



**DUARTE FERNANDES GUERREIRO**

Licenciatura em Ciência e Engenharia Informática

## **ADVENTURE GAME CREATOR**

**CRIAÇÃO E REPRESENTAÇÃO DE HISTÓRIAS  
COM UM STORYBOARD**

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa  
Dezembro, 2023



# ADVENTURE GAME CREATOR

CRIAÇÃO E REPRESENTAÇÃO DE HISTÓRIAS  
COM UM STORYBOARD

**DUARTE FERNANDES GUERREIRO**

Licenciatura em Ciência e Engenharia Informática

**Orientador:** João Paulo Duque Vieira

*COO/CPO, Skills Workflow*

**Coorientador:** Sérgio Marco Duarte

*Professor Auxiliar, NOVA University Lisbon*

## **Adventure Game Creator**

### **Criação e Representação de Histórias com um Storyboard**

Copyright © Duarte Fernandes Guerreiro, 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.

## RESUMO

O Desenvolvimento de Videojogos é das áreas de Desenvolvimento de Software que tem apresentado um maior crescimento nos últimos anos. Uma particularidade especial desta é o vasto interesse e a participação de pessoas de áreas que não a informática na mesma.

Os Jogos de Aventuras são um dos géneros de videogame mais populares e o seu foco principal é a interação do utilizador com o mundo e as personagens à sua volta que servem de meio para contar uma história apelativa. Devido à simplicidade de mecânicas que este tipo de jogos apresenta existem métodos na indústria para separar o trabalho dos programadores e dos escritores e algumas ferramentas que permitem mesmo um utilizador independente sem conhecimentos informáticos criar o seu próprio jogo. No entanto, mesmo estas soluções, devido à complexa especificação dos eventos que requerem tornam o processo de criação de histórias muito exigente e demorado para utilizadores casuais. Sendo a essência destes jogos a sua história é interessante fornecer um meio mais simples e casual que necessite apenas que o utilizador explicita a mesma e que desta forma fiquem definidos os comportamentos e interações para o funcionamento do jogo.

O objetivo é então definir uma estrutura de *Storyboard* que permita expressar histórias de jogos de aventuras diferentes e provar o seu correto funcionamento para permitir que todo o tipo de utilizadores possam contar as suas próprias histórias de uma forma fácil e rápida.

**Palavras-chave:** Desenvolvimento de Videojogos, Jogos de Aventura, *Storyboard*

## ABSTRACT

Video game Development is one of the fastest-growing areas of Software Development in recent years. A unique feature of this area is the wide interest and participation of people from areas other than computer science.

Adventure Games are one of the most popular video game genres, and their main focus is the user's interaction with the world and the characters around them, which serve as a means of telling an appealing story. Due to the simplicity of the mechanics that this type of game presents, there are methods in the industry for separating the work of programmers and writers and some tools that allow even an independent user with no programming background to create their own game. However, even these solutions, due to the complex specification of the events they require, make the process of creating stories very demanding and time-consuming for casual users. Given that the main goal of these games is the story they tell, it's of interest to create a simpler and more casual tool that, given the specification of the story that wants to be told, would be able to define all the logic, expected behaviors, and interactions to create a working game.

The goal, then, is to define a structure, a Storyboard, that can express different video game stories and prove its accuracy to allow all kinds of users to easily and quickly bring their stories to life.

**Keywords:** Videogame Development, Adventure Games, Storyboard

# ÍNDICE

<b>Índice de Figuras</b>	<b>vii</b>
<b>Índice de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Definição do Problema . . . . .	2
1.3 Objetivos . . . . .	2
1.4 Estrutura . . . . .	3
<b>2 Estado da Arte</b>	<b>4</b>
2.1 Criação de Histórias . . . . .	4
2.1.1 Hero's Journey . . . . .	4
2.1.2 Narrativa Visual . . . . .	6
2.1.3 Storyboards . . . . .	7
2.2 Design de VideoJogos . . . . .	9
2.2.1 Lock-and-Key . . . . .	10
2.2.2 Criação de Missões para Jogos gerados Procedimentalmente . . . . .	11
2.3 Criação Assistida de Jogos . . . . .	12
2.3.1 Motores de Jogo . . . . .	13
2.3.2 Jogos . . . . .	18
<b>3 Solução</b>	<b>22</b>
3.1 História . . . . .	22
3.2 Capacidades Teóricas do Storyboard . . . . .	25
3.3 Interpretador da História . . . . .	26
3.4 Comportamentos do Jogador . . . . .	26
<b>4 Implementação</b>	<b>27</b>
4.1 Visão Geral do Sistema . . . . .	27

4.2	Arquitetura do Sistema . . . . .	28
4.3	Criação da História . . . . .	29
4.4	Conversão de Fluxograma-Código . . . . .	32
4.5	Motor de Jogo . . . . .	34
4.5.1	Controlador do Jogador . . . . .	34
4.5.2	Interpretador da História . . . . .	35
<b>5</b>	<b>Avaliação Experimental</b>	<b>37</b>
5.1	Avaliação da Versatilidade do Sistema . . . . .	37
5.1.1	Jogo de Aventura - The Secret of Monkey Island . . . . .	37
5.1.2	Hero's Journey - Hansel e Gretel . . . . .	38
5.1.3	Histórias Instrutivas - Cozinhar uma Pizza . . . . .	40
5.2	Usability Tests . . . . .	41
5.2.1	Metodologia . . . . .	42
5.2.2	Caracterização dos Voluntários . . . . .	42
5.2.3	Testes e Resultados . . . . .	43
5.2.4	Questionário de Utilização . . . . .	57
5.2.5	Observações e Conclusões . . . . .	60
<b>6</b>	<b>Conclusão</b>	<b>62</b>
6.1	Conclusões . . . . .	62
6.2	Trabalho Futuro . . . . .	63
	<b>Bibliografia</b>	<b>64</b>

## ÍNDICE DE FIGURAS

2.1	Excerto do livro explorado em detalhe da autoria de David Wiesner, Flotsam [33]	7
2.2	Recorte de <i>Storyboard</i> utilizado para adaptar o guião de um episódio do desenho animado da <i>Disney</i> , Phineas e Ferb para animação [15]	8
2.3	Tabuleiro de criação de cenário no qual os jogadores devem criar uma colagem das divisões que pretendem para a sua história (em cima) e exemplos de diferentes divisões criadas por jogadores (em baixo) [11]	9
2.4	Abstração de um nível de jogo como um problema de <i>Lock-and-Key</i> da autoria de Mark Brown [3], diamantes representam a localização da chave e quadrados o respetivo cadeado	10
2.5	Esquema da missão e espaço no nível Forest Temple do jogo <i>The Legend of Zelda: The Twilight Princess</i> [9, 10]	12
2.6	Level Editor de desenvolvimento de um jogo em <i>Unreal Engine</i>	13
2.7	<i>Blueprint</i> que ativa a visibilidade de uma lâmpada quando o jogo é iniciado	14
2.8	<i>Unreal Engine Marketplace</i>	15
2.9	Level Editor de desenvolvimento de um jogo em <i>Unity</i>	15
2.10	Exemplo de <i>Script</i> com propriedades visuais	16
2.11	<i>Unity Asset Store</i>	16
2.12	Exemplos da utilização do pacote <i>MoreMountains Feel</i>	17
2.13	Exemplo de definição de comportamentos com <i>Adventure Creator</i>	18
2.14	Exemplo da criação de um nível no videojogo <i>LittleBigPlanet 3</i>	19
2.15	Produto final de um jogo criado em <i>Dreams</i>	19
2.16	Criação do modelo de um avião em <i>Dreams</i>	20
2.17	Criação da lógica de jogo e controlos a aplicar ao avião em <i>Dreams</i>	21
3.1	Exemplo de Fluxograma que representa passos da história de O Capuchinho Vermelho	23
3.2	Exemplo de <i>Storyboard</i> que representa passos da história de O Capuchinho Vermelho	23

3.3	Exemplo de <i>Storyboard</i> que representa passos opcionais e obrigatórios da história de O Capuchinho Vermelho . . . . .	24
4.1	Menu Principal (esquerda) e História a ser jogada (direita) . . . . .	27
4.2	Arquitetura do sistema . . . . .	28
4.3	Modelo de um passo do <i>Storyboard</i> . . . . .	30
4.4	Edição de passos de uma História . . . . .	31
4.5	<i>Schema</i> do <i>JSON</i> para representar os dados do <i>Storyboard</i> . . . . .	33
4.6	Exemplo da reprodução de uma História . . . . .	35
5.1	<i>Storyboard</i> e Cenário da história 1 . . . . .	38
5.2	<i>Storyboard</i> e Cenário da primeira parte da história 2 . . . . .	39
5.3	<i>Storyboard</i> e Cenário da segunda parte da história 2 . . . . .	40
5.4	<i>Storyboard</i> e Cenário da história 3 . . . . .	41
5.5	Gráficos de Caracterização dos Voluntários . . . . .	43
5.6	Gráficos dos resultados da Tarefa A1 . . . . .	44
5.7	Gráficos dos resultados da Tarefa A5 . . . . .	46
5.8	Gráficos dos resultados da Tarefa B1 . . . . .	47
5.9	Gráficos dos resultados da Tarefa B2 . . . . .	47
5.10	Gráficos dos resultados da Tarefa B3 . . . . .	48
5.11	Gráficos dos resultados da Tarefa B4 . . . . .	49
5.12	Gráficos dos resultados da Tarefa B5 . . . . .	50
5.13	Gráficos dos resultados da Tarefa B6 . . . . .	51
5.14	Gráficos dos resultados da Tarefa B7 . . . . .	51
5.15	Gráficos dos resultados da Tarefa B8 . . . . .	52
5.16	Gráficos dos resultados da Tarefa C1 . . . . .	53
5.17	Gráficos dos resultados da Tarefa C2 . . . . .	54
5.18	Gráficos dos resultados da Tarefa C3 . . . . .	54
5.19	Gráficos dos resultados da Tarefa C4 . . . . .	55
5.20	Gráficos dos resultados da Tarefa C5 . . . . .	56
5.21	Gráficos dos resultados da Tarefa C6 . . . . .	57
5.22	Gráficos dos resultados das perguntas relacionadas com a <i>Storyboard</i> . . . . .	58
5.23	Gráficos dos resultados das perguntas relacionadas com o <i>System Usability Scale (SUS)</i> . . . . .	59
5.24	Médias, das diferentes perguntas do questionário <i>SUS</i> . . . . .	60

## ÍNDICE DE TABELAS

2.1	Monomito em Filmes da <i>Pixar</i> [25] . . . . .	6
2.2	Monomito em Curtas-Metragens da <i>Pixar</i> [25] . . . . .	6
5.1	Monomito em Hansel e Gretel . . . . .	39



# INTRODUÇÃO

A indústria dos videojogos é uma área da informática em constante evolução. Desde as técnicas usadas pelos programadores aos requisitos que podem exigir do *hardware* dos utilizadores e até na constituição das equipas que os desenvolvem, existem constantes alterações e, conseqüentemente, as necessidades que estas apresentam também se alteram.

Apesar de, inicialmente, esta área estar reservada exclusivamente para programadores, com sucessivos avanços em termos de *hardware* e de design de videojogos, também o nível de exigência dos clientes aumentou. Atualmente, a maior parte dos jogos são desenvolvidos por equipas que, para além dos programadores, são também constituídas por vários escritores, artistas e designers. Estes especialistas, sem conhecimentos de programação, mas com vastos conhecimentos em outras áreas como a cinematografia, psicologia humana ou escrita narrativa são indispensáveis à criação de videojogos modernos.

Os jogos de aventura são um género de jogo cujas principais características são a interação com o mundo e as personagens em redor para progredir com o enredo da história que é contada ao jogador. Sendo o principal foco deste tipo de jogo a qualidade da história e a forma como esta é contada e, devido à natureza simples de ações que o jogador espera poder realizar neste tipo de aventuras, um bom jogo de aventuras moderno destaca-se não pela qualidade dos seus programadores mas pela qualidade dos seus escritores, artistas e designers.

## 1.1 Motivação

Apesar disto, é ainda assim, necessária a intervenção de programadores no desenvolvimento de jogos de aventura. A sua principal função é implementar os comportamentos expectáveis deste tipo de videojogo e converter a visão dos escritores em código e assim ser possível obter a experiência interativa desejada.

Então, por este género de videojogo ser simples e padronizado do ponto de vista da sua implementação mas, complexo e livre na sua vertente narrativa, é interessante desenvolver uma ferramenta que permita a todo o tipo de utilizadores contar as suas histórias. O objetivo é que qualquer utilizador consiga, rapidamente, especificar uma história numa

estrutura fácil de aprender e de utilizar mas que consiga definir todos os comportamentos, eventos e a lógica intrínseca necessária para que a história decorra da maneira que este pretenda.

Desta forma, ao fornecer uma maneira de auxiliar a criação de jogos de aventura, que remova a necessidade de implementar manualmente em código a visão do escritor e, da qual dependa apenas a capacidade criativa e a qualidade narrativa do mesmo, é possível permitir a qualquer pessoa, mesmo alguém sem conhecimentos de programação, dar vida às suas histórias com um suporte digital interativo que pode partilhar com o mundo. Para além do utilizador comum, esta ferramenta pode também auxiliar a indústria dos videogames profissional. Ao fornecer uma prototipagem mais rápida e simples, irá consequentemente, acelerar o processo de desenvolvimento e melhorar a qualidade deste tipo de jogos ao libertar tempo dos programadores que pode ser despendido em outros aspetos do jogo.

### 1.2 Definição do Problema

O principal foco desta dissertação é provar que é possível desenhar uma estrutura de dependências, um *Storyboard*, capaz de especificar o enredo e os acontecimentos fundamentais de histórias que se pretendam contar. Este *Storyboard* será fundamental para permitir a um vasto número de utilizadores, sem conhecimentos informáticos, desenvolver jogos de aventura.

Um dos grandes desafios deste projeto consiste então, em conseguir abstrair as características, fundamentais, necessárias para especificar o comportamento de um elevado número de histórias diferentes. Para tal, é necessário estudar que propriedades de uma história são indispensáveis para a criação do *Storyboard* e como as relacionar entre si. A partir daí, basta criar um Interpretador do Jogo que consiga consultar a informação presente no *Storyboard* e impor os comportamentos que o utilizador definiu.

O outro principal foco é garantir que a utilização de todo o sistema, mas sobretudo da criação de histórias no *Storyboard*, seja fácil de utilizar, rápido, intuitivo e que a interface com o utilizador seja minimalista e com pouco texto. O objetivo é que todo o processo de criação de uma história, assim que o utilizador esteja familiarizado com as suas opções, permita produzir diferentes histórias rapidamente.

Para testar a versatilidade do projeto final obtido serão apresentados exemplos, de diversos géneros, de diferentes histórias de aventuras, com a utilização deste *Storyboard*, e relatados os resultados obtidos.

### 1.3 Objetivos

Podemos considerar como contribuições do trabalho final desta dissertação:

1. *Storyboard* - Estrutura capaz de especificar os comportamentos esperados para diferentes histórias de aventura;
2. Interpretador de Histórias - Sistema em *Unity* responsável por interpretar e implementar os comportamentos necessários para a história criada pelo jogador;
3. Jogos de Aventura - Para testar o correto funcionamento do projeto final foram criados alguns exemplos de *Storyboard* e respetivos jogos de aventuras que podem ser explorados e alterados pelos utilizadores.

## 1.4 Estrutura

Na secção 2 é apresentado o estado da arte em diferentes áreas relacionadas com o projeto, tais como técnicas de criação de histórias, design de videojogos e criação assistida de jogos.

Na secção 3, é exposto conceptualmente a solução, indicando os motivos que levaram a cada decisão tomada, contrastando as vantagens e desvantagens de cada uma.

Na secção 4 é efetuada uma descrição geral do sistema e como foi efetuada a implementação dos diferentes requisitos da solução alcançada.

Na secção 5 são descritos os resultados obtidos das diferentes testagens efetuadas à aplicação e as conclusões que se podem retirar desses testes

Finalmente, na secção 6, é concluído o documento apresentado as conclusões que se podem retirar do trabalho efetuado e são também apresentadas ideias para trabalho futuro neste tema.

## ESTADO DA ARTE

O Estado da Arte encontra-se dividido em três secções, Criação de Histórias (Secção 2.1) que descreve diferentes técnicas de desenvolvimento de histórias, Design de Videojogos (Secção 2.2) que refere padrões de desenho de níveis de videojogos e Criação Assistida de Jogos (Secção 2.3), que explora algumas ferramentas à disposição dos utilizadores para uma criação assistida de jogos.

### 2.1 Criação de Histórias

Uma característica inerentemente humana é o facto de, desde tempos pré-históricos, contarmos histórias para nos auxiliar na nossa comunicação, seja para uma simples conversa ou para transmitir algum conhecimento [31]. No entanto, apesar de todos contarmos histórias, existem algumas regras fundamentais que devem ser seguidas para conseguir criar uma história que seja cativante [13].

#### 2.1.1 Hero's Journey

O Monomito, a Viagem do Herói ou "Hero's Journey" é uma estrutura bem definida e estudada que se encontra mais ou menos presente em praticamente todas as histórias, especialmente as histórias de aventura, que existem e que são atualmente criadas. O padrão narrativo foi estudado e sumariado por Joseph Campbell que resume a estrutura do mesmo na sua obra "The Hero with a Thousand Faces" como uma história em que "O herói aventura-se da sua vida diária para um mundo do supernatural, é deparado com fabulosos obstáculos e uma vitória decisiva sobre estes é alcançada, o herói volta da sua misteriosa aventura com o poder para auxiliar os seus companheiros com o que obteve" [4].

Segundo Campbell [4], o padrão das histórias de aventura pode ser dividido em 17 etapas diferentes, não tendo obrigatoriamente de estar todas presentes, que se podem agrupar em 3 secções distintas. Na primeira secção, a Partida, o protagonista vive no seu mundo conhecido e é confrontado com algo que o obriga a iniciar a sua viagem contra a sua vontade sendo esta secção decomposta nas seguintes etapas:

1. *The Call to Adventure*
2. *Refusal of the Call*
3. *Supernatural Aid*
4. *The Crossing of the First Threshold*
5. *The Belly of the Whale*

A Iniciação, a segunda secção, começa quando o herói realiza a passagem do seu mundo normal para o mundo desconhecido do fantástico. Nele, é confrontado com inúmeros obstáculos que tem que superar sozinho ou com a ajuda de companheiros que encontra pelo caminho, até alcançar o obstáculo final que o fez iniciar a aventura, passando por apoteose para o conseguir superar e obtendo a sua recompensa depois de o derrotar. Nesta secção é comum encontrar as etapas:

6. *The Road of Trials*
7. *The Meeting with the Goddess*
8. *Woman as the Temptress*
9. *Atonement with the Father*
10. *Apotheosis*
11. *The Ultimate Boon*

Na última secção, o Regresso, o protagonista tem de voltar ao seu mundo original com a recompensa que obteve. É possível agora este não querer regressar ou necessitar de intervenção exterior para poder fazer o trajeto de volta para o seu mundo. O herói, agora transformado pela aventura e conhecimento que obteve na sua viagem, volta então a atravessar a barreira entre mundos, regressando ao local inicial da história com a recompensa que adquiriu. Esta nova capacidade pode agora ser utilizada para benefício de si e dos seus. É característico encontrarmos na secção do Regresso as etapas:

12. *Refusal of the Return*
13. *The Magic Flight*
14. *Rescue from Without*
15. *The Crossing of the Return Threshold*
16. *Master of the Two Worlds*
17. *Freedom to Live*

Para se perceber melhor a influência do Monomito nesta área podemos observar a sua presença em diferentes obras da *Pixar*, um dos, se não o estúdio de filmes de animação com mais sucesso de sempre [28]. No artigo "The Hero's Journey and Three Types of Metaphor in Pixar Animation" [25] o autor identifica em oito dos mais recentes filmes e curtas metragens deste estúdio, apesar de não considerar a etapa oito, "Woman as the Temptress" no seu estudo, praticamente todos os elementos presentes na estrutura do Monomito. É importante também realçar que, em grande número dos casos, os passos encontram-se na ordem exata que foi definida por Campbell [4] como se pode observar na Tabela 2.1 e na Tabela 2.2.

Filme, ano	Etapas do Monomito
Divertida-Mente, (2015)	1; 2; 3; 4; 5; 6; 7; 10; 9; 11; 12; 13; 14; 15; 16; 17
Coco, (2017)	1; 2; 3; 4; 5; 6; 7; 9; 10; 14; 13; 11; 12; 15; 16; 17
Bora Lá, (2020)	1; 2; 3; 4; 5; 7; 6; 10; 11; 12; 9; 13; 14; 15; 16; 17

Tabela 2.1: Monomito em Filmes da *Pixar* [25]

Curta-Metragem, ano	Etapas do Monomito
Parcialmente Nublado, (2009)	1; 4; 6; 10; 11; 17
Guarda-Chuva Azul, (2013)	1; 4; 5; 6; 14; 10; 11; 15; 16; 17
Lava, (2014)	1; 6; 10; 14; 11; 17
Sanjay Super Equipa, (2015)	1; 2; 3; 4; 5; 7; 14; 6; 11; 15; 9; 16; 17
Piper, (2016)	1; 2; 3; 4; 5; 6; 7; 10; 11; 15; 9; 16; 17

Tabela 2.2: Monomito em Curtas-Metragens da *Pixar* [25]

Sendo este padrão de escrita de narrativas uma das principais referências para qualquer pessoa que queira escrever uma história de aventura e um padrão que permite contar praticamente todo o tipo de histórias possíveis, é importante ter a mesma em consideração ao desenvolver o *Storyboard*. O objetivo, passa por conseguir criar uma estrutura de dependências que seja compatível e, se possível, encoraje mesmo os utilizadores menos experientes a seguirem esta de modo a criarem histórias de melhor qualidade.

### 2.1.2 Narrativa Visual

A linguagem natural não representa, no entanto, a única forma que possuímos para comunicar histórias enquanto seres humanos. As chamadas Narrativas Visuais ou Literacia Visual, desde tempos pré-históricos, foram uma parte importante da comunicação do ser humano e um fator ainda mais presente na época digital em que nos encontramos [12]. Não existe uma definição única e universalmente aceite por toda a comunidade literária para Literacia Visual mas, para os efeitos desta dissertação, pode ser geralmente sumariada, como a capacidade dos seres humanos para interpretar diferentes mensagens e informações captadas através da observação de fenómenos visuais [2].

Considerando que o objetivo desta dissertação é fornecer uma ferramenta que permita contar diferentes histórias é importante conhecer as diferentes técnicas que os utilizadores podem querer usar. Para isso, para além de saber da sua existência, é também importante saber como são utilizadas.

Existem inúmeros livros ilustrados, normalmente direcionados para públicos infantis, que conseguem transmitir a sua história, apesar de apresentarem histórias menos conclusivas e deixadas um pouco à interpretação dos seus leitores, sem utilizarem uma única letra. Na obra "Reading the Visual An Introduction to Teaching Multimodal Literacy" [29], o autor expõe diferentes formas de como Literacia Visual é utilizada para contar histórias ou complementar técnicas tradicionais de escrita. Um dos exemplos mais interessantes é exatamente a de livros ilustrados sem qualquer palavra. Neste segmento da obra é explorado como, ao construir metodicamente um cenário muito detalhado, com ênfase às expressões faciais e emoções dos personagens os autores destes livros ilustrados conseguem transmitir a sua narrativa sem o recurso de elementos textuais [29]. O principal caso de estudo do autor para este tipo de obras foca-se no livro "Flotsam" [33] do escritor David Wiesner o qual conta as aventuras de um rapaz na praia que encontra uma câmara fotográfica que lhe permite ver fotografias de locais longínquos.



