o jogo compatível com as sessões de aprendizagem, permitindo assim que os utilizadores possuam uma ferramenta fora do centro.

O jogo deve cumprir com as seguintes características:

- ***Interface* simples:** O público alvo do jogo serão as crianças do 1º ciclo escolar logo a *interface* deve ser simples de compreender e de acompanhar. Uma característica importante da *interface* é o tipo de letra. O tipo de letra deve ser sem sombreados porque dificulta a leitura, por parte das crianças com dislexia. A ferramenta deverá incluir também verificações para garantir que o utilizador está ciente do efeito de cada opção. Um exemplo, deste tipo de verificações, é a verificação de saída do jogo que impede os utilizadores de desligarem o jogo e consequentemente perderem o seu progresso acidentalmente.
- **Diversidade de exercícios:** O jogo deve incluir diferentes exercícios de leitura. A diversidade permite, ao utilizador, treinar diferentes áreas da leitura de um modo independente. O protótipo proposto incluirá dois exercícios diferentes: um exercício de leitura de palavras e um exercício de leitura de textos e análise das temáticas textuais.
- **Aleatoriedade:** devido aos diferentes exercícios é necessário que exista aleatoriedade durante os mesmos de modo a impedir que o utilizador memorize as respostas, contribuindo assim para uma aprendizagem positiva por parte dos utilizadores. O jogo incluirá um número finito de palavras e textos, mas permite ao utilizador introduzir novas palavras e textos no caso de considerar que domina todos os exemplos propostos.
- **Sistema de *Feedback*:** uma vez o jogo se enquadra numa perspetiva de *Edutainment* é necessário que o jogador seja capaz de recolher *feedback* ao longo da sua sessão de utilização. O jogo deverá reagir à resposta do utilizador transmitindo se a resposta está correta ou incorreta. O sistema deve incluir um sistema adicional de informação que permita ao utilizador verificar a sua progressão na forma de um gráfico. O gráfico apenas se encontrara disponível para o nível corresponde à leitura de textos.

Uma vez, definidas as características necessárias para o funcionamento correto do jogo é necessário definir, de seguida, como é que o sistema está modelado nas diferentes áreas.

## 3.2. Modelação do Sistema

O "Reino dos livros", jogo a desenvolver, será desenvolvido seguindo os requisitos apresentados na secção anterior, ou seja, será um jogo sério focado na leitura, disponível para dispositivos Android e para Windows, com diferentes exercícios de aprendizagem. No fim de cada nível será necessário guardar o progresso do jogador permitindo assim disponibilizar toda a informação *a posteriori*.

O público alvo corresponderá ao primeiro ciclo escolar, logo é aconselhado ao aluno jogar na presença de um adulto durante os primeiros anos para o ajudar na utilização da ferramenta. O objetivo é que, no fim, o aluno já se encontre habituado ao jogo e não necessite de ajuda externa. Cada utilizador estará identificado por um ID, permitindo assim uma distinção entre os utilizadores.

Antes de descrever o sistema é necessário entender as diferentes interações entre o utilizador, o jogo e o sistema de recolha de dados, representados na figura 3.1.

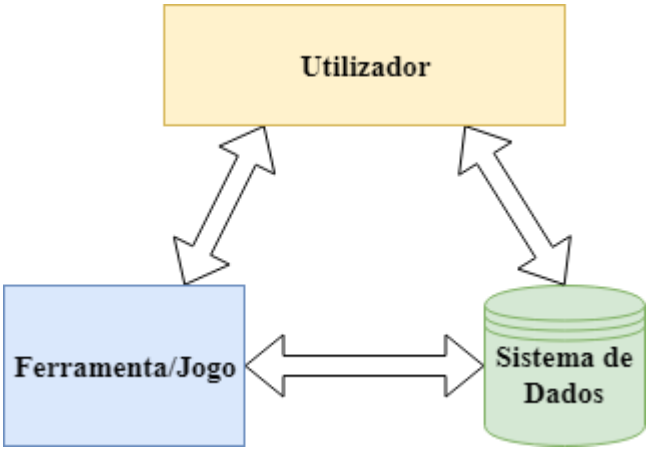


Figura 3.1-Divisão do Sistema em três módulos

Como podemos verificar através da figura, o jogo encontra-se dividido em três módulos que comunicam entre si. Cada módulo é descrito nas secções seguintes.

### 3.2.1. Utilizador

O utilizador é uma das principais componentes do jogo, logo é necessário descrever como é que este interage com os diferentes módulos do jogo. Esta secção considera o utilizador como uma componente externa, ou seja, é descrito quais são os benefícios da utilização da ferramenta e como é que este se encontrará identificado, numa perspetiva global do sistema, exemplificado na figura 3.2.

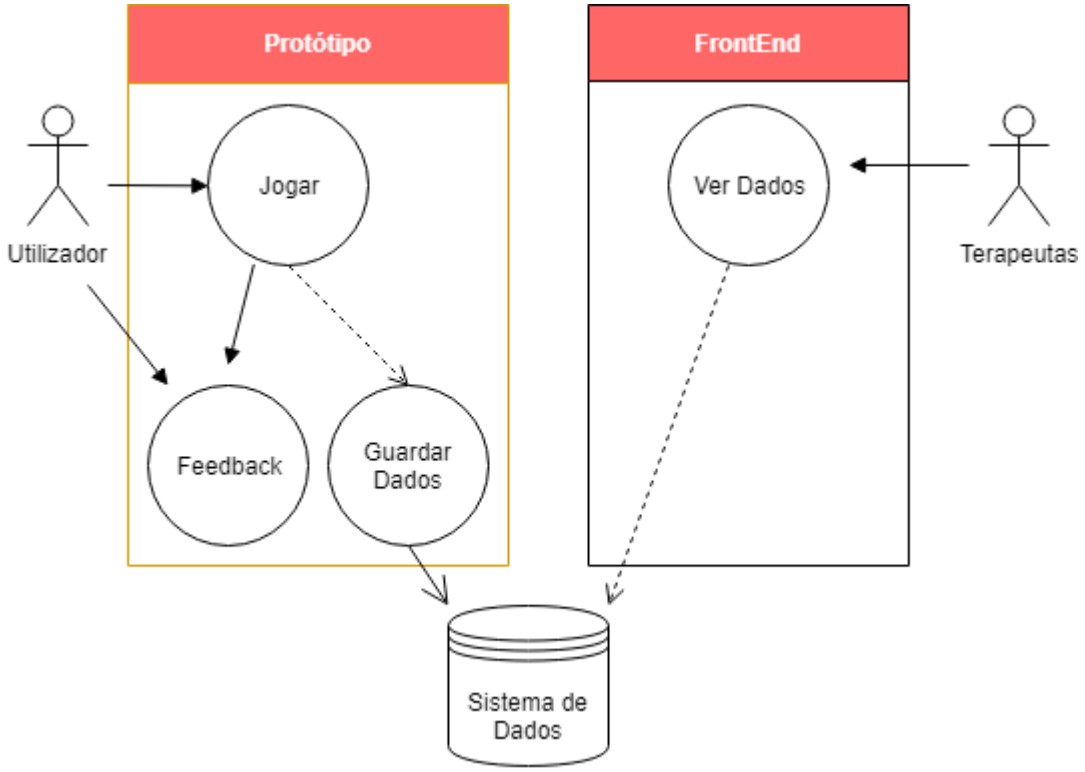


Figura 3.2-Casos de uso do sistema.

Um dos principais problemas, do sistema, é a identificação do utilizador perante o sistema. Uma das primeiras implementações,para resolver o problema, constituiria num sistema de passwords, onde cada utilizador seria obrigado a criar uma conta, de modo a permitir ao jogador, guardar os seus dados. A conta garantiria que o utilizador seria único e fácil de identificar no sistema de dados. Após receber *feedback*, por parte do Centro Diferenças, optou-se por uma alteração no sistema de registo, do utilizador, para tornar mais acessível a utilização por parte das crianças. O novo sistema proposto será constituído por

uma identificação do utilizador, utilizando apenas um número (*ID*). Este *ID* existirá para cada utilizador e será criado através da comunicação entre o jogo e o sistema de dados no início do registo de utilizador. Esta comunicação serve para garantir que o identificador seja sequencial impedindo assim números idênticos durante a identificação dos utilizadores.

Por fim, será necessário referenciar as interações entre a ferramenta e os utilizadores. O utilizador interagirá com o jogo utilizando o *TouchScreen*, no caso de utilizar a versão portátil, para o sistema Android, ou com a utilização de um rato, no caso da versão para Windows. Durante os níveis, os utilizadores receberão *feedback* do jogo na forma de uma indicação no caso de a resposta se encontrar correta ou errada e qual é a sua pontuação ao longo do jogo. No caso da interação do utilizador com o sistema de dados, esta deve permitir ao utilizador inserir o seu identificador único no *Frontend* do sistema de dados e *receber feedback* utilizando uma tabela contendo os seus melhores resultados nos diferentes níveis, mas também mediante um gráfico de velocidade de leitura para cada texto disponível no protótipo. Os benefícios para cada exercício encontraram-se descritos na secção seguinte correspondente ao módulo do jogo.

### 3.2.2. Proposta de Implementação

Nesta secção é descrito como é que a ferramenta deve ser estruturada. O Reino dos Livros será um jogo que incorpora elementos *2D*, no caso dos menus intermédios, mas também conterá elementos *3D* utilizados durante a modelação visual dos níveis. Cada nível pretende treinar um exercício diferente, de modo a criar variedade para o utilizador, mas também para treinar diferentes componentes da leitura permitindo assim retirar diferentes benefícios positivos ao longo da sua utilização. O protótipo será constituído por dois níveis distintos onde cada nível será descrito na sua secção própria.

No início do jogo, deve ser dado ao utilizador a possibilidade de criar um registo para guardar os seus resultados ou continuar se não desejar que os seus resultados sejam registados, no sistema. Esta escolha deverá ser apenas exibida, ao utilizador durante a sua primeira utilização da ferramenta. Após optar se

deseja registar-se, será necessário guardar a opção escolhida, de modo a evitar assim a necessidade de o utilizador escolher, se deseja um registo, sempre que utilizar a ferramenta. É importante separar e explicar as consequências do registo:

- **Com Registo:** No caso de o utilizador desejar criar um registo, devera ser reencaminhado para o menu de registos. Este menu deve requisitar ao utilizador a criação de um nome e a introdução da sua idade. Os dados ficarão guardados no jogo e no sistema de dados. Após a introdução dos dados e confirmação, o jogo deve criar um pedido ao sistema de dados para devolver o identificador único para o utilizador. Nesta secção será obrigatório que a ferramenta tenha acesso à internet para comunicar com o sistema de dados. No caso de não ser possível a comunicação por falta de internet ou por falha durante a comunicação com o sistema de dados, o jogo deve guardar que a comunicação não teve sucesso. Se a comunicação não tiver sucesso, na próxima sessão de jogo, este irá efetuar o mesmo pedido para tentar obter o ID. Este comportamento irá ocorre sempre no início do jogo enquanto o sistema de dados não devolver o identificador do utilizador. Quando o pedido ao sistema de dados ocorrer com sucesso, o utilizador será identificado pelo seu ID e os seus dados começarão a ser guardados pelo sistema de dados numa base de dados externa para posteriormente serem analisados através do *Frontend* do sistema de dados.
- **Sem Registo**: No caso de o utilizador optar por não efetuar o registo inicial, o jogo deve considerar que o utilizador se encontra em modo anónimo. O modo anónimo permitirá que o jogo apenas registe os melhores resultados localmente de modo a que o utilizador continue a ter progressão no jogo. Ao contrário da opção de registo, o modo anónimo não irá permitir disponibilizar informações externas ao utilizador através do *Frontend* uma vez que os seus dados não serão analisados pelo sistema de dados.

O sistema de registo será implementado segundo o esquema da figura 3.3.

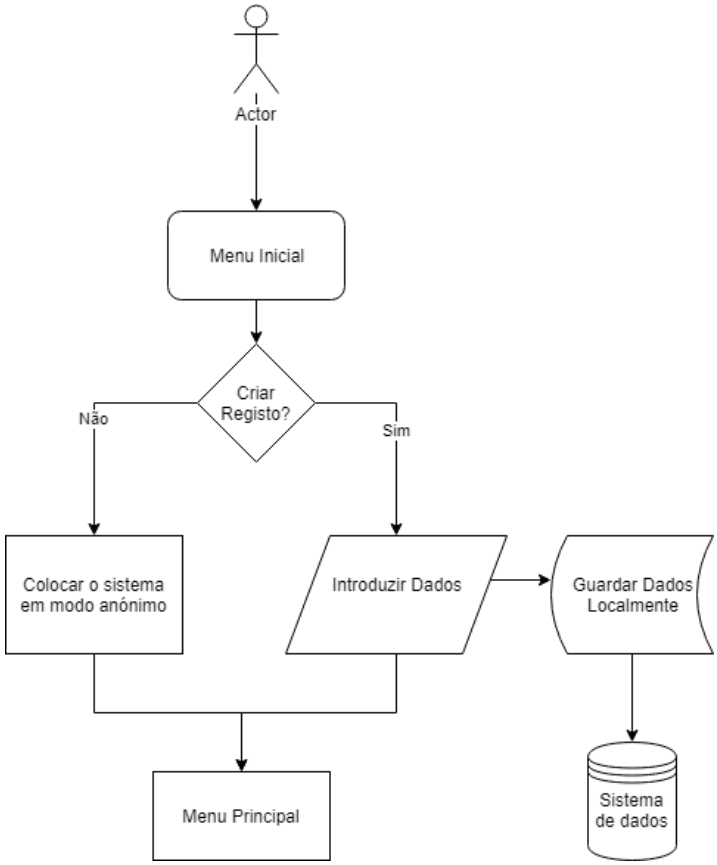


Figura 3.3- Fluxograma referente ao sistema de registo.

Uma vez concluído o sistema de registo, passar-se-á para o menu principal.

Este é o menu mais importante do jogo, onde o utilizador encontrará os seguintes submenus:

- **Menu de Desafio** - neste submenu o utilizador deve escolher qual é o nível que pretenderá jogar.
- **Menu de Créditos** - exposição dos criadores do protótipo ao utilizador.
- **Menu de Perfil** - deve apresentar ao utilizador todos os seus dados assim como a opção de deixar uma opinião sobre o jogo, apagar os seus extras criados no jogo ou permitir apagar o registo.
- **Menu de Sair**- deve permitir ao utilizador sair do jogo de uma maneira segura, guardando os seus dados à saída.

Este menu, figura 3.4, será o mais utilizado por parte do utilizador. Uma vez concluída a possível estruturação dos menus principais, é necessário proceder-se à modelação dos níveis que constituirão o protótipo.

