

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



GESTOR DE NARRATIVAS PARA JOGOS DE
FICÇÃO INTERACTIVA EM TEXTO

Frederico Nuno Nogueira Miranda

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Sistemas de Informação

2012

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



GESTOR DE NARRATIVAS PARA JOGOS DE
FICÇÃO INTERACTIVA EM TEXTO

Frederico Nuno Nogueira Miranda

DISSERTAÇÃO

Trabalho orientado pelo Prof. Doutor Luís Manuel Ferreira Fernandes Moniz

e co-orientado pelo Prof. Doutor Paulo Jorge Cunha Vaz Dias Urbano

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Sistemas de Informação

2012

Agradecimentos

Chegar aqui é o culminar de um longo percurso cheio de altos e baixos, onde a persistência foi a palavra-chave.

Quero agradecer á minha família, em especial à minha mãe que sempre me apoiou em todos os momentos de uma forma incondicional, agradeço á minha avó Maria Antónia pela força, incentivo e boa disposição que me tem dado ao longo desta caminhada.

Estou muito grato ao Prof. Doutor Luís Manuel Ferreira Fernandes Moniz e ao Prof. Doutor Paulo Jorge Cunha Vaz Urbano que desempenharam os seus papéis de orientador e co-orientador de um modo exemplar. Sempre demonstraram disponibilidade, boa disposição e aconselharam ao longo do projecto.

Por último mas não menos importantes, foram os amigos que fiz ao longo deste percurso, Alexandre Gabriel, André Bastos, Bruno Seixas, Carlos Teixeira, Filipe Gil, Geraldo Nascimento, Gonçalo Cruchinho, Henrique Morais, João Costa, João Lobo, João Lopes, Joaquim Reis, Louis Lopes, Rui Flores, Rui Gameiro, Tiago Raimundo e também quero agradecer a todos os meus colegas do LabMAg que me fizeram sentir em casa.

Muito Obrigado!

Para um mundo de videogames melhor.

Resumo

Este relatório tem como objectivo descrever o meu trabalho realizado durante a tese em Gestor de Narrativas para Jogos de Ficção Interactiva em Texto.

Ao contrário dos outros jogos que se baseiam mais na jogabilidade e nos gráficos (como por exemplo jogos de desporto), na ficção interactiva em texto o principal foco é na narrativa e na resolução de puzzles. A interface nestes jogos é simples, baseada apenas em texto, embora também possa conter imagens, animações e vídeos. Nestes jogos o jogador encarna um personagem e interage com o meio que o rodeia através de comandos em texto.

Este tipo de comunicação com o jogo gera vários problemas, como por exemplo, obrigar o jogador a saber um leque de palavras para conseguir interagir na história, ou o jogador perder-se no jogo devido à dimensão deste, tentar vários comandos e nenhum deles levar a uma saída óbvia. Apesar dos jogadores se depararem com problemas nesta área, eles não estão sozinhos, os autores também encontram alguns quando querem criar as suas narrativas. A maioria das ferramentas disponíveis para a criação destes jogos é focada para autores com noções de programação, reduzindo assim o público-alvo.

Para resolver estes problemas nós propomos o FIGN, um conjunto de ferramentas de software, desenhadas para o desenvolvimento de jogos de ficção interactiva em texto, com gestor de narrativas incorporado. Com o FIGN o autor não precisa de saber programar para conseguir construir os seus jogos. Ao usar o FIGN o autor vai poder colocar vários marcos ao longo da narrativa do jogo, chamados de pontos de enredo, que por sua vez, vão auxiliar o gestor de narrativas a controlar o progresso do jogador ao longo do jogo. O autor também vai definir algumas acções do gestor de narrativas para este poder actuar de modo a guiar o jogador pelo enredo da história, como por exemplo, dar sugestões, dar itens ou negar zonas, melhorando a sua experiência de jogo.

Palavras-chave: Ficção Interactiva, Aventuras em Texto, Gestores de Narrativas

Abstract

This document describes the work done during the elaboration of my thesis in Drama Manager with Interactive Fiction Games.

Unlike other games based on gameplay and graphics (for example sports games), in Interactive Fiction the main goal is focused on narrative of the game and puzzle solving.

The interface is simple, usually it is text-only but it might contain pictures, animations and videos. In these classes of games the player embodies a character and interacts with its environment through text commands.

This kind of communication between the player and the game may cause some problems, for example, forcing the player to memorize a certain amount of words to be able to interact with the environment, or the player get lost in the game because of its size, after attempt several text commands and none of them help. Despite the players encounter problems like those, they are not alone, even interactive fiction authors are confronted with them when they try to create their own narratives. Most of the tools available to develop these games are oriented to people with programming skills thereby reducing the target audience.

In order to solve this problems we propose FIGN, a pack of software tools designed to the development of Interactive Fiction Text Games incorporated with a Drama Management. With FIGN the author does not need any programming skill in order to create his own games. Using FIGN the author mark his story with plot points which will help the drama manager to guide the player through the story. The author also define a set of actions to make sure the drama manager can guide the player through the story, for example, giving hints or items, denying zones, resulting in an improved gaming experience.

Keywords: Interactive Fiction, Text Adventures, Drama Manager

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Motivação	1
1.2	Objectivos.....	2
1.3	Contribuições.....	2
1.4	Planeamento	3
1.5	Estrutura do documento.....	4
Capítulo 2.....		6
2.1	Perspectiva Histórica	6
2.2	Ficção Interactiva	7
2.3	Editores para criação de Ficção Interactiva em texto	10
2.3.1	Adrift	11
2.3.2	Alan	12
2.3.3	Hugo	13
2.3.4	Inform 7.....	14
2.3.5	Quest.....	15
2.3.6	TADS	16
2.4	Interpretadores de Ficção Interactiva	17
2.4.1	Interpretador do Adrift	17
2.4.2	Interpretador do Alan	18
2.4.3	Interpretador do Hugo	19
2.4.4	Interpretador do Inform 7	20
2.4.5	Interpretador do Quest.....	21
2.4.6	Interpretador do TADS.....	22
2.4.7	Comparação dos Editores.....	23
2.5	Gestores de Narrativas.....	24
Capítulo 3.....		28

Desenvolvimento do FIGN	28
3.1 Representação do Jogo e do Gestor de Narrativas	28
3.1.1 Representação do Jogo	28
3.1.2 Arquitectura do Interpretador do FIGN	29
3.1.3 Motor de Jogo.....	30
3.1.4 Modelo do Jogador.....	30
3.1.5 Pontos de Enredo.....	30
3.1.6 Gestor de Narrativas.....	33
3.1.7 Representação da Estrutura do Jogo.....	43
3.2 Editor do FIGN.....	45
3.2.1 Preocupação na interface do editor	45
3.2.2 Separador de informação.....	48
3.2.3 Separador da criação de zonas	48
3.2.4 Separador de Itens	51
3.2.5 Separador de <i>NPCs</i>	52
3.2.6 Separador de Pontos de Enredo.....	53
3.2.7 Separador de Gestor de Narrativas.....	54
3.2.8 Separador de Interface.....	55
Capítulo 4.....	57
Discussão e Trabalho Futuro.....	57
4.1 Conclusões.....	57
4.2 Trabalho futuro	58
Capítulo 5.....	60
Bibliografia	60

Lista de Figuras

1	Screenshot do jogo Zork	7
2	Mapa do jogo Zork	8
3	Screenshot do jogo Myst	9
4	Screenshot do jogo Monkey Island	9
5	Editor do Adrift	11
6	Editor do Alan	12
7	Editor do Hugo	13
8	Editor do Inform 7	14
9	Editor do Quest	15
10	Editor do TADS	16
11	Interpretador do Adrift	17
12	Interpretador do Alan	18
13	Interpretador do Hugo	19
14	Interpretador do Quest	21
15	Interpretador do TADS	22
16	Arquitectura do Interpretador	29
17	Mapa do jogo criado para demonstrar o FIGN	31
18	Exemplo da criação de um ponto de enredo	34
19	Exemplo do algoritmo minmax	35
20	Exemplo do algoritmo médiamax	36
21	Criação da árvore de pesquisa	37
22	Primeira fase da criação das subárvores	38
23	Segunda fase da criação das subárvores	38
24	Árvore de pesquisa pronta	39
25	Decisão do gestor de narrativas	41
26	Ação do gestor de narrativas	42
27	Exemplo da estrutura das zonas do jogo	43

28	Exemplo da estrutura dos itens do jogo	44
29	Exemplo da estrutura dos pontos de enredo no jogo	44
30	Screenshot do NetBeans	45
31	Filechooser	46
32	Sugestões de ajuda	47
33	Exemplo de criação de zonas e opções	48
34	Visibilidade das opções disponíveis	50
35	Exemplo de opção que não precisa de itens	50
36	Exemplo de criação de itens	51
37	Exemplo de criação de NPC	52
38	Exemplo de criação de pontos de enredo	53
39	Exemplo da escolha da profundidade de pesquisa	54
40	Exemplo da personalização da interface	55

41 **Lista de Tabelas**

1	Planeamento	3
1	Planeamento	4
2	Comparação entre editores	23

Capítulo 1

Introdução

1.1 Motivação

A Ficção interactiva (FI) [2, 3, 4], conhecido em inglês como *Interactive Fiction*, é um género de jogos também rotulado como “aventuras em texto” ou “jogos de ficção interactiva em texto”. Descrever este estilo de jogos pode parecer um pouco entediante, mas começar um jogo é como entrar na história de um livro, transporta-nos para um mundo onde o jogador pode vaguear livremente e influenciar a história. Na FI é possível encontrar várias obras literárias incluindo utopias, vinganças, horrores, parábolas, intrigas, como por exemplo *Anchorhead*, um jogo que foi nomeado para 6 prémios Xyzzy, no concurso anual de ficção interactiva.

Na FI o jogador encarna um personagem e controla-o através de comandos em texto, participando num diálogo com o jogo. Por baixo desta conversa e a determinar o que o computador vai narrar, encontra-se um interpretador que procura dar resposta a cada frase que o utilizador escreve, comparando as palavras do utilizador com o conjunto de palavras que o interpretador conhece.

Este tipo de jogos pode tornar-se num labirinto para muitos jogadores, após experimentarem vários comandos em texto e nenhum os levar a uma saída óbvia. Esta situação causa frustração, que não é algo que se pretenda obter num jogo. Outra razão para ficar frustrado é o facto de que grande parte das ferramentas usadas para criar jogos de FI ser focada para pessoas com alguma experiência de programação, deixando assim muitas mentes criativas de fora.

Para combater estes problemas criámos o FIGN, um conjunto de ferramentas que permite solucionar estes dois problemas de uma só vez. O FIGN dá hipóteses aos autores de criarem jogos sem serem obrigados a ter noções de programação e utiliza um gestor de narrativas. O gestor de narrativas tem o papel de guiar o jogador pela narrativa sempre que este estiver a perder o rumo na história. Para isso ele vai contar com acções

que pode realizar no jogo para guiar o jogador, caso este precise de ajuda. O autor vai definir estas acções que podem ser: dar sugestões, dar itens ou restringe o acesso do jogador a certas zonas do mapa. Além das acções o autor também vai poder colocar pontos de enredo na sua história, estes pontos servem de referências para o gestor de narrativas controlar o progresso do jogador.

1.2 Objectivos

O principal objectivo deste trabalho é combater os problemas referidos na secção 1.1, ou seja, a falta de gestores de narrativas para apoiar os jogadores, maximizando a sua experiência de jogo e um editor de jogos que não force o autor a saber programar. Para isto decidimos criar um protótipo que fosse capaz de colmatar estes problemas e assim nasceu o FIGN, um conjunto de ferramentas de software desenhadas para o desenvolvimento de jogos de ficção interactiva em texto com gestor de narrativas incorporado.

O editor do FIGN precisa de ter uma interface amigável, que permita aos utilizadores criarem as suas histórias sem necessitarem de conhecimentos de programação.

O interpretador do FIGN vai ter uma interface simples e vai poder contar com um gestor de narrativas.

1.3 Contribuições

As contribuições deste trabalho podem resumir-se nos seguintes pontos:

- Um editor de jogos de ficção interactiva que permite ao autor, sem recorrer a uma linguagem de programação formal, definir todo o cenário do jogo, os personagens intervenientes e as interacções disponíveis.
- Um interpretador de jogo que permite executar um jogo definido pelo autor e controlar as interacções com o jogador.
- Um gestor de narrativas integrado no interpretador, que controla o desempenho do jogador, intervindo em caso de necessidade de modo a proporcionar a melhor experiência de jogo possível.
- Um jogo que permite demonstrar as várias funcionalidades do gestor de narrativas.