Figura 2.1: Excerto do livro explorado em detalhe da autoria de David Wiesner, Flotsam [33]

A grande vantagem deste projeto em relação a este tipo de media é o facto deste não se encontrar restrito a imagens estáticas e serem desta forma muito mais expressivas as emoções através de movimentos dos personagens.

### 2.1.3 Storyboards

A utilização de *Storyboards* tem diversas aplicações possíveis e desde há muito tempo que é utilizada para estudar problemas e planear trabalho. Tal era o caso com Leonardo Da Vinci, que afixava os seus projetos inacabados numa parede para os estudar, Walt Disney que aplicava esta mesma ideia na criação de algo que designava por "parede infinita"

para a organização dos milhares de frames necessários para alcançar "animação completa" nos seus filmes e de forma a ser possível observar a evolução da história com cada novo desenho adicionado à mesma [7]. A técnica ainda é atualmente utilizada por este estúdio, relatado em "Walt Disney, Easter Eggs, and Monsters" [15], que relaciona o processo de criação de um *Storyboard* com a função de um diretor de cinema ao adaptar o que está no guião para o que vai ser desenhado pelos artistas em termos de ângulos de câmara, comportamento e emoções das personagens, como se pode observar na Figura 2.2. Também em contexto empresarial são utilizados *Storyboards* para planeamento organizacional em multinacionais como a *McDonald's* ou a *Ford* [7]. É ainda uma excelente ferramenta de aprendizagem educacional, não só para transmissão de conhecimento como para fomentar criatividade e trabalho em equipa [11, 32].

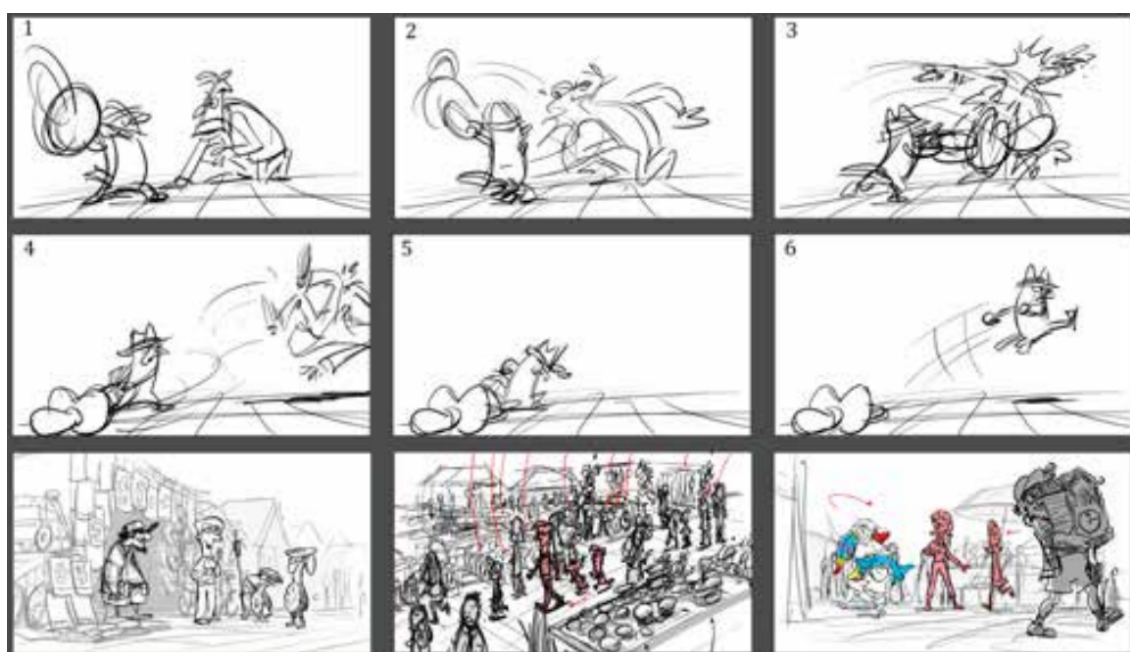


Figura 2.2: Recorte de *Storyboard* utilizado para adaptar o guião de um episódio do desenho animado da *Disney*, *Phineas e Ferb* para animação [15]

No que diz respeito à utilização de *Storyboards* no contexto do objetivo desta tese podemos analisar o trabalho realizado no artigo "Murder on Mansion Hill: Encouraging Collaborative Group Storytelling to Improve Motivational Aspects of Literacy Using Gameplay and Arts-Based Techniques" [11]. Neste, a autora expõe o desenvolvimento do seu jogo, que encoraja os jogadores a trabalhar em duas equipas rivais a criarem a sua história através de pequenas colagens de objetos para criar o seu ambiente de jogo. Depois da criação do cenário, a comunicação é permitida através de encenação por pistas para, como no jogo *Cluedo* [24], descobrir quem, onde e com que objeto foi morta a vítima. Um exemplo desta etapa de criação do espaço da história pode ser observado na Figura 2.3.

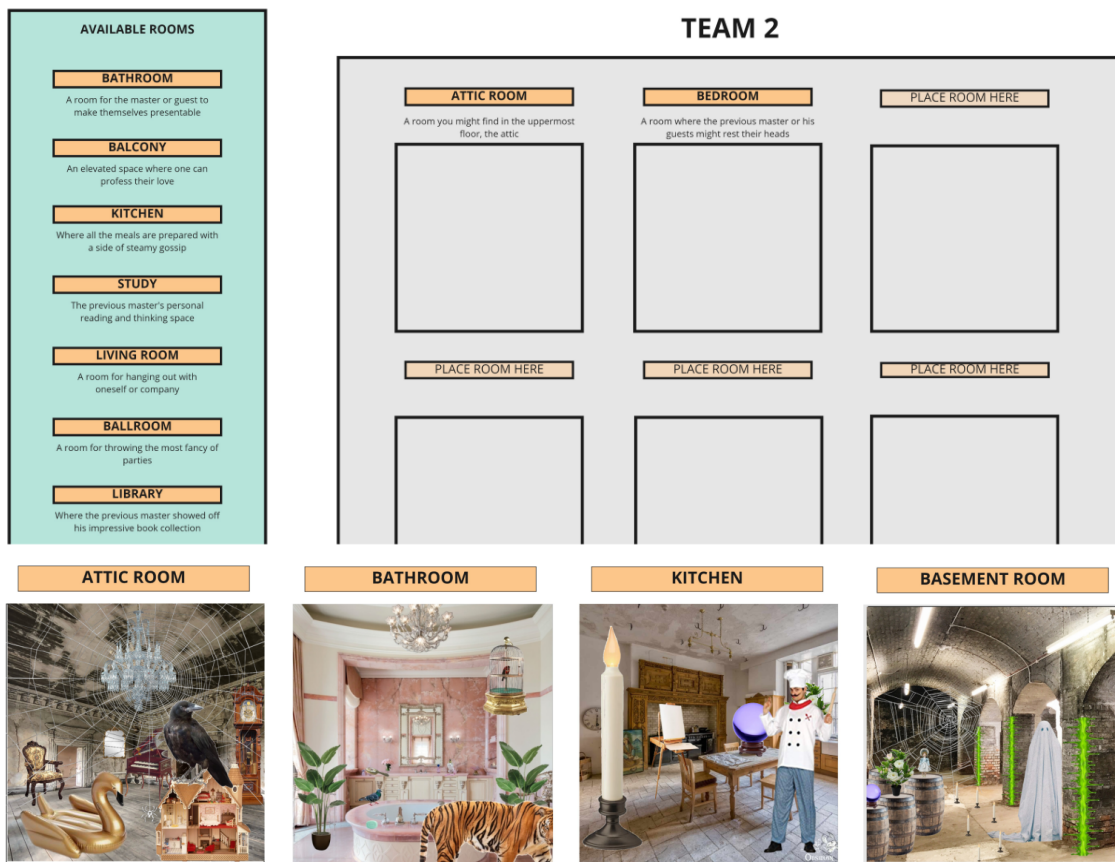


Figura 2.3: Tabuleiro de criação de cenário no qual os jogadores devem criar uma colagem das divisões que pretendem para a sua história (em cima) e exemplos de diferentes divisões criadas por jogadores (em baixo) [11]

Na sua solução, cujo foco principal é promover a criatividade e o interesse pela leitura em jovens, a autora fornece aos jogadores diferentes imagens, categorizadas em diferentes categorias como fundos, itens e personagens que estes devem adicionar ao seu espaço para cada divisão de modo a criar o ambiente perfeito para poder encenar as suas histórias [11]. Para o desenrolar do jogo em si, são seguidas as mesmas regras base do jogo Cluedo [24], com o pormenor de que os jogadores devem comunicar através da encenação apoiando-se nos cenários que criaram, para a criação de uma história mais interessante e didática.

Na criação do nosso *Storyboard* pretendemos seguir também esta abordagem de disponibilizar aos utilizadores todos os componentes que estes possam utilizar, organizados logicamente nos respetivos locais, que devem depois ser selecionados e feito também este trabalho de colagem pelo utilizador como no caso referido, para cada passo da história.

## 2.2 Design de VideoJogos

Ao contrário do que é a norma em design de software, cujo objetivo é criar software fácil de aprender, utilizar e dominar, os principais objetivos de qualquer videojogo são entreter,

manter interessado e desafiar as capacidades do jogador, através de obstáculos que devem ser ultrapassados com técnicas fáceis de aprender e utilizar mas difíceis de dominar [5, 8].

### 2.2.1 Lock-and-Key

Uma abordagem muito usada no design de videogames mas sobretudo nos Jogos de Aventura é o conceito normalmente referido como *Lock-and-Key*. Este, tal como o nome indica, vem da abstração de um obstáculo num videogame como sendo um cadeado que impede o progresso do jogador, vendo-se este obrigado a encontrar a respetiva chave que vai desbloquear este obstáculo e permitir que progrida no jogo. Apesar deste cadeado e chave poderem ser apenas isso, qualquer obstáculo que o jogador não consiga ultrapassar sem antes obter algum item que acrescente ou aumente algum comportamento seu, como a capacidade de poder efetuar um duplo salto ou adquirir a possibilidade de alterar o ambiente que o rodeia, também constituem exemplos de uma estratégia de design de *Lock-and-Key* [19]. Uma abstração simples de um jogo do ponto de vista desta estratégia pode ser observada na Figura 2.4.

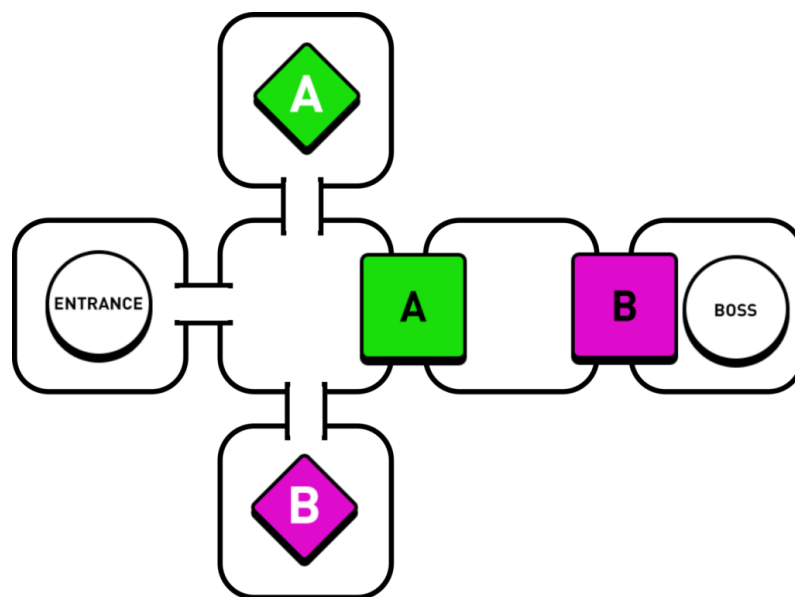


Figura 2.4: Abstração de um nível de jogo como um problema de *Lock-and-Key* da autoria de Mark Brown [3], diamantes representam a localização da chave e quadrados o respetivo cadeado

A maneira mais comum de criar uma situação cativante que beneficie desta estratégia para a criação de um desafio para o jogador, é apresentar primeiro o obstáculo e permitir que o jogador explore as suas opções para tentar resolver o problema com que se depara. Ao aperceber-se que não tem atualmente as capacidades necessárias para superar este obstáculo, o seu objetivo torna-se para a exploração do mundo à sua volta. Nele, irá encontrar outros obstáculos que será capaz de superar, para obter a chave que pode utilizar para progredir no jogo [26].

Esta estratégia de design apresenta ainda uma propriedade fundamental para o objetivo que se pretende alcançar, na criação de jogos de aventuras com criação procedimental e assistida a partir da especificação da história do mesmo. Este é o facto de transmitir ao jogador os objetivos que tem de alcançar. Ao contrário de um jogo de aventura tradicional em que todos os comportamentos e a progressão do jogador estão bem definidos, em cenários gerados procedimentalmente, é preciso outra solução para guiar o jogador [1]. Para definir então o rumo da história, é necessário que cada episódio especificado pelo utilizador na criação da mesma corresponda a um obstáculo que impeça a progressão do jogador. Só depois de este cumprir a ação pretendida no episódio, pode obter a chave que lhe permite avançar e progredir a história para o próximo episódio.

### 2.2.2 Criação de Missões para Jogos gerados Procedimentalmente

A utilização de técnicas de geração procedimental para a criação de jogos é usada em inúmeros cenários diferentes com diferentes objetivos. No entanto, para a criação de jogos de aventuras, é necessário algum cuidado especial na sua utilização devido à necessidade de criar narrativas que dependem de uma ordem fixa de eventos para transmitir a história como pretendida. Esta ordem, pode não ser alcançada, dependendo de como o algoritmo de geração procedimental dispôs as diferentes peças do puzzle [1, 10].

Para impor ordem na geração do ambiente para o utilizador, a técnica mais comum é a utilização de abordagens de *Lock-and-Key* (2.2.1) como é o caso nos artigos "The Quest in a Generated World" [1] e "Generating Missions and Spaces for Adaptable Play Experiences" [10]. No primeiro caso, o autor [1] expõe a sua proposta de solução no jogo que cria para este propósito "Charbitat", em que executa a geração do mundo e das missões durante a exploração do jogo. Sempre que o jogador chega a uma nova área que ainda não foi gerada, o jogo analisa o seu estado atual e decide qual a configuração que deve apresentar, geralmente, e para garantir uma abordagem de *Lock-and-Key*, vai criar um obstáculo ao progresso do jogador antes de disponibilizar, em outra área, a respetiva chave ao jogador assegurando-se sempre que é possível progredir no jogo.

No segundo artigo, é explorada a hipótese de dividir a criação destes jogos em duas fases, ao gerar a disposição das missões e criar o mapa como suporte para estas. A justificação para esta abordagem é apresentada na Figura 2.5 que compara a planta do mapa do nível "Forest Temple do jogo The Legend of Zelda: The Twilight Princess" (à direita) com a abstração das suas missões num grafo de dependências (à esquerda) e o facto de se poder realizar um mapeamento do grafo das missões para o espaço do mapa [9].

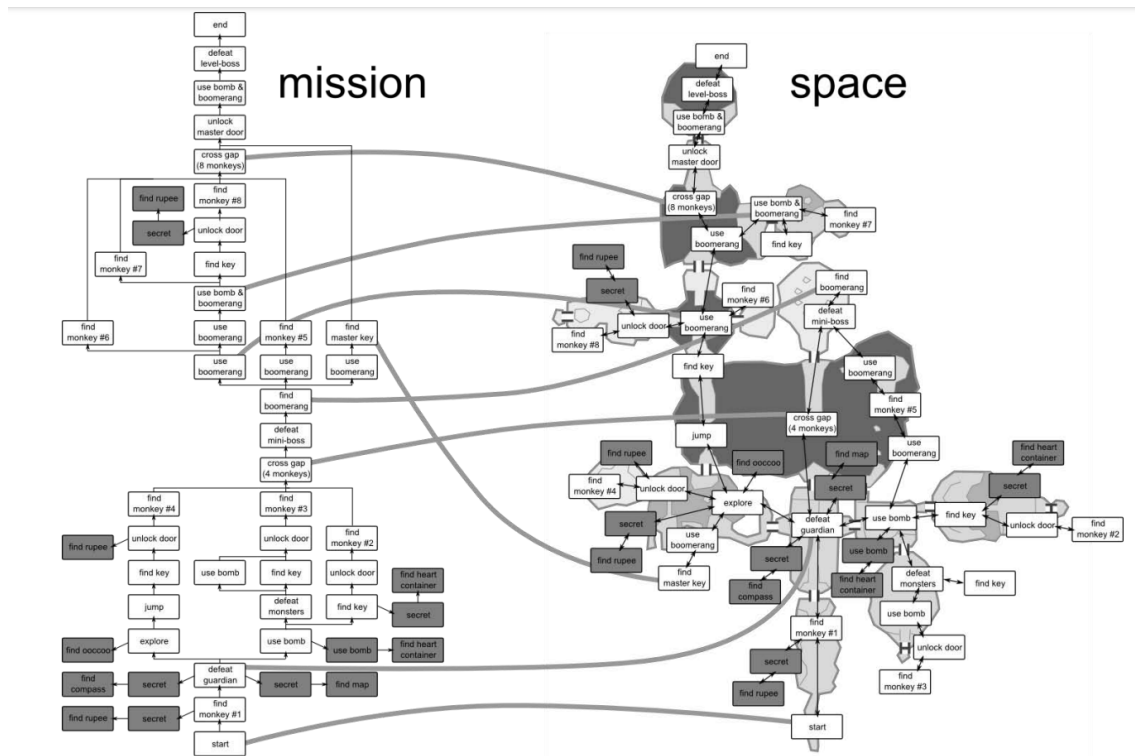


Figura 2.5: Esquema da missão e espaço no nível Forest Temple do jogo The Legend of Zelda: The Twilight Princess [9, 10]

O autor sugere a utilização de *generative grammars*, mais especificamente *graph grammars* para gerar as missões que podem ser representadas como um grafo direcionado constituído por um início, um objetivo e tarefas intermédias. Ao definir um conjunto robusto de regras que permita emular o comportamento presente em puzzles *Lock-and-Key* ao especificar, por exemplo, que os cadeados devem aparecer antes das chaves e que os cadeados permitem o jogador avançar mas que para obter as chaves o jogador deve recuar para as encontrar, esta gramática seria capaz de gerar diferentes missões com desafios únicos e permitiria criar o grafo apresentado na Figura 2.5 e outros semelhantes, a partir do qual se poderia depois gerar o espaço do jogo [10], tarefa essa que já não diz respeito a este tema de dissertação.

### 2.3 Criação Assistida de Jogos

Existem diferentes níveis de ferramentas no que toca a uma criação assistida de jogos dependendo também da capacidade técnica do utilizador e do resultado que este pretende alcançar. Desde motores de jogo que fornecem apenas uma base muito completa das propriedades fundamentais para desenvolvimento de jogos profissionais, até criação assistida de jogos sem a necessidade de escrever código e jogos que permitem alterar comportamentos implementados com novas regras e criar novas experiências.

### 2.3.1 Motores de Jogo

Quando se fala em motores de jogo referimos-nos à coleção de módulos de código que não especificam o comportamento do jogo ou os pormenores do seu estado mas que, são responsáveis por lidar com a parte de geração de gráficos 2D e 3D, som, deteção de colisões e outras propriedades físicas básicas [18]. Ao fornecerem aos programadores bibliotecas capazes de lidar com estas propriedades universais para o desenvolvimento de qualquer videogame, os motores de jogo asseguram um desenvolvimento de jogos mais rápido e completo.

Apesar de ser possível criar jogos sem a utilização de motores de jogo, qualquer empresa focada no desenvolvimento de videogames irá ter o seu próprio motor de jogo com características focadas para o género de jogos que produz e mesmo pequenos desenvolvedores independentes, recorrem à criação dos seus próprios motores ou à utilização de motores de jogo como *Unity* ou *Unreal Engine* para desenvolverem os seus jogos [30].

#### 2.3.1.1 Unreal Engine

O *Unreal Engine*, exemplo do editor na Figura 2.6, é um motor de jogo criado pela *Epic Games* para criação de videogames que fornece técnicas de processamento de gráficos, simulação de físicas básicas, iluminação e sombras, *scripting visual*, animação de personagens e inúmeras outras propriedades que qualquer equipa de desenvolvedores profissionais ou um programador independente necessita para criar o seu jogo [30].

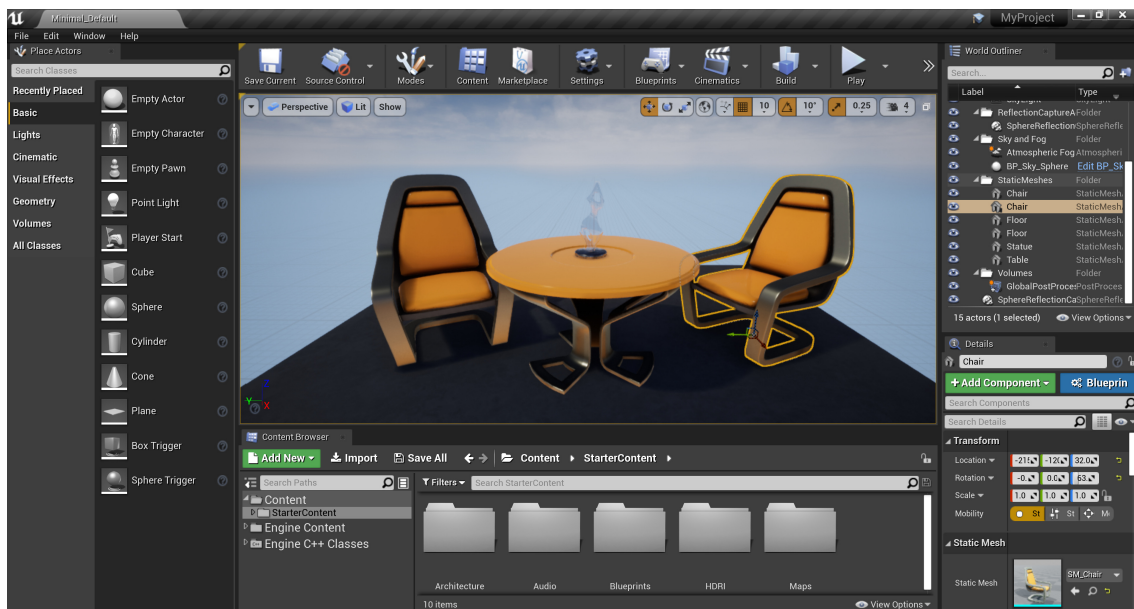


Figura 2.6: Level Editor de desenvolvimento de um jogo em *Unreal Engine*

A maior parte das linguagens de programação são textuais, no entanto, o *Unreal Engine 4* introduz o conceito de *Blueprints*, uma forma diferente de programação visual ao utilizar

uma interface baseada em nós. As *blueprints* são também consideradas uma forma de *scripting* porque existem num ambiente especial e têm um objetivo bem definido sendo então uma forma de *scripting visual* que permite a utilizadores programar comportamentos nos seus jogos sem a necessidade de escrever código e necessitando apenas de entender alguns conceitos de programação de jogos muito básicos [30, 27]. Um exemplo muito simples de uma *blueprint* que ativa uma luz no momento em que o jogo é iniciado pode ser observada na Figura 2.7. Para criar esta funcionalidade apenas foi necessário criar uma nova *blueprint*, seleccionar na cena a propriedade do objeto que se quer alterar e indicar no editor que quando o jogo começar se pretende alterar essa propriedade seleccionada.