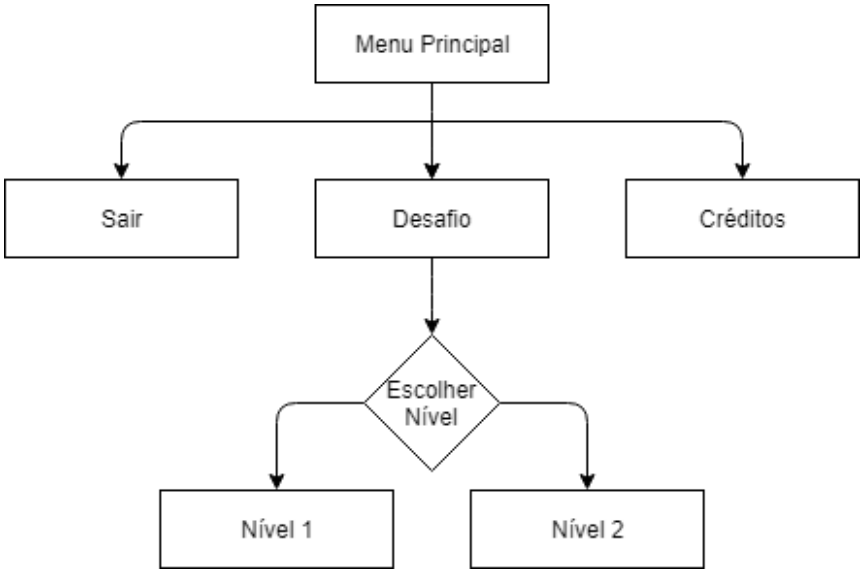


Figura 3.4-Menu principal do protótipo.

#### 3.2.2.1. Proposta para o Nível 1

O primeiro nível será o exercício corresponde à leitura de palavras. O nível terá como objetivo averiguar a velocidade de leitura dos utilizadores. A modelação inicial encontra-se no ilustrada no seguinte diagrama, figura 3.5.

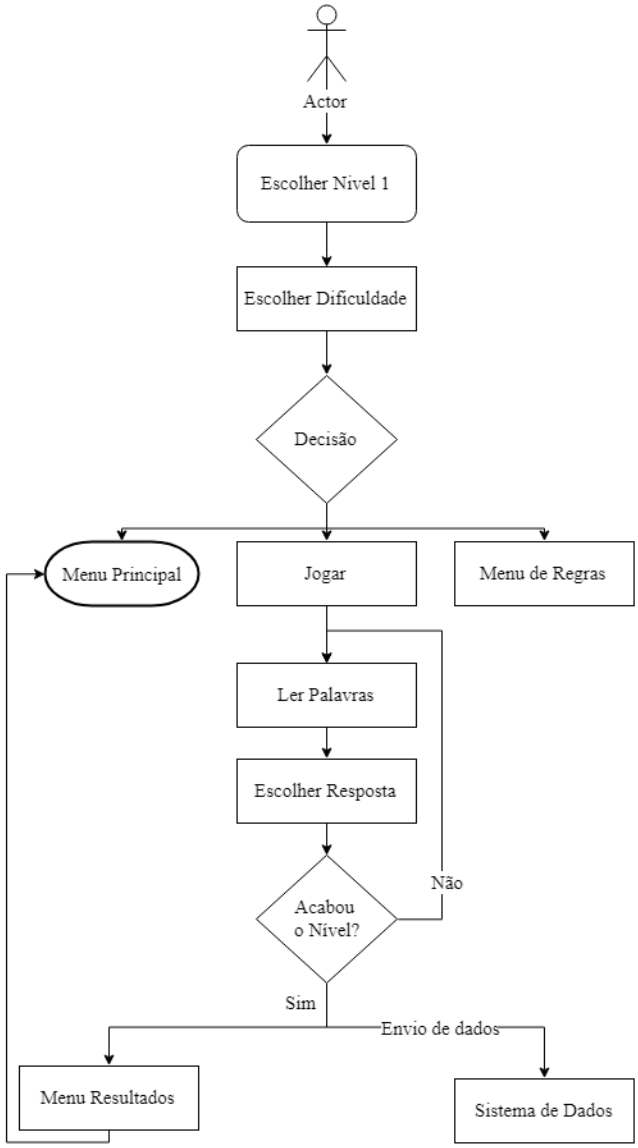


Figura 3.5-Diagrama de modelação do nível 1

Como é possível observar, na figura 3.5, o utilizador começará por escolher uma dificuldade. Neste nível existirão quatro dificuldades correspondentes aos anos de escolaridade:

- **Dificuldade Fácil:** Nesta dificuldade deverão existir palavras apenas com uma sílaba. Esta dificuldade, correspondente ao primeiro ano, tem com meta para a velocidade de leitura, que os utilizadores devem atingir, de 30 palavras por minuto.

- **Dificuldade Média:** Apenas inclui palavras com duas sílabas e corresponderá ao segundo ano, onde a velocidade de leitura será de 50 palavras por minuto.
- **Dificuldade Difícil:** Incluirá palavras com três sílabas e corresponde ao terceiro ano com uma meta na velocidade de leitura de 70 palavras por minuto.
- **Dificuldade Muito Difícil:** nesta dificuldade serão incluídas todas as palavras incluídas nas dificuldades anteriores de modo a consolidar a aprendizagem do aluno. A velocidade de leitura pretendida será de 85 palavras por minuto.

As velocidades iniciais correspondentes à figura 1.1, na secção 1.2, foram adaptadas de acordo com o *feedback* do Centro Diferenças.

Após a escolha da dificuldade, será mostrado ao utilizador um menu intermédio para se proceder ao início do jogo. Este menu também incluirá uma opção para mostrar as regras deste nível.

O objetivo de nível será ler, em voz alta, as palavras enquanto estas aparecem no ecrã. O número de palavras será aleatório, mas compreendido num intervalo de um mínimo de 5 palavras a 20 palavras no máximo. A disposição das palavras será sequencial no sentido dos ponteiros do relógio. No fim de um número aleatório de palavras será perguntado ao utilizador qual foi a última palavra que apareceu. A última palavra apresentada será a resposta correta. A velocidade com que é mostrada cada palavra dependerá da dificuldade escolhida.

Este exercício irá ser utilizado porque permite os utilizadores treinarem a sua atenção, devido às palavras aparecerem e desaparecerem, obrigando assim os utilizadores a prestarem atenção durante o nível, mas também permitir treinar os utilizadores para acompanharem o ritmo de leitura.

Para o utilizador pontuar, terá que acertar na palavra correta e ganhará 10 pontos no caso de estar correta ou 0 pontos no caso de ser incorreta a resposta.

Após a validação da escolha, o jogo deverá repetir-se até o utilizador responder a todas as perguntas. O número de perguntas será influenciado pela dificuldade.

Por fim, devem ser mostrados os seus resultados. Os resultados devem apresentar o número de perguntas propostas, o número de respostas certas e o número de respostas erradas. No caso de o utilizador possuir um registo, os resultados serão enviados também para o sistema de dados. Se o utilizador conseguir uma taxa de sucesso superior a 50% ser-lhe-á atribuída uma estrela no menu correspondente a este nível de modo a indicar que conseguiu completar o nível com sucesso. Se conseguir ultrapassar o seu melhor resultado, o resultado deve ser guardado como o novo resultado a ultrapassar.

No caso de o utilizador completar o jogo, será disponibilizada uma opção extra para adicionar novas palavras ao nível.

#### 3.2.2.2. Proposta para o Nível 2

O segundo nível terá como objetivo a identificação das temáticas textuais assim como avaliar a velocidade de leitura de textos. O nível irá ser estruturado segundo o seguinte diagrama, figura 3.6.

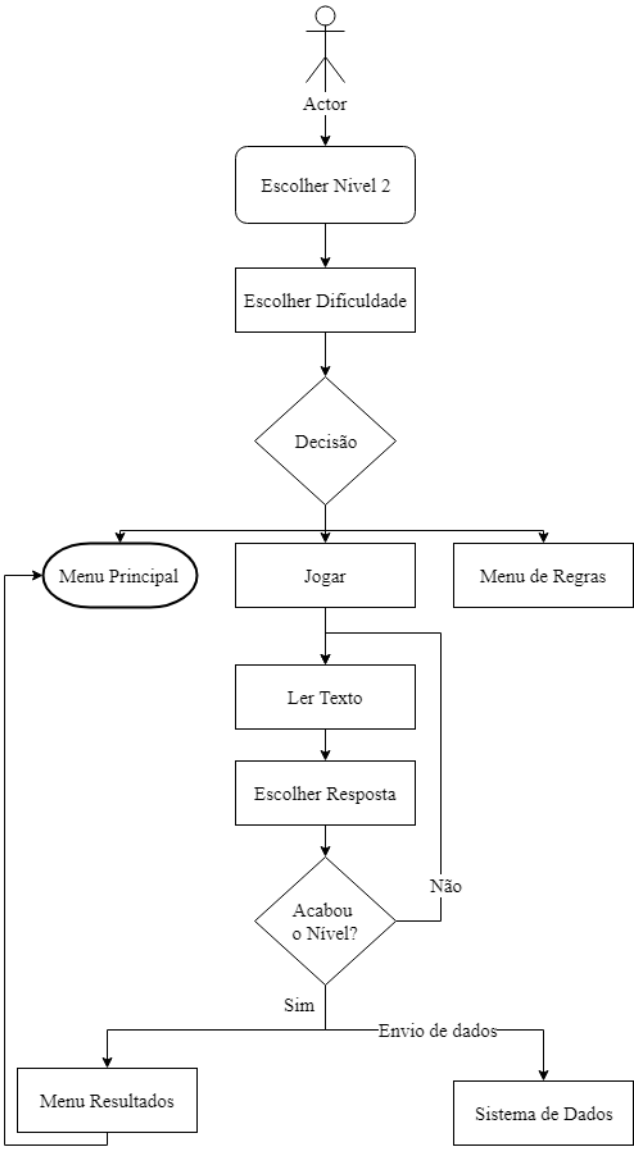


Figura 3.6-Diagrama de modelação do nível 2.

A estruturação do início do nível será semelhante ao nível anterior proposto, onde o utilizador terá que escolher uma dificuldade. As dificuldades devem incluir diferentes temáticas textuais para permitir o treino de diferentes textos. As

temáticas textuais, pertencentes a cada dificuldade, encontraram-se descritas nos pontos seguintes semelhantes ao nível anterior e variam segundo a dificuldade:

- **Dificuldade Fácil** - textos narrativos, textos descritivos, textos informativos e poemas. A velocidade de leitura nesta dificuldade é igual a 30 palavras por minuto.
- **Dificuldade Média** - Inclui as mesmas temáticas que a dificuldade fácil. A velocidade de leitura nesta dificuldade é igual a 50 palavras por minuto
- **Dificuldade Difícil** - textos narrativos, textos informativos, textos descritivos, notícias, cartas e convites. A velocidade de leitura nesta dificuldade é igual a 70 palavras por minuto
- **Dificuldade Muito Difícil** - textos narrativos, textos descritivos, retratos, notícias, cartas, convites, avisos e textos de enciclopédias. A velocidade de leitura nesta dificuldade é igual a 85 palavras por minuto

Os textos utilizados serão fornecidos pelo Centro Diferenças, uma vez, que são textos utilizados préviamente durante as sessões de acompanhamento das crianças no centro.

Após a escolha da dificuldade, será necessário a presença de um menu secundário para o início de jogo assim como uma opção para a exemplificação das regras correspondestes a este nível.

Este exercício foi escolhido para este nível pois permite incorporar a leitura de textos com o treino da velocidade de leitura dos utilizadores. De modo a facilitar a avaliação dos utilizadores, o nível será dividido em duas etapas. A divisão irá permitir aos utilizadores avaliarem a sua velocidade leitora, mas também os conhecimentos que possuem respetivamente às temáticas textuais. As etapas serão caracterizadas da seguinte maneira:

O início do nível, corresponderá á etapa 1, onde o jogador terá que ler um texto cronometrado. Uma vez concluída a leitura, o utilizador deverá confirmar o fim da leitura para proceder para a etapa seguinte. Nesta etapa será necessária

uma verificação para evitar que os jogadores prossigam para a etapa seguinte sem ler os textos. O tempo cronometrado e o número de palavras presentes no texto serão utilizados para calcular a velocidade de leitura do utilizador.

No fim do nível, correspondente à etapa 2, serão apresentadas três temáticas textuais para o jogador escolher qual é a temática textual correspondente ao texto que leu na etapa anterior. Esta etapa incluirá uma notificação para fornecer *feedback* ao utilizador, para verificar-se a sua resposta estava correta ou incorreta.

O procedimento correspondente às etapas 1 e 2 será repetido um número finito de vezes, na mesma ordem, consoante a dificuldade escolhida anteriormente. De modo a evitar a memorização dos textos por parte do utilizador, o mesmo texto não será escolhido duas vezes seguidas.

Se o utilizador acertar corretamente a temática textual irá receber 10 pontos e 0 pontos, no caso de a resposta estar errada. No início de cada etapa 2 será exposto ao utilizador a sua velocidade de leitura calculada, durante a leitura, na etapa 1.

Por fim, será mostrado ao jogador os seus resultados que irá conter o número de perguntas total, o número de respostas certas e erradas. Como um dos objetivos deste nível é também avaliar a velocidade leitora, será disponibilizado uma lista com as velocidades de leitura que obteve nos diferentes textos presentes, correspondentes à etapa 1. No caso de o jogador optar por um registo, será enviado juntamente com os resultados, a lista das velocidades obtidas para o sistema de dados. Posteriormente os dados enviados serão e mostrados ao utilizador através de um gráfico de linhas. Cada texto terá o seu gráfico específico e estarão disponíveis através do *Frontend* do sistema de dados.

### 3.2.3. Sistema de Dados

Por fim será descrito a modelação relativamente ao sistema de dados referido nas secções anteriores.

Uma vez que o jogo Reino dos Livros é um jogo sério, numa perspetiva de *Edutainment*, é necessário que existam avaliações de desempenho ao longo do jogo e, para tal, será desenvolvido um sistema de dados externo ao jogo. Este módulo será o mais importante para a avaliação dos utilizadores e será implementado segundo o seguinte diagrama de sequência, figura 3.7.