1.4 Planeamento

Na tabela seguinte segue-se o planeamento do projecto desenvolvido. Neste projecto a construção do interpretador e do editor foram quase sempre feitas em simultâneo, pois cada vez que uma das ferramentas era alterada, havia necessidade de actualizar a outra também.

Tarefas	Data de início	Data de Fim	Descrição
Investigação	20/09/2011	30/09/2011	Recolha de informação para o trabalho.
Protótipo do interpretador e do editor	01/10/2011	12/10/2011	Elaboração do primeiro protótipo funcional.
Demo do jogo	13/10/2011	14/10/2011	Elaboração de um pequeno demo do jogo para testes.
Aperfeiçoamento do interpretador	15/10/2011	26/10/2011	Adição de funcionalidades como interacção com itens, uso de itens para mudar de zona.
Aperfeiçoamento do editor	27/10/2011	10/11/2011	Melhoramento da interface, adição de funcionalidades como por exemplo, configurar itens.
Nova iteração sobre o interpretador	11/10/2011	18/11/2011	Melhoramento da interface, adição de funcionalidades, como por exemplo, interacção com <i>NPCs</i> .
Relatório preliminar	18/11/2011	15/12/2011	Elaboração do relatório preliminar.
Investigação	16/12/2011	25/01/2012	Investigação sobre Gestores de Narrativas.
Elaboração do Gestor de Narrativas.	26/01/2012	30/04/2012	Sucessivas iterações sobre o interpretador e o editor e várias fases de testes.

Criação do jogo	1/05/2012	16/05/2012	Elaboração de um jogo para demonstrar as funcionalidades do Gestor de Narrativas.
Finalização do Gestor de Narrativas	17/05/2012	29/06/2012	Pequenos ajustes no interpretador e no editor.
Elaboração de um artigo	30/06/2012	15/07/2012	Elaboração de um artigo.
Elaboração do relatório final	16/07/2012	26/09/2012	Elaboração do relatório final.

Tabela 2 – Planeamento do projecto

1.5 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 2 – Trabalho Relacionado

Este capítulo descreve os trabalhos relacionados na área em que o projecto desenvolvido se insere.

- Capítulo 3 – Trabalho Realizado

Este capítulo descreve o trabalho desenvolvido no âmbito do conjunto de ferramentas FIGN. A arquitectura, componentes e ferramentas usadas para o seu desenvolvimento.

- Capítulo 4 – Discussão e Trabalho Futuro

Neste capítulo são apresentadas as conclusões e perspectivas para trabalho futuro.

Capítulo 2

2.1 Perspectiva Histórica

Desde o início dos tempos que o homem conta histórias[1], o que inicialmente começou como histórias de caça e contos dos antepassados rapidamente tomou a forma de mitos e lendas que constituíram a identidade da tribo.

Algumas histórias apareceram para ensinar, outras para avisar, umas para tentar resolver os grandes mistérios do mundo e outras apenas para simples entretenimento.

Ao longo do tempo, independentemente da língua, da raça, da religião, o homem passou de geração em geração uma riqueza cultural nas suas histórias. Enquanto algumas foram caindo no esquecimento outras têm sido recontadas século após século, moldando os nossos pensamentos, religiões, filosofias e até o próprio mundo.

As histórias tiveram um grande impacto no homem, levaram-no a explorar, lutar, ter esperança e até sonhar. Elas foram inspiração para arte, musica, tecnologia, e para mais e mais histórias.

Hoje em dia com a TV, a internet, a rádio, os jornais, os livros, temos uma fonte quase infinita de temas que cobrem todos os assuntos que se possam imaginar. Toda esta panóplia de escolhas á nossa disposição tornou-nos mais exigentes. Quando começamos a ler um livro e na história há algo que não nos agrada, como por exemplo, o enredo, um personagem, uma situação, o que nos leva a acabar de o ler? Afinal de contas, temos muitas outras histórias à nossa espera.

2.2 Ficção Interactiva

Ficção Interactiva (FI) é um termo amplo. Estritamente falando, Ficção Interactiva consiste em influenciar o resultado de uma história, como os livros do tipo “*Chose Your Own Adventure*” com histórias ramificadas.

Em geral os jogos de aventura são programas que nos contam uma história, deixam-nos encarnar um personagem e vamos vivendo a história através das nossas acções. Em muitos jogos de FI temos que resolver puzzles para poder prosseguir na narrativa, como por exemplo “Como é que destranco a porta?”. Em alguns jogos também é necessária a interacção com *NPCs (non-player characters)* para avançar na história.

Ao contrário de muitos jogos que se baseiam mais na destreza física do jogador e dependem muito dos reflexos (como o Quake), em FI o principal foco é na narrativa e na resolução de puzzles apelando à destreza mental.

Em FI consegue-se ter dois tipos de jogos, gráficos e em texto. As aventuras em texto apareceram primeiro, joga-las é como ler um livro, mas onde é necessário usar comandos em texto para fazer o protagonista interagir com meio ambiente. A interface é em geral apenas texto, mas também pode conter imagens, animações e vídeos (como a interacção com o jogo é em texto ainda se mantém na categoria de aventuras em texto). Um dos primeiros jogos deste género foi Zork: The Great Underground Empire – Part I [12], (ver Figura 1). Neste jogo o principal objectivo é encontrar os 20 tesouros de Zork, para os encontrar é preciso resolver vários puzzles e passar por labirintos[13].

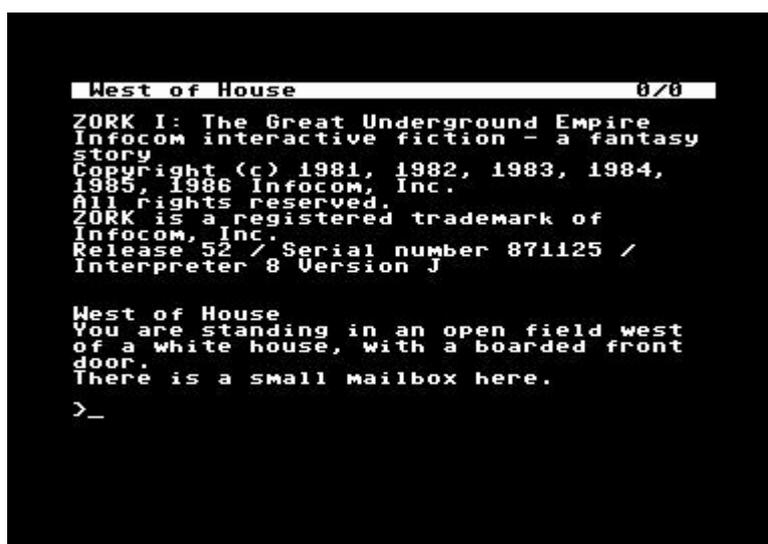


Figura 1 – Screenshot do jogo Zork: The Great Underground Empire - Part I

ZORK I

The Great Underground Empire

Legend

- Normal Passageway
- One-way passageway
- Passageway requiring special equipment or problem solving
- ||| Narrow passageway (passage limit)
- Passageway returning to room of origin

NOTES: All horizontal passages leave the room in the direction shown. Vertical passages are labeled "U" for UP and "D" for DOWN. To avoid unnecessarily giving away problems, this map lists only those objects immediately visible upon entering a room. Only objects which can be taken or used are listed; objects which are merely part of a room are not. Where more than one direction leads to the same place, all are not necessarily shown.

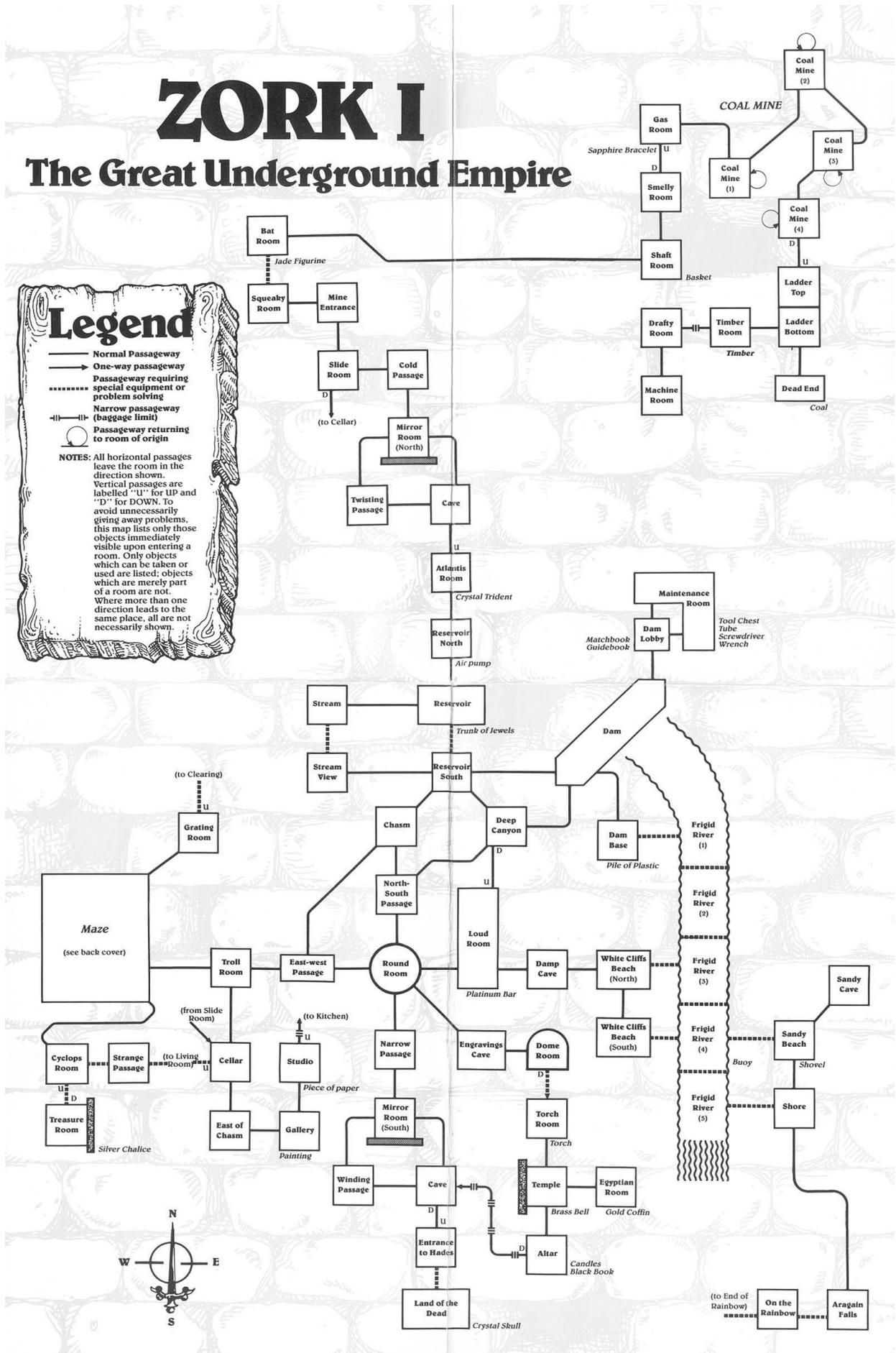


Figura 2 – Mapa parcial do jogo Zork: The Great Underground Empire - Part I

Como é possível ver na Figura 2, a dimensão do mapa onde o jogador se deslocava é vasta e as probabilidades de um jogador se perder num mapa destas dimensões são grandes.

Ao contrário dos jogos de aventura em texto as aventuras gráficas contam histórias através de imagens, passaram de um *parser* de texto para uma interface do tipo *point and click*, ou seja, em vez do jogador comunicar através de comandos em texto limita-se a clicar no ecrã com o cursor do rato. Um exemplo deste estilo de jogos é o Myst[14] (Figura 3) que foi um dos primeiros jogos de aventuras gráficas na primeira pessoa, neste caso o jogador via o mundo através dos olhos do protagonista, ou Monkey Island[15] (Figura 4) onde muitas das opções que se podiam executar faziam lembrar as opções disponíveis nos jogos de aventura em texto.



Figura 3– Screenshot do jogo Myst



Figura 4 – Screenshot do jogo Monkey Island

2.3 Editores para criação de Ficção Interactiva em texto

Os jogos de ficção interactiva[2] atingiram o seu pico de popularidade de entre 1979 e 1986. Para criar um jogo bastava ter uma mente criativa, saber escrever e saber usar as ferramentas que existiam para a criação destes jogos. Devido ao aumento da popularidade foram criadas várias ferramentas, cada uma com as suas próprias características. Algumas eram focadas para autores com experiência em programação, outras conseguiam abstrair o autor de tais conceitos. De entre as várias ferramentas algumas das mais conhecidas[2][5] são:

- Adrift [5]
- Alan [6]
- Hugo [7]
- Inform 7 [8]
- Quest [9]
- TADS [10]

2.3.1 Adrift

Ferramenta que permite a criação rápida e fácil de jogos de ficção interactiva em texto, permite que o autor se concentre na criação do jogo sem ter de perder tempo a aprender a programar. Os jogos são criados através da adição de zonas, objectos, personagens e tarefas. Na Figura 5 podemos ver um exemplo da criação de um jogo nesta plataforma, onde as zonas criadas aparecem em rectângulos azuis ligados entre si, mostrando os caminhos que as ligam.