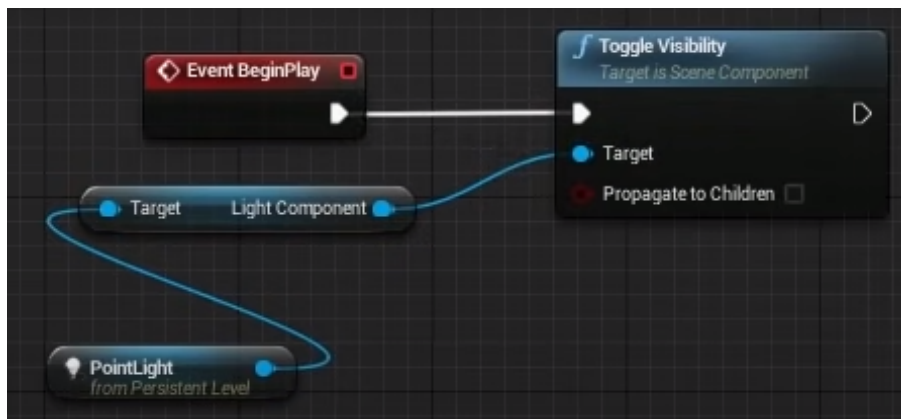


Figura 2.7: *Blueprint* que ativa a visibilidade de uma lâmpada quando o jogo é iniciado

Devido à possibilidade ainda de se poder definir comportamentos mais complexos do que aqueles apresentados na Figura 2.7, programados numa linguagem mais tradicional como C++ e depois referenciados como nesse exemplo, a utilização de *blueprints* permite não só que pessoas com poucos conhecimentos de programação consigam criar alguns comportamentos simples sozinhos, como ainda permite no caso de estúdios de desenvolvimento de videojogos, a divisão de tarefas entre programadores que definem a lógica das tarefas e designers que podem depois utilizar estas classes criadas e desenvolver o jogo como pretendem. Devido à sua natureza visual e interface simples qualquer pessoa, independentemente dos seus conhecimentos de programação, consegue após um curto processo de aprendizagem, utilizar *blueprints* para especificar os comportamentos que pretenda [30, 27].

Outra possibilidade disponível aos utilizadores é o *Unreal Engine Marketplace*, como apresentado na Figura 2.8, uma extensão do *Unreal* que permite a desenvolvedores colocar à venda conteúdo que criaram, no mercado, a outros utilizadores como efeitos sonoros, texturas, modelos, animações e até projetos inteiros que os utilizadores podem depois usar como pretendem. A utilização destes componentes pode possivelmente poupar centenas de horas de tempo de desenvolvimento de um projeto e é uma excelente forma de aprender boas práticas de desenvolvimento de videojogos, ao permitir a utilizadores novatos estudar a maneira de trabalhar de utilizadores mais experientes [30].

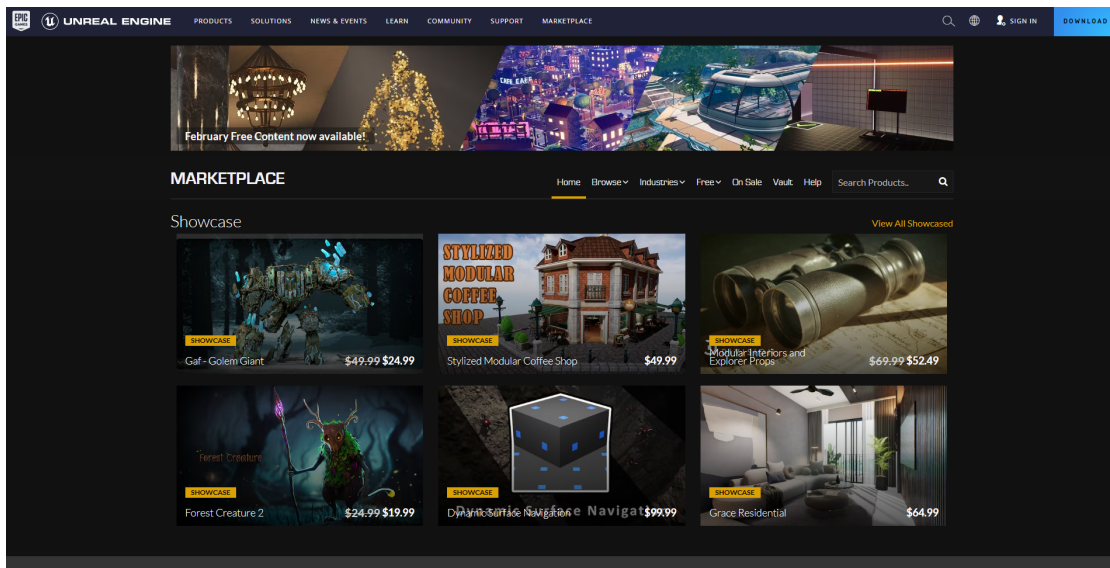


Figura 2.8: Unreal Engine Marketplace

### 2.3.1.2 Unity

O *Unity*, exemplo do editor na Figura 2.9, também conhecido como *Unity3D*, é um motor de jogo para criação de jogos com ferramentas responsáveis por processamento de gráficos, áudio, físicas simples e outras propriedades universais para o desenvolvimento dos mesmos. A primeira versão, inicialmente disponível apenas para *Mac OS X*, foi disponibilizada ao público em 2005 com o objetivo de fornecer um motor de jogo com ferramentas profissionais para desenvolvedores amadores de forma a democratizar a indústria de desenvolvimento de videogames [16, 14].

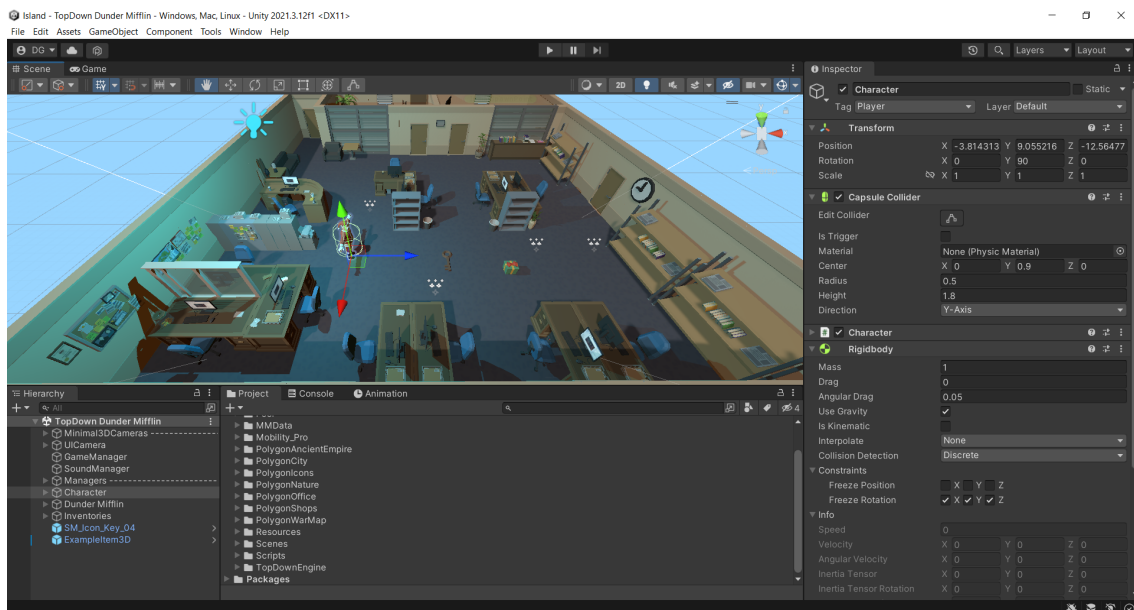


Figura 2.9: Level Editor de desenvolvimento de um jogo em Unity

Tal como é o caso com o desenvolvimento de jogos em *Unreal Engine*, *Unity* também usa uma linguagem de *scripting*, mas não possui qualquer tipo de *scripting visual* como as blueprints do *Unreal*. O motor de jogo apresenta no entanto algo chamado "propriedades visuais" que permitem editar variáveis públicas em *scripts* diretamente no editor sem a necessidade de alterar código [16]. Um exemplo desta funcionalidade pode ser observada na Figura 2.10, na qual se pode facilmente modificar, inclusive durante a execução do jogo, o parâmetro da velocidade do personagem, para poder facilmente testar diferentes valores para, por exemplo, encontrar uma velocidade que corresponda melhor à animação do seu movimento.

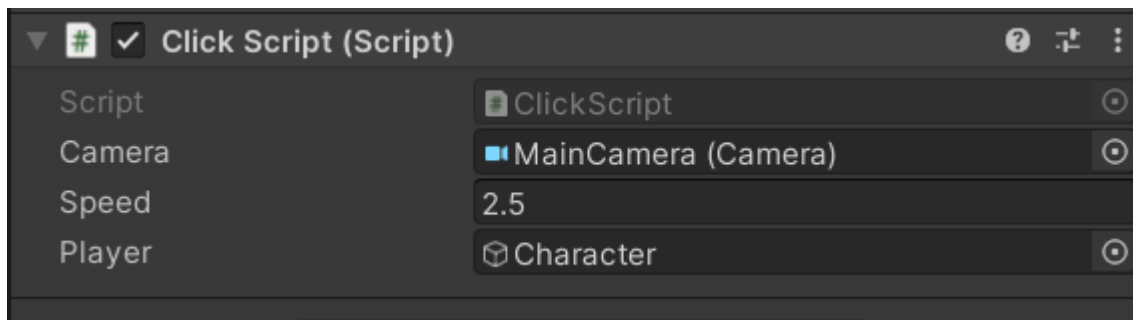


Figura 2.10: Exemplo de *Script* com propriedades visuais

O *Unity* também possui o seu mercado público, o *Unity Asset Store*, presente na Figura 2.11, lançado em 2010 e incluído também no próprio *IDE*, que permite aos seus utilizadores colocar à venda e comprar, diferentes projetos no mercado [16], desde pequenos adereços a conjuntos de ferramentas para assistir na criação de jogos, como é o caso dos pacotes da *MoreMountains* [23] e *Adventure Creator* [6] que serão abordados de seguida.

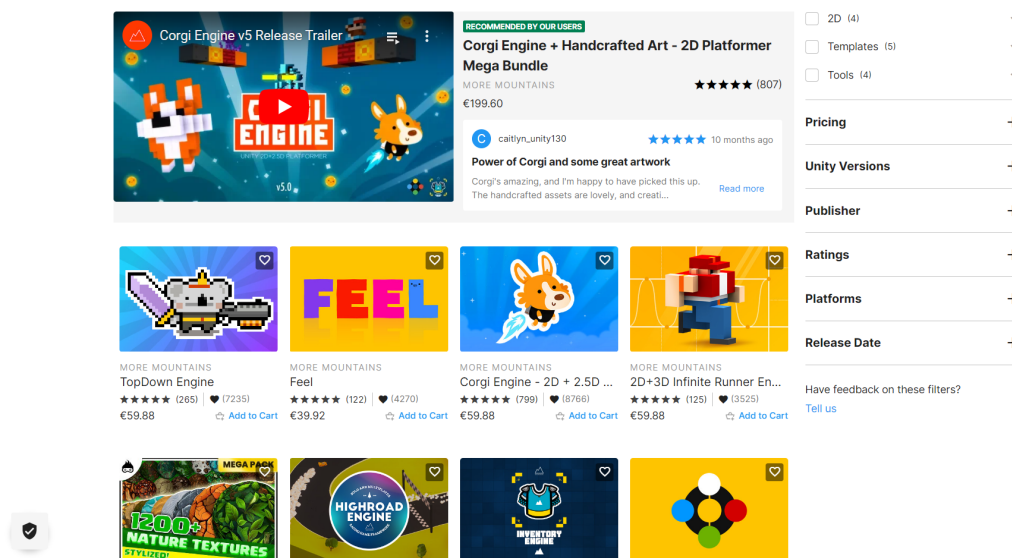


Figura 2.11: *Unity Asset Store*

A *MoreMountains* é um estúdio da autoria de Renaud Forestié que disponibiliza algumas ferramentas que desenvolveu ao longo dos anos a trabalhar na área de criação de videojogos, como por exemplo o *Feel*, um pacote que desenvolveu para facilmente adicionar *feedbacks* a qualquer jogo através de efeitos e micro-interações com objetos com que o jogador interage, o *Inventory Engine* que auxilia na criação e manutenção de um sistema de inventário e itens simples e o *TopDown Engine* uma ferramenta extremamente útil para qualquer pessoa que queria criar jogos em *Unity* com uma vista *Top-Down* [23]. Estas ferramentas disponibilizadas pela *MoreMountains* não só auxiliam a parte de programação do jogo, ao fornecerem alguns comportamentos com sugestões de implementações que podem, ou não, ser expandidas pelo utilizador para adicionar novas funcionalidades pretendidas para o seu jogo, como acrescentam ainda centenas de pequenos pormenores que apesar de não serem fundamentais para o correto funcionamento do jogo, têm um impacto enorme na perceção e satisfação dos jogadores no produto final.

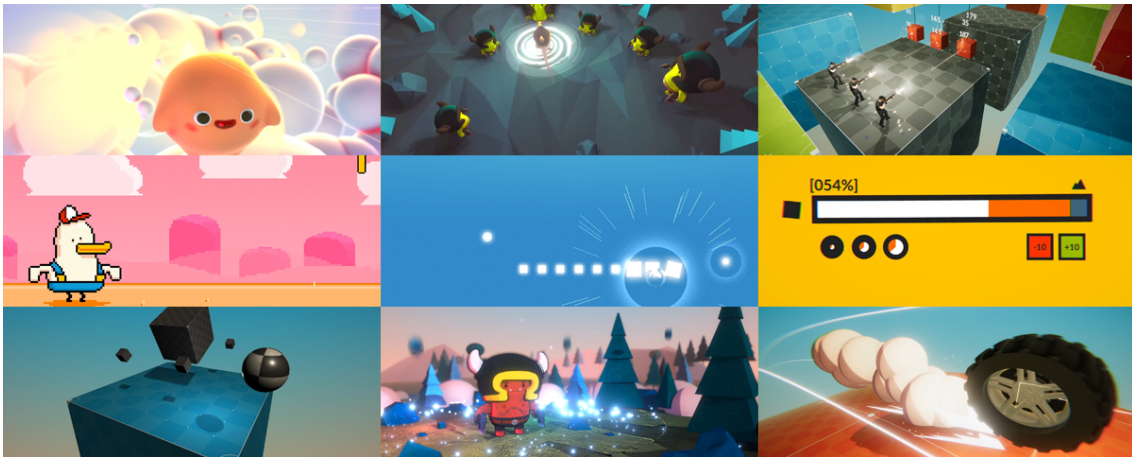


Figura 2.12: Exemplos da utilização do pacote *MoreMountains Feel*

Outra ferramenta também disponível no *Unity Asset Store* para assistir na criação de videojogos é o *Adventure Creator*, que é descrito como sendo "um conjunto de ferramentas com tudo o que é necessário para criar o seu próprio jogo de aventuras. De jogos *point-and-click* 2D como *Monkey Island* a épicos cinemáticos como *The Walking Dead* do estúdio *Telltale*" [6]. As principais características desta ferramenta são o facto de permitir definir comportamentos simples mas comuns em jogos de aventura, como transição entre cenas e interação com o mundo, através da sua abordagem de construção com ações e a sua implementação de *scripting visual* em *Unity*.

Pode-se observar a utilização destas funcionalidades na Figura 2.13 em que, na parte da esquerda, é definida, com a abordagem de construção com ações, a transição para outra cena, após a colisão com uma área definida e, na direita, utilizando *scripting visual*, é definido o estado do jogo e o diálogo a ser apresentado ao jogador dependendo da visibilidade de um obstáculo na cena. Tal como no caso das *Blueprints* no *Unreal Engine*, esta ferramenta permite então a qualquer utilizador criar jogos relativamente complexos

sem a necessidade de escrever uma única linha de código.

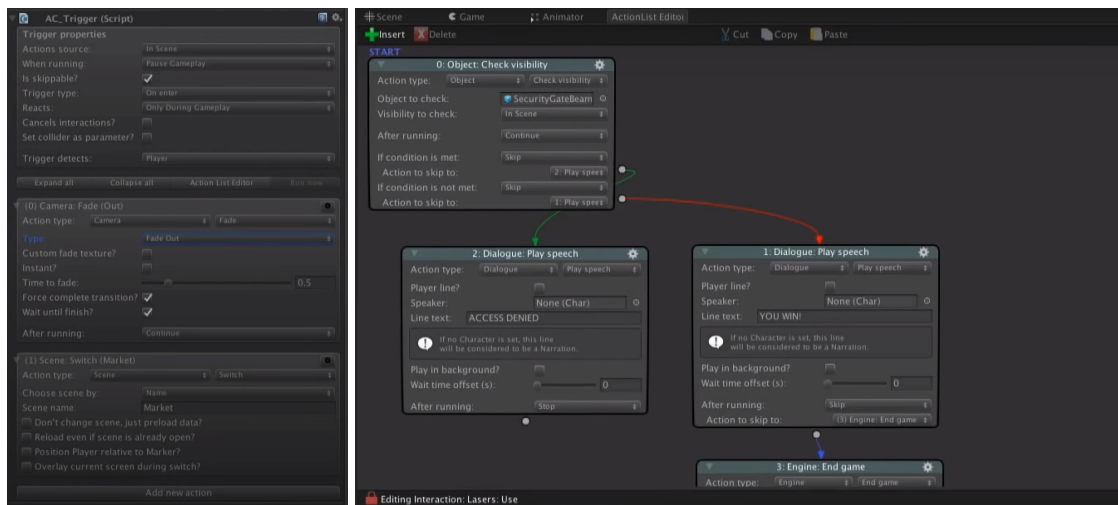


Figura 2.13: Exemplo de definição de comportamentos com *Adventure Creator*

Todas estas ferramentas, apesar da imensa ajuda que fornecem a um utilizador que queira desenvolver o seu jogo, devido a permitirem qualquer jogo possível ser desenvolvido, não conseguem apresentar o nível de simplicidade e rapidez pretendido nesta dissertação para um utilizador sem conhecimentos na área de videojogos criar e jogar a sua história.

### 2.3.2 Jogos

Para além das ferramentas mencionadas nas secções anteriores que auxiliam utilizadores a criar jogos, existem alguns videojogos que permitem aos seus jogadores, com diferentes níveis de sucesso neste processo, customizar, modificar ou criar novos comportamentos de forma a alterar o jogo em si e até criar experiências completamente diferentes da fornecida inicialmente.

Um estúdio de jogos conhecido exatamente por criar este tipo de experiências para os seus utilizadores é a *Media Molecule*, fundada em 2006 e responsável pelo desenvolvimento da série de jogos *LittleBigPlanet* [22], com três lançamentos diferentes, de 2008 a 2014 e *Dreams* [21] em 2020, ambos para a *PlayStation*.

Os jogos da série *LittleBigPlanet*, para além de disponibilizarem aos seus jogadores campanhas *single-player*, destacam-se pelo facto de fornecerem aos utilizadores um sistema de criação de níveis de jogo totalmente adaptável. Este modo de jogo permite aos jogadores construir os seus próprios cenários e definir alguma lógica para criação de comportamentos simples nos seus jogos, mas sempre por cima das mecânicas de jogo já existentes do jogo base [17].

Na Figura 2.14, podemos observar alguns pormenores da criação de um nível no último jogo da série, *LittleBigPlanet 3*, na figura da esquerda podemos observar o menu em que o utilizador define os parâmetros do comportamento que o objeto selecionado

deve executar no cenário do jogo. Neste caso, a corrente deve diminuir de tamanho ao longo de quatro segundos e, na figura da direita, podemos observar o menu de seleção de material para os blocos de construção que o utilizador utilizou para criar o cenário.

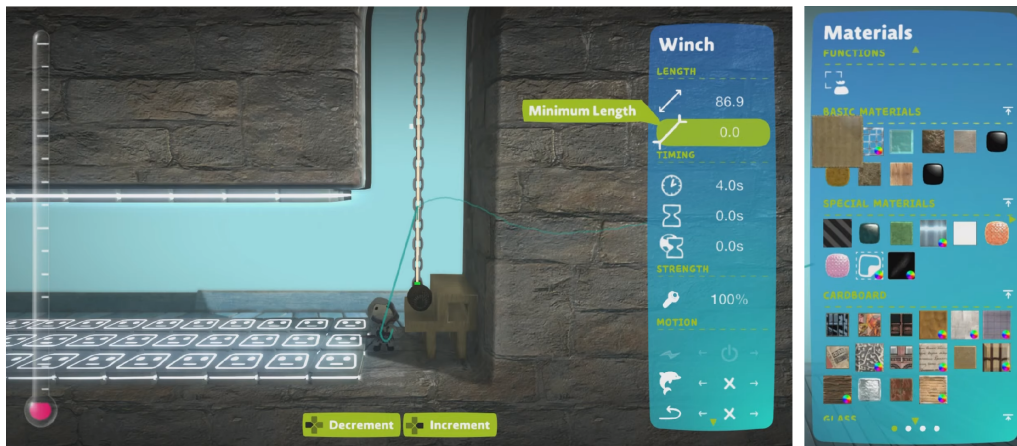


Figura 2.14: Exemplo da criação de um nível no videojogo *LittleBigPlanet 3*

Devido ao sucesso obtido com os diferentes jogos da série *LittleBigPlanet*, a *Media Molecule* disponibilizou em 2020 o jogo *Dreams*. Este jogo é caracterizado por ser puramente um sistema de criação de jogos, que possui um conjunto extenso de ferramentas à disposição dos seus utilizadores para criação de objetos, animações, música e programar visualmente quaisquer comportamentos necessários. É caracterizado também, por apresentar uma abordagem de aprendizagem muito típica de um videojogo, que permite criar qualquer jogo que o utilizador pretenda [17]. Um exemplo de um jogo criado unicamente em *Dreams*, pode ser observado na Figura 2.15, na qual, o jogador deve navegar a sua aeronave através de um colorido cenário repleto de obstáculos e alcançar diferentes *checkpoints* gerados aleatoriamente antes do limite de tempo acabar.



Figura 2.15: Produto final de um jogo criado em *Dreams*

O desenvolvimento de jogos em *Dreams* apresenta algumas semelhanças à do desenvolvimento de jogos de forma mais tradicional, juntamente com muitas das características da interface de outros jogos do estúdio, devido às limitações do sistema em que foi lançado.

O processo de criação dos objetos que vão ser utilizados no jogo como o cenário, personagens, animações e sons apresenta a mesma interface baseada em seleção do componente e escolha de alteração que se pretende efetuar e moldagem da forma do objeto tal como nos jogos da série *LittleBigPlanet*. Um exemplo pode ser observado na Figura 2.16, na qual, de momento, o utilizador está a efetuar pequenos ajustes na asa do avião que está a criar para o jogo na Figura 2.15.



Figura 2.16: Criação do modelo de um avião em *Dreams*

No que diz respeito à criação da lógica, controlos e dos comportamentos do jogo que são aplicados aos objetos, ao contrário do que era o caso com outros jogos do estúdio, que só permitiam alguns comportamentos básicos com opções algo simples e que não permitiam ao utilizador afastar-se muito do jogo base, em *Dreams*, o utilizador tem à sua disposição *scripting visual* tal como nos exemplos mencionados anteriormente que lhe permite criar qualquer comportamento que necessite de implementar para o jogo que pretenda. Na Figura 2.17 podemos observar, apesar de não ser possível visualizar as funções em si por uma questão de clareza visual do jogo, um excerto da utilização de *scripting visual* responsável pelo movimento da aeronave do jogo presente na Figura 2.15.