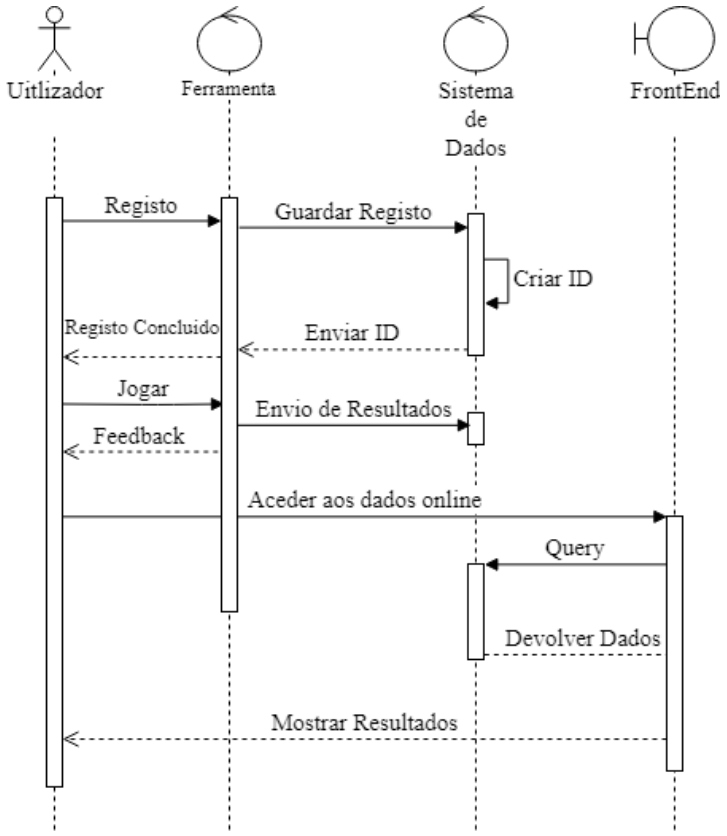


Figura 3.7-Diagrama de Sequência das comunicações entre os módulos do sistema.

O sistema de dados será responsável pela gestão de toda a informação relativa ao utilizador, no caso de optar por um registo inicial. No caso de não optar pelo registo, este módulo não estará disponível para aquele utilizador, embora o jogo se comporte de igual maneira para ambos os casos. O sistema será

importante uma vez que não é possível avaliar presencialmente todas as crianças, como no caso das sessões de ajuda tradicionais, logo o sistema devera ser autónomo de modo a substituir o registo externo por parte dos terapeutas.

O sistema será dividido em duas partes, relativamente ao utilizador e ao jogo, uma vez que ambos irão fazer pedidos de dados diferentes.

Relativamente ao jogador, serão enviados os dados tais como a sua idade e o nome escolhido, durante o registo, mas também os dados relativos ao seu desempenho nos diferentes níveis. Posteriormente o sistema de dados arquivará os dados utilizando uma base de dados. A gestão dos dados será feita pelo *Backend* do sistema de dados. O *Backend* do sistema de dados será implementado utilizando MySQL e PHP e é responsável pelos procedimentos de programação para transmitir a informação guardada na base de dados para o *Frontend*. Um dos requisitos mais procurados pelos terapeutas, no centro, seria a criação de um conjunto de gráficos correspondentes aos diferentes textos presentes no protótipo. Os gráficos permitiriam ilustrar as diferentes velocidades de leitura, para cada texto, que o utilizador irá obter ao longo das diferentes tentativas, referentes ao nível 2, facilitando assim o trabalho por parte dos terapeutas relativamente à leitura de textos. Estes gráficos serão mostrados no *Frontend* do sistema de dados.

Uma vez recolhidos os dados, estes serão processados e apresentados ao utilizador através de uma página online. A página constituirá todo o *Frontend* do sistema de dados de modo a permitir uma fácil visualização, dos dados.

Resumindo, relativamente ao jogo, existirá uma comunicação entre o sistema de dados e o jogo durante o registo do utilizador para impedir a criação de identificadores duplicados no registo assim como durante o envio de dados. No *Frontend* existirá também uma tabela correspondente aos melhores resultados que a criança obteve nas diferentes dificuldades e níveis.

De modo a facilitar a modelação da base de dados, pertencente ao sistema de dados, foi desenhado o seguinte modelo de relacionamentos e entidades presente na figura 3.8.

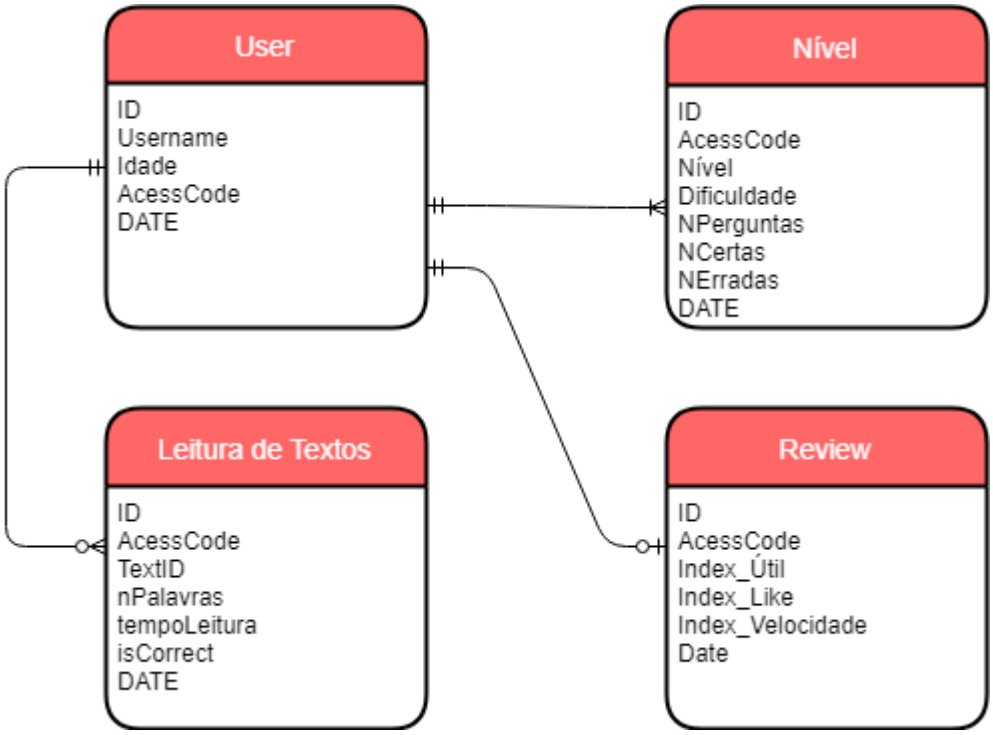


Figura 3.8-Diagrama de entidades e relacionamentos.

4

# 4. Implementação da Ferramenta

Neste capítulo do documento, será explicado o desenvolvimento do protótipo, que é baseado nos requisitos necessários presentes no capítulo anterior. Irá abordar-se também os *softwares* utilizados durante a criação do protótipo assim como as escolhas feitas durante o desenvolvimento do mesmo.

## 4.1. *Game-Engine*, Unreal Engine

Uma vez escolhidos os requisitos, é necessário proceder-se à escolha de um *Game-Engine* que nos permita cumprir os requisitos pretendidos. O *Game-Engine* é uma *framework* de criação de jogos que permite a programação das diferentes componentes presentes no jogo tais como os diferentes níveis, menus, mas também os comportamentos que os diferentes objetos irão ter ao longo do jogo. O objetivo principal é simplificar e diminuir o tempo necessário para o desenvolvimento de jogos.

Atualmente existem diferentes *Game-Engines* presentes no mercado, o que dificulta a escolha de um *Game-Engine*, para este trabalho, se apenas

observarmos a capacidade do mesmo para cumprir os requisitos [25]. Para simplificar e ajudar a limitar a escolha foram analisados os seguintes pontos:

- **Preço de utilização** - Apenas os *Game-Engines* gratuitos foram analisados, para este projeto, permitindo assim o desenvolvimento possível do projeto sem custos adicionais.
- **Multiplataforma** - O jogo pretende atingir o maior número de pessoas que se enquadrem no público alvo, logo é necessário que o *Game-Engine* seja capaz de exportar o protótipo implementado para diferentes plataformas escolhidas. As plataformas alvo são: sistema Android e sistema Windows.
- **Documentação de Ajuda** - sendo um *Game-Engine* uma *framework* com inúmeras funcionalidades préviamente implementadas é necessário que exista uma documentação de ajuda suficiente para permitir compreender as diferentes funcionalidades e métodos do *Game-Engine*.
- **Componente multidimensional** - O jogo irá conter elementos de diferentes dimensões :3D no caso do desenho dos níveis e 2D durante a criação de menus. É necessário que o *Game-Engine* permita incorporar as diferentes dimensões uma vez que alguns *Game-Engines* foram apenas implementados com uma dimensão (3D ou 2D) como foco.

Analisandos os pontos, préviamente referidos e a tabela 2.2 na secção 2.2.2, optou-se pela escolha do Unreal Engine, embora seja semelhante ao Unity3D, porque além de permitir cumprir os requisitos propostos, já tinha sido utilizado anteriormente pelo Autor para a criação de outros projetos com objetivos semelhantes.

## 4.2. Análise do *Game-Engine*

Uma vez escolhida a *framework* para o desenvolvimento do projeto, antes de ser explicada a implementação do protótipo, é necessário fazer uma breve introdução do Unreal Engine. A versão utilizada do *software* foi a 4.18.3 e

permaneceu a mesma de modo a evitar possíveis incompatibilidades durante o desenvolvimento do projeto, embora a ferramenta seja atualizada com regularidade com novas funcionalidades e melhor desempenho. Para facilitar a compreensão da ferramenta por parte do leitor, serão apresentadas diferentes imagens, acompanhadas de uma breve descrição, para facilitar a ilustração das diferentes componentes utilizadas na *framework* ao longo da implementação.

### 4.2.1. *Layout* Inicial da Ferramenta

Quando o *Game-Engine* é inicializado, semelhante à figura 4.1 a primeira *interface* é mostrada ao utilizador. Para evitar a confusão devido aos diferentes submenus presentes na *interface* inicial, a imagem encontra-se divida em quatro partes independentes, mas que durante a criação de um jogo todas se interligam e cumprem um papel importante.

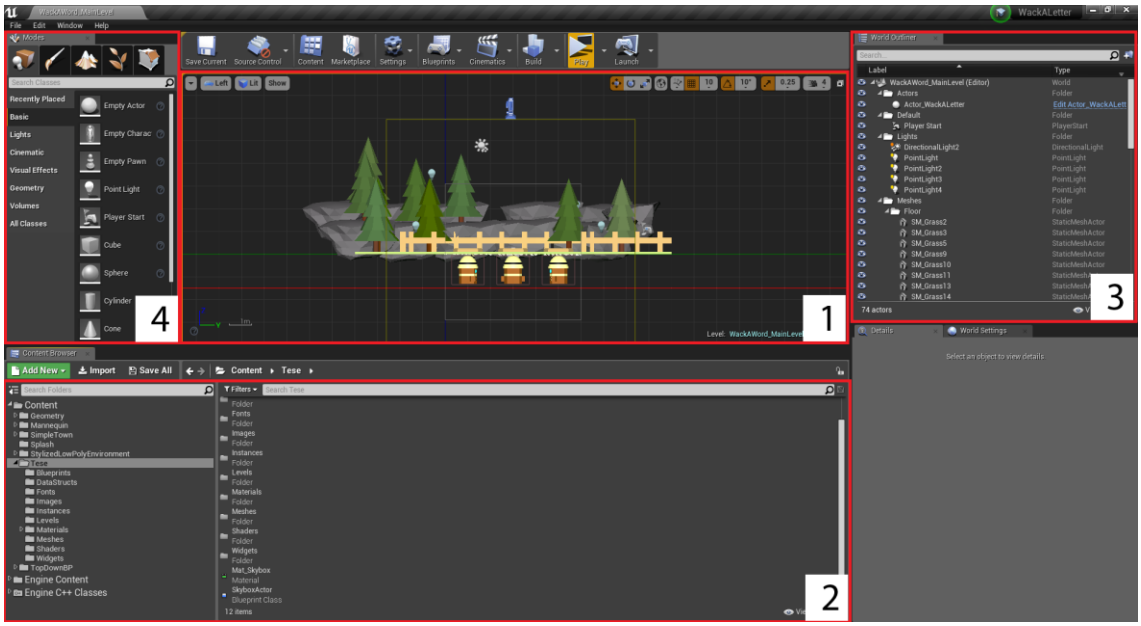


Figura 4.1- Interface inicial do Unreal Engine

Os pontos que se encontram na figura 4.1 são os seguintes:

1. ***Design* do jogo**: nesta secção o utilizador pode começar a colocar todos os seus objetos que desejar para a construção de um nível. Na imagem encontra-se um exemplo tirado durante a criação do primeiro nível. Inclui também, no canto superior esquerdo, um

conjunto de ferramentas de *debug* e controlo de desempenho assim como opções como a rotação da câmara e controlo de iluminação para observar como é que o desenho do jogo se enquadra em diferentes perspetivas. É a secção mais importante das presentes na imagem porque representa o que é mostrado ao utilizador.

2. **Gestão de ficheiros:** aqui encontram-se todos os ficheiros associados ao projeto para permitir a utilização dos mesmos ao utilizador. É bastante acessível uma vez que inclui diferentes exemplos de *assets*, que representam elementos préviamente criados como modelos 3D, texturas, sons e entres outros, para utilizar no *design* do jogo, mas permite também ao utilizador criar o seu sistema de pastas para organizar o projeto como considerar mais benéfico. No caso de o utilizador desejar utilizar um *asset* para o design do jogo apenas tem de o arrastar para a secção 1 da imagem para o colocar no nível.

3. **Gestão do nível**: esta seção é responsável por apresentar ao utilizador a hierarquia relativa aos objetos que o nível está a utilizar naquele momento. Permite ao utilizador gerir a organização do seu nível através de um sistema de pastas idêntico ao gestor de ficheiros e aceder rapidamente aos valores padrão inerentes a cada objeto utilizado.

4. **Lista de objetos base**: contém uma lista de diferentes objetos préviamente implementados que são muito utilizados no *Game-Engine*, pelos *developers* e a pela comunidade de utilizadores do Unreal Engine, para o utilizador escolher. É bastante útil porque permite testar diferentes funcionalidades comuns para a criação de jogos como iluminação, camaras, blocos provisórios como cubos, cilindros e outras formas geométricas entre outros, mas acessíveis e fáceis de adaptar pelo utilizador para incluir no nível.

### 4.2.2. *Interface* de Programação

Para o desenvolvimento deste projeto, foi utilizando uma linguagem de programação incluída no *Game-Engine*, chamada de Blueprints [26]. O sistema de Blueprints é um dos pontos que distingue esta *framework* das *frameworks* semelhantes, uma linguagem visual baseada no conceito de blocos (*nodes*) que permite ao utilizador criar os seus mecanismos utilizando algoritmos visuais através de diferentes blocos interligados de uma maneira simples. Esta linguagem é uma linguagem orientada pelos objetos, baseada em C++, onde cada bloco é equivalente ao código que seria produzido em C++ nativo. Embora a linguagem seja bastante eficiente, para projetos de grande escala, as *Blueprints* perdem desempenho, relativo à programação clássica, devido a ser necessário interpretar os diferentes blocos primeiro antes de executar o código, em C++, respetivo embora para este projeto o impacto no desempenho é mínimo.[27] [28]

Resumindo cada *Actor*, nomenclatura utilizada pelo **Unreal Engine** para representar um objeto no seu todo, será implementado através de *Blueprints* como na figura 4.2 e 4.3.