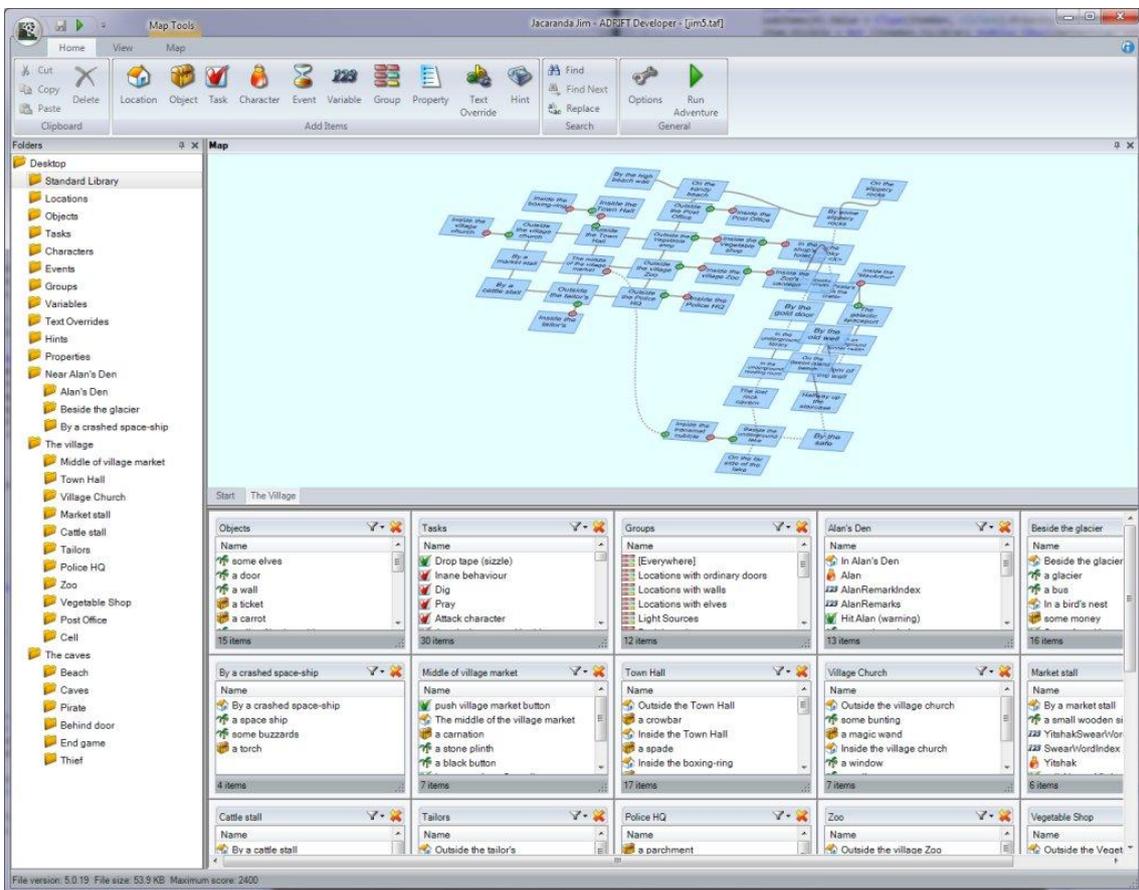


Figura 5 – Screenshot da criação de um jogo usando o Adrift

2.3.2 Alan

Esta ferramenta tem a sua própria linguagem, a linguagem Alan, através dela o autor pode construir os mais variados jogos de ficção interactiva em texto que desejar. Como se pode ver na Figura 6, temos um exemplo da criação de um jogo de aventura em texto onde a sintaxe da linguagem encontra-se a vermelho.

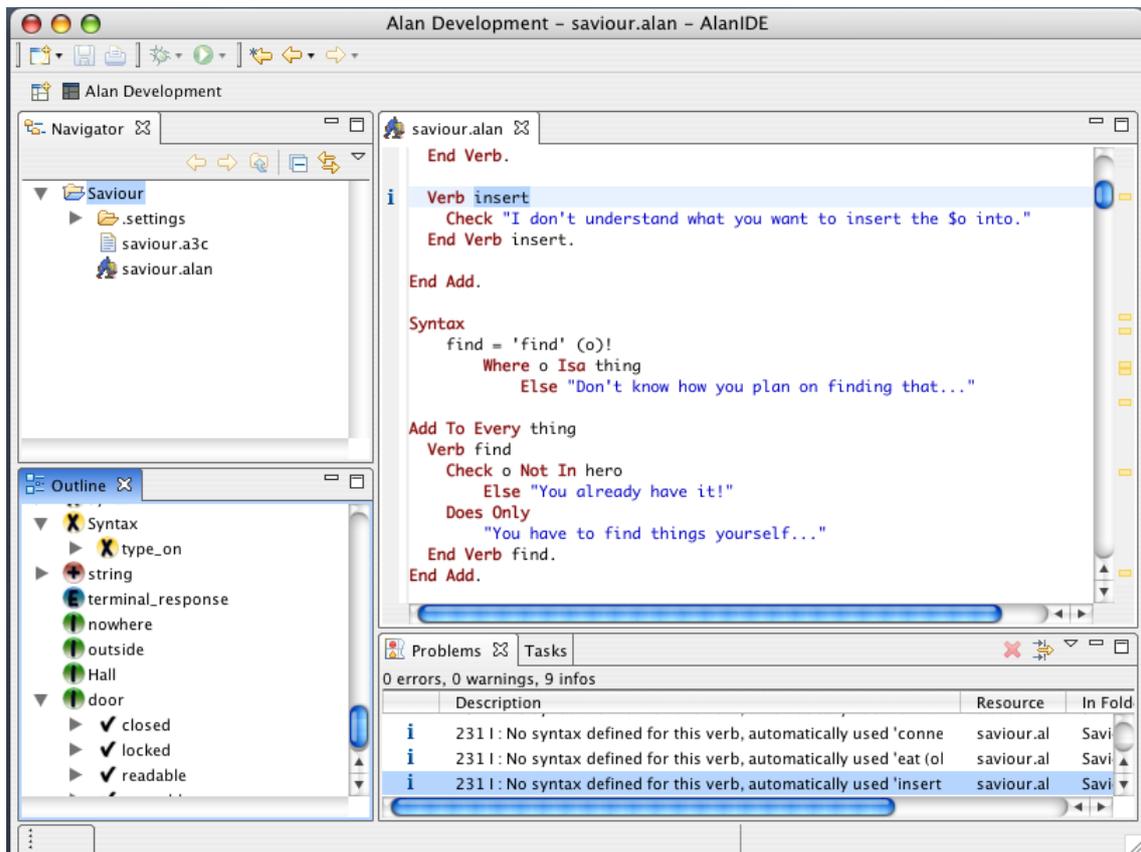


Figura 6 - Screenshot da criação de um jogo usando o Alan

2.3.3 Hugo

Esta ferramenta possibilita a criação de jogos de ficção interactiva através de algum conhecimento de programação para usufruir da sua linguagem de alto nível. Esta ferramenta também permite o uso de imagens, musica, efeitos sonoros e vídeo. Além de jogos também é possível criar apresentações, tutoriais, protótipos e outras aplicações.

Um exemplo de um jogo criado com esta ferramenta é o Future Boy[11]. Na Figura 7 podemos observar um screenshot desta ferramenta.

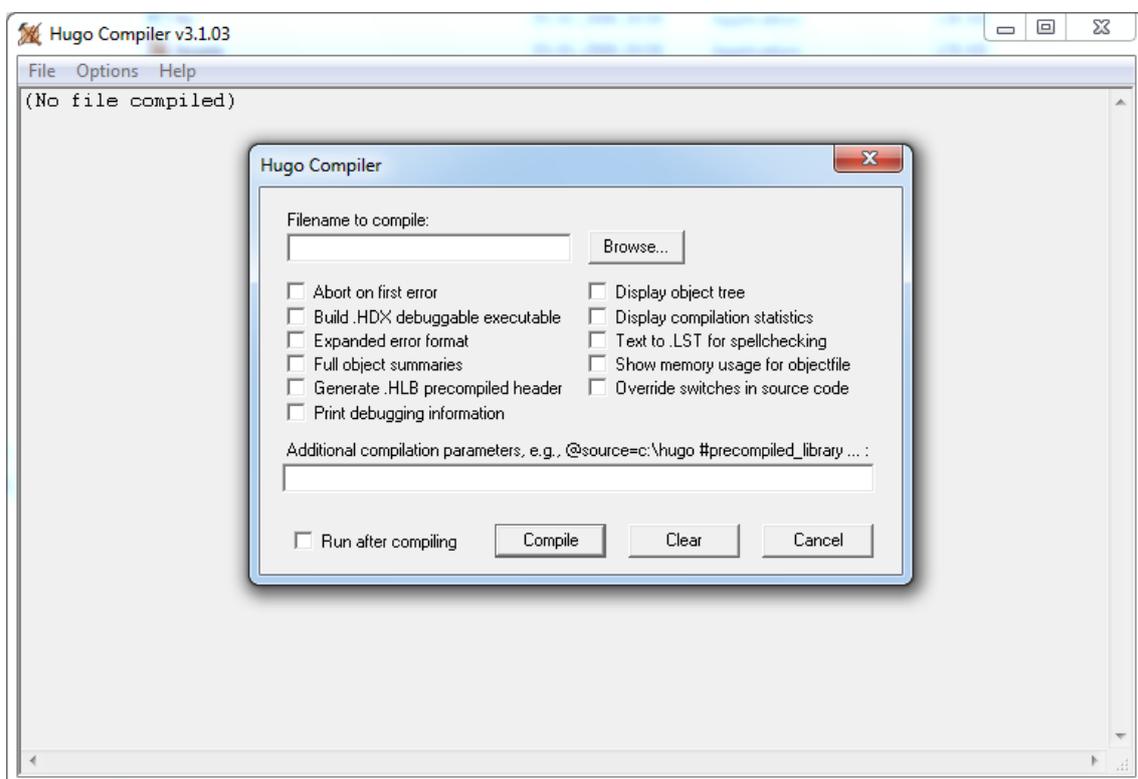


Figura 7 – Screenshot do compilador do Hugo

2.3.4 Inform 7

Esta ferramenta faz uso de uma sintaxe baseada na linguagem natural, como se pode ver na Figura 8 que demonstra a criação de um jogo. Do lado esquerdo encontra-se o código da história, neste caso para definir uma zona o autor escreve “Nome da zona *is a room*” e quando é compilado, é associada a noção de *room* à zona definida pelo autor. No lado direito encontra-se o mapa do jogo criado. Uma das grandes vantagens do inform 7 é de ter um manual de instruções incorporado, chamado livro de receitas (na *tab Documentation*) com um vasto número de exemplos para auxiliar na construção de jogos.



Figura 8 – Screenshot da criação de um jogo usando o Inform 7

2.3.5 Quest

A ferramenta Quest permite ao autor criar jogos de ficção interactiva em texto sem necessitar de ter noções de programação devido à sua interface simples e intuitiva. Com esta ferramenta é possível adicionar imagens, música, efeitos sonoros e até vídeos do YouTube e Vimeo.

É possível ver um exemplo da edição de um jogo na Figura 9.