Figura 2.17: Criação da lógica de jogo e controlos a aplicar ao avião em *Dreams*

Considerando o que permite ser desenvolvido dentro dos próprios jogos, as ideias abordadas pelo estúdio responsável por estes videojogos é algo que vai ter uma grande influência no projeto desenvolvido para esta dissertação. Ao contrário do que era o caso com os outros exemplos apresentados, mas não por isso menos importantes o seu estudo, o público alvo destes jogos são também utilizadores mais casuais, que pretendem desenvolver jogos e histórias menos complexas, muito rapidamente e de uma forma mais divertida. O objetivo desta dissertação, no entanto, é permitir uma construção ainda mais rápida e simples, mas também por isso, um pouco menos manipulável por parte do utilizador de experiências interativas na forma de Jogos de Aventura. Não se pretende que o utilizador tenha que se preocupar com realizar qualquer tipo de *Scripting Visual*, criação de modelos ou qualquer outra tarefa remotamente difícil ou demorada.

### 3.1 História

Como foi explorado na secção 2.1.3, *Storyboards* são utilizados de diferentes formas para representar e contar histórias de uma maneira clara e concisa. A sua utilização permite desconstruir algo de grandes dimensões e elevada complexidade em pequenas etapas mais fáceis de compreender e modificar. Considerando essa informação, a interação do utilizador com a estrutura da história seria, preferencialmente, sempre apresentada na estrutura de *Storyboard*.

No caso de uma história, cada entrada do *Storyboard* é um passo da mesma que precisa de representar toda a informação relativa a esse passo. Os passos de uma história são etapas sucessivas desta em que o estado da história é alterado. É importante, no entanto, que a criação e leitura desta estrutura seja rápida e simples para o utilizador e para isso, é necessário trocar, possivelmente, alguma perda de expressividade para obter uma solução mais prática. Desta forma, foi abordada a hipótese de representar cada passo de uma história como tendo um objeto ou personagem no mundo envolvente, com que o jogador tem de interagir para desencadear o passo, os requisitos necessários, se algum, do estado da história ou objeto que necessita de utilizar, para o mesmo poder ser iniciado, as alterações que são efetuados ao estado da história, os objetos, se algum, que recebe da conclusão do passo e a reação do personagem ao realizá-lo com sucesso.

A utilização de *Storyboards* tradicionais apresenta, no entanto, uma limitação. O facto de ser uma estrutura linear é perfeito para representar passos de histórias tradicionais, que ocorrem numa ordem cronológica fixa mas, apresenta um problema quando se pretende representar uma história que apresente caminhos divergentes ou concorrentes como pode ser o caso em Jogos de Aventura. Para ultrapassar esta situação, decidiu-se optar pela utilização de um Fluxograma em que cada entrada deste corresponderia a um passo da história do *Storyboard*.

Esta solução consiste em fornecer ao utilizador a capacidade de adicionar novos passos ao Fluxograma, cada um destes passos contém a informação referente a um passo da história, o objeto ou personagem com que se deve interagir, os requisitos deste passo, a

reação do personagem, o objeto que deve ser adquirido para o inventário e ainda, caso o utilizador pretenda, diálogo apresentado na conclusão do passo. O motivo pela qual é interessante usar um Fluxograma é permitir ao utilizador criar ligações, setas direcionadas, entre os vários passos criados. Desta forma, é possível criar dependências entre passos de uma história de uma forma simples, com representação visual e sem a necessidade de uma especificação de dependências trabalhosa.

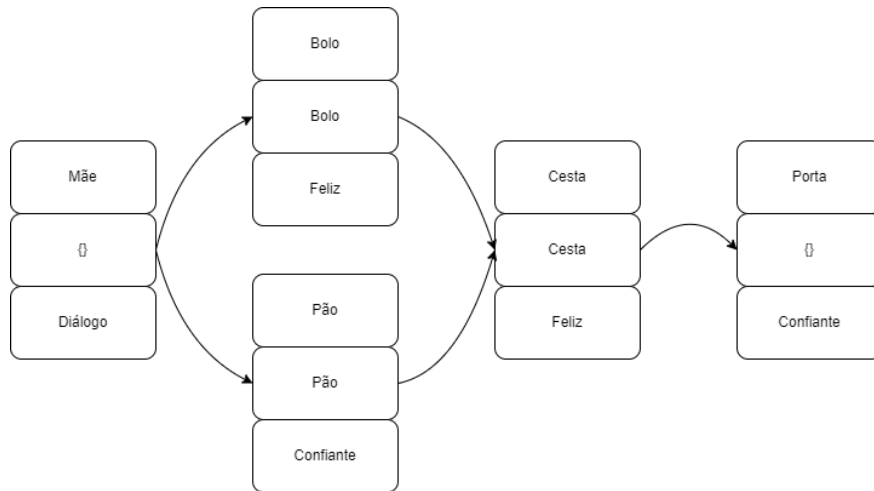


Figura 3.1: Exemplo de Fluxograma que representa passos da história de O Capuchinho Vermelho

Na figura 3.1 são especificados, primeiro, os objetos ou personagens com que o jogador tem de interagir nos primeiros cinco passos da história, de seguida, os objetos que são recebidos ao recolher o pão, bolo e a cesta e, finalmente, a reação do personagem no momento em que é efetuado cada passo. A utilização de setas indicadoras da direção do fluxo permite inferir as dependências entre passos da história, de forma que, não é necessário representar os Requisitos e Alterações do estado da história na entrada de cada passo.

Ativador	Mãe	Pão	Bolo	Cesta	Porta
Requisitos	{ }	{Falar com Mãe}	{Falar com Mãe}	{Adquiriu Pão, Adquiriu Bolo} {Pão, Bolo}	{Guardou Pão e Bolo}
Alterações	{Falar com Mãe}	{Adquiriu Pão}	{Adquiriu Bolo}	{Guardou Pão e Bolo}	{Saiu de Casa}
Recebe	{ }	Pão	Bolo	Cesta com Pão e Bolo	{ }
Reação	Diálogo	Confiante	Feliz	Feliz	Confiante

Figura 3.2: Exemplo de *Storyboard* que representa passos da história de O Capuchinho Vermelho

Na figura 3.2 observa-se o *Storyboard* alternativo ao Fluxograma presente na figura 3.1. Para além dos elementos presentes no Fluxograma, devido à ausência das setas entre

passos para definir dependências, é necessário incluir os pré-requisitos que os passos sem ser o inicial necessitam e as alterações efetuadas ao estado da história. Nesta outra solução, a existência de ciclos, caminhos divergentes ou concorrentes é mais dificilmente criada e detetada pelo utilizador.

O objeto recebido pelo jogador após a conclusão com sucesso de alguns passos, pode parecer trivial do ponto de vista da interpretação do *Storyboard* criado. É, no entanto, impactante para o jogador quando este joga a história ao necessitar da sua utilização ativa para completar os passos que dependem da utilização de objetos. Se especificar, por exemplo, a necessidade de seleccionar e utilizar uma chave para abrir uma porta.

O *Storyboard*, tal como qualquer outro programa, deve permitir ao utilizador indicar o início e o fim da sua execução. Dessa forma é necessário permitir identificar o passo, ou passos, iniciais, pela dependência de um objeto inicial e o passo, ou passos, finais, ao adquirir um objeto final após a sua conclusão com sucesso.

Finalmente, para providenciar uma maior flexibilidade de opções para contar histórias ao utilizador, as dependências entre passos podem ser opcionais ou obrigatórias. Se um passo possuir múltiplas dependências, estas podem ser identificadas, individualmente, como sendo opcionais ou obrigatórias deste passo. Qualquer passo, para poder ser efetuado, necessita que todas as suas dependências obrigatórias estejam concluídas antes da sua tentativa e não depende das suas dependências opcionais. Passos que apresentem apenas dependências opcionais, necessitam que, pelo menos uma, destas esteja concluída para poder ser efetuado.

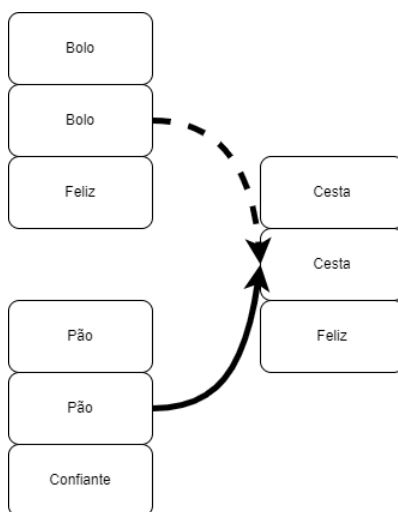


Figura 3.3: Exemplo de *Storyboard* que representa passos opcionais e obrigatórios da história de O Capuchinho Vermelho

Na Figura 3.3 observa-se um excerto dos *Storyboards* apresentados anteriormente mas, neste caso, pretende-se ilustrar uma construção da história em que é obrigatório obtermos o objeto indicado no passo com a linha preenchida e em que o passo com a linha a tracejado é opcional. Ao jogar esta versão da história, o jogador é obrigado a realizar o passo em que recolhe o Pão e pode, ou não, recolher o Bolo. Na versão apresentada, previamente, na

Figura 3.2, o jogador podia realizar também, ambos os passos, ou apenas um deles, para prosseguir a história.

## 3.2 Capacidades Teóricas do Storyboard

Para todos os efeitos, o *Storyboard* desenvolvido é uma *Domain Specific Language (DSL)* que permite a representação, manipulação e leitura de diferentes histórias. A interface apresentada ao utilizador, o Fluxograma manipulado por este, tem como objetivo permitir a utilização do *Storyboard* sem a necessidade de escrever código tal como numa *Visual Programming Language (VPL)*.

Para além da *Storyboard* e do Interpretador da História, é importante entender o conceito de Inventário e dos objetos que este possui. Todos os passos da história apresentam requisitos de objetos e adquirem outros para serem usados em passos seguintes. Estes objetos podem ser objetos com representação visual, podendo ser apresentados no inventário ao jogador, como outros personagens, mobílias ou adereços ou podem não possuir uma representação visual, serem objetos abstratos, apenas para garantir uma abordagem *Lock-and-Key*. Todos estes objetos adquiridos ao longo do desenrolar da história são adicionados e removidos do Inventário, este pode ser visto como uma coleção, em memória, do estado atual da história, uma representação dos passos já concluídos.

O resultado gerado da criação da história pelo utilizador deve conter a informação de todos os passos da história. Informação essa que é processada pelo Interpretador da História 3.3, quando o jogador a pretender jogar. Com o objetivo de permitir ao utilizador contar qualquer história que pretenda, a *DSL* apresenta as funcionalidades típicas de uma linguagem de programação:

- Manipulação de Memória - O inventário do jogador, da responsabilidade do Interpretador da História manter, representa a memória. Quando o jogador completa um novo passo da história, são retirados do inventário os objetos necessários para completar o passo e adicionados os objetos adquiridos da realização deste;
- Ciclos e Condições - As ligações entre passos da história no Fluxograma permitem criar ciclos e condições lógicas dependentes da utilização, e consumo, de objetos obtidos pelo jogador anteriormente e das dependências dos passos. Desta forma é possível impedir o jogador de cumprir um passo se não possuir os objetos necessários ou se não os utilizar para a realização deste;
- Atribuições de Memória - Ao utilizar passos com recolha de objetos o utilizador está a atribuir esses objetos ao inventário (memória) aquando a conclusão do passo;
- Início e Fim - O *Storyboard* permite ao utilizador especificar o início ou inícios da história ao identificá-los desta forma. O fim ou finais da história são identificados de uma forma semelhante, não se pretenderia que todos os caminhos que acabem sem um novo passo do diagrama fossem fins da história;

Analogamente com outras *DSLs* e linguagens de programação, a disponibilidade de estes comportamentos indicam que, com a utilização do *Storyboard* desenvolvido e, considerando uma história como sendo um programa, deverá ser possível representar qualquer história.

### 3.3 Interpretador da História

Para processar o *Storyboard* criado é possível seguir um de dois caminhos, desenvolver um Interpretador ou um Compilador para a *DSL* utilizada. Para esta dissertação foi escolhido desenvolver um Interpretador capaz de computar os resultados necessários da leitura do *Storyboard* em vez de um Compilador devido ao ambiente em que se desenvolveu o projeto.

Ao representar cada passo de uma história com todos os parâmetros mencionados anteriormente é necessário manter um estado da história, isto é, as alterações que são obtidas após cada passo da história ser realizado. Para representar este estado da história pode ser guardado um inventário que será atualizado após a conclusão de cada passo. Esta estratégia, ao contrário de uma outra solução, em que, por exemplo, cada passo guardaria os passos que necessitava estarem concluídos para poder ser executado e em que se registaria no interpretador os passos já concluídos, segue a técnica de *Lock-and-Key* 2.2.1 ao abordar o requisito para realizar um passo como se de um item que o jogador necessita de obter anteriormente. Esta diferença permite uma implementação uniforme entre passos que necessitem que o utilizador utilize um item do seu inventário com passos que não necessitem.

### 3.4 Comportamentos do Jogador

Como em qualquer Jogo de Aventura, o jogador é capaz de movimentar o seu personagem, interagir com o mundo que o rodeia e possuir um inventário onde guarda itens que pode usar para interagir com os objetos ou personagens do mundo.

## IMPLEMENTAÇÃO

### 4.1 Visão Geral do Sistema

O trabalho desenvolvido nesta dissertação está integrado num jogo que permite ao seu utilizador criar e jogar as histórias que construa no estilo de um Jogo de Aventura.

O sistema permite ao utilizador gerar um novo projeto, no qual pode criar uma história, passo a passo, adicionando novos nós a um Fluxograma que devem ser preenchidos selecionando diferentes objetos ou personagens e reações de uma lista pré-definida e arrastados para o passo. Cada passo do Fluxograma permite indicar, a ordem das dependências da história, o objeto ou personagem com que se deve interagir para desencadear o passo, a animação do personagem principal e do personagem com que se interage, um pequeno diálogo a ser apresentado ao jogador e, se for o caso, o objeto que deve ser adquirido após a conclusão deste. Nesta fase, o utilizador também deve criar o cenário que necessita para que a história que desenvolveu se desenrole.

O jogo permite ainda que o utilizador selecione um projeto já criado e guardado na Base de Dados que pode depois jogar para experimentar a sua história ou voltar a editá-la. Alguns exemplos destes comportamentos podem ser observados na Figura 4.1



Figura 4.1: Menu Principal (esquerda) e História a ser jogada (direita)

## 4.2 Arquitetura do Sistema

A Arquitetura do Sistema desenvolvido pode ser dividida em três secções distintas cada uma responsável por uma etapa diferente do sistema. A Criação e Edição de uma História por parte do utilizador, uma Base de Dados para armazenamento das histórias criadas e o Processamento do Jogo que permite interpretar e jogar a história selecionada. Na Figura 4.2 é possível observar uma representação da Arquitetura do Sistema.

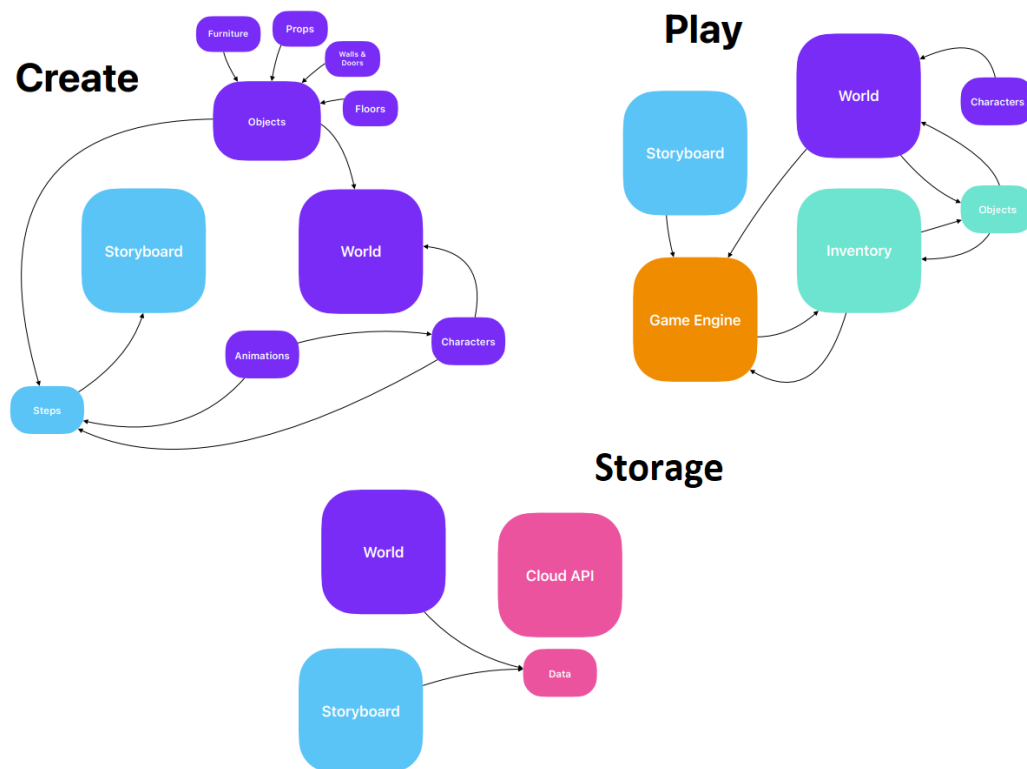


Figura 4.2: Arquitetura do sistema

Uma explicação mais detalhada de cada componente é apresentada a seguir:

- **Create:**
  - **Objects** - coleção de Objetos variados como mobílias, adereços ou elementos estruturais que o utilizador tem à sua disposição para permitir a construção do Cenário e dos diferentes Passos da História;
  - **Characters** - coleção de Personagens humanoides, que o utilizador pode utilizar, à semelhança dos Objetos, para caracterizar o Cenário e Passos da História. A estas, no entanto, podem-se aplicar Animações e permitem definir o aspeto do personagem principal;

- *Animations* - coleção de diferentes Animações aplicáveis a Personagens humanóides e aos Passos da História. As Animações são acionadas após a conclusão de cada Passo da História e quando se interage com um Personagem com Animação definida;
- *Step* - entidade responsável por representar um Passo da História definida pelo utilizador. Cada Passo possui um Objeto ou Personagem com quem o jogador deve interagir, a Animação que deve ser reproduzida após a conclusão, um diálogo a ser apresentado e, possivelmente, um objeto para dar ao utilizador;
- *Storyboard* - coleção de todos os Passos da História criada pelo utilizador. Responsável por converter o diagrama apresentado, em dados para serem interpretados pelo Interpretador da História. Processa as diferentes ligações entre Passos em dependências dos mesmos;
- *World* - local construído pelo utilizador e constituído pelos Objetos e Personagens por ele selecionados. Cenário onde se encontra o personagem e onde a História se vai desenrolar;
- *Storage* - Base de Dados onde são armazenadas as Histórias e Cenários criados pelos diferentes utilizadores. Estes dados são depois transmitidos de volta para a aplicação para permitir que sejam editados ou jogados;
- *Play*:
  - *Game Engine* - Interpretador da História que está a ser jogada. Responsável pelo movimento e ações do jogador, interpretação da História e atualização do estado da mesma ao alterar o Inventário do jogador após a conclusão de cada Passo;
  - *Inventory* - Inventário do Jogador. Coleção que armazena os itens que o jogador obtém ao longo da História e que necessita de utilizar.

### 4.3 Criação da História

Quando contamos uma história em linguagem natural, independentemente do seu objetivo, não somos obrigados a seguir uma estrutura padronizada e bem definida para sermos bem sucedidos nesta tarefa. É possível até contar uma mesma história de maneiras completamente diferentes mas conseguindo transmitir sempre a mesma mensagem. Tal como explorado na secção 2.1.3, um diretor de cinema ou animador profissional, precisam de realizar algumas adaptações ao converter um guião ou texto que se pretende adaptar a um filme ou a um projeto de animação.

Foi necessário então, criar um *Storyboard*, que auxiliasse um utilizador sem experiência neste tipo de atividades a criar uma adaptação da sua história, que conseguisse ser processado por uma máquina. A abstração proposta nesta Dissertação permite ao utilizador contar uma história ao dissecar a mesma nos seus elementos mais básicos, os seus passos.

Para contar uma história o utilizador é apresentado com uma interface que lhe permite adicionar novos passos num Fluxograma. Estes passos são representados com assistência de um *asset* disponível na *Unity Asset Store* o *UI Node Connect 4*, que controla toda a parte gráfica de criação de ligações entre diferentes passos do Fluxograma. O *asset* em questão pode ser observado na Figura 4.3.

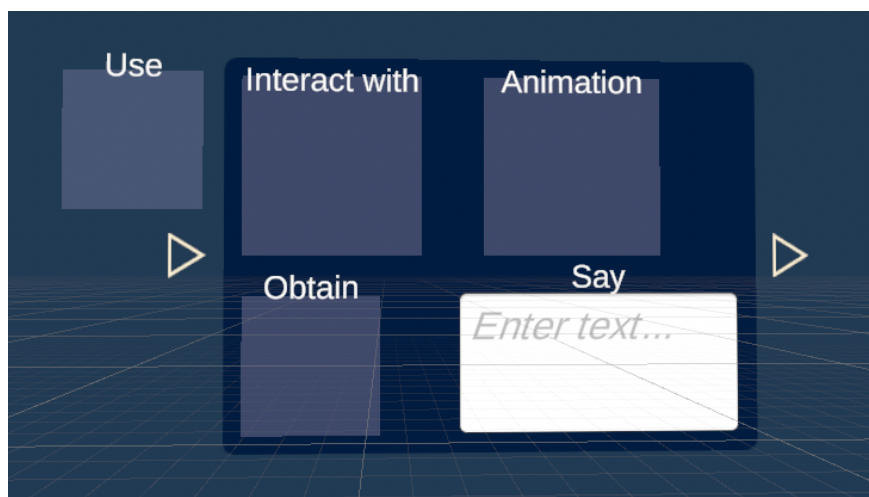


Figura 4.3: Modelo de um passo do *Storyboard*

Em cada passo, o utilizador é responsável por indicar com que elemento no mundo, um objeto ou personagem, pretende que o jogador interaja para desencadear esse passo, necessitando para isso de seleccionar e arrastar de uma lista pré-definida de objetos ou personagens um desses elementos para o passo pretendido. Ao premir o botão esquerdo do rato é efetuado um *Raycast* da posição da câmara ao local onde se encontra o ponteiro, determinado o objeto ou personagem atingido e seleccionado. Ao seleccionar este da lista é criada uma cópia dele que pode ser arrastada e, ao ser desseleccionada, é efetuado um novo *Raycast* da câmara ao local onde se encontrava a cópia que, caso se encontre sobre um passo do Fluxograma coloca a cópia nesse passo.

Os passos da história apresentam também uma cor identificadora para assinalar passos que possuem um objeto para dar ao utilizador. Para indicar que um objeto deve ser atribuído ao jogador após a conclusão do passo, o utilizador, tal como anteriormente, necessita de arrastar o objeto que pretenda para o local no passo para isso reservado. Esta funcionalidade tem um pormenor importante para permitir contar histórias interessantes. Se o objeto ou personagem com que se interage para desencadear o passo seja diferente do obtido, ou não haja nenhum, após a conclusão do passo este é obtido, ou não. No entanto, caso o objeto ou personagem com que se deve de interagir for igual ao obtido este desaparece da cena após a conclusão do passo e é adicionado ao inventário. Desta forma podemos apresentar os dois comportamentos expectáveis para obter objetos num jogo de aventuras, a interação com um objeto ou personagem para adquirir um item, ou a recolha deste diretamente do cenário.