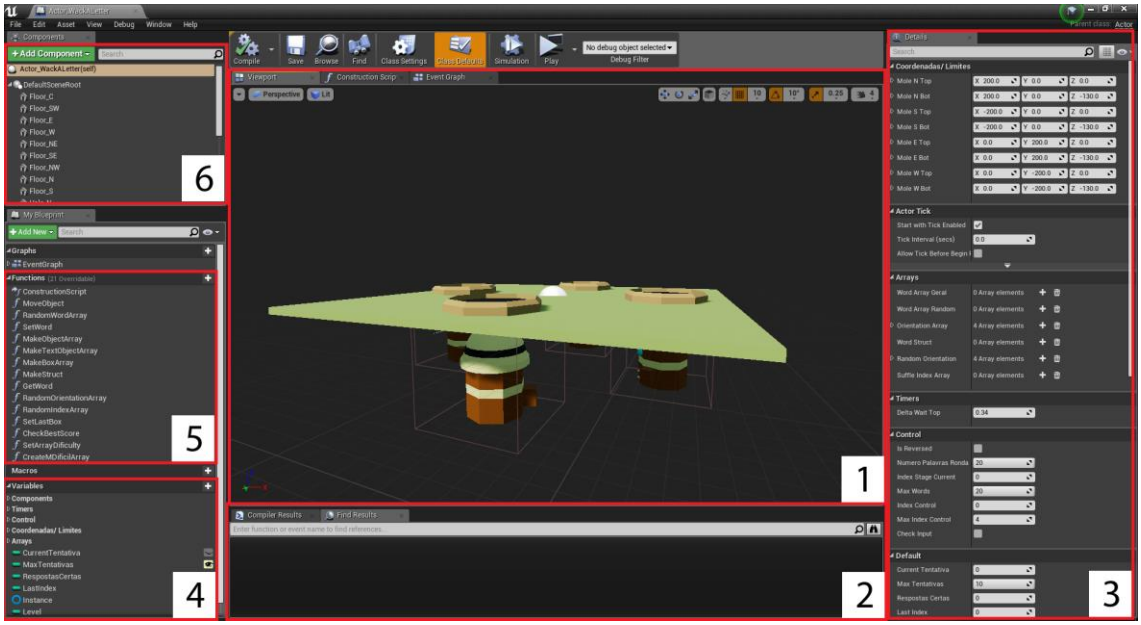


Figura 4.2-Interface de Programação.

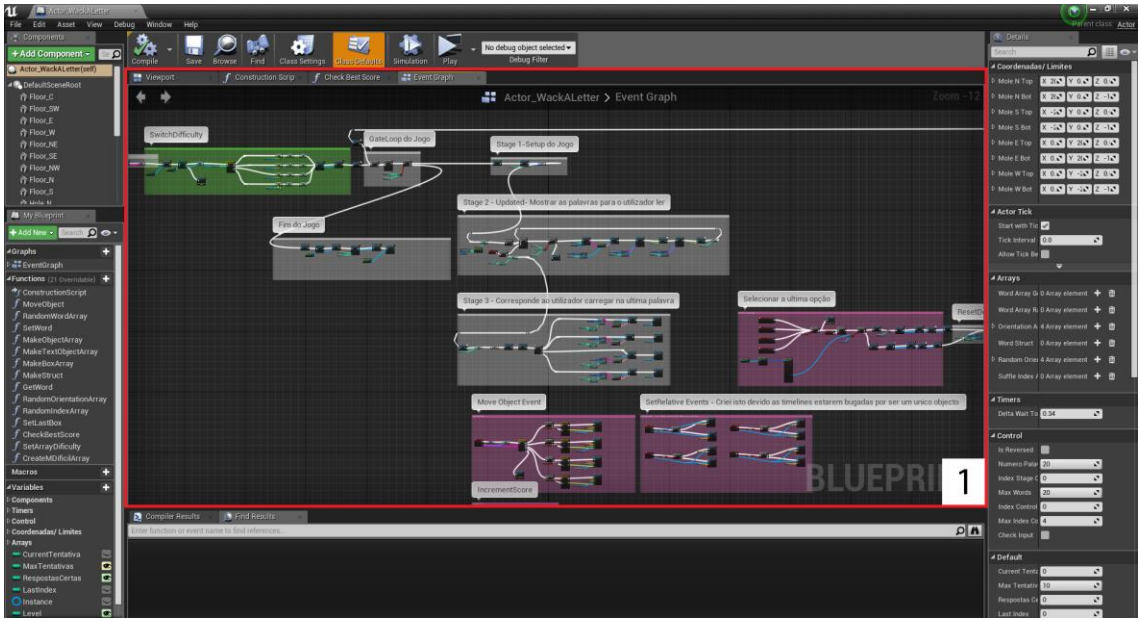


Figura 4.3- Programação de Blueprints

A interface de programação, figura 4.2 e 4.3, incluem as mesmas secções à exceção da secção 1, para ajudar o utilizador durante o desenvolvimento.

1. ***Interface* Visual**: Esta secção encontra-se divida em duas partes. Na primeira parte, figura 4.2, permite ao programador observar uma representação visual, em *3D*, de como o *Actor* será visto no nível. Permite também ao utilizador colocar todas as componentes secundárias (*Child*) que estarão relacionadas com a *Root*, componente principal do *Actor*. Referente á segunda parte, figura 4.3, será apresentado ao utilizador o *EventGraph* para começar a implementar os mecanismos que o Actor deve cumprir através de funções e eventos.

2. **Janela do Compilador:** Permite ao utilizador receber *feedback* do compilador para averiguar se ocorrerão erros durante a compilação do *Actor*. Por outro lado, permite ao utilizador procurar todas as instâncias que um bloco específico apareceu no *EventGraph* para diminuir o tempo que o utilizador demoraria a procurar manualmente todos os blocos que precisa.

3. ***Default Values***: corresponde a uma lista que mostra todos os valores de inicialização utilizados pelo *Actor*. É utilizado para permitir ao utilizador conseguir verificar todos os valores definidos, *default*,

de uma maneira coesa e concentrada de modo a aplicar alterações aos valores se necessário, sem perder tempo que seria gasto se o utilizador tivesse que ir a todas as variáveis mudar presencialmente no código.

4. **Variáveis**: permite ao utilizador criar novas variáveis assim como gerir todas as variáveis globais do *Actor*. No caso das funções, inclui uma divisória extra que contêm apenas as variáveis locais presentes na função.

5. **Lista de funções:** permite ao utilizador criar as funções necessárias para os mecanismos do *Actor* e de igual modo, listar todas as funções implementadas. Quando o utilizador acabar de implementar a sua função, numa janela do EventGraph, específica para funções, apenas tem que arrastar da lista para a secção 1 para criar um bloco (*node*) referente à função implementada para interligar com o restante esquema.

6. **Hierarquia de componentes:** começa com a componente principal, *Root*, que é única e apenas existe uma em cada *Actor*, e permite ao utilizador acrescentar novas componentes, disponibilizadas pelo *Game-Engine*, ao seu *Actor* para gerar os mecanismos necessários ou por vezes ajudar no debug. Embora o utilizador não seja obrigado a incluir componentes adicionais, como no caso de um *Actor* que apenas funcione como controlador, é sempre incluída na hierarquia a componente principal, *Root*, devido à *Root* ser responsável pela transformada absoluta (translação - T, rotação - R e fator de escala - S) do objeto no nível.

Por fim, é possível verificar que o Unreal Engine disponibiliza inúmeras funcionalidades acessíveis para os utilizadores e se necessário, é possível incluir *plugins*, ou seja, bibliotecas externas de funções para o utilizador incluir no seu projeto. Para a implementação deste protótipo foi escolhido utilizar um *plugin* para executar a comunicação entre o jogo e o sistema de dados, explicado na secção 4.2.3.

### 4.2.3. VARest - Plugin para o Unreal Engine

O registo de dados é um dos requisitos principais do jogo, logo é necessário criar uma ligação entre o jogo e a base de dados. Infelizmente, as *Blueprints* não dispõem de nenhum bloco para efetuar envio de dados para o exterior do jogo o que obrigou a utilizar para a implementação da comunicação, entre o protótipo e o sistema de dados, o *plugin* VARest. Este plugin encontra-se disponível gratuito na loja da Unreal Engine [29].

O VARest utiliza uma arquitetura *REST* [26] que permite separar o jogo do sistema de dados e criar uma comunicação entre o jogo e o exterior através de pedidos HTTP/HTTPS para o envio de dados. A vantagem principal é conseguir incluir diretamente na linguagem de *Blueprints* a capacidade de efetuar pedidos HTTP/HTTPS facilmente. Uma vez que o envio de dados é assíncrono, o *plugin* também disponibiliza *a priori* eventos para saber se o pedido foi enviado com sucesso ou insucesso de modo a não bloquear o jogo enquanto o pedido é feito.

Para o envio dos dados contidos dentro do pedido HTTP foi utilizado a linguagem JavaScript Object Notation (JSON) [26] porque é uma linguagem leve e permite compactar a informação em pares com o nome da variável e o seu valor e integrar em diferentes sistemas mesmo que não se encontrem implementados com a mesma linguagem de programação. Neste caso é utilizado para comunicar entre *Blueprints* (C++) e PHP.

O procedimento utilizado para criar uma mensagem JSON através de *Blueprints* e o VARest para ser enviado no pedido HTTP encontra-se na figura 4.4.

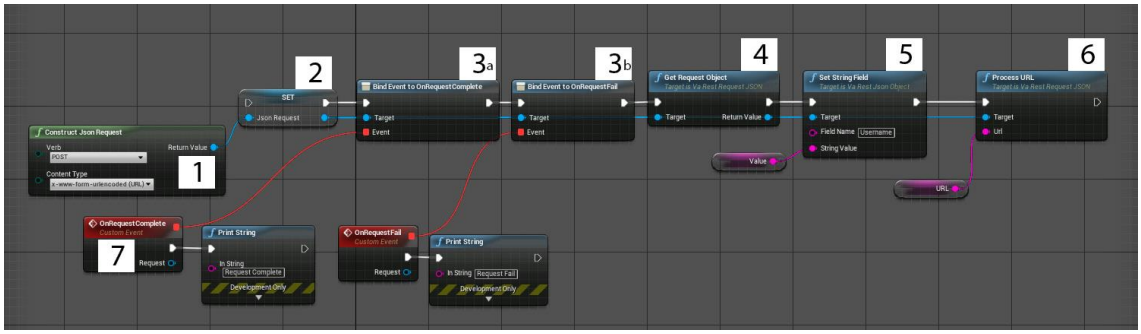


Figura 4.4-Procedimento para criar uma mensagem JSON no jogo.

De seguida, encontra-se o procedimento presente na figura 4.4 explicado nos seguintes pontos:

1. **Construct JSON Request:** permite inicializar um pedido HTTP. Neste bloco é necessário especificar o verbo HTTP que se pretende utilizar. Os verbos disponíveis, no *plugin*, são: *GET, POST, PUT, DELETE*. Para a criação do protótipo foi utilizado o verbo *POST* durante a comunicação. [27]
2. **SET Variable**: Guardar o pedido recém-criado numa variável auxiliar.
3. **Bind Event:** permite associar a dois eventos para evitar o jogo ter de esperar pelo pedido uma vez que é assíncrono.
    a. **OnRequestComplete** - executa o código associado se o pedido por enviado com sucesso.
    b. **OnRequestFail** - executa o código associado se o envio do pedido falhar.
4. **Set Request Object**: permite aceder ao objeto JSON associado ao pedido HTTP criado no ponto 1.
5. **Set String Field:** este bloco permite criar os pares de dados que serão enviados na mensagem JSON. Para criar um par é necessário introduzir o nome da variável no campo *Fieldname* que corresponde à chave do par e por fim colocar o valor da variável no campo *StringValue* para completar o par de dados.
6. **Process URL:** por fim é necessário enviar o pedido criado. A mensagem será enviada para o caminho online correspondente ao campo *URL*.
7. **Resposta do Pedido**: Se necessário, é possível guardar a resposta do pedido através do campo *RequestContent* presente em ambos os eventos anteriormente explicados.

Este procedimento é idêntico para o envio dos diferentes dados do jogo uma vez que a única alteração é o número de vezes que será utilizado o bloco *SetStringField* para criar os diferentes pares de variáveis.

## 4.3. Implementação do protótipo

Nesta secção do documento serão explicados os métodos utilizados ao longo do desenvolver do protótipo e o funcionamento das diferentes componentes do jogo. A implementação segue os requisitos propostos na secção 3.2.2 do capítulo 3 com algumas alterações baseadas no *feedback* recolhido no Centro Diferenças.

### *4.3.1. Game-Instance*

Antes de se proceder à explicação do protótipo é necessário explicar o que é a *Game-Instance* e como é que influenciou o desenvolver do jogo.

A *Game-Instance* é a classe mais importante do jogo, uma vez que permite guardar todas as variáveis importantes para o funcionamento do jogo, ou seja, funciona como um registo global para que qualquer *Blueprint* seja capaz de conseguir aceder a estes valores. A vantagem principal foi permitir guardar variáveis entre níveis porque sempre que existe uma troca de níveis, o nível antigo é destruído e como consequência todos os *Actors* nele presente são destruídos. Assim é possível manter a informação porque a *Game-Instance* encontra-se sempre disponível. Esta classe é responsável por implementar as funções relativas ao *Save Game*. Apenas existe uma *Game-Instance* ao longo do jogo, ou seja, o objeto associado a esta classe é criado quando o jogo é inicializado e é destruído quando é fechado o jogo, o que obriga à criação de um *Save Game* para guardar todas os estados booleanos ,*flags* de controlo, *strings* e outras variáveis importantes da *Game-Instance* para que na próxima utilização, o jogo se mantenha igual.

### 4.3.2. Sistema de Login

Sempre que o jogo é inicializado, o sistema começa por efetuar uma verificação se existe um ficheiro .SAV na pasta de *Save Games*. Um *Save Game* é um objeto que inclui as diferentes variáveis relacionadas com o utilizador como os seus melhores resultados, nome, idade e muitos outros, mas também guarda a

informação relativa a que níveis foram concluídos assim como as flags de controlo associadas ao jogo. Este objeto é atualizado inúmeras vezes ao longo da sessão de utilização. Quando termina a sessão, é criado o ficheiro .sav que irá conter todas as informações relativas ao objeto *Save Game*.