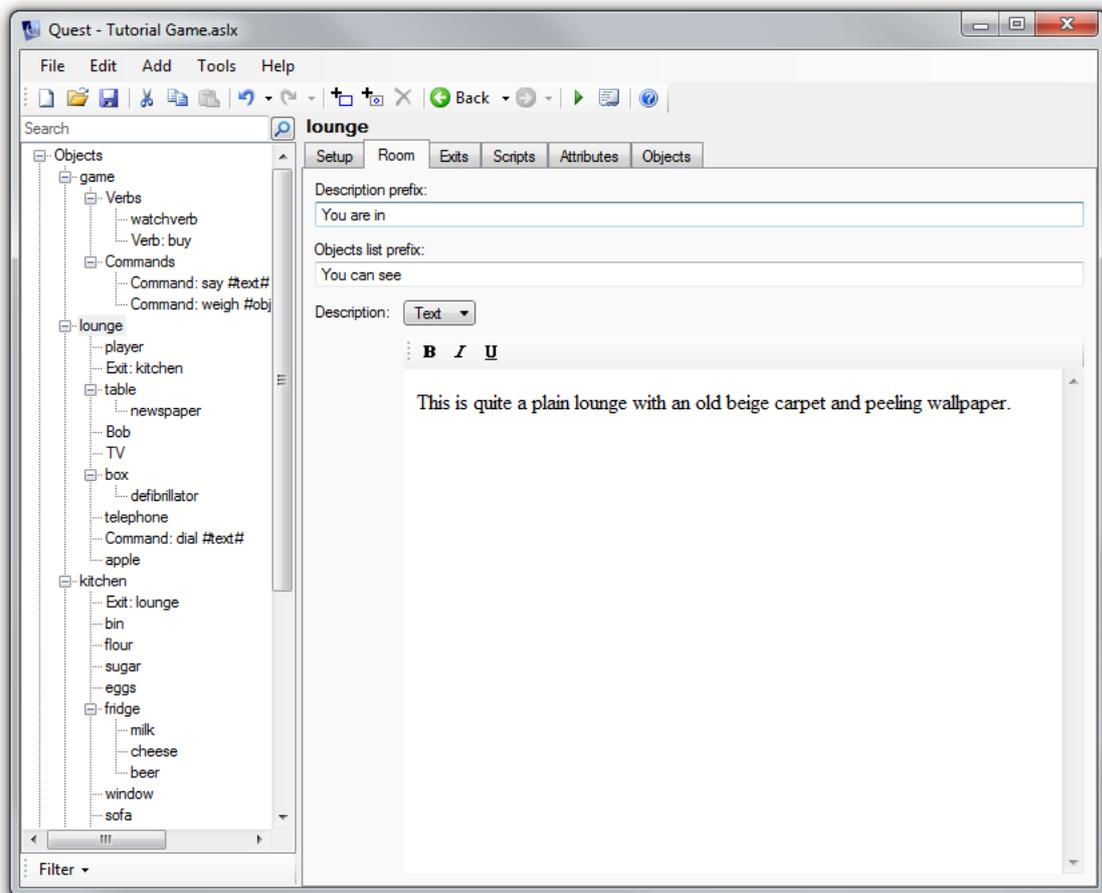


Figura 9 - Screenshot da criação de um jogo usando o Quest

2.3.6 TADS

O TADS é uma ferramenta que permite criar jogos de ficção interactiva através da sua própria linguagem, semelhante a C++ e Javascript (ver Figura 10). Esta ferramenta já tem o HTML TADS integrado o que lhe possibilita editar os jogos tradicionais constituídos apenas por texto e os jogos mais modernos que tiram partido das funcionalidades multimédia do HTML TADS como por exemplo, imagens, música e hiperligações.

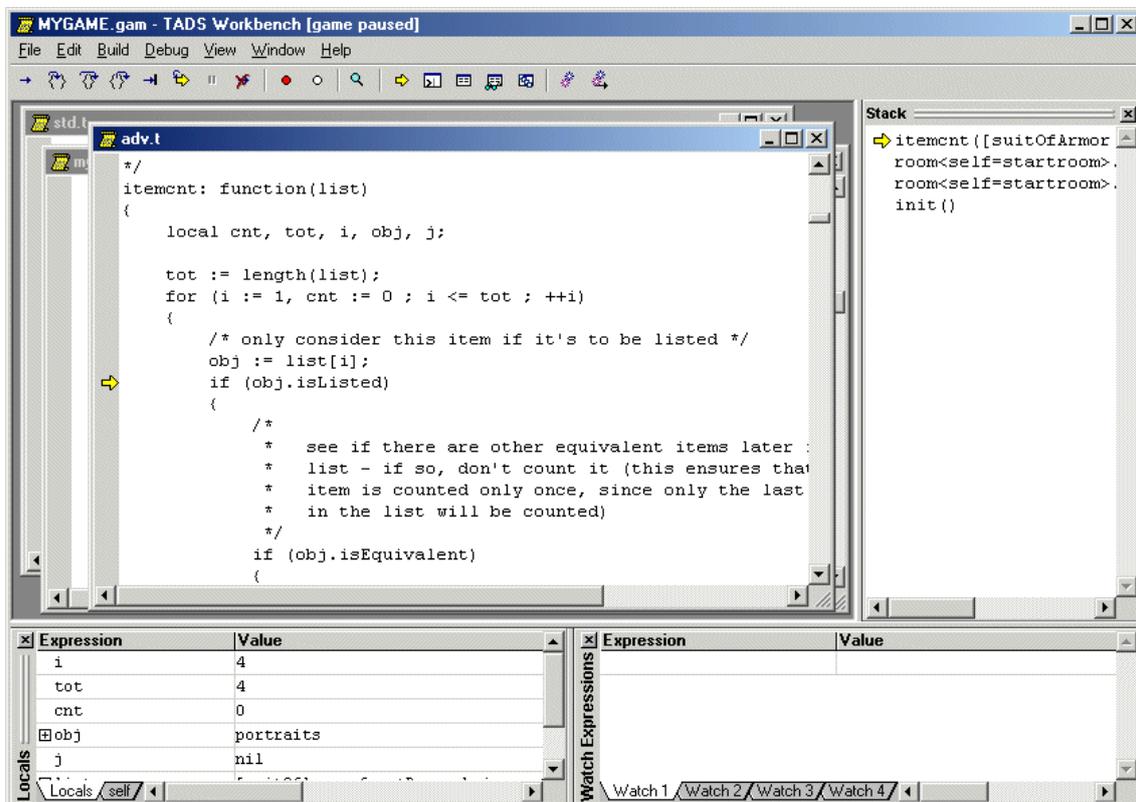


Figura 10 - Screenshot da criação de um jogo usando o TADS

2.4 Interpretadores de Ficção Interactiva

Na ficção interactiva as histórias são criadas através de um grande número de programas, muitos deles nem partilham a mesma linguagem. Para conseguir colmatar este problema os jogos são compilados para um formato independente da plataforma. Os interpretadores, por sua vez são construídos especificamente para certos sistemas operativos, comportando-se como máquinas virtuais que conseguem executar os jogos. Uma das grandes vantagens deste conceito de máquinas virtuais é a compatibilidade entre plataformas, os jogos apenas precisam de ser criados uma única vez para poderem ser executados em qualquer computador.

2.4.1 Interpretador do Adrift

O interpretador do Adrift neste momento apenas suporta Windows e Linux, embora estejam a desenvolver interpretadores para suportar também Mac e navegadores da web. Uma das suas grandes vantagens consiste em ter um mapa em 3d (ver Figura 11) que mostra os locais onde o jogador já esteve e pode ser rodado permitindo diferentes pontos de vista.

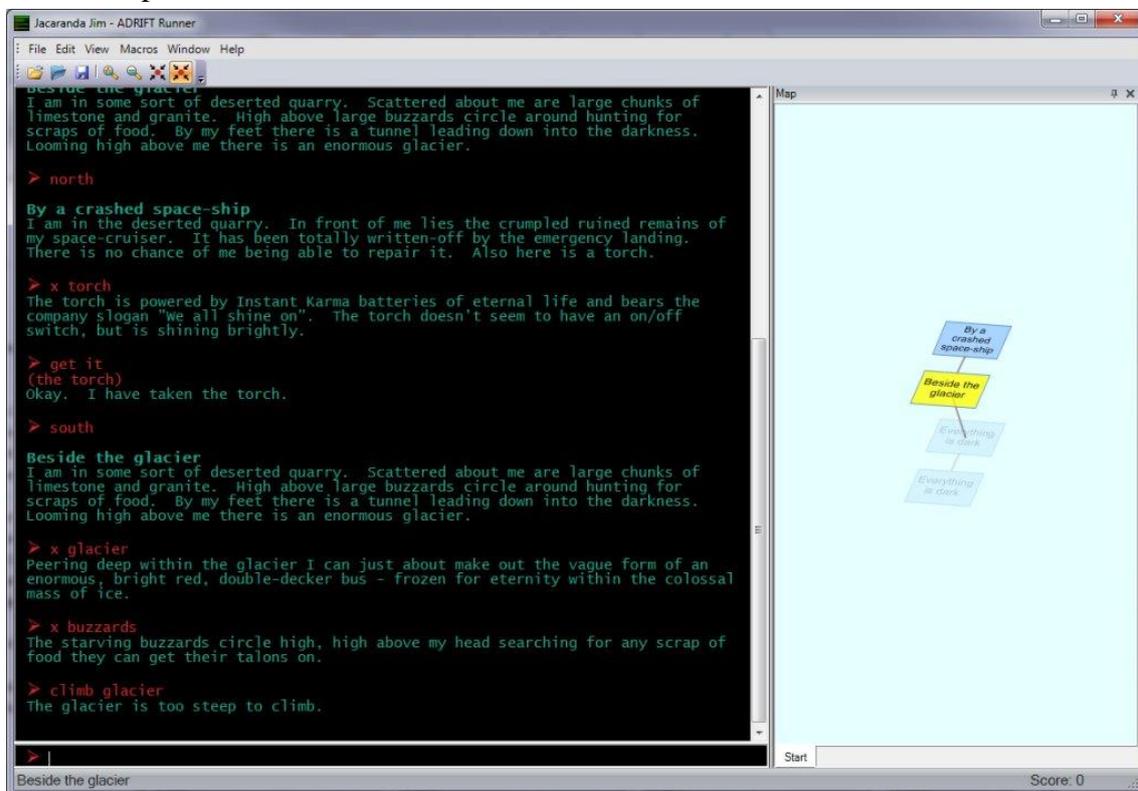


Figura 11 – Interpretador do Adrift

2.4.2 Interpretador do Alan

Para executar os jogos criados com a ferramenta Alan é necessário usar o WinArun. Na Figura 12 podemos ver um screenshot do WinArun a executar o jogo *The Hollywood Murders*. Suporta Windows, Linux e Mac.

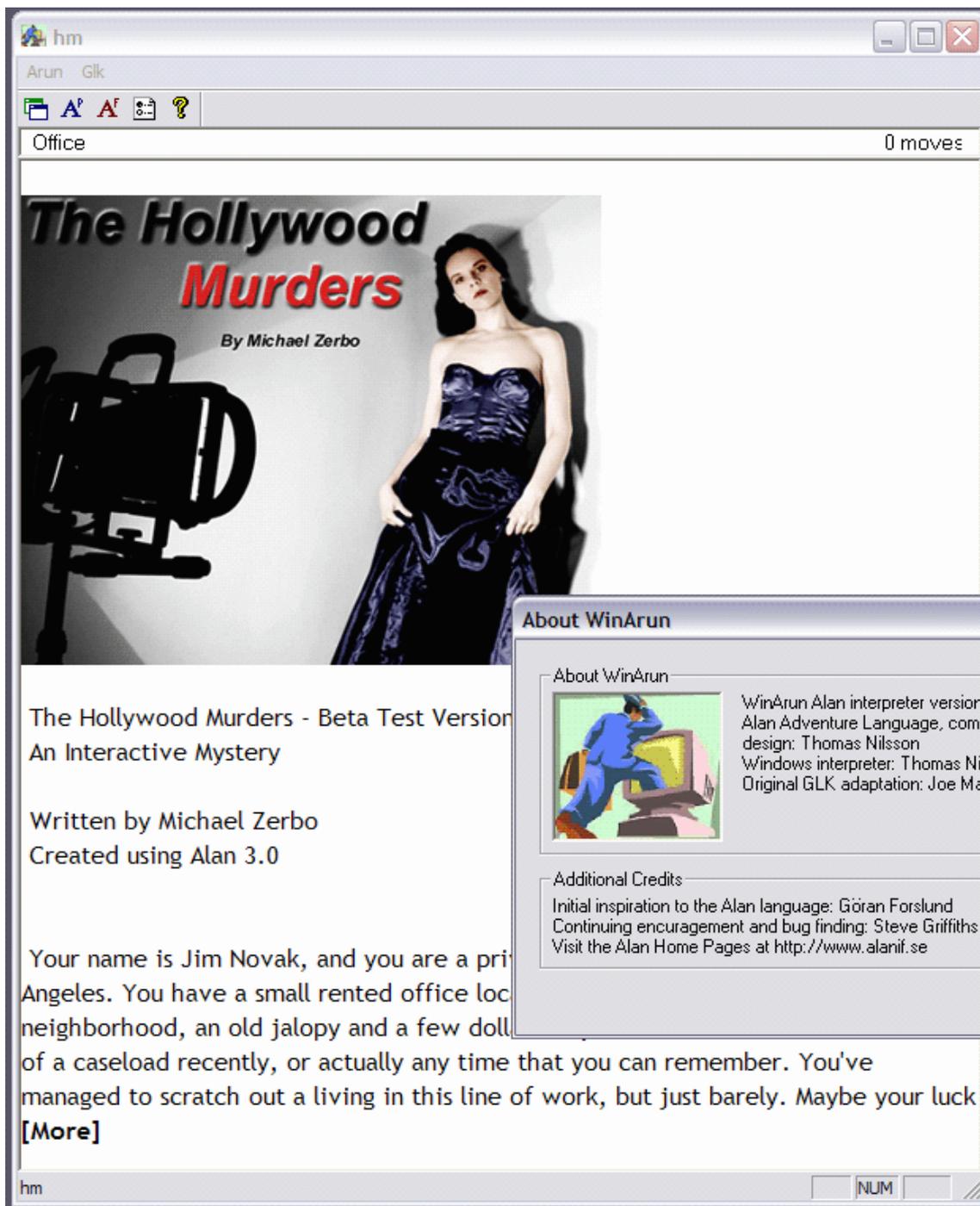


Figura 12 – Interpretador do Alan (WinArun)

2.4.3 Interpretador do Hugo

O interpretador do Hugo suporta Windows, Linux, Mac. Podemos observar um exemplo de um jogo feito com o Hugo, o Future Boy que foi nomeado para 9 Prêmios Xizzy (melhor jogo, melhor escrita, melhor história, melhor cenário, melhores puzzles, melhores *NPCs*, melhor puzzle individual, melhor *NPC* individual, melhor uso do meio) na Figura 13.