Caso o utilizador pretenda que o jogador utilize um item, previamente recolhido, para

completar este passo, por exemplo uma chave que abra um cofre, deve também indicar esta restrição de uma forma semelhante ao descrito anteriormente para o local devidamente indicado no passo.

O utilizador pode também indicar a reação do personagem principal ao concluir com sucesso cada passo, ao selecionar e arrastar uma das animações disponibilizadas para o mesmo. Caso o objeto com que o jogador deve interagir num determinado passo seja um humanoide, uma animação pode também ser atribuída a esse objeto diretamente no passo e, no momento da realização com sucesso do passo em que é interagido, o humanoide irá sair da sua animação de repouso e ativar a animação que lhe foi atribuída. O processo de atribuição da animação ao passo é exatamente igual ao explicado anteriormente para os outros objetos mas, para a atribuição aos personagens é verificado se o alvo do *Raycast* é uma personagem e, caso seja, é adicionado o identificador da animação selecionada ao personagem.

Cada passo possui ainda uma pequena caixa de texto no qual pode ser introduzido um diálogo que é apresentado ao jogador após a conclusão com sucesso do passo.

Finalmente, as ligações entre os passos devem ser efetuadas, selecionando e arrastando a saída do passo para a entrada do que se pretende anteceder. O *asset* mencionado anteriormente, *UI Node Connect 4*, é responsável por implementar a parte visual deste comportamento mas o processamento desta informação, posteriormente, é descrito na secção seguinte. Ao selecionar uma ligação existente é possível alterar o seu tipo entre uma ligação obrigatória ou opcional. A interface apresentada ao utilizador para a criação da sua história é apresentada na Figura 4.4.

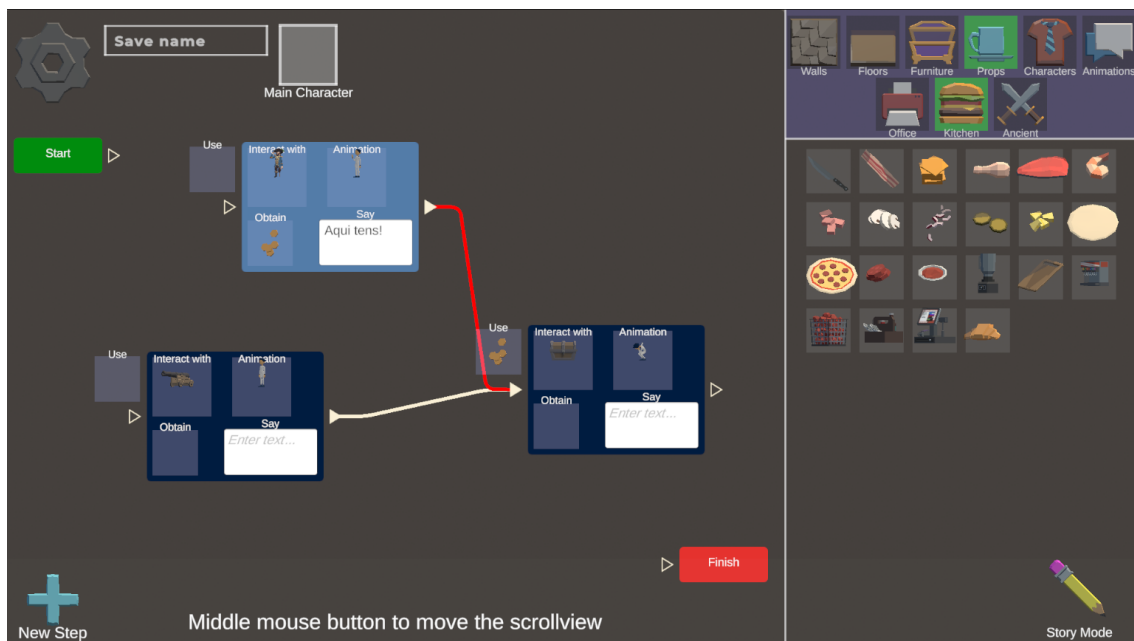


Figura 4.4: Edição de passos de uma História

No canto inferior esquerdo é possível observar o botão que permite adicionar novos

passos. No centro da imagem, os três passos que constituem a história do exemplo apresentado, com as respectivas animações, objetos com que o jogador deve interagir, diálogo a ser apresentado, o objeto adquirido e as dependências, entre os diferentes passos da história. A dependência obrigatória encontra-se sinalizada pela cor vermelha. Do lado direito da figura, é possível observar o menu com alguns dos diferentes objetos que o utilizador pode selecionar para construir a sua história.

### 4.4 Conversão de Fluxograma-Código

A conversão do Fluxograma é a etapa responsável por transformar o Fluxograma criado pelo utilizador para uma estrutura em código que pode ser processada pelo Interpretador da História e vice-versa.

Quando o utilizador pretende guardar a sua história o sistema itera todos os passos do Fluxograma e cria uma nova entrada para cada um. Na primeira iteração, cada passo tem um identificador único que é determinado pela ordem pela qual é processado, o nome do objeto ou personagem que o utilizador arrastou para com que o jogador interaja, o nome do objeto recebido após a conclusão e o nome do objeto necessário para o passo ser efetuado, caso existam, a animação, selecionada pelo utilizador, que deve ser reproduzida no personagem principal após a execução com sucesso, assim como a animação do personagem com que se interage, caso seja um personagem e as coordenadas globais do passo no Fluxograma para permitir reconstruir o mesmo quando o utilizador o pretenda na posição correta.

De seguida, os passos são novamente iterados e, para cada ligação que sai do passo, é registado o par de valores com o identificador dos passos envolvidos nessa ligação e uma variável *boolean* se é obrigatória ou opcional. Este par de valores representa a chave na estratégia de *Lock-and-Key* e são obtidos após a conclusão com sucesso do passo. Nesta iteração são também consultadas as ligações incidentes na porta de entrada do passo, as suas dependências. Para cada entrada é, igualmente, registado o par de valores com o identificador dos passos envolvidos nessa ligação e uma variável *boolean* se é obrigatória ou opcional. Este valor representa a fechadura na estratégia de *Lock-and-Key* e são as restrições necessárias para o passo ser efetuado.

Esta estrutura em código, que já pode ser processada pelo Interpretador da História (secção 4.5) ao longo do jogo, é guardada com o nível e enviada para a Base de Dados na forma de um *JSON* onde todas as histórias guardadas estão disponíveis para serem carregadas, editadas ou jogadas.

```
"Storyboard": {
  "type": "array",
  "description": "The current Storyboard",
  "items": {
    "type": "object",
    "properties": {
      "Id": {
        "type": "string"
      },
      "ColliderName": {
        "type": "string"
      },
      "Requirements": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "_itemName": {
              "type": "string"
            },
            "_amount": {
              "type": "integer"
            }
          }
        }
      },
      "Acquired": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "_itemName": {
              "type": "string"
            },
            "_amount": {
              "type": "integer"
            }
          }
        }
      },
      "ItemDependentSteps": {
        "type": "array",
        "items": {
          "type": "integer"
        }
      },
      "StepCoordinates": {
        "type": "array",
        "items": {
          "type": "number"
        }
      }
    }
  }
}
```

Figura 4.5: *Schema* do JSON para representar os dados do *Storyboard*

Na Figura 4.5 é apresentado o *schema* dos JSONs criados da adaptação dos vários Fluxogramas para serem armazenados na Base de Dados e recolhidos para serem interpretados pelo Interpretador da História.

Caso o utilizador pretenda consultar uma história existente que carregou da Base de Dados, esta precisa de ser apresentada novamente na forma de Fluxograma para permitir a sua fácil leitura e edição. Para tal, o sistema precisa de iterar cada um dos passos da história carregada e criar um novo passo do Fluxograma. Este passo é então inicializado com o seu identificador único, nas coordenadas associadas. É encontrado o objeto ou personagem pelo nome na lista de objetos pré-definidos e associado ao passo e procede-se da mesma forma para atribuir as animações corretas, diálogo e, nos passos que também tenham objetos obtidos ou como requisito, o mesmo ocorre para estes.

Depois de todos os passos terem sido gerados, é consultada a lista de requisitos dos respetivos passos da história e efetuadas as ligações da porta de entrada deste passo à

porta de saída dos passos cujos requisitos são o seu identificador único. Depois de todos os passos da história terem sido iterados, o Fluxograma encontra-se repostado ao estado em que o utilizador o observou pela última vez, pronto para ser consultado ou editado.

## 4.5 Motor de Jogo

A história e cenários previamente gerados têm como objetivo definir o ponto de partida e os acontecimentos que deverão ser desencadeados quando o jogador reproduzir a história. É a função do Motor de Jogo garantir que a ordem destes é cumprida e permitir ao jogador jogar o produto final. Este é constituído por duas partes principais, o Controlador do Jogador e o Interpretador da História.

### 4.5.1 Controlador do Jogador

O Controlador do Jogador encontra-se encarregue pelo movimento, animações do personagem principal, comunicação com o jogador e de informar o Interpretador da História do objeto com que está a interagir para este poder processar o pedido.

Para controlar todas as ações do personagem o jogador necessita apenas dos botões esquerdo e direito do seu rato. Sempre que pressiona o botão direito do rato é efetuado um *Raycast* da posição da câmara ao local onde se encontra o ponteiro e determinado o objeto atingido. É inquirido com o Interpretador da História se existe algum passo da história cujo objeto desencadeador seja o em questão e, caso a resposta seja falso, o personagem irá andar, a uma velocidade constante e com uma animação própria, em linha reta, até ao local em que o ponteiro se encontrava, onde irá parar e regressar à animação base inativa. Caso a resposta seja que de facto o objeto atingido pertence a um passo da história, é calculado se o jogador se encontra perto suficiente para poder interagir com o objeto em questão ou seja, a uma distância vetorial inferior a 3 unidades no sistema de coordenadas do *Unity*. Caso se encontre demasiado longe, irá dirigir-se em direção ao objeto tal como anteriormente mas, se o objeto se encontrar no seu raio de alcance, irá ser consultado com o Interpretador como prosseguir.

O Interpretador devolve o código da animação apropriada para o personagem principal apresentar, de acordo com o passo poder ser executado ou não. Para utilizar um objeto do inventário num objeto ou personagem que está no mundo, o jogador necessita de selecionar o objeto do seu inventário, com o botão esquerdo do rato, e arrastá-lo até ao objeto pretendido para completar a interação que irá ser processada também pelo Interpretador mas agora com informação adicional do objeto que está a ser utilizado para completar o passo.



Figura 4.6: Exemplo da reprodução de uma História

Na Figura 4.6 é apresentada a conclusão de um passo de uma história criada com o *Storyboard*, não é possível representar na imagem, mas o personagem principal e o personagem com que se interagiu reproduzem as suas respetivas animações e é apresentado o diálogo que o utilizador definiu para este passo.

#### 4.5.2 Interpretador da História

O Interpretador, é responsável por manter um conceito de inventário, no qual estão guardados os objetos adquiridos pelo jogador ao longo da sua sessão. É também encarregado de consultar a história do jogo selecionado sempre que o jogador tenta interagir com um objeto ou personagem, verificar se este representa um passo da história e, se o jogador apresentar os requisitos necessários para o realizar, completar o passo, atualizando o estado do inventário do jogador.

Como foi mencionado na secção anterior, sempre que o jogador realiza uma ação na qual o destinatário é um objeto ou personagem do cenário é necessário inquirir com o Interpretador se existe algum passo da história na qual este objeto está envolvido. A este pedido, o Interpretador simplesmente tem de iterar os diferentes passos da história até encontrar um cujo objeto ou personagem seja o mesmo que o enviado pelo Controlador do Jogador ou ficar sem passos para iterar.

A principal tarefa do Interpretador é, no entanto, processar os diferentes pedidos do jogador quando, de facto, se trata de uma tentativa de progredir a história. Estes pedidos, recebidos do Controlador, indicam o objeto ou personagem com que o jogador está a interagir e, caso este tenha selecionado e arrastado um objeto do seu inventário para completar o passo, o nome do objeto fornecido pelo jogador. Em primeiro lugar, o Interpretador consulta as dependências que o jogador necessita para poder completar este passo, estas tanto podem ser objetos recolhidos pelo jogador e presentes no seu inventário visível que este necessita de utilizar para completar passos ou simplesmente objetos conceptuais para respeitar a estratégia de *Lock-and-Key*. Depois, verifica se existe

no inventário do jogador todos os objetos necessários das dependências obrigatórias ou, se não existirem dependências obrigatórias neste passo, pelo menos uma das dependências opcionais que, ao confirmar-se, significa que o passo pode ser efetuado. A única exceção a esta regra acontece se o passo necessitar de um objeto do inventário do jogador recolhido por este que tem em sua posse mas não selecionou e arrastou para completar o passo.

Quando é confirmado que o passo pode ser concluído, o Interpretador efetua o processo de atualizar os objetos no inventário, retirando primeiro os objetos que eram requisitos deste passo e depois adicionando os novos objetos adquiridos da conclusão do mesmo. O Interpretador verifica também se o objeto interagido é o mesmo que o obtido e, caso o seja, remove este, no cenário. As últimas etapas da conclusão com sucesso de um passo da história são devolver ao Controlador o código da animação associado ao passo para este poder reproduzir a animação no personagem principal, caso o passo envolva a interação com um personagem, a animação que esta deve apresentar e, ao mesmo tempo, o diálogo a ser apresentado ao jogador.

Caso os requisitos para a conclusão do passo não estejam cumpridos, o Interpretador procura, se existir, o próximo passo que tenha também este objeto ou personagem como alvo. O processo mencionado anteriormente é repetido até não existirem mais passos com este objeto ou personagem na história ou assim que um deles tenha sido concluído com sucesso. É necessário efetuar este processo caso o jogador precise de interagir mais do que uma vez com o mesmo objeto ou personagem mas em partes diferentes da história. Quando não existirem mais passos com este objeto ou personagem para iterar e nenhum tiver sido realizado com sucesso então o Interpretador indica esta informação ao Controlador ao devolver como animação o código da animação pré-definida para demonstrar que não foi possível completar o passo.

## AVALIAÇÃO EXPERIMENTAL

### 5.1 Avaliação da Versatilidade do Sistema

Para colocar à prova as capacidades práticas do modelo de Storyboard desenvolvido, foram reproduzidas algumas histórias, ou excertos das mesmas, e avaliados os resultados obtidos.

Foram definidos três tipos diferentes de histórias que deviam tentar ser reproduzidos para estes testes. Histórias de Jogos de Aventura, *Hero's Journey* e Histórias Instrutivas. Para cada um destes, foi escolhida a história mais conhecida que melhor se enquadrasse com os cenários que era possível serem gerados de acordo com os diferentes objetos e personagens que tínhamos em posse para o desenvolvimento do projeto. Nenhum dos critérios para as histórias selecionadas foi a facilidade de adaptação para o *Storyboard*, o objetivo destes testes era exatamente tentar encontrar quaisquer dificuldades que surgissem na adaptação das diferentes histórias.

O cenário, na qual as diferentes histórias foram reproduzidas, apresentou algumas limitações para este exercício visto encontrar-se limitado a uma tabela de 10x10 em que cada objeto ou personagem ocupava uma célula da mesma. Caso o utilizador necessite de mais espaço para contar a sua história, poderá dividir a mesma em diferentes partes e criar um cenário e *Storyboard* para cada uma dessas, tal como é explorado na segunda história testada 5.1.2. Apesar destas restrições, os resultados são ilustrativos das possibilidades do *Storyboard*.

#### 5.1.1 Jogo de Aventura - The Secret of Monkey Island

Considerando o resultado desta dissertação, em que qualquer história que o utilizador crie será sempre apresentado no estilo de um Jogo de Aventura, o primeiro teste tentou recriar o conhecido jogo "The Secret of Monkey Island". Este jogo, publicado em 1990 pela *Lucasfilm Games*, foi um sucesso internacional e segue as aventuras de Guybrush nas Caraíbas que aspira a tornar-se pirata.

A adaptação desta história para a estrutura desenvolvida foi extremamente simples de efetuar visto que, como seria de esperar, o exercício de adaptar a estrutura de um Jogo de

Aventuras, segmentado por passos que o jogador tem que completar, para uma estrutura de dependências em que cada uma destas é um passo da história é bastante evidente.

O exercício foi, no entanto, interrompido pela limitação espacial do cenário que tinha de ser apresentado ao jogador. Este problema pode ser ultrapassado com a divisão da história em sub-partes, tal como é explorado no próximo exemplo da História 5.1.2. Na adaptação efetuada, o *Storyboard* permite ao jogador realizar os passos dos primeiros atos da história. Deve falar com o pirata de vigia, roubar um balde e carne da cozinha, ser ridicularizado pelos líderes dos piratas, voltar a falar com o pirata de vigia, falar com os irmãos Fettucini, ajudá-los a experimentar o seu lançamento de canhão ao equipar o balde que recolheu como um capacete e a receber o seu pagamento pelos serviços prestados. O *Storyboard* e jogo criados para este teste podem ser observados na Figura 5.1.

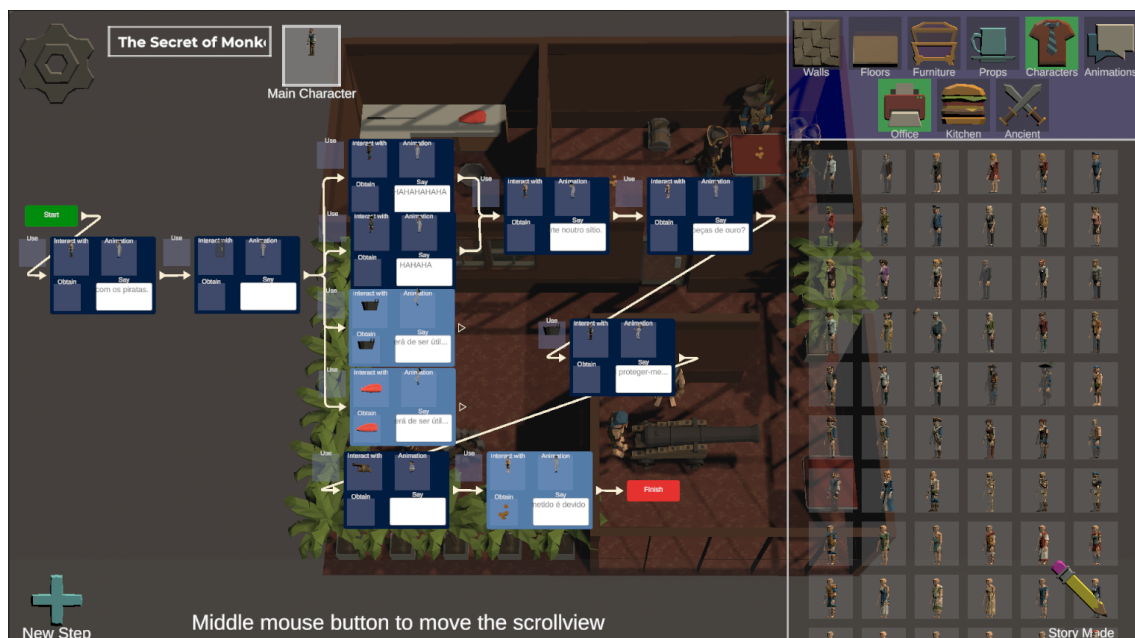


Figura 5.1: *Storyboard* e Cenário da história 1

Uma das principais características do jogo "The Secret of Monkey Island", que o fizeram ter tanto sucesso quando foi inicialmente publicado, era o seu humor espirituoso nos diálogos dos personagens. Com uma utilização inteligente do *Storyboard* desenvolvido é possível replicar esta característica, realizando vários passos da mesma interação com o personagem cada uma com um diálogos original diferente. Por motivos ilustrativos, devido à complexidade que isso adicionaria à Figura 5.1 este exercício focou-se na tentativa de replicação do enredo da história e não nos seus pormenores.

### 5.1.2 Hero's Journey - Hansel e Gretel

O segundo tipo de história que se pretendeu adaptar para o *Storyboard* foi uma história tradicional que seguisse o padrão do *Hero's Journey* ou Monomito. A história escolhida

para tal foi o conto popular "Hansel e Gretel", compilado e publicado pela primeira vez pelos irmãos Grimm em 1812.

História	Etapas do Monomito
Hansel e Gretel	1; 2; 3; 4; 5; 8; 9; 10; 11; 13; 14; 15; 17

Tabela 5.1: Monomito em Hansel e Gretel

Esta história precisou de uma pequena adaptação para o *Storyboard*. No conto popular, ambos os irmãos desempenham um papel de personagem principal, na primeira metade Hansel e na segunda Gretel. O *Storyboard* foca-se sempre numa só personagem principal. Por esse motivo, foi necessário dividir a história em duas partes. A primeira focada em Hansel e a segunda em Gretel.

O resultado obtido permite ao jogador contar e jogar a história do início ao fim com todos os passos. Na primeira história pode agarrar pedras para as largar pelo caminho, de forma a encontrar o caminho de regresso a casa, depois, ao agarrar num pouco de pão pode largar as migalhas que faz com que se percam e encontrem a casa da bruxa, ficando depois preso na jaula dentro da casa desta. Na segunda parte da história, agora como Gretel, cozinha pratos para engordar o irmão, até surgir a oportunidade de empurrar a bruxa para dentro do fogão quando esta os pretendia comer, libertar o irmão da sua jaula e regressar a casa com as riquezas da bruxa para o pai. Os *Storyboards* e jogos criados para este teste podem ser observados nas Figuras 5.2 e 5.3.