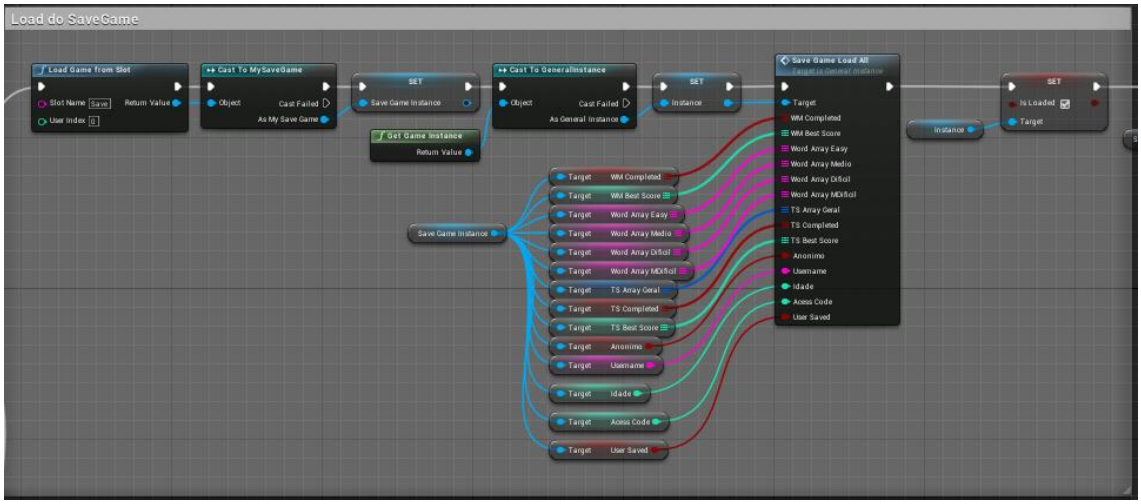


Figura 4.5 - Load de um Savegame.

No caso de existir um *Save Game* na pasta significa que o utilizador já jogou préviamente, logo o procedimento ilustrado na figura 4.5, é ler o ficheiro .SAV presente na pasta e sincronizar os dados antigos com os dados presentes na ferramenta utilizando a função ***SaveGame Load All***. Está função atualiza a *Game-Instance* atual com os valores guardados no ficheiro. Simultaneamente é ativado a flag de controlo *isLoaded* para indicar que existe um registo prévio, para não propor ao utilizador o menu de registo para a criação de um novo registo.

Para efeitos de *backup* e para evitar possíveis perdas de sincronismo entre os valores guardados localmente na *Game-Instance* e os valores presentes na base de dados, o sistema de login cria um pedido HTTP, procedimento presente na secção 4.2.3, com os dados relativos aos melhores resultados do utilizador para atualizar os valores presentes na base de dados, de modo a ficarem coerentes com o jogo.

Quando o pedido é recebido pelo *Backend* do sistema de dados, para fazer a atualização dos dados, são executadas as seguintes *SQL queries*. As *queries* [26] correspondem a uma pergunta formal que é feita à base de dados para aceder às informações presentes na base de dados e fazer operações como introduzir, selecionar, apagar e atualizar os dados.

```
1.  $query_update_WM='UPDATE user SET WMCOMPLETED="'.$wmcompleted.'",
2.  WME="'.$wmbestscore_array[0].'",
3.  WMM="'.$wmbestscore_array[1].'",
4.  WMD="'.$wmbestscore_array[2].'",
5.  WMMD="'.$wmbestscore_array[3].'"
6.  where acesscode="'.$acesscode.'"';
7.
8.  $query_update_TS='UPDATE user SET TSCOMPLETED="'.$tscompleted.'",
9.  TSE="'.$tsbestscore_array[0].'",
10. TSM="'.$tsbestscore_array[1].'",
11. TSD="'.$tsbestscore_array[2].'",
12. TSMD="'.$tsbestscore_array[3].'"
13. where acesscode="'.$acesscode.'"';
```

No caso de não existir nenhum ficheiro .SAV válido, a flag *isLoaded* é colocada a falso.

Por fim o utilizador é enviado para o menu principal na secção 4.3.4 ou para o menu de registo presente na secção 4.3.3. O caminho escolhido depende do estado atual da flag *isLoaded.*

### 4.3.3. Sistema de Registo

Como referido anteriormente, o protótipo necessita de um sistema de registos para criar a identificação do utilizador.

O utilizador apenas consegue interagir com este menu, figura 4.6, se o sistema de *login* colocar a flag *isLoaded* préviamente a falso.

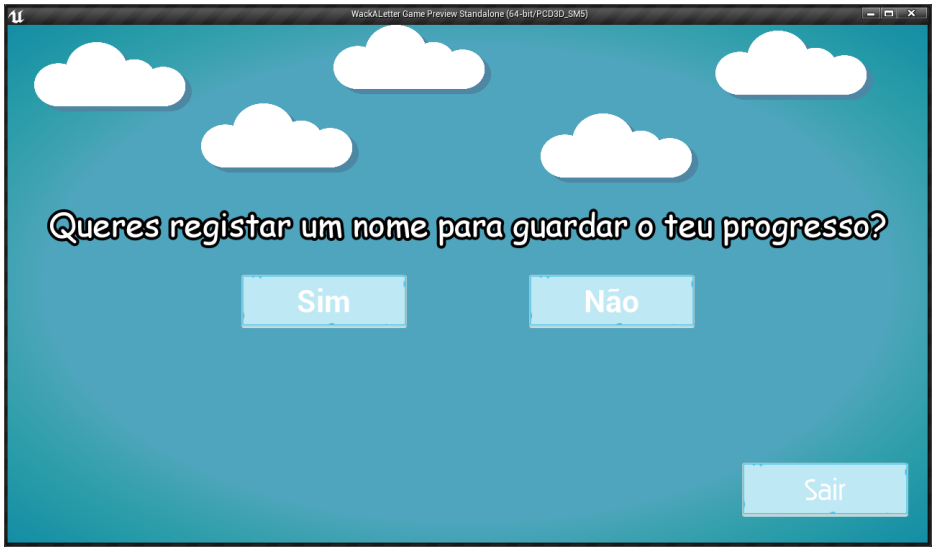


Figura 4.6-Menu introdutório para o registo do utilizador.

Como não existe um registo prévio, verificado pela flag *isLoaded*, é necessário perguntar ao utilizador se quer efetuar um registo para permitir guardar o seu progresso na base de dados. As consequências destas escolhas encontram-se explicadas na secção 3.2.1. Assim é possível não restringir a jogabilidade à criança e garantir a sua privacidade. Se o utilizador optar por não querer um registo, a flag *Modo Anónimo* será colocada a *true* para transmitir ao jogo que não é necessário enviar os resultados no fim dos níveis para a base de dados e ajustar o menu de perfil para não incluir informações referentes aos dados criados durante o registo tais como: o nome, a idade e código de acesso. Assumindo que o utilizador deseja criar um registo é mostrado o seguinte menu na figura 4.7.

Figura 4.7-Menu de registo do utilizador.

Neste menu, é necessário introduzir um nome e a respetiva idade. O nome é apenas um nome ilustrativo visto que o que vai identificar o utilizador é o seu *ID*, criado no fim do registo. A idade permite posteriormente restringir os dados a uma faixa etária se necessário.

Após a introdução dos dados, é necessário carregar no botão "Continuar" para validar e verificar se foram introduzidos corretamente. O jogo disponibiliza algumas verificações de segurança como não introduzir números durante o registo do nome ou nomes com demasiados caracteres, mas se o nome respeitar as condições anteriores, o utilizador tem a liberdade de criar o nome que desejar. No caso de os dados não serem válidos, por exemplo introduzir um nome vazio, é mostrado uma mensagem de erro ao utilizador para que este possa alterar para os valores corretos. O botão "Voltar" permite ao utilizador cancelar o registo e voltar ao menu anterior com a escolha de registo.

Após a validação dos dados, são guardados na *Game-Instance*, secção 4.3.1, para estarem disponíveis em qualquer parte do jogo como explicado.

Quando o registo é concluído, é necessário criar um pedido ao sistema de dados para obter o *ID* associado a este registo. O *ID* é obtido através do *Request Response* utilizando o procedimento descrito na secção 4.2.3. Juntamente com o pedido de registo será enviado também o nome escolhido e a idade introduzida para ficarem associados ao ID criado.

Para efeitos de protótipo, o *ID* é gerado através da contagem do número de utilizadores já registados na base de dados e incrementado uma vez. Isto permite que os *ID* não se repitam, mas torna-os sequenciais. Para gerar o código de acesso (*ID*) e o registo do utilizador são utilizadas as seguintes *queries*.

```
1.  $query_select= 'SELECT id from user where 1';
2.  $results = mysqli_query($connect,$query_select)or die(mysqli_error($con-
    nect));
3.  $rows=mysqli_num_rows($results);
4.  $acesscode =  $rows+ 1;
5.
6.  $query_checkUser="INSERT into user (USERNAME,IDADE,ACESSCODE,DATE)
7.  VALUES ('".$username."' ,'".$idade."','".$acesscode."','".$date."')";
8.
9.  $results = mysqli_query($connect,$query_checkUser)or die(mysqli_error($con-
    nect));
```

O sistema de dados envia de volta o *ID* como resposta ao pedido, que é guardado na *Game-Instance*. A partir deste momento o jogo apenas envia o ID juntamente com os restantes dados de modo a associar os dados ao utilizador e evitar estar sempre a enviar o nome escolhido, que pode causar problemas se existirem nomes iguais. De seguida a flag *UserSaved* é colocada a *true* para representar que o registo do utilizador foi completo com sucesso.

No caso de algum pedido falhar, é colocada a flag *UserSaved* a *false* para que o sistema envie, de novo, os dados na próxima utilização para conseguir receber um *ID* novo.

### 4.3.4. Menu Principal

Nesta secção é ilustrado o menu principal, figura 4.8, que é acedido pelo utilizador após a conclusão de todo o processo de registo ou ausência de registo.

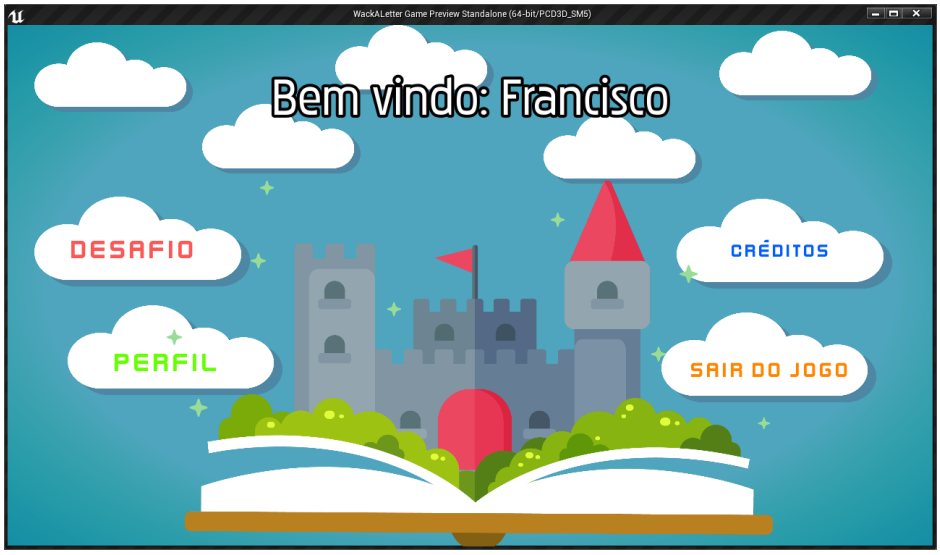


Figura 4.8-Menu principal do jogo.

Este menu serve para introduzir o utilizador às diferentes funcionalidades presentes no jogo através de um conjunto de botões que permitem aceder aos diferentes submenus. Como é possível observar, o menu respeita a estruturação definida na secção 3.2.

Para o utilizador começar a jogar, primeiro tem que carregar no botão correspondente ao "Desafio" para acender aos submenus correspondentes aos níveis. A figura 4.9 ilustra o submenu do primeiro nível.

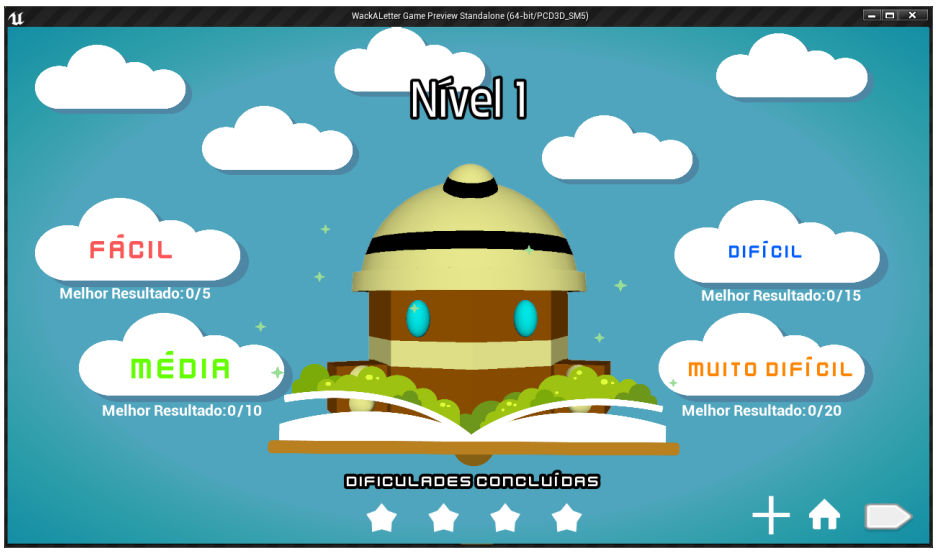


Figura 4.9-Menu de exemplo correspondente ao nível.

Este menu permite escolher a dificuldade do nível. A dificuldade é escolhida quando o utilizador carregar na nuvem correspondente à opção desejada. Permite mostrar ao utilizador os seus melhores resultados nas diferentes dificuldades, sem que seja necessário retroceder para o menu de perfil.

No caso de o utilizador completar um nível com uma taxa de sucesso superior ou igual a 50% é atualizado a cor da estrela correspondente à dificuldade para indicar que conseguiu completar o nível com sucesso naquela dificuldade.

O ícone correspondente à casa permite ao utilizador voltar ao menu principal e a o ícone da seta permite ao utilizador navegar para o menu do nível dois. Ambos os menus, correspondentes aos níveis, se comportam da mesma maneira.

No caso do primeiro nível é disponibilizado uma opção extra para permitir ao utilizador introduzir novas palavras que serão incluídas no nível. Para as introduzir é necessário carregar no ícone que correspondente ao "+" para ser reencaminhado para o submenu correspondente à figura 4.10.

Figura 4.10-Menu para adicionar uma nova palavra.

Este submenu permite introduzir uma nova palavra para a dificuldade que escolher. A dificuldade "muito difícil" está bloqueada porque esta dificuldade apenas utiliza palavras presentes nas outras dificuldades. Para efeitos de prova de conceito este menu encontra-se disponível sem qualquer segurança para que

os terapeutas conseguissem adaptar o jogo para uma criança em especifico, se necessário, mas como trabalho futuro seria importante restringir este menu através de um possível código ou uma implementação semelhante para garantir que as crianças não seriam capazes de introduzir palavras muito difíceis para elas contribuindo assim para um desinteresse por parte das crianças.

No caso de ser introduzida uma palavra errada ou muito difícil ou na dificuldade errada, no menu de perfil é disponibilizado uma opção para apagar todos os extras, neste caso as palavras, permitindo assim que o jogo volte ao seu estado inicial.