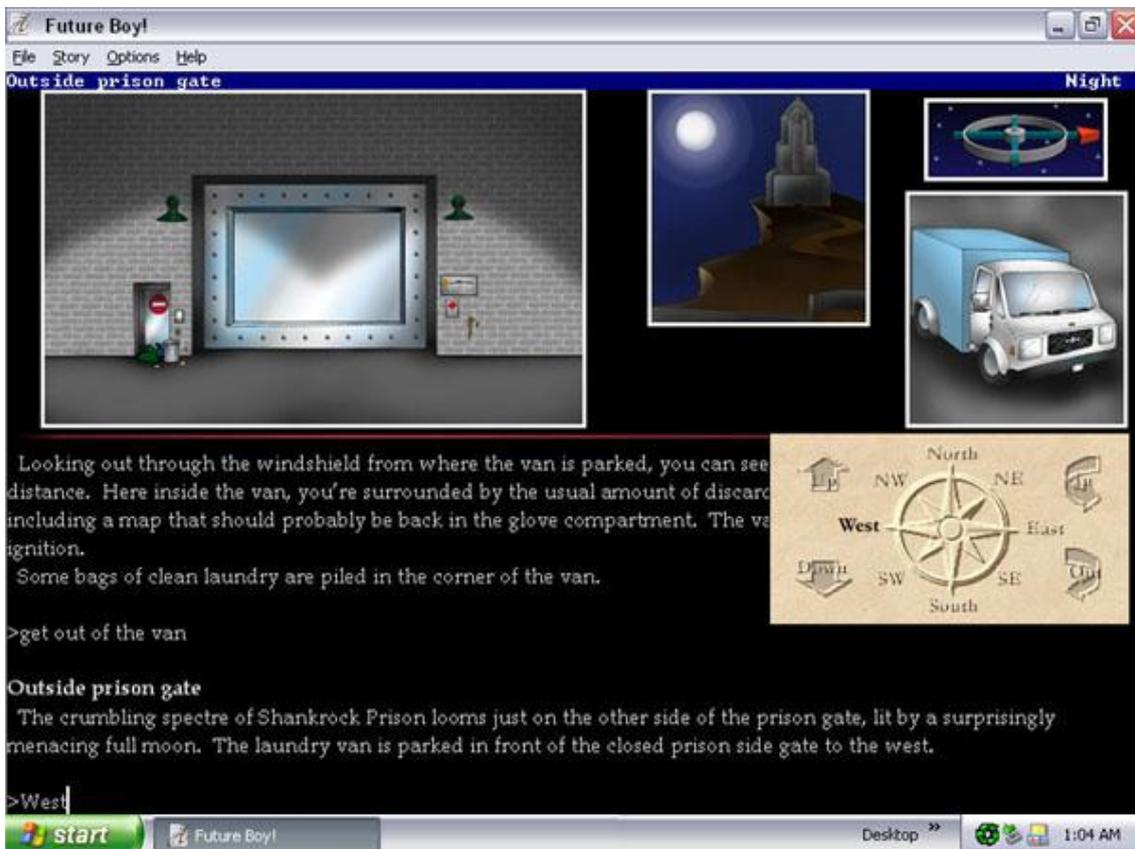


Figura 13 – Interpretador do Hugo

2.4.4 Interpretador do Inform 7

O Inform 7 não tem nenhum interpretador em particular, para executar os seus jogos pode ser usado qualquer um destes interpretadores:

- Glulxe
- Git
- Zoom
- Spatterlight
- Frotz
- Gargoyle.

Este conjunto de interpretadores tem características muito semelhantes, sendo as suas principais diferenças as plataformas onde podem ser executados:

- Glulxe:
 - Suporta Windows
- Git:
 - Suporta Windows
- Gargoyle:
 - Suporta Windows, Linux e Mac.
- Spatterlight:
 - Suporta Mac
- Zoom:
 - Suporta Linux e Mac
- Frotz:
 - Suporta Windows e Linux

2.4.5 Interpretador do Quest

O Quest tem 2 interpretadores, um deles pode ser instalado no computador (apenas suporta Windows) e o outro é um interpretador online, que se encontra numa fase beta. Com este interpretador online é possível jogar em qualquer sistema operativo. Na Figura 14 podemos ver um exemplo de um jogo a ser executado no interpretador offline, embora a interface do interpretador online seja muito semelhante à da Figura 14. Uma das grandes vantagens desta interface é o facto de não obrigar o jogador a decorar um conjunto de palavras para poder interagir com o interpretador.

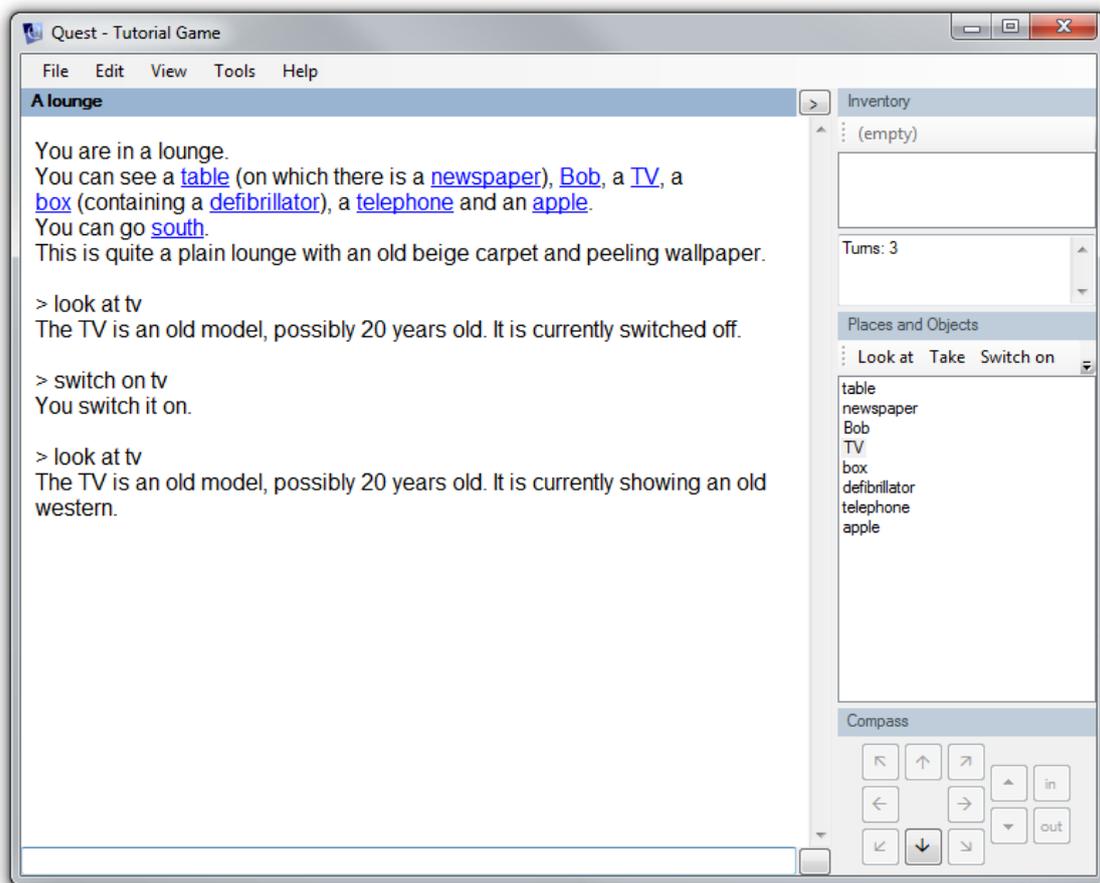


Figura 14 – Interpretador do Quest

2.4.6 Interpretador do TADS

O interpretador do TADS suporta Windows, Linux e Mac. Suporta imagens, animações, efeitos sonoros, e outras características de multimédia referidas anteriormente sobre o seu editor (ver Figura 15).

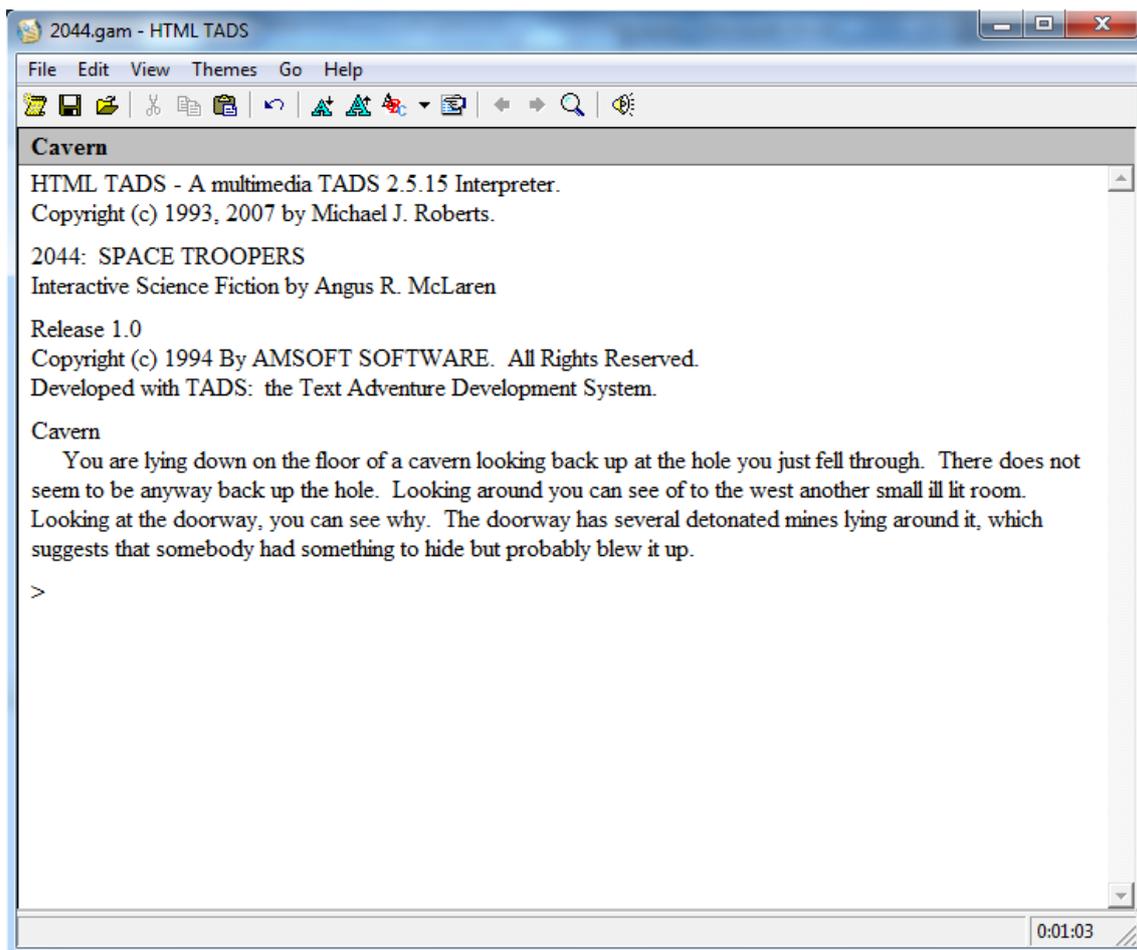


Figura 15 – Interpretador do TADS

2.4.7 Comparação dos Editores

Todas estas ferramentas têm servido de ponte para a criação de várias obras literárias, embora algumas das suas limitações tornam o seu uso bastante penoso. A maior diferença que as separa é a abstracção da programação, apenas o Adrift e o Quest conseguem proporcionar uma interface que permite ao autor criar uma história sem ter conhecimentos na área de programação e o Inform 7 consegue suavizar bastante essas noções. A tabela que se segue resume as principais características dos editores.