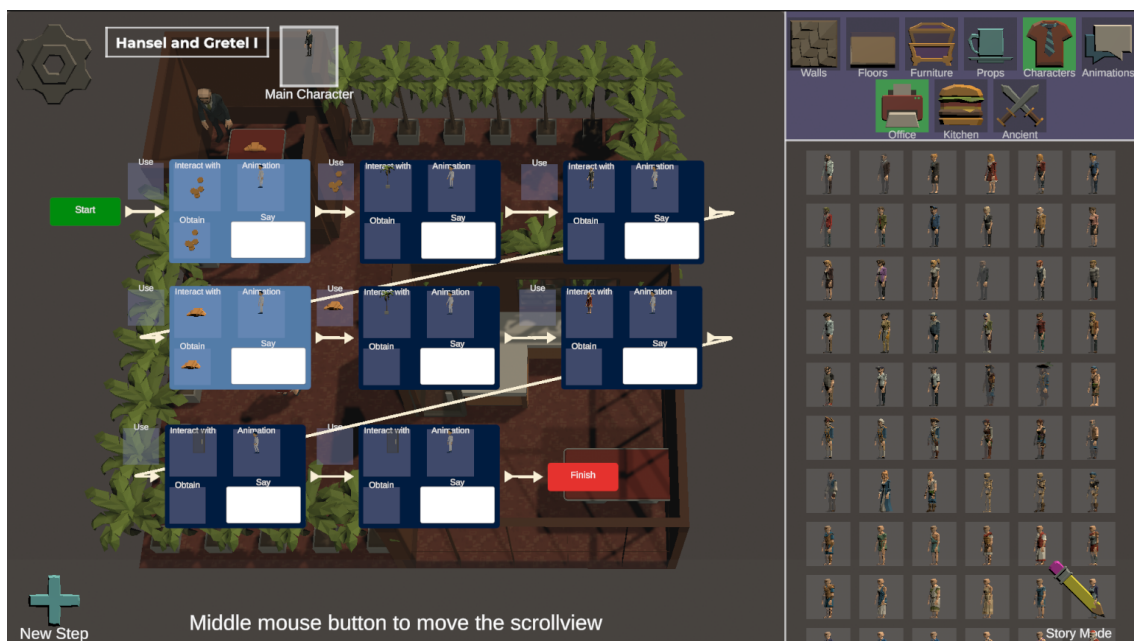


Figura 5.2: *Storyboard* e Cenário da primeira parte da história 2

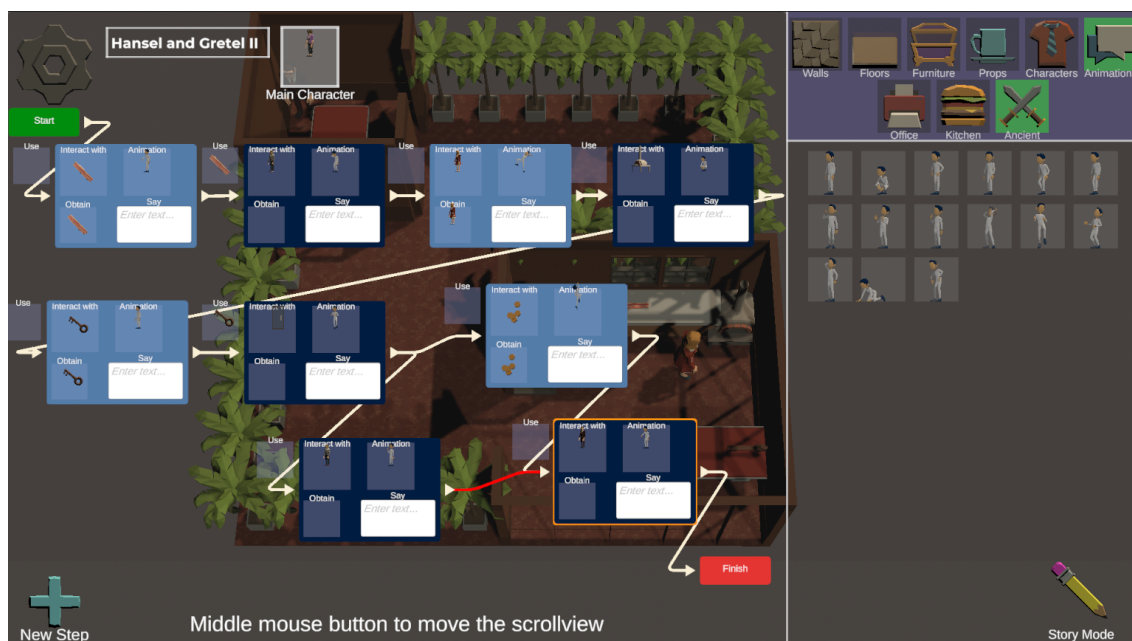


Figura 5.3: *Storyboard* e Cenário da segunda parte da história 2

A experiência resultante para o jogador deste teste reflete, de uma forma fiel, os acontecimentos da história original dividida nas duas partes. O jogo criado permite transmitir a essência da história e, caso seja utilizado como um meio, para auxiliar a transmissão da história de forma oral, permite uma experiência mais cativante para os utilizadores.

### 5.1.3 Histórias Instrutivas - Cozinhar uma Pizza

O último tipo de história que se tentou criar para testar as capacidades do *Storyboard* foi uma história instrutiva, uma história em que se ensinasse quem jogasse a realizar alguma tarefa. O exemplo escolhido para este tipo de histórias consistiu em criar uma história que instrua o jogador em como cozinhar uma Pizza.

Para a criação do *Storyboard*, foi necessário fazer o trabalho semelhante ao que um cozinheiro teria de fazer se quisesse escrever a sua receita num livro de receitas. Contar, passo a passo, o que um cozinheiro deve fazer para conseguir replicar a sua pizza. No *Storyboard*, o utilizador pode ainda colocar na secção de diálogo, qualquer comentário pertinente a alguma técnica ou porções que tenham de ser cumpridas na sua receita e o resultado final é apresentado na forma de uma experiência interativa. No caso concreto da história criada, o jogador encontra-se numa cozinha com todos os ingredientes que necessita para fazer a receita. Para realizar a história deve primeiro lavar as suas mãos, colocar o molho de tomate e depois o queijo na base da pizza, cortar o bacon com a faca de cozinha, colocar este na pizza e depois colocar esta no forno. O *Storyboard* e jogo criados para este teste podem ser observados na Figura 5.4



### 5.2.1 Metodologia

Para a realização de cada teste, foi necessária a presença de dois elementos, o investigador, a supervisionar a realização dos testes, e o voluntário que seria apresentado com uma série de tarefas que deveria realizar. As sessões decorreram presencialmente ou online, com o auxílio de uma plataforma de videoconferência para permitir a comunicação entre as duas partes e para o voluntário poder partilhar o seu ecrã com o investigador.

É apresentada ao voluntário uma curta explicação do objetivo do sistema que vai testar e um formulário de consentimento que indica a finalidade dos testes. O voluntário pode então consentir para iniciar a realização dos mesmos. Após esta etapa, o investigador partilha com o voluntário a aplicação e este é instruído para iniciar a mesma, de forma a permitir o início dos testes.

Para o efeito desta dissertação foram realizados dois grupos de tarefas para testar as duas principais funcionalidades a avaliar, a criação de uma história e permitir ao utilizador jogá-la. Cada um destes grupos encontra-se dividido em pequenas tarefas que o voluntário deve realizar de forma sequencial para alcançar o objetivo de cada grupo. Em cada tarefa, o investigador observa o comportamento do voluntário e preenche o formulário com pequenas observações relativas à conclusão de cada tarefa ou perguntas que efetua ao voluntário. No fim de cada grupo de tarefas, o voluntário é também inquirido sobre diferentes perguntas sobre este.

Todos os dados recolhidos são efetuados manualmente, pelo investigador, com alguns destes sendo dados objetivos, como o tempo de conclusão de cada tarefa e outros subjetivos como o grau de intuitividade do sistema, satisfação com os resultados obtidos e percepção do funcionamento do sistema. Todos os dados obtidos após a conclusão dos testes, são utilizados para retirar conclusões sobre as alterações que devem ser efetuadas ao sistema e para poderem ser apresentadas conclusões da sua utilização prática.

### 5.2.2 Caracterização dos Voluntários

Foi registado o contributo de 14 voluntários para a realização dos diferentes testes. Deste grupo, 11 dos participantes pertenciam ao sexo masculino e 3 ao sexo feminino, compreendidos por idades entre os 19 e os 65 com uma mediana de 24 anos. Apenas 2 dos voluntários relataram nunca jogarem jogos e, dos restantes 12, 11 mencionaram preferência por jogar num computador. Os dados de caracterização do grupo pode ser encontrado em maior detalhe nos gráficos presentes na Figura 5.5.

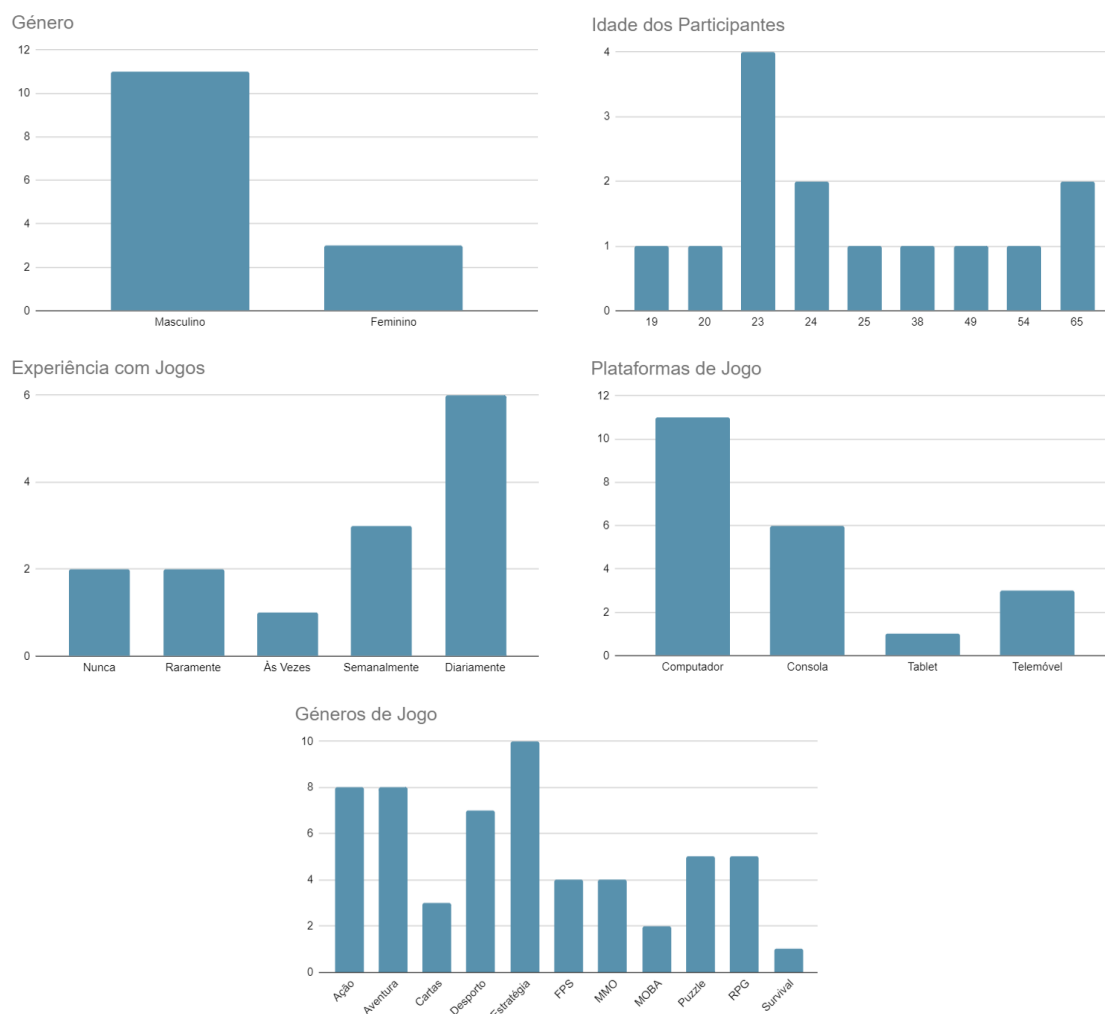


Figura 5.5: Gráficos de Caracterização dos Voluntários

### 5.2.3 Testes e Resultados

Para a experiência completa da utilização do sistema são necessárias três etapas, a criação do cenário de jogo por parte do utilizador, a definição da história do jogo e o utilizador jogar a sua criação. Esta primeira etapa, não diz respeito a esta dissertação, pelo que não vão ser aprofundados os diferentes resultados nem retiradas conclusões da mesma. No entanto, a sua realização, familiarizou o voluntário com alguns comportamentos universais do funcionamento do sistema e, por isso, estes, devem ser apresentados para se permitir obter conclusões fidedignas.

#### 5.2.3.1 Grupo A - Criação do Cenário

O primeiro grupo de testes foca-se na navegação do voluntário pelo sistema e criação do cenário para a história que pretendemos que conte. Será pedido ao utilizador que efetue uma série de ajustes a um cenário para criar um restaurante que permite contar a história.

• **Tarefa A1 - Selecionar a história "Test" e entrar em modo de Criação**

Nesta tarefa o utilizador é instruído para selecionar uma história existente do repositório de histórias e para entrar em modo de Criação. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o utilizador entenda como navegar no menu principal e preparar o sistema para o próximo passo. É também perguntado ao voluntário o seu grau de satisfação com o menu principal e a circulação pelo sistema.

- **A1Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **A1Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **A1Q3 - Avaliação do Menu Principal** Opinião do voluntário sobre o Menu Principal, as opções são "Muito Mau", "Mau", "Normal", "Bom" ou "Muito Bom".
- **A1Q4 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 5 deles de forma independente e 9 com intervenção do investigador. Os utilizadores demoraram entre 15 segundos a 2 minutos e 12 segundos a realizar esta tarefa, com uma média de 1 minuto. Apenas um dos utilizadores ficou insatisfeito com o Menu Principal. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.6.

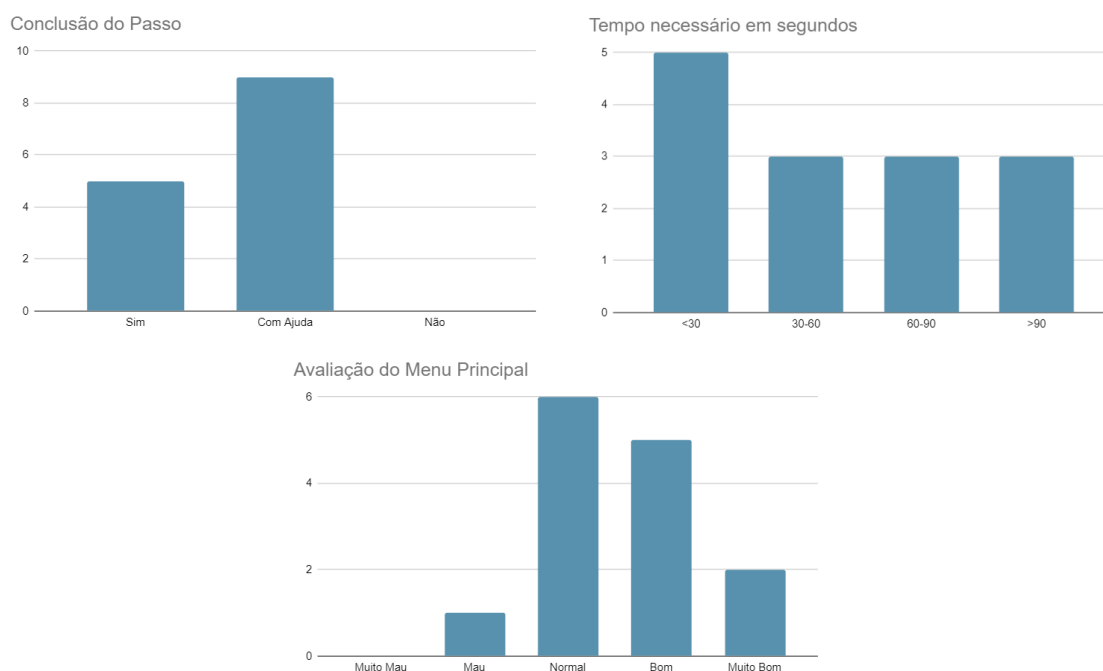


Figura 5.6: Gráficos dos resultados da Tarefa A1

- **Tarefa A2 a A4 - Testes de funcionalidades de edição do cenário sem influência nesta Dissertação**
  
- **Tarefa A5 - Navegar entre as diferentes opções do modo de Criação**

Nesta tarefa o utilizador é instruído a alterar do modo de "Edição de Cenário" para o modo de "Edição de História" de seguida, para "Testagem da História" e, finalmente, de volta para "Edição de Cenário". Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga identificar os diferentes modos da criação de um jogo e como os alcançar. É questionado o utilizador se considera os diferentes modos estarem bem identificados.

- **A5Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
  
- **A5Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
  
- **A5Q3 - Quais dos modos são facilmente identificáveis** Opinião do voluntário sobre os diferentes ícones para identificação dos modos. As opções de resposta são "Edição de Cenário", "Edição de História" ou "Testagem da História". O voluntário pode selecionar nenhuma ou mais do que uma resposta.
  
- **A5Q4 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 6 deles de forma independente e 8 com intervenção do investigador. Os utilizadores demoraram entre 17 segundos a 2 minutos e 5 segundos a realizar esta tarefa, com uma média de 49 segundos. De todos os utilizadores, apenas um pequeno grupo, conseguiu identificar, facilmente, os diferentes modos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.7.

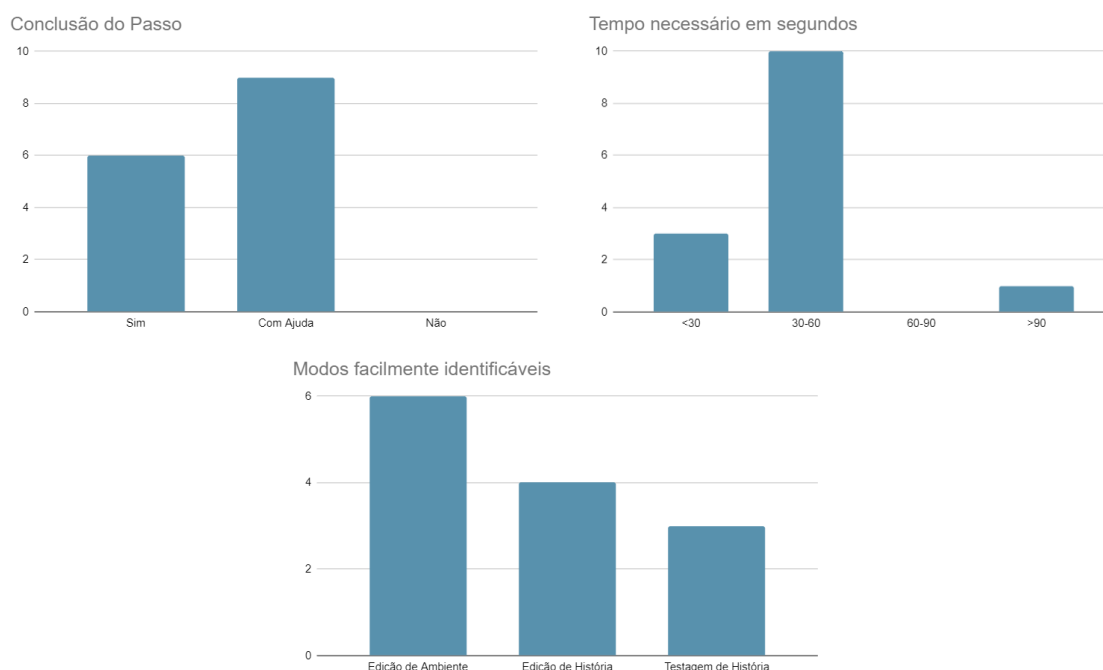


Figura 5.7: Gráficos dos resultados da Tarefa A5

### 5.2.3.2 Grupo B - Criação da História

O segundo grupo de testes realizados tem como ponto de partida o cenário desenvolvido nos passos anteriores e pretende instruir o voluntário em como utilizar o sistema de forma a construir uma pequena história para depois a poder jogar. A história que pretendemos que conte contém apenas 3 passos e permite fazer uma pizza. A história que se pretende que o utilizador replique é uma simplificação da que pode ser observada no Capítulo 5.1.3.

- **Tarefa B1 - Mudar para o modo de "Edição de História"**

Nesta tarefa o utilizador é instruído, tal como anteriormente, para alterar do modo de "Edição de Cenário" para o modo de "Edição de História". Nenhuma informação adicional é indicada e o objetivo desta tarefa é o utilizador preparar o sistema para o próximo passo.

- **B1Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B1Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B1Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada de forma independente. Este resultado era expectável devido a ser uma repetição de uma tarefa já realizada anteriormente, mas necessária para continuar o procedimento.

Os utilizadores demoraram entre 1 segundo a 11 segundos a realizar esta tarefa, com uma média de 4 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.8.

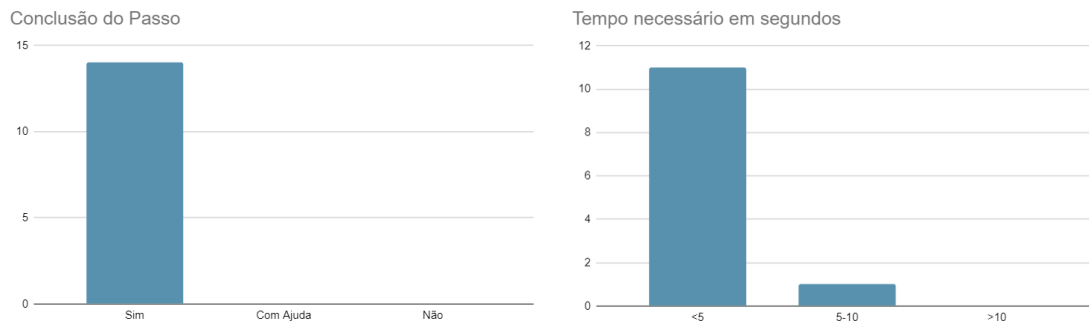


Figura 5.8: Gráficos dos resultados da Tarefa B1

- **Tarefa B2 - Adicionar três passos à história e movimentá-los**

Nesta tarefa é pedido ao utilizador que crie três passos para a história que pretendemos que crie. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga identificar o ícone responsável por criar novos passos e movimentar estes pelo ecrã.

- **B2Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B2Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B2Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 8 deles de forma independente e 6 com intervenção do investigador. Os utilizadores demoraram entre 3 segundos a 1 minuto e 51 segundos a realizar esta tarefa, com uma média de 36 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.9.

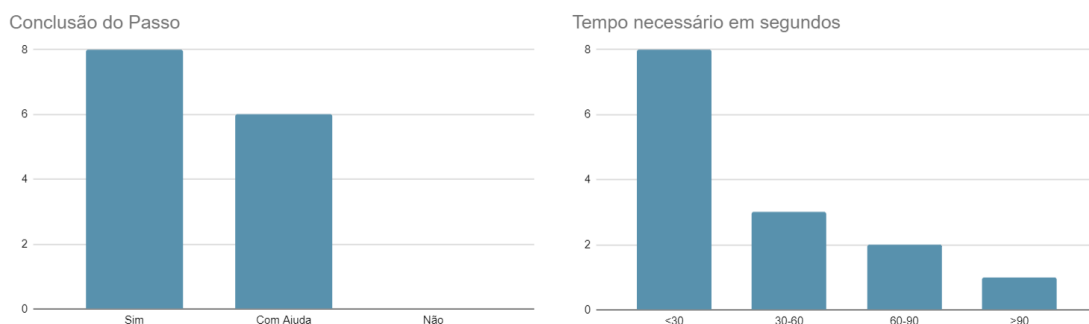


Figura 5.9: Gráficos dos resultados da Tarefa B2

- **Tarefa B3 - Adicionar a diferentes passos cogumelos, pizza e forno**

Nesta tarefa é pedido ao utilizador que adicione a cada um dos passos criados anteriormente os objetos mencionados, de forma a indicar com que objetos no cenário o jogador vai interagir em cada um dos passos. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga identificar os objetos da lista disponível e associar os mesmos aos passos da história criados.

- **B3Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B3Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B3Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 6 deles de forma independente e 8 com intervenção do investigador. Os utilizadores demoraram entre 12 segundos a 1 minuto e 25 segundos a realizar esta tarefa, com uma média de 45 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.10.

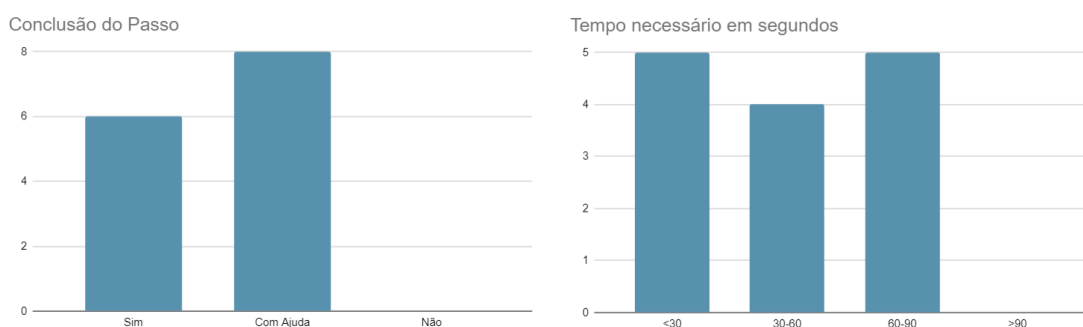


Figura 5.10: Gráficos dos resultados da Tarefa B3

- **Tarefa B4 - Alterar os passos dos cogumelos e pizza para serem eventos de recolha do objeto**

Nesta tarefa é pedido ao utilizador que indique ao *Storyboard* que estes dois eventos devem atribuir o objeto interagido ao jogador, no momento de realização dos testes esta era a abordagem que se utilizava. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga alterar os dois passos indicados para serem passos com recolha de objeto.