### 4.3.5. Implementação do Nível 1

O nível 1 corresponde à leitura de palavras com o objetivo de treinar a velocidade e atenção dos utilizadores ao longo da leitura. Antes do início do nível é mostrado ao utilizador as regras de como se comporta e é jogado o nível. As regras são as seguintes:

1. Ler as palavras e memorizar a última palavra.
2. Carregar na última palavra que apareceu.

O jogo começa por analisar qual foi a dificuldade escolhida através do valor presente na *Game-Instance* referente à dificuldade. Dependendo do valor, o nível adapta-se utilizando uma lista de palavras diferente para cada dificuldade e começa por definir o número máximo de tentativas. Cada tentativa corresponde a uma ronda. O nível repete-se até o utilizador completar um número, préviamente definido pela dificuldade, de rondas.

Anteriormente ao início da primeira ronda o jogo começa por aceder à lista de palavras e constrói uma estrutura temporária com 20 palavras escolhidas aleatoriamente e sem repetição da lista inicial e as diferentes transformadas para o posicionamento das palavras. As transformadas representam a posição nos diferentes eixos (X, Y, Z) assim como a sua rotação e o tamanho no nível. As posições definidas são as seguintes: Norte, Este, Sul e Oeste. Quando o procedimento é

completo, o objetivo seguinte é mostrar as diferentes palavras escolhidas ao utilizador.

Para mostrar as palavras ao utilizador, foi criado um modelo *3D*, figura 4.11, responsável por sair do buraco, mostrar a palavra ao utilizador e voltar a desaparecer. O modelo foi criado utilizado uma ferramenta externa chamada Autodesk Maya [30].

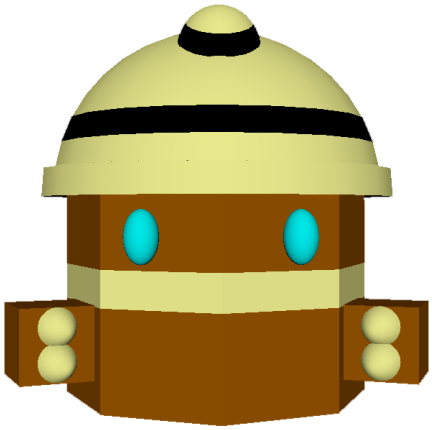

Figura 4.11-Modelo *3D* presente no nível 1.

Este mecanismo já tinha sido implementado em diferentes jogos semelhantes e foi bastante útil porque permite treinar a atenção dos utilizadores. Existe um *delay* entre a subida e a descida do modelo *3D* para garantir que a criança tem tempo para ler a palavra. Este mecanismo encontra-se na figura 4.12.

Figura 4.12 - Mecanismo inicial para mostrar as palavras.

Para a modelação deste comportamento foi utilizado uma funcionalidade do Unreal Engine chamada de *Timelines* [25]. As *Timelines* permitem fazer interpolações entre duas posições e permite definir o tempo entre o ponto inicial, que corresponde à posição inicial dentro do buraco, e o ponto final correspondente à posição final onde é mostrada a palavra. Para este nível foi utilizada uma *Timeline* com um tempo entre pontos de 0.33 segundos, ou seja, o tempo de subida e descida é o mesmo e encontra-se visível na figura 4.14. Este valor foi obtido experimentalmente com o apoio dos terapeutas do centro Diferenças.

No entanto, para adaptar o comportamento às diferentes velocidades de leitura propostas foi adicionado um *delay node* extra com valores calculados préviamente para que o tempo que a palavra se encontra visível seja semelhante à velocidade de leitura pretendida. Como foi utilizado a mesma *Timeline* para fazer o comportamento oposto, do ponto final para o inicial, apenas é necessário executar o pino *Reverse* presente no *node* das *Timelines*. O comportamento implementado, para a subida das palavras, pelas *Timelines* é demonstrado na figura 4.13.

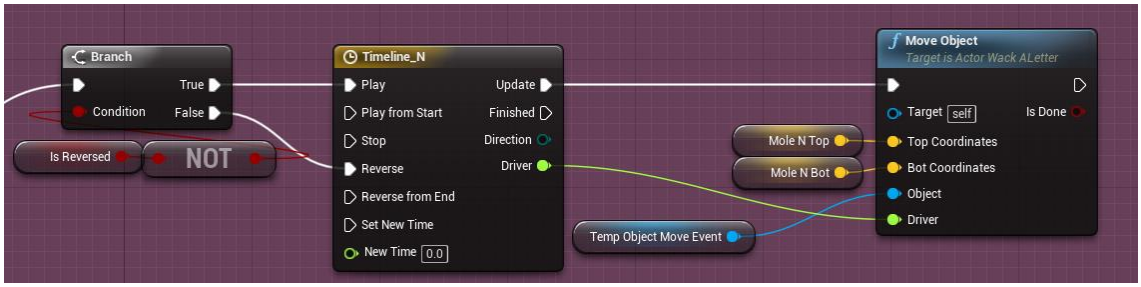


Figura 4.13 - Procedimento das *Timelines* para a subida da palavra.

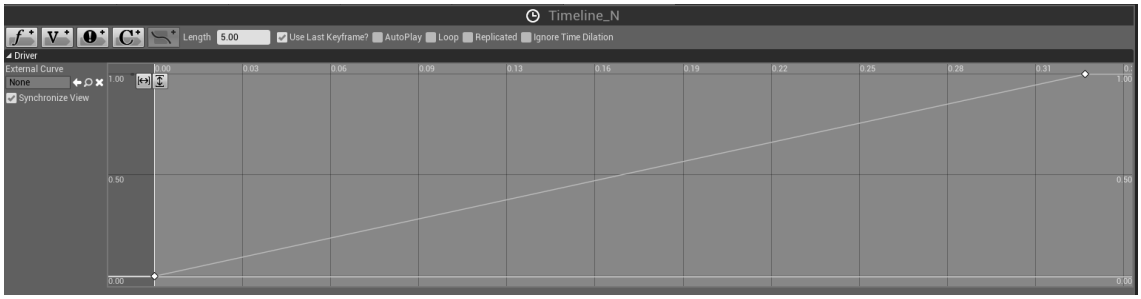


Figura 4.14 - Driver da Timeline que corresponde a uma rampa.

As palavras aparecem segundo os ponteiros do relógio e posição das palavras segue a sequência "Norte->Este->Sul->Oeste" e sempre com o mesmo

ritmo. Ao fim de um número aleatório de palavras mostradas, entre 5 a 20, é perguntado ao jogador qual foi a última palavra que apareceu anteriormente como mostra a figura 4.15 para o jogador carregar.

Figura 4.15 - Escolha da palavra no nível 1

Se o jogador carregar na palavra correta, ganha 10 pontos e recebe *feedback* positivo através de uma mensagem de aviso. No caso de o jogador responder incorretamente recebe 0 pontos e uma mensagem de *feedback* negativo para o jogador compreender que estava errado.

Após a escolha da palavra, é incrementado o número correspondente ao progresso no canto superior esquerdo do HUD[3] para indicar que a ronda acabou e o jogo retorna ao início onde coloca todos os modelos na posição inicial e escolhe novamente 20 palavras aleatórias para mostrar na próxima ronda. O jogo repete-se consoante o número de rondas definidas pela dificuldade.

[3] HUD-Interface responsável por mostrar ao utilizador diferentes valores no ecrã.

Quando o jogador responder a todas as rondas é mostrado ao jogador o seguinte menu de resultados. O menu encontra-se na figura 4.16.

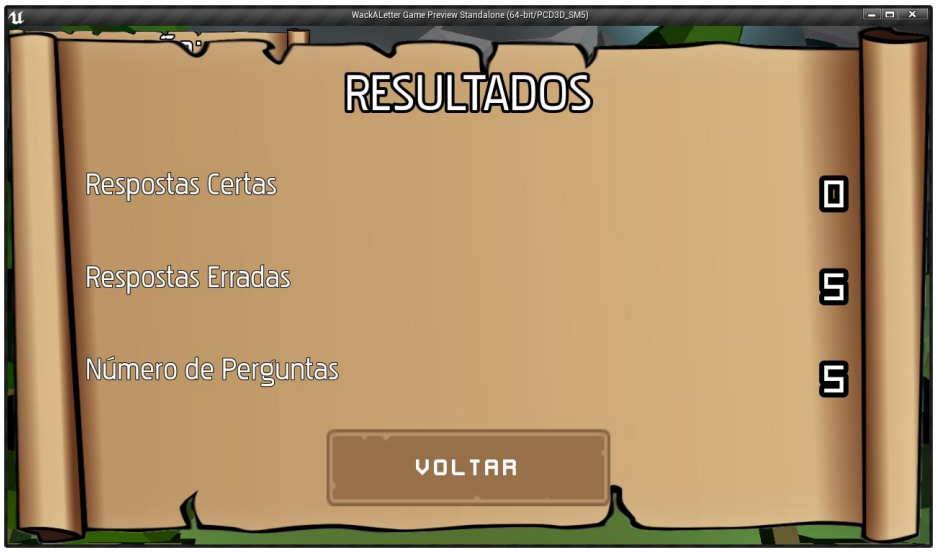


Figura 4.16-Menu de resultados do primeiro nível.

No menu de resultados é mostrado o número de respostas certas que o jogador obteve nas diferentes rondas assim como o número de respostas erradas e o número total de rondas que o jogador jogou.

Simultaneamente enquanto é mostrado os resultados ao jogador é criado um pedido, procedimento na secção 4.2.3, com os resultados, para enviar para o sistema de dados. O pedido é processado pelo *Backend* e introduzido na base de dados através da seguinte *query*.

```
1. $query_insertLevel ="INSERT INTO level (acesscode,nivel,dificuldade,nper-
   guntas,ncertas,nerradas,date)
2. VALUES
3. ('".$acesscode."' , '".$nivel."' , '".$dificuldade."' , '".$npergun-
   tas."' , '".$ncertas."' ,'".$nerradas."' , '".$date."')";
```

Por fim é avaliado se os resultados obtidos são melhores do que os resultados anteriores guardados na *Game-Instance* e atualizados no caso de o utilizador conseguir melhorar. Se for a primeira vez que o utilizador conseguiu uma taxa de sucesso superior a 50% é lhe atribuído uma estrela no menu principal.

Se o jogador desejar sair do jogo, deve carregar no ícone no canto inferior direito para colocar o jogo em pausa e permitir ao jogador sair do nível ou do jogo por completo com segurança. No caso de o jogador desistir, os dados correspondentes a este nível não serão guardados.

### 4.3.6. Implementação do Nível 2

O nível 2 corresponde à leitura de textos. O objetivo principal deste nível é treinar a velocidade de leitura dos utilizadores e treinar a associação entre os textos e a sua temática textual respetiva. À semelhança do nível anterior, o nível começa com um pequeno menu introdutório para explicar as regras do mesmo.

As regras são as seguintes:

1. Ler um texto cronometrado.
2. Carregar na temática textual correta para o texto anteriormente lido.

Antes de o utilizador começar a jogar, é necessário executar a inicialização do objeto principal, *BP_Ship_Controller*, que é responsável por gerir este nível. Na inicialização é criada uma estrutura auxiliar para guardar todas as temáticas validas para a dificuldade escolhida assim como o tempo de leitura para cada ronda através da função implementada *Make Struct*. De seguida, é criado um *array* com todos os textos validos que cumpram as temáticas corretas para que sejam mostrados ao longo do nível e evitar futuras comparações para testar a validade dos textos escolhidos.

Quando a inicialização é concluída, é executado pelo *BP_Ship_Controller* a função *Start* para dar início à primeira parte do nível correspondente à leitura de textos.

O texto apresentado é escolhido com a função *Get Question* que devolve um dos textos validos, da estrutura auxiliar, assim como o número de palavras que o texto contém e qual é a temática correta. Simultaneamente é escolhido aleatoriamente dois temas errados da mesma lista de temáticas validas para servir de opções erradas. Assim que o texto é escolhido é necessário atribuir a cada

objeto, correspondente a um barco no nível, uma temática. Apenas um dos barcos é que tem a resposta correta enquanto que os restantes barcos têm as respostas erradas escolhidas anteriormente.

Por fim o texto é mostrado ao utilizador na figura 4.17.O texto será cronometrado para posteriormente calcular a velocidade de leitura do utilizador tendo em conta o número de palavras do texto lido assim como o tempo que demorou a ler. Se necessário é possível parar o jogo e o tempo se carregar no ícone superior direito. Para evitar que o utilizador pare o tempo e leia o texto, o texto é não se encontra disponível enquanto o jogo estiver parado.

Figura 4.17 - Exemplo de um texto escolhido para o utilizador ler.

Quando o utilizador considerar que acabou de ler o texto corretamente, é necessário carregar no botão *Continuar* para parar o tempo e progredir para a segunda parte deste nível. Quando a primeira parte é concluída, relativa à leitura, é calculado a velocidade de leitura que o utilizador obteve com a função *Calcular Velocidade*.

A segunda parte é relativa às temáticas textuais onde o utilizador deve carregar no barco correspondente à temática textual correta para o texto que acabou de ler, como em exemplo na figura 4.18. No canto superior esquerdo da HUD o utilizador consegue ver os seus pontos assim como o número de tentativas que ainda disponibiliza, equivalente ao número de rondas para esta dificuldade, assim como a velocidade de leitura que obteve durante a leitura.

Figura 4.18-Escolha da temática correta relativa ao texto anteriormente lido.

Se o utilizador carregar na resposta correta é lhe atribuído 10 pontos ao total de pontos, mas no caso de a resposta estar incorreta não lhe são atribuídos pontos. Ambos as escolhas são acompanhadas com uma mensagem de *feedback*, positivo ou negativo, para mostrar se a resposta estava correta ou incorreta.

O nível é concluído quando o utilizador completar todas as rondas disponíveis. No fim o utilizador acede ao seguinte menu, figura 4.19, para ver os seus resultados.

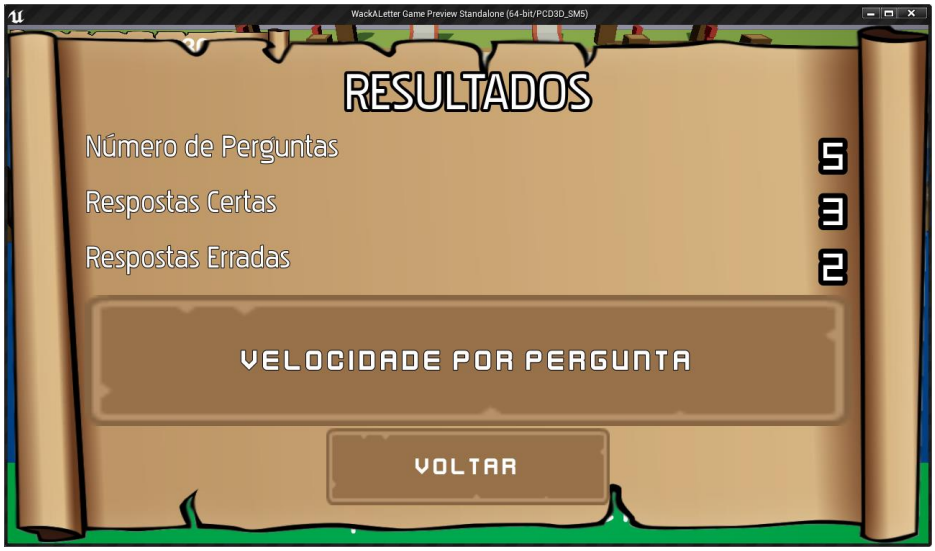


Figura 4.19-Menu de resultados do segundo nível.