Sistema	Adrift	Alan	Hugo	Inform 7	Quest	TADS
Imagens	Suporta	Suporta	Suporta	Suporta	Suporta	Suporta
Musica	Não Suporta	-	Suporta	-	Suporta	Suporta
Efeitos Sonoros	Não Suporta	-	Suporta	-	Suporta	-
Vídeos	Não Suporta	-	Suporta	-	Suporta	Não Suporta
Linux	Suporta	Suporta	Suporta	Suporta	Não Suporta	Suporta
Mac	Não Suporta	Suporta	Suporta	Suporta	Não Suporta	Suporta
Windows	Suporta	Suporta	Suporta	Suporta	Suporta	Suporta
Gestor de Narrativas	Não Suporta	Não Suporta	Não Suporta	Não Suporta	Não Suporta	Não Suporta
Interface amigável	Suporta	Não Suporta	Não Suporta	Suporta	Suporta	Não Suporta

Tabela 1 – Principais diferenças entre os editores

Os interpretadores são todos muito semelhantes, alguns têm características únicas como o interpretador do Adrift que mostra o mapa do jogo ao jogador, ou o

interpretador do Quest que possibilita uma interacção não só escrita mas também através do uso do rato. Apesar destas diferenças, nenhum deles suporta gestores de narrativas.

2.5 Gestores de Narrativas

Um dos maiores obstáculos nos jogos que giram á volta da narrativa é a forma de moldar a experiência do jogador à estrutura da narrativa sem lhe tirar a sensação de livre arbítrio no jogo. O jogo ideal seria aquele onde o jogador pudesse explorar tudo ao seu próprio ritmo, que o permitisse sentir que tem impacto na história, ao contrário da maioria dos jogos que tende a criar narrativas levemente ramificadas onde são alternados períodos onde o jogador pode interagir e tomar decisões, com períodos de *cutscenes*[17][35]. As *cutscenes*, por vezes referidas com outros termos como *cinematics* ou *in-game movies*, são sequências sobre as quais o jogador não tem controlo, sendo usadas para avançar no enredo.

Como já foi referido, em muitos casos uma simples história pode tornar-se num labirinto sem saída óbvia para o jogador e para combater este problema é necessária a presença de um gestor de narrativas[18, 19, 20, 21, 22, 23, 24, 25, 26, 27].

No jogo, o gestor de narrativas controla a actividade do jogador através de pontos de enredo. Os pontos de enredo são simplesmente momentos importantes na narrativa onde é verificado o progresso do jogador. Quando as condições de um ponto de enredo se reúnem, o gestor de narrativas reconhece-o e recorre a uma função de avaliação personalizada pelo autor, para determinar a medida que deve tomar. As condições para um ponto de enredo ser reconhecido, podem ser por exemplo, o jogador estar num certo lugar e ter consigo um objecto específico.

Para tomar estas decisões o gestor de narrativas projecta os possíveis caminhos que o jogador pode fazer no jogo e recorre à função de avaliação para descobrir qual dos caminhos maximiza a experiência de jogo. As medidas do gestor de narrativas podem não só influenciar o comportamento do jogador, como também podem influenciar o mundo do jogo, como por exemplo, bloquear um caminho forçando o jogador a optar por outro, ou alterar o local onde o jogador apanha um determinado objecto. Em certas alturas pode até acontecer que a melhor acção que o gestor de narrativas possa tomar seja abster-se de qualquer medida não interferindo no jogo.

A ideia de um gestor de narrativas foi primeiramente proposta por Bates[18] em 1992 e desenvolvida por Weyhrauch[19] na sua dissertação em 1997. Weyhrauch desenvolveu uma função de avaliação para a história do jogo de ficção interactiva *Tea*

for Three e desenvolveu um algoritmo de procura em árvore que aplicava a função de avaliação. Para um gestor de narrativas a função de avaliação não pode ser apenas um algoritmo de procura adversária, pois o jogador não sabe que está a jogar com o gestor de narrativas e não tem intenções de o vencer. Tendo isto em conta Weyhrauch propôs o algoritmo avg-max que é em tudo semelhante ao minmax, com excepção de que não interessa prever a pior jogada, aliás todas as jogadas do jogador vão ter o mesmo peso, logo é calculada a média em vez do valor mínimo. Os seus testes foram efectuados com utilizadores simulados que executavam acções aleatoriamente. O seu trabalho validou que é possível criar uma função, onde o autor consegue definir a estética da história, isto é, um conjunto de arcos narrativos definidos pelo autor como sendo o percurso ideal para uma boa experiência de jogo.

Em 2003 Michael Mateas e Andrew Stern [20][21][16] criaram um gestor de narrativas baseado em *beats*, para o drama interactivo *Façade*. Neste caso os *beats* são a menor unidade de alteração no valor dramático, em que os valores dramáticos correspondem a atributos dos personagens e da história, como por exemplo, amor, confiança e tensão. Mas este gestor de narrativas não se enquadra em grandes mundos simulados (como por exemplo o Grand Theft Auto), apenas em pequenos mundos onde todas as acções no mundo da narrativa contribuem para a história.

Em 2004 Ari Lamstein e Michael Mateas [20] pegaram no trabalho de Weyhrauch para responderem a uma questão que tinha ficado pendente, “seria possível um gestor de narrativas guiar um jogador de modo eficiente ao longo da narrativa de um jogo?”. Para isto usaram apenas uma parte da narrativa do jogo Anchorhead para realizar os testes. O facto de Weyhrauch exigir uma pesquisa em profundidade completa, tornou-a intratável em tempo real. Porém ao tratar o gestor de narrativas como um problema de pesquisa em árvore é possível usar muitas técnicas já existentes para colmatar este problema, como por exemplo a pesquisa por amostragem, a pesquisa em profundidade limitada, entre outras.

Em 2006 Mark J. Nelson et al. [22] propuseram um sistema de aprendizagem por reforço, para um gestor de narrativas baseado na optimização declarativa. Neste projecto eles dão ao autor a possibilidade de definir o que constitui uma boa narrativa (ou seja, a estética da narrativa), usando um conjunto de pontos de enredo e uma função de avaliação. Os pontos de enredo são avaliados pelo autor que lhes atribui uma pontuação numérica deixando o gestor de narrativas calcular quais são os melhores arcos da narrativa que o jogador deve percorrer. Para tomar estas decisões, o gestor de narrativas tem que optimizar o uso das suas acções consoante os pontos de enredo e a função de avaliação. Após utilizarem o seu sistema no jogo Tea for Three e numa parte

do jogo Anchorhead chegaram á conclusão que obtinham melhores resultados em jogos onde o mundo virtual era maior (Anchorhead).

Em 2009 Manu Sharma et al.[23][24] criaram um gestor de narrativas baseado em casos chamado C-DraGer. A arquitectura do C-DraGer consiste em 3 módulos: motor de jogo, modelação do jogador e gestor de narrativas. O motor de jogo é responsável por executar o jogo e interagir com o jogador, a modelação do jogador é responsável por analisar as suas acções e criar um modelo de interesses deste e o gestor de narrativas, influencia o progresso da narrativa no jogo tornando-a mais apelativa ao jogador. Na parte da modelação do jogador o C-DraGer usa raciocínio baseado em casos, verificando em todos os casos que já conhece qual se assemelha mais com as características do jogador actual, assumindo que jogadores com características semelhantes de jogo também vão percorrer caminhos semelhantes e cria um modelo de interesses do jogador. No final de cada jogo, cada jogador classifica os pontos de enredo que percorreu, actualizando assim a modelação do jogador. As decisões do gestor de narrativas não vão depender apenas da modelação do jogador, mas também de parâmetros usados numa função de avaliação, definidos pelo designer do jogo. Neste caso, também influenciados por Weyhrauch optaram pelo avg-max para função de avaliação.

Capítulo 3

Desenvolvimento do FIGN

Neste capítulo, vamos começar por descrever em pormenor a arquitectura do FIGN e seguir iremos apresentar o editor do FIGN que permite que se crie um jogo sem ser necessário ter conhecimentos de programação. Criámos um jogo com o editor que vai servir para ilustrar os vários componentes do FIGN que também será usado para ilustrar o editor.

3.1 Representação do Jogo e do Gestor de Narrativas

3.1.1 Representação do Jogo

Um jogo de ficção interactiva em texto não é composto por simples conjuntos de parágrafos interligados, mas sim um mundo virtual constituído por zonas, objectos e personagens. Normalmente o jogador interage neste mundo através de comandos em texto, desloca o seu personagem através da direcção dos pontos cardeais, como por exemplo norte, sul, este, oeste e por vezes também através dos colaterais, nordeste, sudeste, noroeste e sudoeste.

Ele pode interagir com objectos através de comandos em texto semelhantes a “apanhar livro”, adicionar objectos ao seu inventário para mais tarde utilizar na resolução de puzzles e interagir com outros personagens através de comandos como “falar com (nome do personagem)”. Cada objecto e personagem pode ter vários níveis de interacção, através do comando “examinar” é possível recolher mais informação e posteriormente aumentar as possibilidades de interacção.

Descrição de um exemplo de jogo

Para este trabalho foi criado um jogo para exemplificar algumas das funcionalidades do gestor de narrativas. A história é simples, um arqueólogo vai visitar a aldeia fantasma de Colmeal em busca da razão para a desertificação da aldeia e na esperança de encontrar algum vestígio de ruínas romanas. Ao longo da história o arqueólogo começa a estranhar algumas contradições que vai encontrando, até que descobre a verdadeira razão da desertificação da aldeia quando passa a noite numa das casas abandonadas. Durante a noite ele descobre que na realidade a aldeia não estava deserta, toda a população tinha sido transformada em zombies.

3.1.2 Arquitectura do Interpretador do FIGN

É impossível falar do gestor de narrativas sem falar do interpretador do FIGN, para a sua criação optámos por uma arquitectura semelhante à do C-DraGer, sendo constituído por 3 módulos principais: o motor de jogo, o modelo do jogador e o gestor de narrativas (ver Figura 16). O motor de jogo é responsável por executar o jogo e interagir com o jogador, o modelo do jogador é responsável pela avaliação do progresso do jogador ao longo do seu percurso no jogo e o gestor de narrativas é responsável por tornar a história mais apelativa ao jogador.

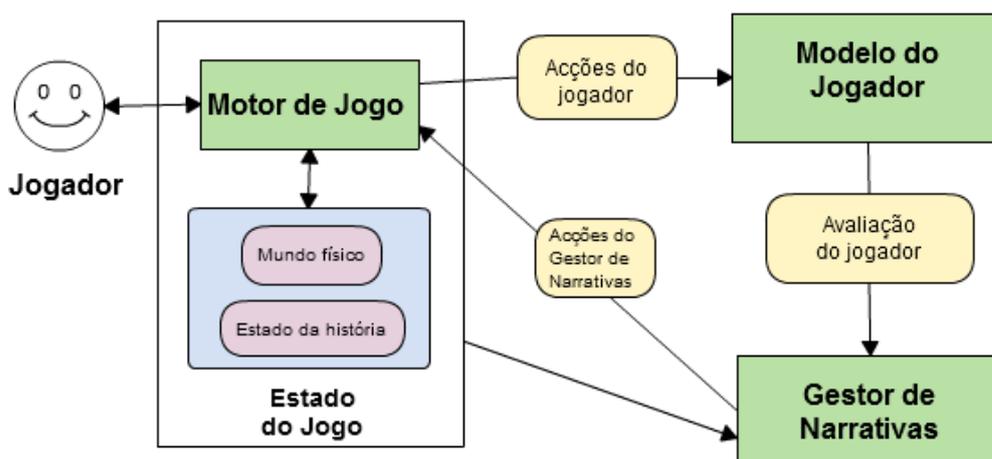


Figura 16 – Arquitectura do interpretador do FIGN

3.1.3 Motor de Jogo

O motor de jogo é responsável por:

- Executar o jogo.
- Interagir com o jogador sempre que este realiza alguma acção.
- Actualizar o estado do jogo.
- Actualizar o modelo do jogador sempre que este executa uma acção.
- Informar o gestor de narrativas sempre que o jogador muda de estado, para que o gestor de narrativas verifique se o novo estado em que o jogador se encontra corresponde a um ponto de enredo.
- Executar as acções do gestor de narrativas.