- **B4Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B4Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.

- **B4Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 9 deles de forma independente e 5 com intervenção do investigador. Os utilizadores demoraram entre 2 segundos a 1 minuto e 27 segundos a realizar esta tarefa, com uma média de 33 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.11.

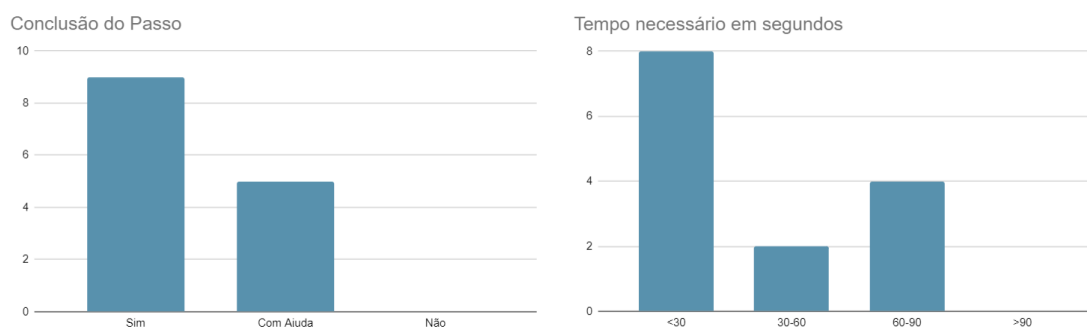


Figura 5.11: Gráficos dos resultados da Tarefa B4

- **Tarefa B5 - Adicionar a animação de dança a um dos passos**

Nesta tarefa é pedido ao utilizador que altere a animação associado a um passo à sua escolha para a de uma dança. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga navegar até ao separador das animações, selecionar a pretendida e trocar a animação pré-definida de um dos passos por esta.

- **B5Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B5Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B5Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 11 deles de forma independente e 3 com intervenção do investigador. Os utilizadores demoraram entre 10 segundos a 1 minuto e 22 segundos a realizar esta tarefa, com uma média de 28 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.12.

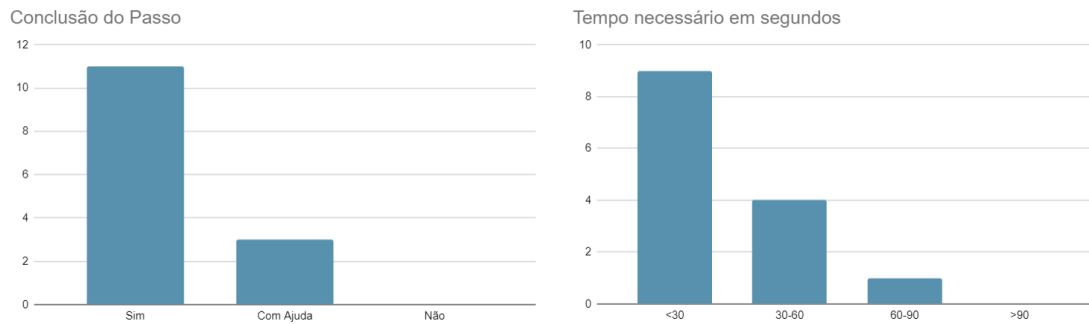


Figura 5.12: Gráficos dos resultados da Tarefa B5

- **Tarefa B6 - Criar ligação entre a saída do passo dos cogumelos para a entrada do passo da pizza**

Nesta tarefa é pedido ao utilizador que crie uma dependência na história entre o passo que apresenta os cogumelos como objeto que deve ser interagido com o passo em que a pizza deve ser interagida. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga especificar a ordem de realização de estes dois passos da história.

- **B6Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B6Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B6Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 9 deles de forma independente e 5 com intervenção do investigador. Os utilizadores demoraram entre 1 segundo a 1 minuto e 32 segundos a realizar esta tarefa, com uma média de 29 segundos. É importante realçar que alguns utilizadores já se encontravam familiarizados com esta característica do Storyboard pois tinham-se deparado com este comportamento enquanto tentavam realizar os passos anteriores, daí a variação entre tempos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.13.

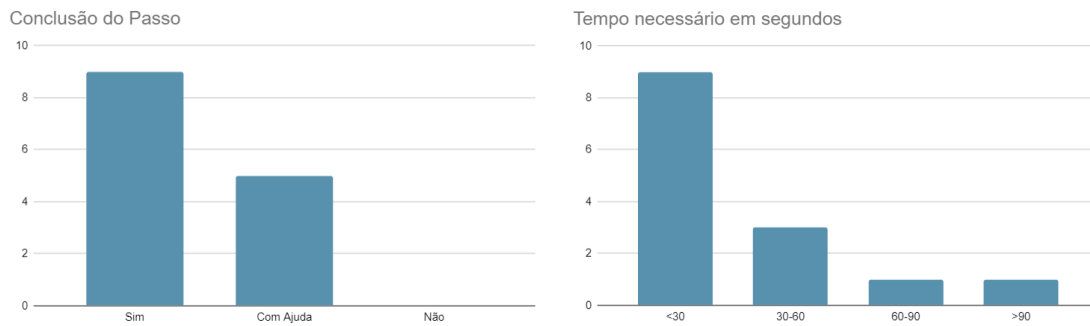


Figura 5.13: Gráficos dos resultados da Tarefa B6

- **Tarefa B7 - Criar ligação entre a saída do passo da pizza para a entrada do passo do forno**

Nesta tarefa é pedido ao utilizador que, tal como no passo anterior, crie uma dependência na história mas com o passo da pizza e do forno. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga especificar a ordem de realização de estes dois passos da história.

- **B7Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B7Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B7Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada de forma independente. Os utilizadores demoraram entre 1 segundo a 19 segundos a realizar esta tarefa, com uma média de 4 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.14.

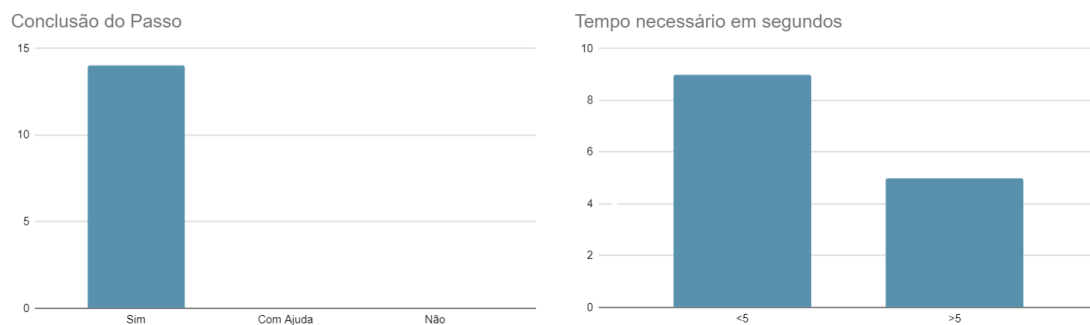


Figura 5.14: Gráficos dos resultados da Tarefa B7

- **Tarefa B8 - Guardar a História**

Nesta tarefa é pedido ao utilizador que guarde as alterações que efetuou no modo de Criação. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o voluntário consiga selecionar nas opções do modo de Criação a opção de guardar o seu progresso.

- **B8Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **B8Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **B8Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários, excepto 1, devido a um erro, entretanto corrigido, conseguiram concluir com sucesso a tarefa apresentada, 11 deles de forma independente e 2 com intervenção do investigador. Os utilizadores demoraram entre 4 segundos a 39 segundos a realizar esta tarefa, com uma média de 17 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.15.

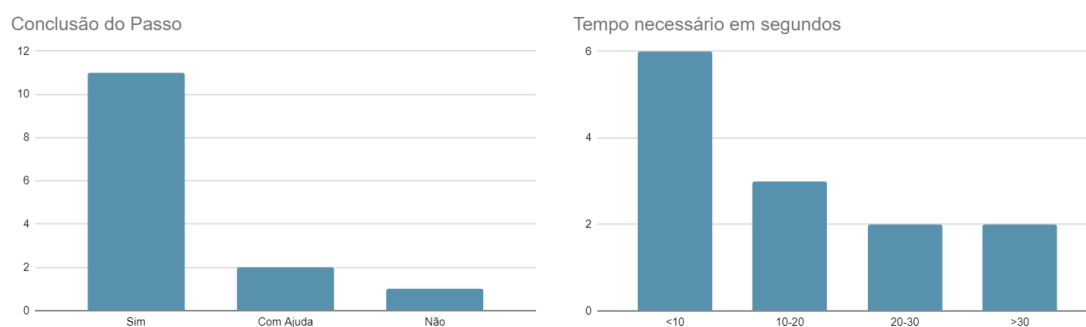


Figura 5.15: Gráficos dos resultados da Tarefa B8

### 5.2.3.3 Grupo C - Jogar a História

Para o último grupo de testes pretendemos que o utilizador jogue a história que desenvolveu nos outros dois grupos de tarefas anteriores. Para esta última etapa contamos apenas com 13 voluntários, devido a um problema que impossibilitou a participação de um destes.

- **Tarefa C1 - Selecionar a história "Test" e entrar em modo de Jogar**

Nesta tarefa o utilizador é instruído para selecionar uma história existente do repositório de histórias e para entrar em modo de Jogar. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o utilizador prepare o sistema para o próximo passo.

- **C1Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".

- **C1Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **C1Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 10 deles de forma independente e 3 com intervenção do investigador. Os utilizadores demoraram entre 5 segundos a 54 segundos a realizar esta tarefa, com uma média de 19 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.16.

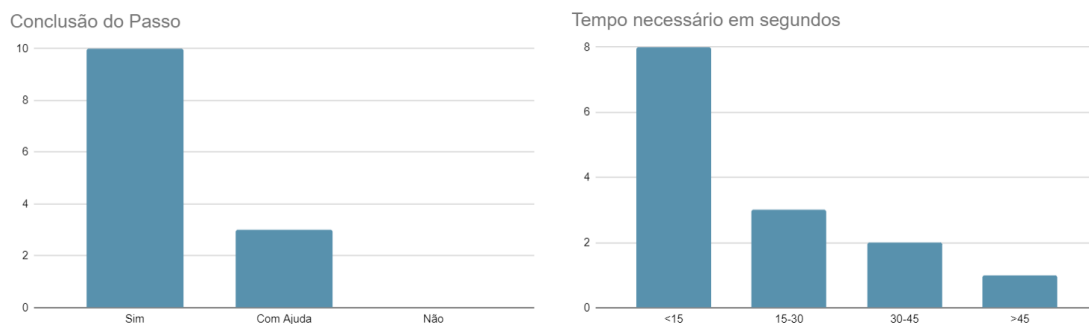


Figura 5.16: Gráficos dos resultados da Tarefa C1

- **Tarefa C2 - Movimentar o personagem pelo cenário**

Nesta tarefa o utilizador é instruído para movimentar o personagem pelo cenário do Jogo. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o utilizador consiga mover o seu personagem.

- **C2Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **C2Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **C2Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 11 deles de forma independente e 2 com intervenção do investigador. Os utilizadores demoraram entre 1 segundo a 36 segundos a realizar esta tarefa, com uma média de 8 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.17.

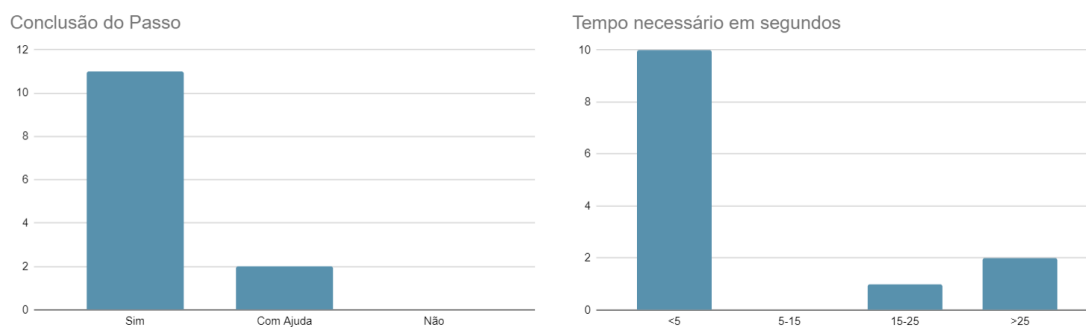


Figura 5.17: Gráficos dos resultados da Tarefa C2

• **Tarefa C3 - Recolher os cogumelos**

Nesta tarefa o utilizador é instruído para recolher os cogumelos que se encontram no cenário do Jogo. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o utilizador consiga mover o seu personagem até ao objeto pretendido e recolher este.

- **C3Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **C3Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **C3Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 12 deles de forma independente e 1 com intervenção do investigador. Os utilizadores demoraram entre 1 segundo a 80 segundos a realizar esta tarefa, com uma média de 13 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.18.

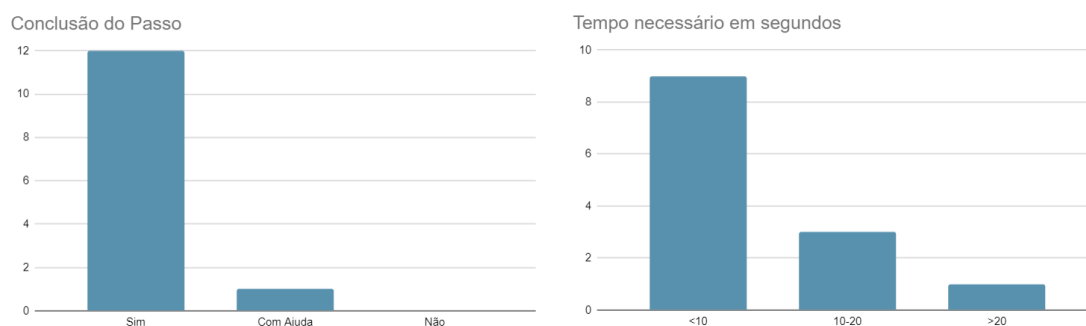


Figura 5.18: Gráficos dos resultados da Tarefa C3

• **Tarefa C4 - Tente Interagir com o Forno**

Nesta tarefa o utilizador é instruído para tentar interagir com o forno que se encontra no cenário do Jogo. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o utilizador tente interagir com o forno mas se aperceba pela reação do personagem que ainda não pode fazer isso. É questionado o utilizador se considera que a reação do personagem é apropriada para assinalar que não está a proceder pela ordem correta da história.

- **C4Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **C4Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **C4Q3 - A Reação do Personagem transmitiu-lhe que não estava a seguir a história pela ordem correta?** Resposta pode ser "Sim" ou "Não".
- **C4Q4 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada de forma independente. Os utilizadores demoraram entre 3 segundo a 15 segundos a realizar esta tarefa, com uma média de 7 segundos. Apenas um utilizador considerou que a reação do personagem não lhe transmitiu que estava a agir de forma incorreta. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.19.

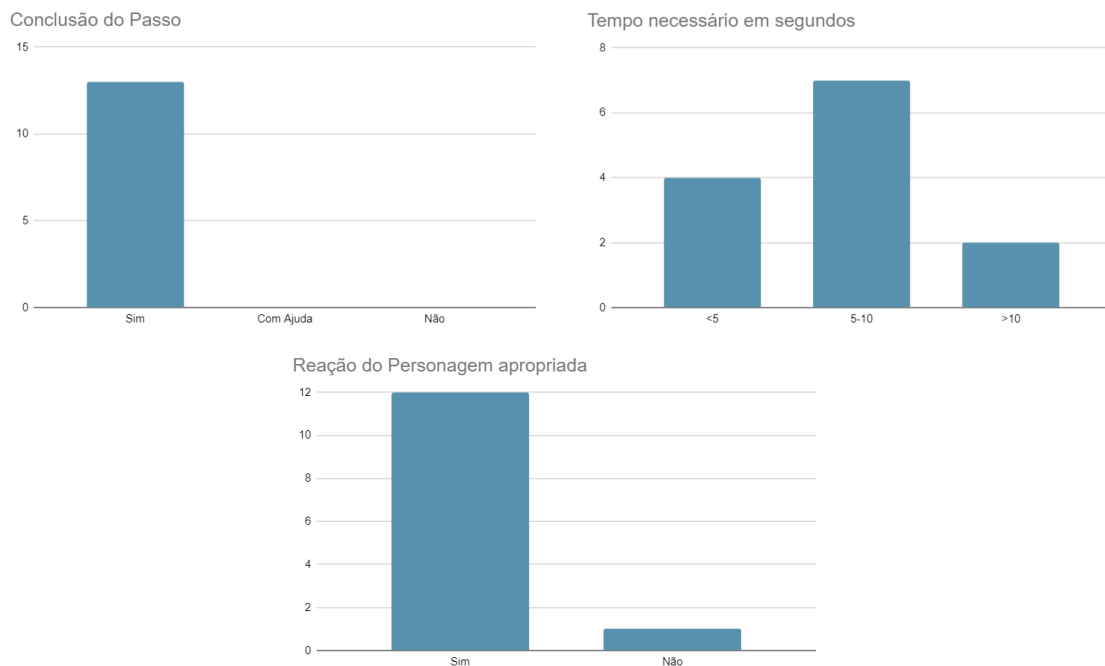


Figura 5.19: Gráficos dos resultados da Tarefa C4

- **Tarefa C5 - Colocar os cogumelos na pizza**

Nesta tarefa o utilizador é instruído para colocar os cogumelos na pizza que se encontra no cenário do Jogo. Nenhuma informação adicional é indicada e o objetivo desta tarefa é que o utilizador consiga utilizar os itens que tem em sua posse no objeto pretendido para completar esse passo. É questionado o utilizador se considera a utilização de itens no jogo um processo intuitivo ou não.

- **C5Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **C5Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **C5Q3 - A utilização de itens é intuitiva?** Resposta pode ser "Sim" ou "Não".
- **C5Q4 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 10 deles de forma independente e 3 com intervenção do investigador. Os utilizadores demoraram entre 5 segundos a 51 segundos a realizar esta tarefa, com uma média de 23 segundos. Dos voluntários, 2 consideraram a utilização de itens pouco intuitiva. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.20.

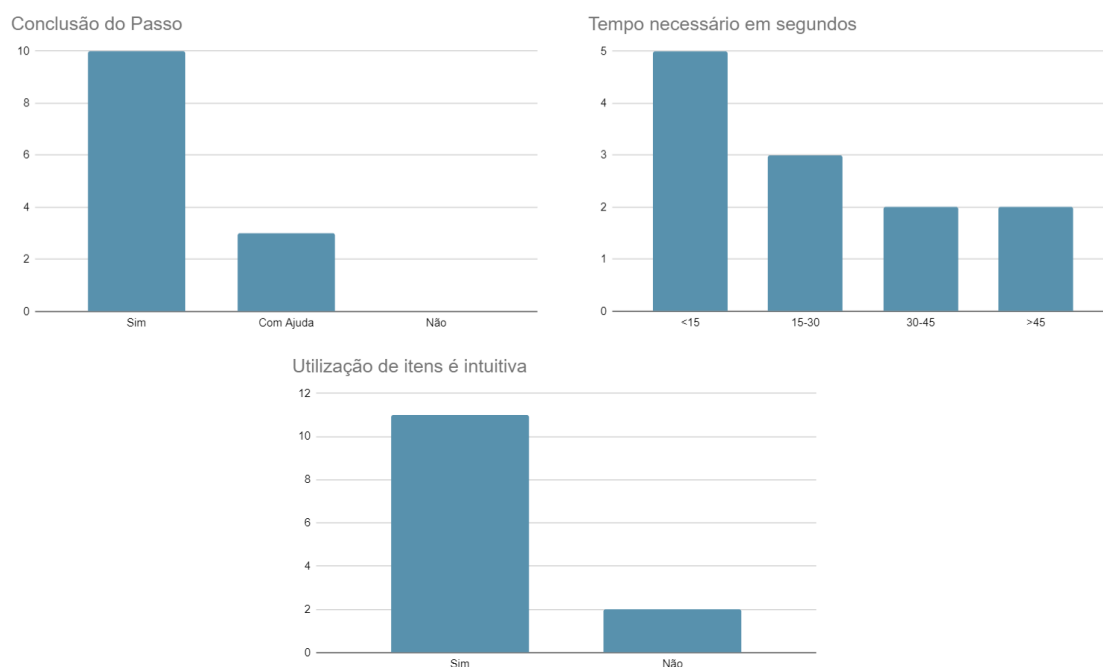


Figura 5.20: Gráficos dos resultados da Tarefa C5

- **Tarefa C6 - Colocar a pizza no forno**

Nesta tarefa o utilizador é instruído para colocar a pizza no forno que se encontra no cenário do Jogo. Nenhuma informação adicional é indicada e o objetivo desta tarefa

é que o utilizador, tal como anteriormente, utilize o item que tem em sua posse no objeto pretendido para completar o passo e concluir a história.

- **C6Q1 - Conseguiu Completar?** Resposta pode ser "Sim", "Com Ajuda" ou "Não".
- **C6Q2 - Tempo Necessário** Tempo, em segundos, que o voluntário demorou a completar a tarefa.
- **C6Q3 - Observações** O investigador pode colocar alguma observação feita por si ou pelo voluntário na realização da tarefa.

Todos os voluntários conseguiram concluir com sucesso a tarefa apresentada, 12 deles de forma independente e 1 com intervenção do investigador. Os utilizadores demoraram entre 1 segundo a 25 segundos a realizar esta tarefa, com uma média de 8 segundos. Os resultados obtidos desta tarefa podem ser observados nos gráficos da Figura 5.21.

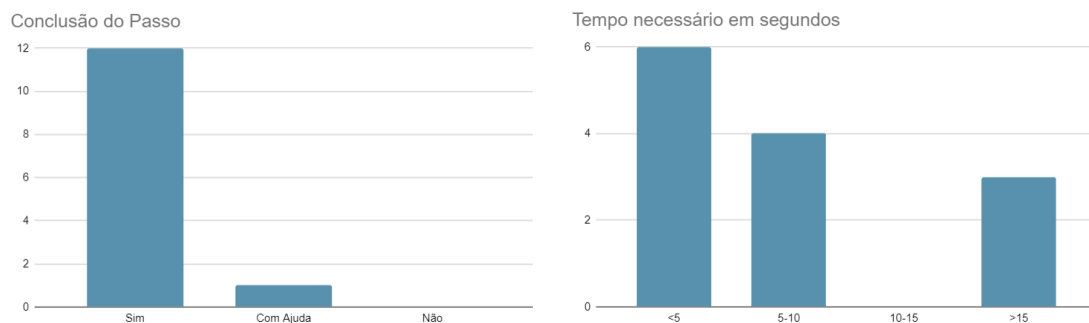


Figura 5.21: Gráficos dos resultados da Tarefa C6

#### 5.2.4 Questionário de Utilização

Após a conclusão de todos os testes os voluntários foram apresentados com duas séries de perguntas, uma primeira para avaliar a compreensão do utilizador sobre a criação e interpretação de histórias e a segunda um questionário de *System Usability Scale (SUS)*.

Na sua maioria, os utilizadores indicaram ser capazes de entender o objetivo de utilizar os diferentes passos da história para criar a história, interpretar o *Storyboard* e conseguir criar uma nova história sem supervisão. A noção de passos da história padrão e de passos com recolha de objeto foi, no entanto, um ponto de confusão. Os resultados da primeira série de perguntas podem ser observados na Figura 5.22.