Como é possível observar, os resultados são semelhantes ao outro nível, mas neste caso, é mostrado ao utilizador uma lista com todas as velocidades que obteve ao longo das rondas.

Por fim é necessário criar um pedido, procedimento na secção 4.2.3, para enviar todos os valores referentes aos resultados para o sistema de dados para futura validação e introdução na base de dados. Neste nível, os terapeutas consideraram importante avaliar a velocidade que o aluno obteve num texto específico sem consideração se o aluno obteve uma resposta errada. Para criar esta avaliação extra, o *Backend* fornece os dados ao *Frontend* do sistema de dados para que este, ilustre um gráfico com as últimas 10 velocidades obtidas para um texto. Para aceder a este gráfico, o utilizador necessita de introduzir o seu código de acesso no *Frontend* para aceder à sua página de resultados, e escolher qual é o texto que pretende ver a avaliação. Para obter os últimos 10 resultados para criar o gráfico foi utilizada a seguinte *query*:

```
$sqlSearch = "SELECT id,np,tl,correct from text_reading where acesscode=? and
textid=? ORDER BY id DESC LIMIT 10";
```

A *query* devolve o número de palavras, o tempo de leitura e se a resposta do utilizador foi correta, numa ordem descendente. A *query* está limitada às últimas 10 entradas na base de dados. Por fim é necessário inverter a ordem através da função *Array_Reverse* do PHP.

Um exemplo de um gráfico disponível encontra-se na figura 4.20.

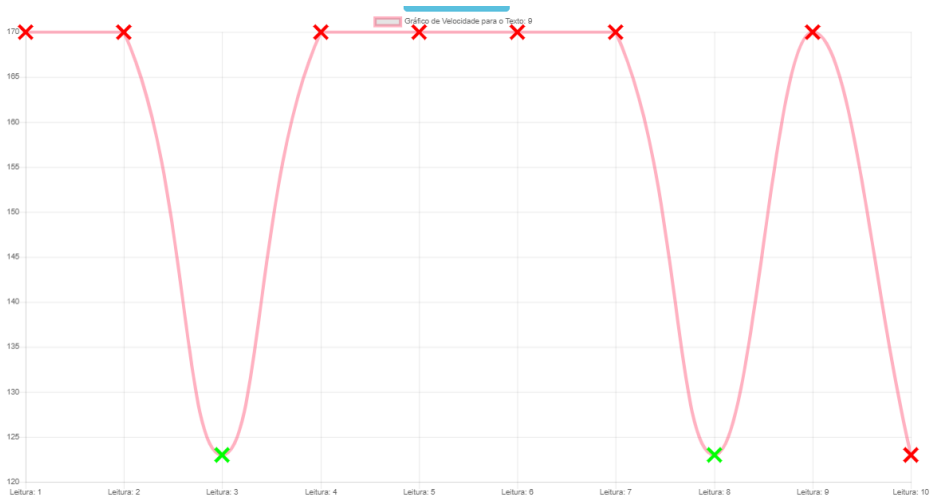


Figura 4.20 - Exemplo de um gráfico correspondente ao texto 9.

No gráfico estão representadas as últimas 10 leituras com as correspondentes velocidades de leitura expressas em palavras por minuto assim com um ponto a verde se o aluno conseguiu responder corretamente naquela leitura ou um ponto vermelho no caso de insucesso.

### 4.3.7. Menu de Perfil

Se o utilizador escolher o botão correspondente ao perfil é apresentado o menu da figura 4.21.

Figura 4.21-Menu de Perfil.

Este menu é responsável por mostrar todas as informações importantes relativas ao utilizador. Podemos observar o nome registado assim como a idade

introduzida e o seu código de identificação. O código é muito importante uma vez que permite ao utilizador aceder aos seus resultados no *Frontend*, logo deve estar bastante visível. Permite também observar os melhores resultados nos diferentes níveis num local só para facilitar a visualização dos mesmos por parte do utilizador. Estes dados serão disponibilizados também através do *Frontend* do sistema de dados como na figura 4.22.

.

Figura 4.22-Resultados *online* do utilizador.

O menu inclui os seguintes botões:

- O botão *LimparExtras* permite apagar todos os extras criados pelo utilizador para os diferentes níveis. Como o publico alvo são crianças é necessário que exista uma verificação de segurança para impedir apagar os extras acidentalmente.
- O botão *Critica* permite ao utilizador criar uma pequena avaliação através de três perguntas sobre o jogo numa escala numérica de 1 a 5 onde 1 correspondente à pior classificação e o 5 à melhor classificação.

É disponibilizado a opção de mudar de utilizador onde o utilizador pode optar por mudar para outro registo que é bastante útil se utilizar a ferramenta em diferentes dispositivos de modo a permitir assim continuar o seu progresso em qualquer dispositivo. Infelizmente esta funcionalidade não ficou completa e

o menu correspondente ao mudar de utilizador ficou responsável apenas por permitir ao utilizador apagar o seu registo local, permitindo assim criar um novo registo.

### 4.3.8. Menu de Créditos

A última opção do menu principal correspondente ao menu de créditos onde o utilizador pode consultar os autores da ferramenta assim como as diferentes licenças relativas ao conteúdo feito por terceiros utilizado na ferramenta. É disponibilizado também a lista dos autores dos diferentes textos.

# 5. Analise de Resultados

Neste capítulo é apresentada a analise dos resultados obtidos através do protótipo ao longo do período de teste assim como a metodologia utilizada para validar os dados adquiridos. Os resultados foram obtidos em parceria com os terapeutas do Centro Diferenças durante algumas secções de acompanhamento de crianças.

## 5.1. Recolha de Dados

Como anteriormente referido, o sistema de dados tem uma enorme importância, visto ser responsável por adquirir todos os dados sobre a utilização da ferramenta permitido assim tirar conclusões sobre o rendimento das crianças O foco principal é permitir responder à pergunta de investigação através do estudo dos dados obtidos dos jogadores nas diferentes tentativas.

De modo, a garantir a validade dos dados é necessário utilizar um método de validação adequado aos dados em estudo.

O método escolhido para o estudo dos dados encontra-se dividido da seguinte maneira:

- **Amostra de Participantes**: Para efetuar um estudo estatístico é necessário proceder-se à criação de um grupo de teste. O grupo deve ser constituído por um número de crianças pertencentes ao publico alvo

que queremos analisar para que os dados adquiridos sejam os mais fidedignos possíveis e representativos da realidade. O método mais adequado seria a criação de dois grupos de crianças onde apenas um grupo utilizaria o jogo ao longo do seu treino de leitura e o grupo seguinte não utilizaria o jogo. No fim, ambos os grupos teriam que resolver um teste, para averiguar a sua velocidade de leitura, de modo a retirar conclusões sobre a utilização benéfica dos jogos nas áreas educativas. A construção da amostra seria semelhante à amostra criada no estudo analisado na secção 2.1.5. Contudo não foi possível utilizar este método devido ao curto intervalo de tempo que foi possível utilizar a ferramenta no centro, logo será utilizado uma amostra de 8 crianças, para efeitos de teste, onde as crianças serão divididas em 4 grupos de duas pessoas onde cada criança irá jogar uma dificuldade correspondente ao seu ano escolar.

- **Método de aquisição de Dados**: para a recolha de dados, foi criado um sistema de analise e registo de dados utilizando o protótipo juntamente com uma base de dados para que seja possível guardar toda a informação recolhida para uma análise *a posteriori* do publico alvo. Sempre que um utilizador concluir um nível serão enviados os dados, referentes aos seus resultados, para a base de dados de modo a permitir criar um registo de progressão para a criança. Infelizmente durante o período de teste do *software*, o sistema de dados não se encontrava disponível *online* o que impossibilitou a adquisição de dados remotamente e automaticamente. Contudo foram recolhidos alguns dados de utilizadores durante uma sessão de teste presencial com o auxílio do computador com o jogo instalado para permitir a recolha de dados.
- **Analise de dados**: para a analise, serão utilizados os dados recolhidos para a criação de gráficos representativos da amostra de utilizadores que testaram o *software*. Será analisado a taxa de sucesso dos utilizadores nos diferentes níveis assim como a sua velocidade no segundo nível terminando com uma análise geral do que é possível observar dos gráficos.

## 5.2. Analise de Resultados

Neste capítulo será apresentada a analise dos dados obtidos no período de teste em parceria com o centro Diferenças. Durante o período de teste, o número de leituras durante o nível 2 foi reduzido para 5 leituras para cada dificuldade. Os dados serão divididos consoante o nível onde foram obtidos.

### 5.2.1. Resultados do Nível 1

Neste nível o objetivo era averiguar se os utilizadores conseguiam acompanhar o ritmo de leitura proposto. Como tal, os dados retirados foram os resultados obtidos correspondentes ao número de perguntas certas, número de perguntas erradas e o número de perguntas total para cada dificuldade. Neste nível cada criança jogou 5 vezes para cada dificuldade associada.

Baseado nos dados obtidos das 8 crianças do grupo de teste obteve-se os seguintes gráficos circulares:

Figura 5.1-Gráfico circular da taxa de sucesso para a primeira dificuldade.

Figura 5.2-Gráfico circular da taxa de sucesso para a segunda dificuldade

Analisando os resultados obtidos paras as duas dificuldades mais fáceis, figuras 5.1 e 5.2, podemos concluir que de um modo geral as crianças conseguiram acompanhar o ritmo de leitura e responder às perguntas corretamente. No entanto analisando os gráficos para as dificuldades mais altas, figuras 5.3 e 5.4, os resultados são semelhantes, mas apresentam uma taxa de sucesso inferior.

Figura 5.3-Gráfico circular da taxa de sucesso para a terceira dificuldade.

Figura 5.4-Gráfico circular da taxa de sucesso para a quarta dificuldade.

Nas dificuldades mais altas, as crianças necessitam de responder a um número maior de rondas com velocidades superiores às restantes. Este incremento no número de respostas contribui para um desinteresse maior por parte das crianças que infelizmente reflete-se no aumento do número de respostas erradas devido à falta de atenção ao fim de algumas tentativas. Conclui-se então que o número de tentativas deve ser igual para todas as dificuldades e apenas variar a velocidade de leitura em cada dificuldade de modo a normalizar a experiência do utilizador, sem tornar as dificuldades mais altas aborrecidas. Deste modo, podemos verificar que o nível 1 teve sucesso no grupo de teste embora a amostra seja pequena.

### 5.2.2. Resultados do Nível 2

No segundo nível, o objetivo era analisar se os utilizadores conseguiam atingir as velocidades de leituras propostas assim como testar os seus conhecimentos em relação às temáticas textuais. Os dados obtidos incluem o número de respostas certas, incorretas e total de respostas do mesmo modo que o nível anterior embora neste nível seja avaliado também a velocidade de leitura obtida nas diferentes leituras. Os dados foram divididos em quatro gráficos onde cada gráfico

é representativo de um par de crianças que jogou naquela dificuldade para comparar se conseguiram obter a velocidade proposta. Como os terapeutas apenas consideraram importante o treino da velocidade, só será avaliado os dados referentes às velocidades de leitura e não a taxa de sucesso das crianças relativamente à identificação das temáticas textuais.

Os gráficos obtidos foram os seguintes, utilizando apenas 5 leituras por dificuldade:

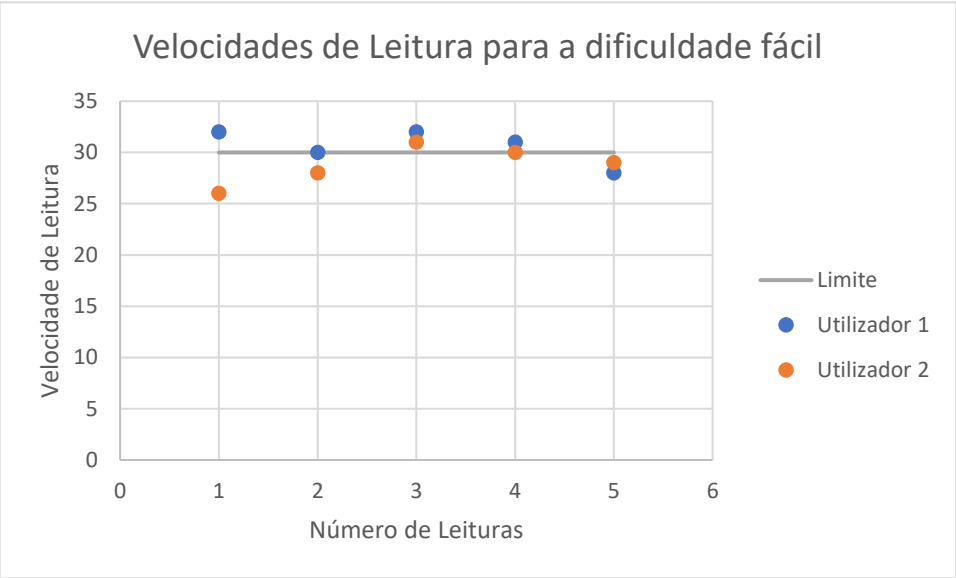


**Figura 5.5-Gráfico de velocidades de leitura para a primeira dificuldade.**

Relativamente à primeira dificuldade, figura 5.5, as crianças tinham que ler quatro temáticas de textos diferentes, onde a meta para a velocidade de leitura era as 30 palavras por minuto. Como esta dificuldade foi implementada para as crianças mais novas, os textos continham poucas palavras e de um modo geral, analisando o gráfico podemos concluir que ambas as crianças mantiveram velocidades de leitura perto da velocidade proposta variando por vezes devido ao número de palavras que cada texto tinha.