3.1.4 Modelo do Jogador

O modelo do jogador é simples: verifica quantas acções o jogador deve fazer e quantas é que realmente efectuou, calculando assim a percentagem do progresso realizado. O progresso do jogador é medido apenas nas zonas por onde ele já passou, por exemplo, se o jogador passa por 2 zonas, deve fazer 5 acções mas apenas realizou 2, então o seu progresso é de 40%.

O progresso do jogador é inversamente proporcional á necessidade de ser ajudado pelo gestor de narrativas, por isso, quando o progresso é baixo, como por exemplo 40%, o gestor de narrativas vai ter uma probabilidade de actuar maior, quando é alto, vai ser menos provável ajudar, já que o jogador deve estar no bom caminho. Deste modo o gestor de narrativas controla o progresso do jogador através do modelo do jogador e dos pontos de enredo.

3.1.5 Pontos de Enredo

Os pontos de enredo são simplesmente momentos na narrativa onde é verificado o progresso do jogador e o autor pode definir o número de pontos de enredo que quiser. Estes pontos são associados a estados de jogo. Tendo em conta que cada estado de jogo

é composto por uma zona e pelos itens que o jogador tem consigo no inventário nesse momento, é fácil de detectar o jogador quando entra num estado de jogo que tem um ponto de enredo associado. No entanto, a maneira como são usados nos artigos referidos no capítulo 2 obrigam o autor da narrativa a definir um grande número de pontos e a definir quais são as sequências de pontos de enredo que podem proporcionar uma experiência mais agradável ao jogador. Por sua vez o gestor de narrativas procura guiar o jogador pelas melhores sequências colocando o jogador a seguir “um rasto de migalhas” como na história de Hansel e Gretel[36].

Para o nosso protótipo decidimos tomar uma abordagem ligeiramente diferente. Quisemos automatizar estes passos para retirar trabalho ao autor. Para começar, não deixamos o autor escolher sequências de pontos de enredo, pois acreditamos que quando alguém cria uma história não linear, com vários finais, não tem intenções de criar partes do jogo que não queira que o jogador explore. Deste modo o autor apenas vai criar pontos de enredo para momentos do jogo onde o jogador possa fugir ao rumo da narrativa, tornando a criação de pontos de enredo muito menos penosa.

Na Figura 17 podemos ver os pontos de enredo que existem no jogo criado para este projecto.

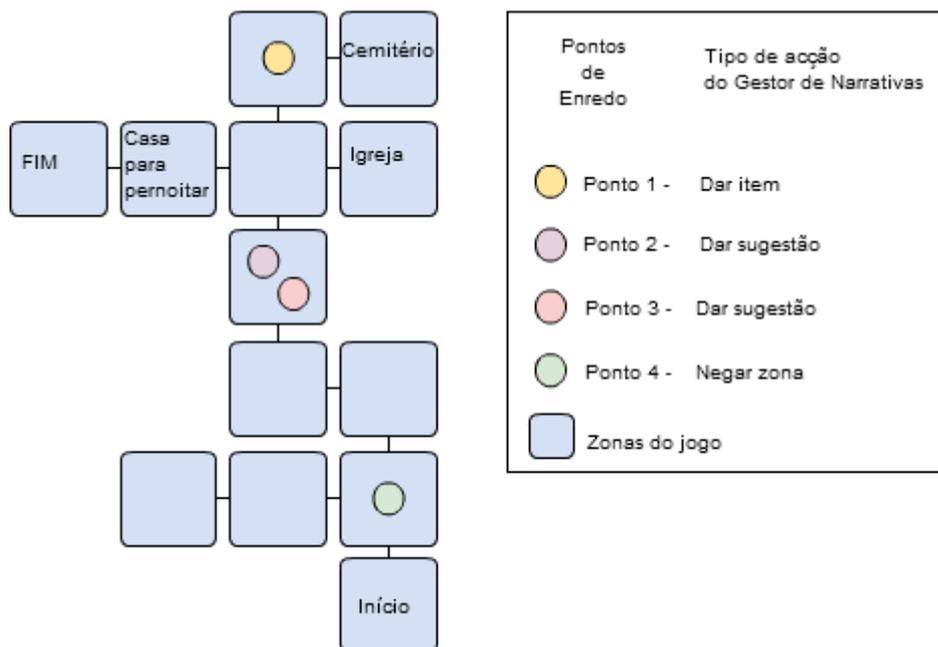


Figura 17 – Mapa do jogo criado para demonstrar o uso do FIGN

Resumindo a história, o jogador é um arqueólogo, começa na zona “Início” e tem que percorrer a aldeia á procura de pistas para a sua desertificação. Ao percorrer a aldeia o jogador encontra pistas na Igreja e no Cemitério, e após encontrar as pistas pode pernoitar numa casa em frente á Igreja, onde é atacado ao meio da noite por zombies e o jogo acaba.

No percurso ideal, o jogador completava o jogo e o gestor de narrativas nem precisava de actuar, independentemente do jogador passar ou não passar pelos pontos de enredo. Mas como o jogador nem sempre segue o caminho ideal, por vezes perde-se, então é necessário ser guiado.

Cada ponto de enredo tem um conjunto de regras, estas regras podem ser por exemplo, zonas que o jogador tem que visitar, zonas por onde ele ainda não passou, itens que ele tem que ter obrigatoriamente consigo. Quando estas regras são reconhecidas o gestor de narrativas verificar se deve actuar ou não.

O primeiro ponto de enredo com que o jogador se depara é o Ponto 4, que lhe vai negar a passagem para a zona que está à esquerda do ponto de enredo. Este ponto foi colocado com o único e exclusivo propósito de demonstrar o gestor de narrativas a negar uma zona, como já mencionei anteriormente, este tipo de acções deve ser evitado.

O jogador continua a percorrer a vila e passa por 2 pontos de enredo. O Ponto 2 é reconhecido se o jogador apanhar um papel que está na Igreja, ainda não tiver visitado o Cemitério e estiver na zona onde está o Ponto 2 (caso alguma destas regras não se verifique o ponto de enredo é ignorado). Se todas estas regras se verificarem e o gestor de narrativas decidir que deve actuar, vai dar a sugestão ao jogador de ir verificar o cemitério. Qual a razão deste ponto de enredo estar colocado nesta zona em particular? Simples, é a zona onde o jogador começa a fugir do rumo ideal para ter uma boa experiência neste jogo. O Ponto 3 é semelhante: se o jogador já esteve no cemitério, tem no seu inventário apenas um relógio e chegou à zona do Ponto 3, o gestor de narrativas vai determinar se deve actuar ou não, e caso opte por actuar vai sugerir ao jogador que visite a Igreja.

O Ponto 1 é reconhecido se o jogador entrar no Cemitério mas não apanhar o relógio que lá se encontra e passar pela zona do Ponto 1. Se o gestor de narrativas achar que deve actuar, dá o item ao jogador. Na secção 3.1.6 é possível ver a criação deste ponto de enredo no editor.

3.1.6 Gestor de Narrativas

O Gestor de Narrativas vai ter o papel de guiar o jogador, do melhor modo possível, para lhe proporcionar uma agradável experiência de jogo. Quando chega a altura de tomar uma decisão, isto é, o jogador chega a uma zona onde se encontra um ponto de enredo e todas as suas condições são satisfeitas, é determinada a melhor opção a tomar. Independentemente da melhor opção ser ajudar o jogador com uma acção ou não interferir, o gestor de narrativas nunca mais vai voltar a considerar tomar qualquer medida nesse ponto de enredo. Para tal dispõe de três acções que cumprem propósitos diferentes:

- Dar uma sugestão ao jogador.
- Dar um objecto ao jogador.
- Negar-lhe o acesso a uma zona, quer de modo temporário quer de modo permanente.

Fornecer uma sugestão - A acção que consiste em dar uma sugestão não tem qualquer impacto no estado do jogo mas sim na cabeça do jogador. Este tipo de acção consiste em influenciar a probabilidade do jogador escolher um caminho em vez de outro, influenciando assim o percurso que este vai tomar na narrativa. Estas acções permitem manipular o jogador de um modo menos intrusivo.

Negar uma zona – A acção de negar uma zona, quer seja de modo temporário, quer seja de modo permanente, deve ser usada com atenção, porque das três acções, esta é a que tem um impacto mais negativo no jogador caso ele se aperceba que está a ser manipulado. Nos testes do C-DraGer com vários jogadores, foi possível confirmar que este tipo de acção foi a que teve um impacto mais negativo[23]. O bloqueio do acesso a certas zonas pode ser útil em jogos em que o mapa é grande e labiríntico. Por um lado os bloqueios passam mais despercebidos, por outro, pode facilitar o percurso do jogador.

Fornecer um item – O gestor de narrativas só pode fornecer itens que existam no jogo e que o jogador não tenha conseguido ainda alcançar. Esta acção exige cuidados porque há acções do jogador que não foram concretizadas e que se o fossem levariam a

obter os itens dados pelo gestor de narrativas. Este problema pode causar inconsistências no jogo e por consequência estragar a experiência do jogador. Tendo em conta isto, cada vez que o gestor de narrativas fornece um item ao jogador, este vai indicar ao motor de jogo que é preciso procurar qual era a acção original onde o item era fornecido e vai alterá-la. A alteração consiste em alterar o texto da acção original e alterar o tipo de acção, ou seja, em vez de continuar a ser uma acção que fornece o item ao jogador, passa a ser uma acção meramente informativa, que apenas vai mostrar o novo texto.

Na figura 18 podemos ver um exemplo da criação de um ponto de enredo e da acção que o gestor de narrativas pode ter nesse ponto, utilizando o editor do FIGN.

Se esta acção for usada pelo Gestor de Narrativas vai mostrar ao jogador o texto “Um cão apareceu do nada a correr e deixou cair qualquer coisa a meus pés. Baixei-me para apanhar e noto que é um relógio antigo. Guardei-o na mochila.”, vai colocar o item relógio no inventário do jogador e vai alterar o texto da acção original para “Afinal era só um pedaço de vidro, embora a cova parece ter a terra remexida, estranho...”.

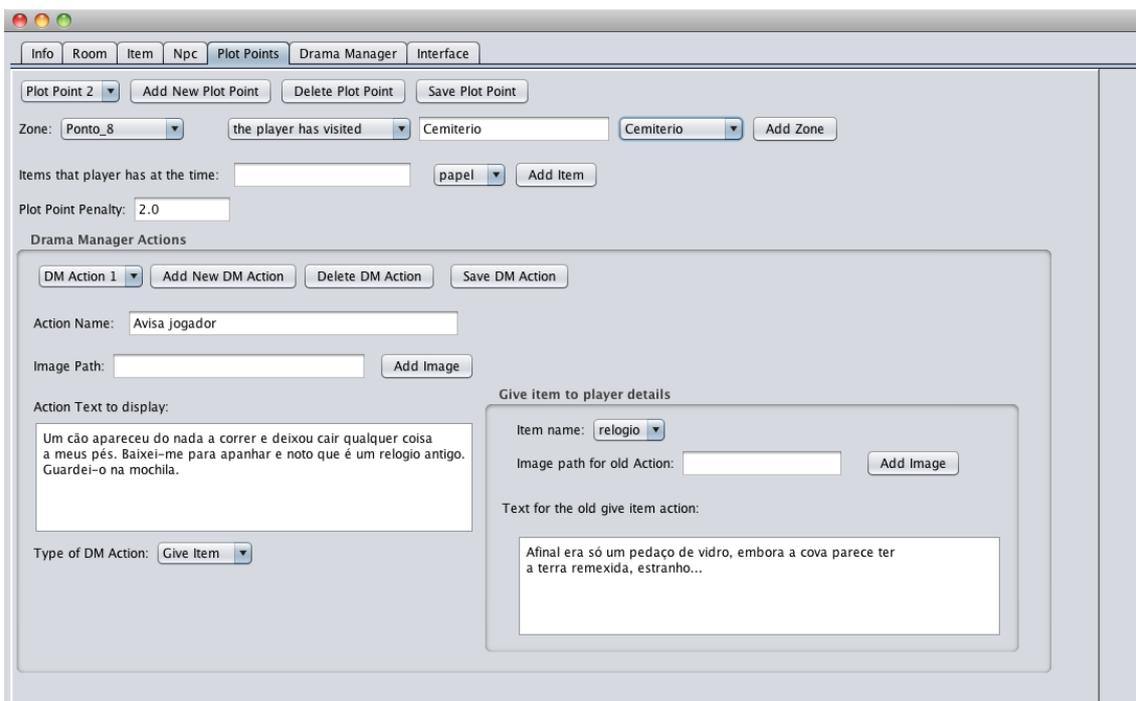


Figura 18 – Exemplo da criação de um ponto de enredo assim como a respectiva acção para o Gestor de Narrativas

Neste exemplo também é possível observar o estado de jogo onde o ponto de enredo vai ser inserido. Neste caso, o ponto de enredo vai ser associado ao estado de jogo composto pela zona chamada Ponto_8 e inventário vazio.