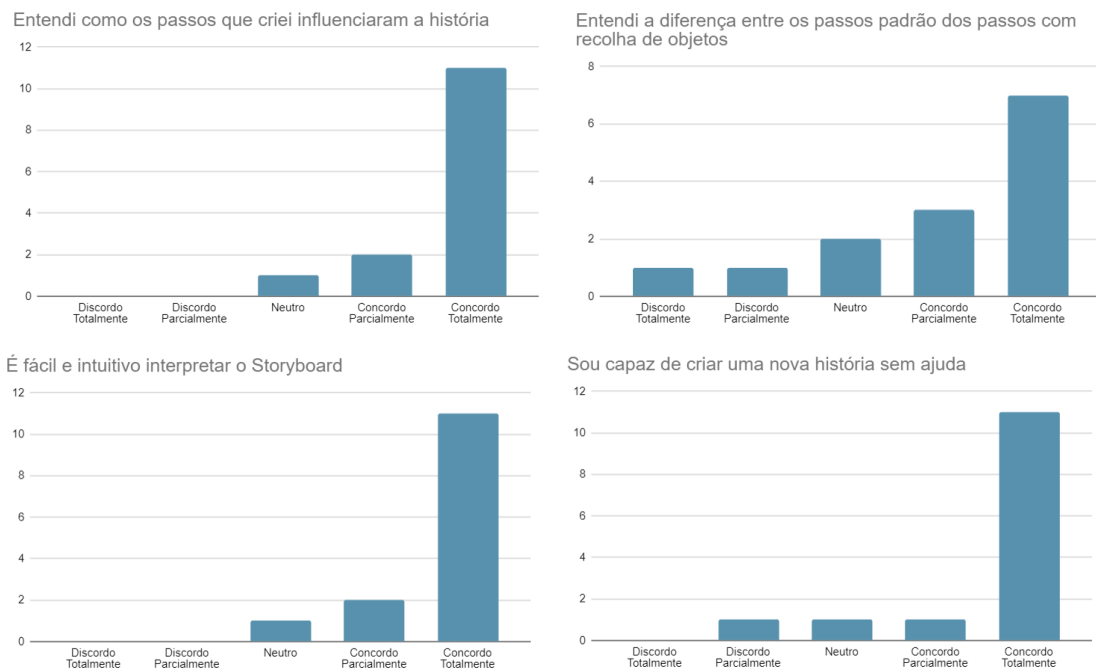


Figura 5.22: Gráficos dos resultados das perguntas relacionadas com a *Storyboard*

### 5.2.4.1 System Usability Scale (SUS)

O *System Usability Scale (SUS)*, é uma ferramenta criada por John Brooke em 1986 para avaliar a utilização de produtos ou serviços. Consiste numa série de 10 perguntas com 5 escolhas de resposta que variam de *Discordo Totalmente* (1) a *Concordo Totalmente* (5). Para poder avaliar o resultado de um questionário *SUS* é preciso seguir um processo simples.<sup>1</sup> Para perguntas ímpares, subtrair 1 ao valor da resposta. Para perguntas pares, subtrair o valor da resposta a 5. Adicionar o valor de todas as respostas do utilizador e multiplicar o valor por 2,5. O valor obtido vai estar abrangido entre 0 e 100 mas não reflete uma percentagem da qualidade do sistema. O valor médio de um questionário *SUS* é 68 pontos em 100, pelo que qualquer valor inferior a 68 encontra-se abaixo da média e qualquer valor superior a este encontra-se acima da média. Os resultados das perguntas individuais relacionadas com os testes *SUS* podem ser observados nos gráficos presentes na Figura 5.23.

<sup>1</sup>Measuring Usability with the System Usability Scale (SUS), <https://measuringu.com/sus/>, Acedido : 2023

## 5.2. USABILITY TESTS

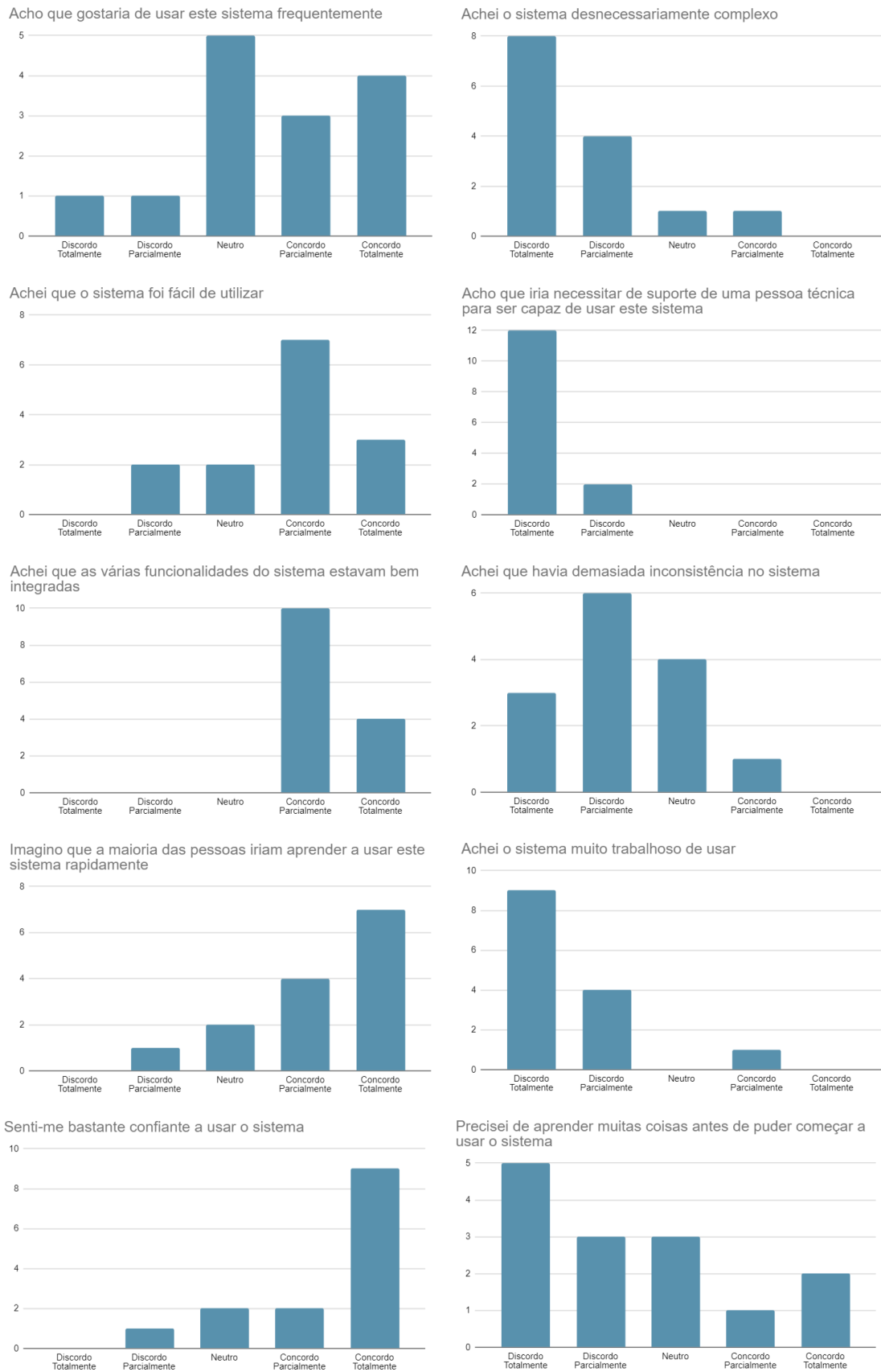


Figura 5.23: Gráficos dos resultados das perguntas relacionadas com o *System Usability Scale (SUS)*

Os resultados médios, de cada pergunta são apresentados na Figura 5.24.

Pergunta	Média
Acho que gostaria de usar este sistema frequentemente.	3.57
Achei o sistema desnecessariamente complexo.	1.64
Achei que o sistema foi fácil de utilizar.	3.79
Acho que iria necessitar de suporte de uma pessoa técnica para ser capaz de usar este sistema.	1.14
Achei que as várias funcionalidades do sistema estavam bem integradas.	4.29
Achei que havia demasiada inconsistência no sistema.	2.21
Imagino que a maioria das pessoas iriam aprender a usar este sistema rapidamente.	4.21
Achei o sistema muito trabalhoso de usar.	1.50
Senti-me bastante confiante a usar o sistema.	4.36
Precisei de aprender muitas coisas antes de poder começar a usar o sistema.	2.43

Figura 5.24: Médias, das diferentes perguntas do questionário *SUS*

O resultado final dá uma média de 78 pontos o que coloca o sistema desenvolvido acima da média. Os principais pontos negativos do sistema estão relacionados com existir alguma inconsistência neste e a necessidade dos utilizadores terem de aprender muito sobre o seu funcionamento antes de o poderem utilizar corretamente.

### 5.2.5 Observações e Conclusões

O utilizador médio, a usar a aplicação desenvolvida, pela primeira vez, demorou 3 minutos e 23 segundos a selecionar a história e efetuar as alterações que lhe foram requisitadas no cenário, 3 minutos e 16 segundos a criar a história e 1 minuto e 18 segundos a completar a história criada, para um tempo de utilização total médio de 7 minutos e 57 segundos. Nos diferentes exemplos em que os utilizadores foram obrigados a repetir etapas, seleção de uma história do repositório, seleção do modo em Criação e criar ligações entre passos da história registaram-se melhorias significantes nos tempos necessários para o utilizador as completar.

Na primeira seleção de uma história, os utilizadores demoraram entre 15 segundos a 2 minutos e 12 segundos a realizar esta tarefa, com um tempo médio de 1 minuto. Na segunda tentativa os mesmos utilizadores demoraram entre 5 segundos a 54 segundos a realizar esta tarefa, com um tempo médio de 19 segundos. O tempo médio entre estas duas tentativas foi 3 vezes mais rápido da segunda vez.

Na primeira tentativa de indicar as dependências entre os passos de uma história, os utilizadores demoraram entre 1 segundo a 1 minuto e 32 segundos a realizar esta tarefa, com uma média de 29 segundos. Na segunda tentativa, os utilizadores demoraram entre

1 segundo a 19 segundos a realizar esta tarefa, com uma média de 4 segundos. O tempo médio entre tentativas foi cerca de 7 vezes mais rápido da segunda vez.

Também na navegação entre os diferentes modos de edição do modo de Criação os utilizadores apresentaram uma melhoria significativa do tempo para efetuar a segunda tentativa mas, na primeira tarefa, tinham também que trocar entre mais modos.

As observações que foram efetuadas tanto pelos investigadores como pelos voluntários confirmaram algumas falhas no sistema apresentado aos utilizadores. Um foco dos utilizadores foi a existência de alguma inconsistência do funcionamento de processos no sistema. Algumas funcionalidades requeriam que o utilizador seleccionasse e arrastasse diferentes elementos do sistema, com o botão esquerdo do seu rato e, outras, necessitavam que o utilizador seleccionasse o elemento ao pressionar e largar o botão esquerdo do rato e depois onde o colocar, pressionando novamente o botão.

A abordagem de desenvolvimento que se efetuou no projeto da dissertação de apresentar uma interface simples, baseada em ícones e com pouco texto, apresentou uma dificuldade aos utilizadores que a estavam a testar pela primeira vez. Esta abordagem permite a utilizadores familiarizados com o funcionamento do sistema facilmente e rapidamente realizar as ações que pretendem mas dificulta a aprendizagem por parte de um novo utilizador. Praticamente todos os utilizadores apresentaram alguma dificuldade a efetuar muitas das tarefas pela primeira vez, mas indicaram que conseguiriam usar o sistema sem a necessidade de serem supervisionados numa utilização futura.

Para além dos dados retirados da interpretação dos valores medidos para a realização de cada tarefa e dos questionários efetuados, alguns voluntários manifestaram pedidos e sugestões de mudanças que consideraram melhorar o funcionamento do programa desenvolvido. Graças a estes dados e sugestões de utilizadores, várias alterações de interface com o utilizador e de funcionamento de alguns sistemas foram efetuadas de forma a melhorar o projeto desenvolvido. A principal, e mais simples de todas estas alterações, passou por manter a abordagem de uma interface simples, focada em ícones, mas acrescentar pequenas legendas aos diferentes elementos, de forma a que um utilizador experiente conseguisse rapidamente desempenhar a função que pretendesse e um novo utilizador aprendesse através da leitura o que cada elemento representava.

## CONCLUSÃO

### 6.1 Conclusões

A dificuldade principal da elaboração desta dissertação foi desenvolver um bom equilíbrio entre expressividade fornecida ao utilizador para representar a sua história e a quantidade de funções que teria que desempenhar para fazer a mesma.

Para alcançar este objetivo, a utilização de *Storyboards* apresentou ser uma solução perfeita para histórias lineares cujos eventos ocorriam numa ordem cronológica fixa. No entanto, para histórias em que era necessário introduzir alguma variabilidade na ordem em que os eventos pudessem ocorrer, um Fluxograma permitiu expressar e manipular informação de uma forma muito mais direta para o utilizador, mantendo na mesma a estrutura de um *Storyboard*.

Desta forma, foi desenvolvido um conjunto de ferramentas para conseguir suportar estes comportamentos para os utilizadores. Para permitir a criação de novas histórias, desenvolveu-se uma interface na qual o utilizador podia acrescentar os passos que necessitasse. A estes passos, para o utilizador conseguir dar-lhes significados, foi feita a recolha de diversos objetos e personagens que podiam ser atribuídos para especificar com o que devia ser interagido para desencadear o passo. Para poder indicar o sentido da história, foi apresentada uma maneira de o utilizador poder criar ligações entre os passos. Finalmente, foi desenvolvida toda a lógica necessária para processar as criações do utilizador, em experiências interativas na forma de Jogos de Aventura.

O resultado final, permitiu a utilizadores sem conhecimentos de programação, que usaram o programa pela primeira vez, contarem as suas histórias, partilharem as mesmas e jogarem diferentes histórias de aventura feitas por si e por outros utilizadores, após o tempo de aprendizagem inicial da navegação pelas diferentes possibilidades do sistema.

Os resultados obtidos da criação e adaptação de diferentes histórias para o modelo de jogo que foi criado permitiram identificar algumas restrições desta solução. Algumas histórias reais apresentam mais do que um personagem ativo a desencadear eventos para avançar a história, algo que não é permitido representar nesta estrutura. Outra limitação evidenciada no processo de criação de diferentes histórias é não ser possível

o aparecimento de objetos na cena a meio da história. É possível o desaparecimento de objetos quando o jogador os coleciona mas todos os objetos que estão em cena estão sempre visíveis. Não é permitido especificar, por exemplo, que um objeto esteja invisível até um acontecimento ser realizado e passe a ser visível daí em diante. É importante referir, novamente, que estas adaptações e considerações, foram feitas tendo sempre em mente o melhor equilíbrio encontrado entre expressividade disponibilizada e menor esforço do utilizador para a obter.

Algumas destas restrições são, no entanto, facilmente ultrapassadas, no sistema atual, se nos servirmos da possibilidade de dividir as histórias em diferentes partes. Desta forma, com divisões inteligentes da história, é possível ter diferentes personagens principais em cada parte da história ou novas cenas com objetos que se tornaram visíveis nesta parte da história.

Esta dissertação produziu as seguintes contribuições:

- **Storyboard** - DSL visual desenvolvida, com a qual o utilizador interage para definir os passos da sua história. Estrutura responsável por traduzir uma história de código para o aspeto visual apresentado ao utilizador e o contrário;
- **Interpretador de Histórias** - Interpretador dos dados criados pelo *Storyboard*, desenvolvido em *Unity*, responsável por interpretar a história criada pela *DSL*;
- **Jogos de Aventura** - As diferentes histórias exploradas no capítulo 5.1, "The Secret of Monkey Island", "Hansel and Gretel" (I e II) e "Restaurant" estão todos disponíveis no programa e podem ser explorados e alterados pelos utilizadores;

## 6.2 Trabalho Futuro

Um dos aspetos que necessita urgentemente de ser aprofundado no seguimento desta dissertação é a realização de uma prova formal, para demonstrar que o *Storyboard* desenvolvido para a criação e representação de histórias, permite contar qualquer história possível. Isto é, realizar a prova formal para comprovar que a *DSL* criada é *Turing Complete*.

Outro aspeto que deve ser explorado para enriquecer e facilitar a experiência dos utilizadores é a utilização de *Large Language Models (LLM)* para auxiliar a criação e reprodução de diálogo em cada evento da história. Com a crescente popularidade e acesso atual de programas de *LLM* para auxiliar na realização de todo o tipo de objetivos, a tarefa do utilizador seria imensamente simplificada. Este processo poderia, por exemplo, consistir na escolha de fragmentos de ações cujo sistema completaria as lacunas ou até mesmo um método completamente automatizado de criação de diálogo baseado no restante contexto da história já especificada. Este diálogo proveniente seria facilmente traduzido para diferentes línguas e poderia ainda ser automaticamente sintetizado em voz gerada por Inteligência Artificial para obter resultados ainda melhores.

## BIBLIOGRAFIA

- [1] C. Ashmore e M. Nitsche. «The Quest in a Generated World». Em: *Proceedings of Digital Games Research Association Conference, Situated Play* (2007) (ver p. 11).
- [2] M. D. Avgerinou e R. Pettersson. «Toward a Cohesive Theory of Visual Literacy». Em: *Journal of Visual Literacy* 30.2 (2011), pp. 1–19. DOI: [10.1080/23796529.2011.11674687](https://doi.org/10.1080/23796529.2011.11674687). eprint: <https://doi.org/10.1080/23796529.2011.11674687>. URL: <https://doi.org/10.1080/23796529.2011.11674687> (ver p. 6).
- [3] M. Brown. *How my Boss Key dungeon graphs work*. URL: <https://www.patreon.com/posts/how-my-boss-key-13801754> (acedido em 2023-01-29) (ver p. 10).
- [4] J. Campbell. *The Hero with a Thousand Faces*. 3rd Edition. The Collected Works of Joseph Campbell. HarperCollins Publishers;Fontana, 1949. ISBN: 9780586085714 (ver pp. 4, 6).
- [5] C. Clanton. «An Interpreted Demonstration of Computer Game Design». Em: *CHI 98 conference summary on Human factors in computing systems* (1998) (ver p. 10).
- [6] A. Creator. *Adventure Creator*. URL: <https://adventurecreator.org/> (acedido em 2023-01-27) (ver pp. 16, 17).
- [7] G. L. Denison. «Storyboarding: A Brief Description of the Process.» Em: *Annual International Convention of the Council for Exceptional Children 73rd, Indianapolis* (1995) (ver p. 8).
- [8] H. Desurvire, M. Caplan e J. A. Toth. «Using heuristics to evaluate the playability of games». Em: Association for Computing Machinery, 2004-04, pp. 1509–1512. ISBN: 1581137036. DOI: [10.1145/985921.986102](https://doi.org/10.1145/985921.986102) (ver p. 10).
- [9] J. Dormans. «Adventures in level design». Em: Association for Computing Machinery, 2010-06, pp. 1–8. ISBN: 9781450300230. DOI: [10.1145/1814256.1814257](https://doi.org/10.1145/1814256.1814257) (ver pp. 11, 12).
- [10] J. Dormans e S. Bakkes. «Generating Missions and Spaces for Adaptable Play Experiences». Em: *IEEE Transactions on Computational Intelligence and AI in Games* (2011) (ver pp. 11, 12).

- [11] S. Downie. *Murder on Mansion Hill: Encouraging Collaborative Group Storytelling to Improve Motivational Aspects of Literacy Using Gameplay and Arts-Based Techniques*. 2021, pp. 46–61. DOI: [10.1007/978-3-030-89394-1\\_4](https://doi.org/10.1007/978-3-030-89394-1_4) (ver pp. 8, 9).
- [12] R. Emanuel e S. Challons-Lipton. «Visual Literacy and the Digital Native: Another Look». Em: *Journal of Visual Literacy* 32.1 (2013), pp. 7–26. DOI: [10.1080/23796529.2013.11674703](https://doi.org/10.1080/23796529.2013.11674703). eprint: <https://doi.org/10.1080/23796529.2013.11674703>. URL: <https://doi.org/10.1080/23796529.2013.11674703> (ver p. 6).
- [13] K. Fog, C. Budtz, P. Munch e S. Blanchette. *The Four Elements of Storytelling*. Springer Berlin Heidelberg, 2010, pp. 31–46. DOI: [10.1007/978-3-540-88349-4\\_2](https://doi.org/10.1007/978-3-540-88349-4_2) (ver p. 4).
- [14] W. Goldstone. *Unity Game Development Essentials*. 1ª ed. Packt Publishing, 2009. ISBN: 9781847198181 (ver p. 15).
- [15] W. Grieb. «Walt Disney, Easter Eggs, and Monsters». Em: *Knowledge Quest* (2016) (ver p. 8).
- [16] J. K. Haas. «A History of the Unity Game Engine». Em: *Worcester Polytechnic Institute* (2014) (ver pp. 15, 16).
- [17] S. Kalberg. «Game Creation System. Dreams (PS4)». Em: *theseus.fi* (2020) (ver pp. 18, 19).
- [18] M. Lewis e J. Jacobson. «Game engines in scientific research». Em: *Communications of the ACM* 45 (2002-01), pp. 27–31 (ver p. 13).
- [19] *Lock and Key Dungeons*. BorisTheBrave.Com. 2021 (ver p. 10).
- [20] J. M. Lourenço. *The NOVAthesis L<sup>A</sup>T<sub>E</sub>X Template User's Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (ver p. ii).
- [21] M. Molecule. *Dreams*. [Videojogo]. 2020 (ver p. 18).
- [22] M. Molecule. *LittleBigPlanet*. [Videojogo]. 2008 (ver p. 18).
- [23] MoreMountains. *MoreMountains*. URL: <https://moremountains.com/> (acedido em 2023-01-27) (ver pp. 16, 17).
- [24] A. E. Pratt. *Cluedo*. [Jogo de Tabuleiro]. 1949 (ver pp. 8, 9).
- [25] A. Prokhorov. «The Hero's Journey and Three Types of Metaphor in Pixar Animation». Em: *Metaphor and Symbol* 36.4 (2021), pp. 229–240. DOI: [10.1080/10926488.2021.1919490](https://doi.org/10.1080/10926488.2021.1919490) (ver p. 6).
- [26] S. Rogers. *Level Up! The Guide to Great Video Game Design*. 2ª ed. 2014 (ver p. 10).
- [27] M. Romero e B. Sewell. *Blueprints Visual Scripting for Unreal Engine 5*. 3ª ed. 2022 (ver p. 14).
- [28] G. Satell. «The Little Known Secret To Pixar's Creative Success». Em: *Forbes* (2015) (ver p. 6).

## BIBLIOGRAFIA

---

- [29] F. Serafini. *Reading the Visual An Introduction to Teaching Multimodal Literacy*. Teachers College Press, 2014 (ver p. 7).
- [30] T. Shannon. *Unreal Engine 4 for design visualization: Developing stunning interactive visualizations, animations, and renderings*. 2017 (ver pp. 13, 14).
- [31] W. Storr. *The Science of Storytelling*. 2019. ISBN: 978-1-4197-4303-0 (ver p. 4).
- [32] R. Wahdid e A. Aziz. «Storyboarding: A model technique for the language learning process». Em: *LLT Journal: A Journal on Language and Language Learning* (2022) (ver p. 8).
- [33] D. Wiesner. *Flotsam*. Clarion Books, 2006 (ver p. 7).





Adventures in Quantum Computing