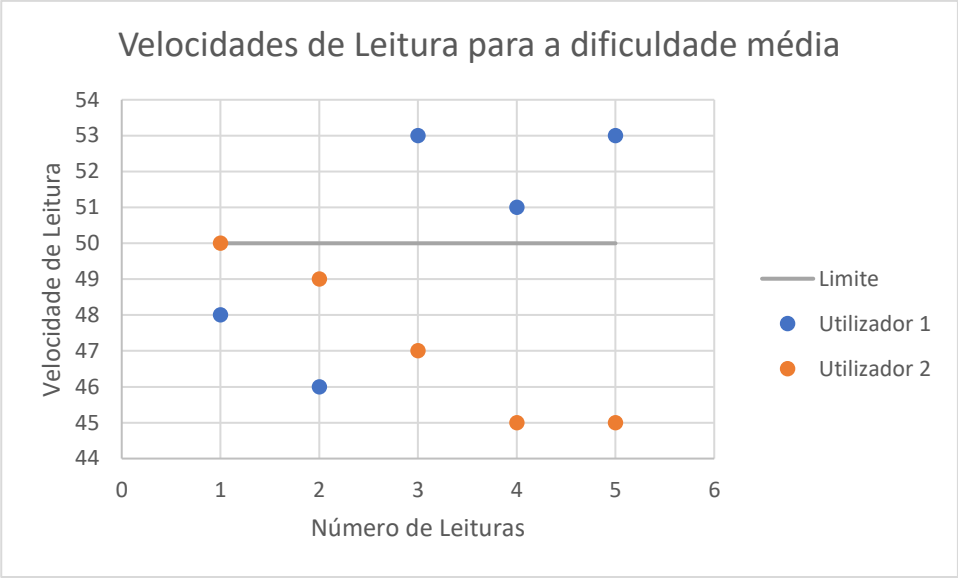


Figura 5.6-Gráfico de velocidades de leitura para a segunda dificuldade.

Analisando o segundo gráfico,figura 5.6, correspondente à segunda dificuldade podemos observar que embora as temáticas sejam as mesmas que para a primeira dificuldade as crianças conseguiram manter a sua velocidade de leitura perto da velocidade proposta existindo mais uma vez variação na velocidade para os diferentes textos.

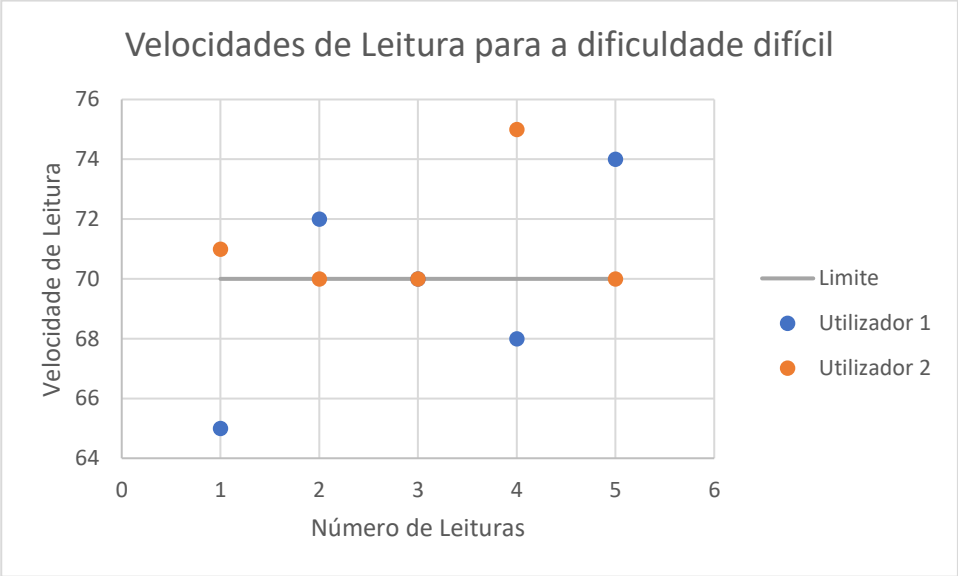


Figura 5.7-Gráfico de velocidades de leitura para a terceira dificuldade.

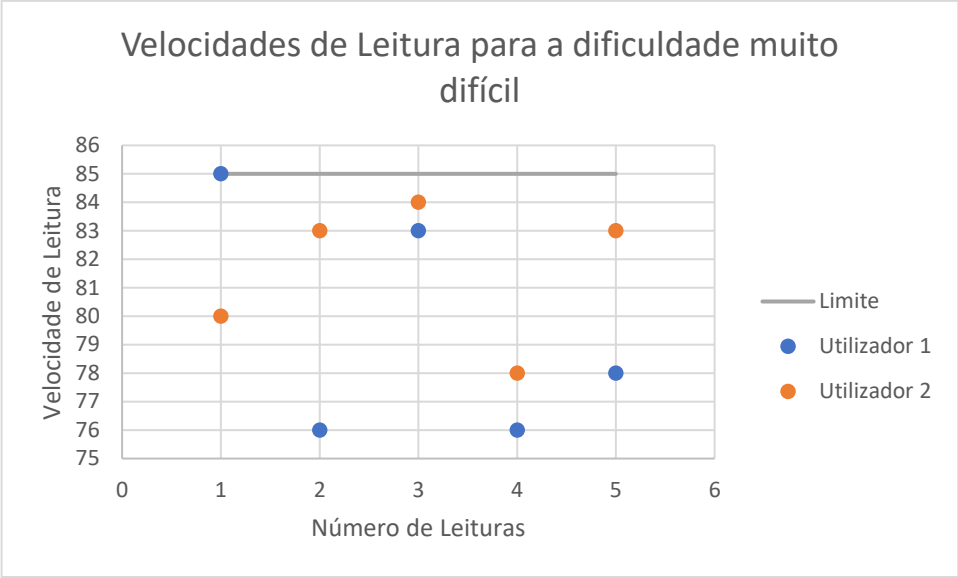


**Figura 5.8-Gráfico de velocidades de leitura para a quarta dificuldade.**

Contudo nas dificuldades mais altas, figura 5.7 e 5.8, existe uma maior diferença entre as velocidades obtidas e as velocidades propostas. Esta diferença é devido ao aumento na dificuldade dos textos, através de um aumento no número de temáticas textuais, assim como o número de palavras que cada texto contém. No entanto as crianças mantiveram a sua velocidade perto das velocidades propostas.

Concluído, o nível 2 cumpriu o seu objetivo de avaliar se as velocidades de leitura em diferentes textos das crianças eram semelhantes às velocidades propostas que cada criança deve ter para o seu nível de escolaridade. Embora os resultados sejam positivos é importante referenciar que não é possível retirar conclusões absolutas sobre o funcionamento benéfico do jogo nas crianças uma vez que foi analisada uma amostra pequena de utilizadores e não foi feita uma comparação entre os resultados obtidos pelas crianças que treinaram a sua leitura através do jogo com um grupo de crianças que utilizaram os métodos tradicionais de aprendizagem.

6

# 6. Conclusões e Trabalhos futuros

Neste capítulo é feita uma recapitulação final do que foi referido nos capítulos anteriores do documento. Inicialmente procedeu-se a uma pesquisa sobre o tema para investigar como é que os jogos se comportam na atualidade e como é que podemos enquadrar os jogos na área da educação através dos jogos sérios.

Surge então a proposta "Reino dos Livros" que tem como objetivo ajudar a melhorar a velocidade de leitura de crianças no primeiro ciclo escolar.

Para a verificação do desempenho dos jogadores relativamente às velocidades padrão de leitura foi feito um estudo dos seus resultados de modo a retirar conclusões para permitir responder à pergunta de investigação proposta no início da dissertação.

Os dados obtidos apresentam resultados positivos que indicam que os alunos demonstrarão melhores capacidades de leitura. No entanto, devido à amostra ser constituída por um número reduzido de crianças alvo não podemos retirar conclusões finais sobre os benefícios da utilização de jogos sérios na educação. Contundo, com base nos dados obtidos, pode-se concluir que o jogo teve benefícios positivos nas crianças porque fornecerão um novo método divertido de estudo.

Embora o protótipo cumpra os requisitos necessários, na reunião final com o Centro Diferenças, foram apontadas algumas alterações para o melhoramento geral do jogo.

As alterações importantes e consequentes trabalhos futuros foram as seguintes:

- **Normalização do número de rondas**: os terapeutas consideraram que nas dificuldades mais altas, o número de rondas proposto era muito alto, o que tornava o jogo aborrecido ao fim de algumas rondas para as crianças. De modo a resolver o problema é proposto uma normalização onde cada dificuldade terá um número fixo de rondas, igual para todas as dificuldades, variando apenas então o número de palavras mostradas ao longo do nível 1 e os textos lidos no nível 2 para cada dificuldade.
- **Criação de uma personagem e história**: os níveis atualmente não se encontram ligados tematicamente, logo para melhorar a imersão do jogador seria importante criar uma temática geral ou história para o jogo acompanhado de uma personagem criada pelo jogador. Foi desenhado um protótipo para a personagem, mas infelizmente não foi possível animar a personagem o que levou a não ser incluída no protótipo. O protótipo para a personagem foi desenhado no Autodesk Maya e encontra-se na figura 6.1:

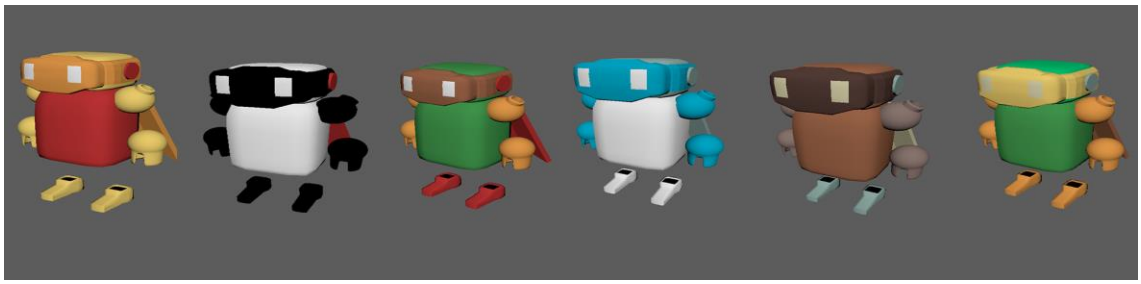

Figura 6.1-Protótipo de uma personagem para o jogo.

- **Sistema de recompensas**: para cativar as crianças a utilizar mais o jogo seria criado um sistema de recompensas que seriam atribuídas ao jogador no fim dos níveis. A distribuição das recompensas deve ser gradual para não causar dependência e tornar-se o foco principal para a utilização do jogo por parte das crianças.
- **Melhoramento do Sistema de dados:** embora o sistema de dados cumpra os requisitos pedidos, seria importante reforçar o sistema e melhorar o aspeto estético do *Frontend*, aumentar as funcionalidades do *Backend* e permitir aos utilizadores acederem a mais dados no jogo. O ponto mais importante seria introduzir mais textos e palavras para os diferentes níveis e permitir que quando um terapeuta adicione um novo texto seja atualizado em todos os dispositivos sem a necessidade de uma atualização da ferramenta.

# Referências


[1] Centro de Desenvolvimento DIFERENÇAS, "Perturbação Específica da Aprendizagem | DIFERENÇAS." [Online]. Available: http://diferencas.net/?page_id=633. [Accessed: 21-Feb-2018].

[2] "Português | Direção-Geral da Educação." [Online]. Available: http://www.dge.mec.pt/portugues.

[3] G. Frasca, "Videogames of the Oppressed: Videogames as a Means for Critical Thinking and Debate," *Sch. Lit. Commun. Cult.*, no. April, p. 118, 2001.

[4] "OpenGL Overview - The Khronos Group Inc." [Online]. Available: https://www.khronos.org/opengl/.

[5] "Getting Started with DirectX Graphics (Windows)." [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/hh309467(v=vs.85).

[6] S. D. Mikulcik, "Application of Neural Networks for Intelligent Video Game Character Artificial Intelligences," 2016.

[7] R. S. Alsawaier, "The effect of gamification on motivation and engagement," *Int. J. Inf. Learn. Technol.*, vol. 35, no. 1, pp. 56–79, Jan. 2018.

[8] I. Boughzala and H. Michel, "Introduction to the serious games, gamification and innovation minitrack," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2016–March, p. 817, 2016.

[9] "GAMIFICATION | meaning in the Cambridge English Dictionary." [Online].

Available:
https://dictionary.cambridge.org/dictionary/english/gamification.

[10] "EDUTAINMENT | meaning in the Cambridge English Dictionary." [Online]. Available:
https://dictionary.cambridge.org/dictionary/english/edutainment.

[11] S. Franceschini, S. Gori, M. Ruffino, S. Viola, M. Molteni, and A. Facoetti, "Action Video Games Make Dyslexic Children Read Better," *Curr. Biol.*, vol. 23, no. 6, pp. 462–466, Mar. 2013.

[12] J. Sánchez and R. Olivares, "Problem solving and collaboration using mobile serious games," *Comput. Educ.*, vol. 57, no. 3, pp. 1943–1952, Nov. 2011.

[13] P. V. Ute Ritterfeld, Michael Cody, *Serious Games: Mechanisms and Effects*. 2009.

[14] D. Drummond, A. Hadchouel, and A. Tesnière, "Serious games for health: three steps forwards," *Adv. Simul.*, vol. 2, no. 1, p. 3, Dec. 2017.

[15] "Water Busters – Ecogamer." [Online]. Available: https://www.ecogamer.org/conservation/water-busters/.

[16] M. Lewis and J. Jacobson, "Introduction," *Commun. ACM*, vol. 45, no. 1, 2002.

[17] J. Gregory, "Game Engine Architecture," 2009.

[18] "Game Engine Analysis and Comparison |." [Online]. Available: https://www.gamesparks.com/blog/game-engine-analysis-and-comparison/.

[19] "6 Top Game Engines in 2017 | DiscoverSDK Blog." [Online]. Available: http://www.discoversdk.com/blog/6-top-game-engines-in-2017.

[20] "Unity - Multiplatform - Publish your game to over 25 platforms." [Online]. Available: https://unity3d.com/pt/unity/features/multiplatform.

[21] "Unity - Manual: Unity User Manual (2017.3)." [Online]. Available: https://docs.unity3d.com/Manual/index.html.

[22] "If You Love Something, Set It Free." [Online]. Available: https://www.unrealengine.com/en-US/blog/ue4-is-free.

[23] "Game Engine Technology by Unreal." [Online]. Available: https://www.unrealengine.com/en-US/faq.

[24] "Unreal Engine 4 Documentation | Unreal Engine." [Online]. Available: https://docs.unrealengine.com/latest/INT/.

[25] E. Christopoulou and S. Xinogalos, "Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices," *Int. J. Serious Games*, vol.

4, no. 4, Dec. 2017.

[26] "Blueprints Visual Scripting." [Online]. Available: https://docs.unrealengine.com/en-us/Engine/Blueprints.

[27] "What is the cost of using Blueprint instead of C++? - UE4 AnswerHub." [Online]. Available: https://answers.unrealengine.com/questions/23167/blueprint-overhead.html.

[28] "Guidelines for Programming for Blueprints." [Online]. Available: https://docs.unrealengine.com/en-us/Engine/Blueprints/TechnicalGuide/Guidelines.

[29] "VaRest by Vladimir Alyamkin in Code Plugins - UE4 Marketplace." [Online]. Available: https://www.unrealengine.com/marketplace/varest-plugin. [Accessed: 25-Feb-2019].

[30] "Maya | Computer Animation & Modeling Software | Autodesk." [Online]. Available: https://www.autodesk.com/products/maya/overview.