Embora seja pertinente que o gestor de narrativas ajude o jogador, é importante impedir que ele se torne demasiado manipulativo, para isso é aplicada uma penalidade às suas acções. Neste exemplo podemos observar o campo Plot Point Penalty, onde o autor pode definir a penalidade aplicada ao gestor de narrativas caso ele queira actuar. Esta penalidade é suavizada caso o progresso recebido do modelo do jogador seja baixo, permitindo o gestor de narrativas actuar apenas se o jogador estiver com dificuldades a progredir no jogo.

O gestor de narrativas recebe o estado do jogo e o modelo do jogador e decide qual a acção a fazer através de simulação do jogo e avaliando os estados de acordo com uma função de avaliação da qualidade do jogo.

Como já foi anteriormente mencionado, um gestor de narrativas não pode simplesmente usar um algoritmo de pesquisa adversária, pois o jogador não sabe que está a jogar com ele e não tem intenções de o vencer. Para além de que as jogadas não são alternadas, como no Xadrez, uma jogada do jogador pode envolver várias acções até encontrar um ponto de enredo e passar a ser a vez do gestor de narrativas, como por exemplo, o jogador pode ver um quadro, examinar uma gaveta, guardar um bloco de notas que lá se encontrava e só depois é que seria a vez do gestor de narrativas.

Tendo isto em conta, Weyhrauch propôs o algoritmo avg-max (médiamax), enquanto o minmax[29] se enquadra em jogos onde o jogador alterna com uma inteligência artificial e onde esta procura a pior jogada do jogador (ver Figura 19), o médiamax procura a média das jogadas que o jogador pode realizar (ver Figura 20) porque não é possível determinar qual delas o jogador vai escolher.

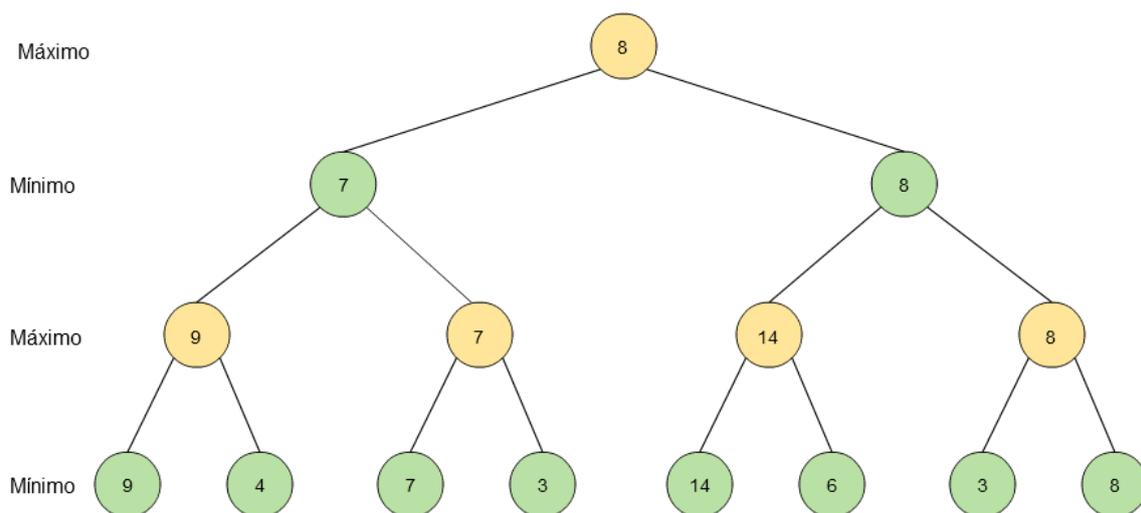


Figura 19 – Exemplo do algoritmo minmax

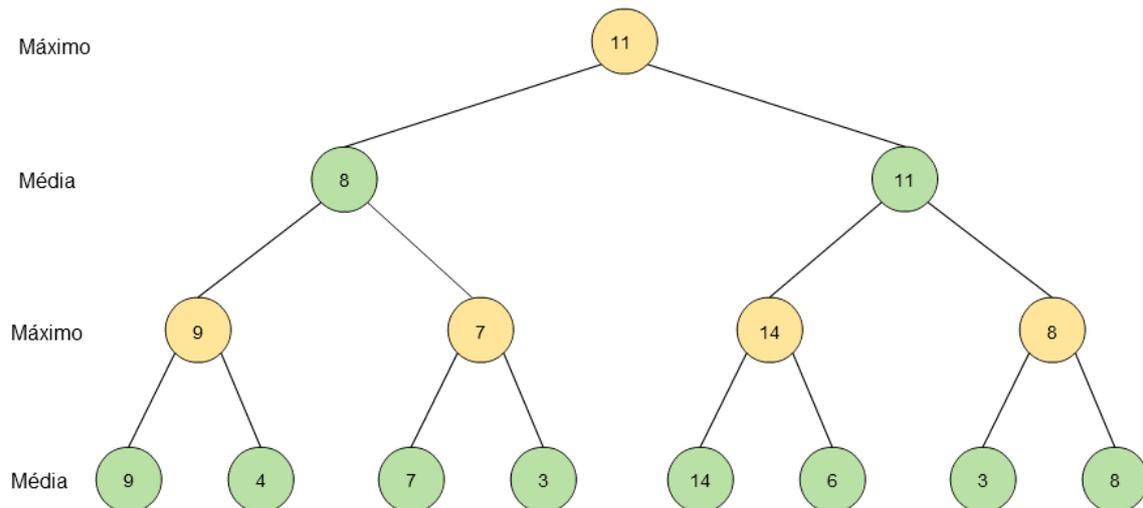


Figura 20 - Exemplo do algoritmo médiamax

Neste caso o algoritmo médiamax tem o seguinte comportamento:

Função médiamax (Nó)

$X = \text{Max} (\text{Nó})$

Média (Nó)

➔ O nó tem sucessores?

Não – Retorna o valor para o nó pai

Sim – Enquanto tiver sucessores $X = \text{Max} (\text{Nó Sucessor})$

$X = \text{média do valor dos nós sucessores}$

Retorna X para o nó pai

Max (Nó)

➔ O nó tem sucessores?

Não – Retorna o valor para o nó pai

Sim – Enquanto tiver sucessores $X = \text{Média} (\text{Nó Sucessor})$

$X = \text{valor mais alto dos nós filhos}$

Retorna X para o nó pai

Mas, ao contrário de Weyhrauch, restringimos limite de profundidade da árvore de pesquisa onde ele vai actuar. Deste modo o tempo gasto pelo gestor de narrativas não é perceptível ao jogador, caso contrário o jogo ficaria parado enquanto o gestor de narrativas estivesse à procura da melhor acção para tomar.

Antes da aplicação do algoritmo médiamax, o gestor de narrativas começa por criar uma árvore de pesquisa, com profundidade limitada pelo valor definido pelo autor multiplicado por 5 (no separador Drama Manager do editor) e depois aplica o médiamax sobre essa árvore.

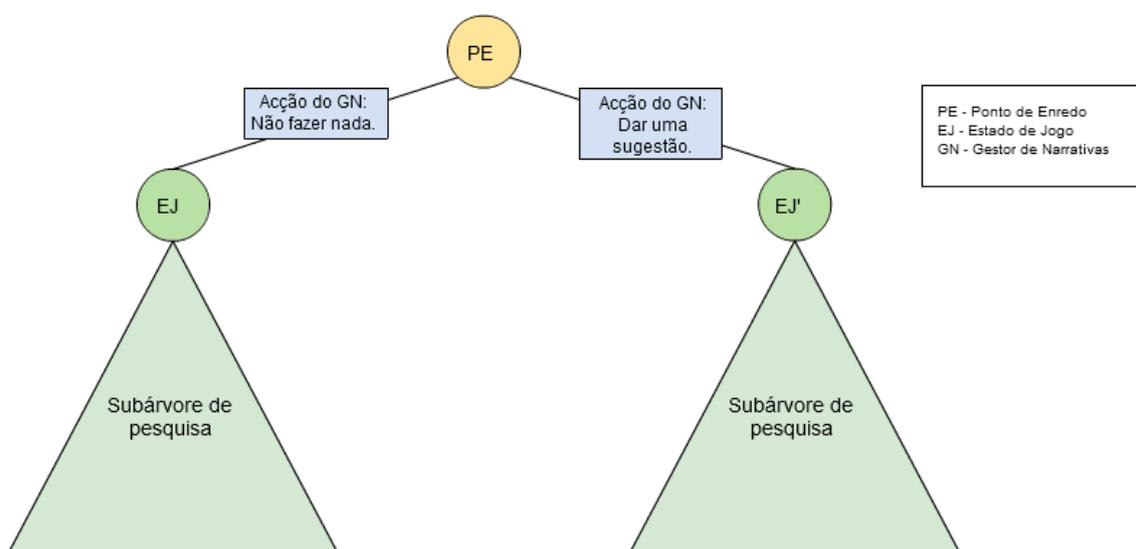


Figura 21 – Criação da árvore de pesquisa

A criação da árvore de pesquisa começa, com a criação do nó raiz correspondente ao ponto de enredo que foi reconhecido. Os filhos do nó raiz, vão ser os estados resultantes da acção do Gestor de Narrativas, que no exemplo da Figura 21 correspondem aos estados EJ e EJ'. Após este passo, o resto da árvore é criada em duas fases.

Na primeira fase é aplicada uma função sobre o nó EJ e sobre o nó EJ' que vai procurar que ligações esses estados de jogo têm com outros estados, continuando este processo N vezes, onde N é o valor definido pelo autor (no separador Drama Manager do editor). Na Figura 22 encontra-se um exemplo para N=3, onde apenas existem 3 níveis de profundidade. A função pegou no estado EJ e procurou que ligações este estado tinha, e fez o mesmo para os estados sucessores.

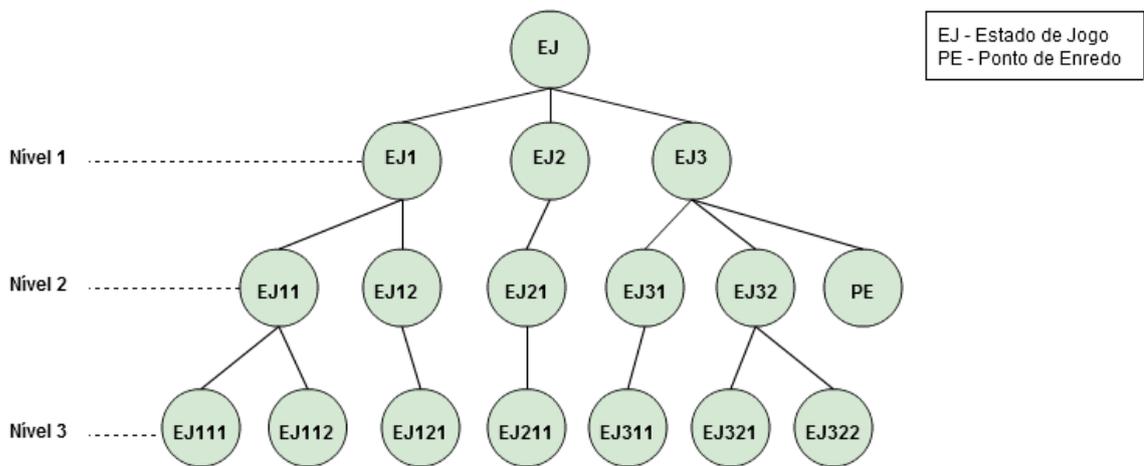


Figura 22 – Primeira fase da criação das subárvores

Quando a função chega à profundidade definida pelo autor, neste exemplo $N=3$, são criados nós que contêm os vários caminhos possíveis, isto é, vai criar um nó com o caminho EJ, EJ1, EJ11, EJ111, outro nó com o caminho EJ, EJ1, EJ11, EJ112, para todos os caminhos. Após esta etapa, começa a fase 2, que consiste em aplicar a fase 1 mais 4 vezes, sempre utilizando o estado final presente nestes nós para continuar a procurar mais N estados, criando assim uma subárvore com a profundidade $5N$. Seguindo o exemplo ficamos com uma subárvore semelhante à da Figura 23.

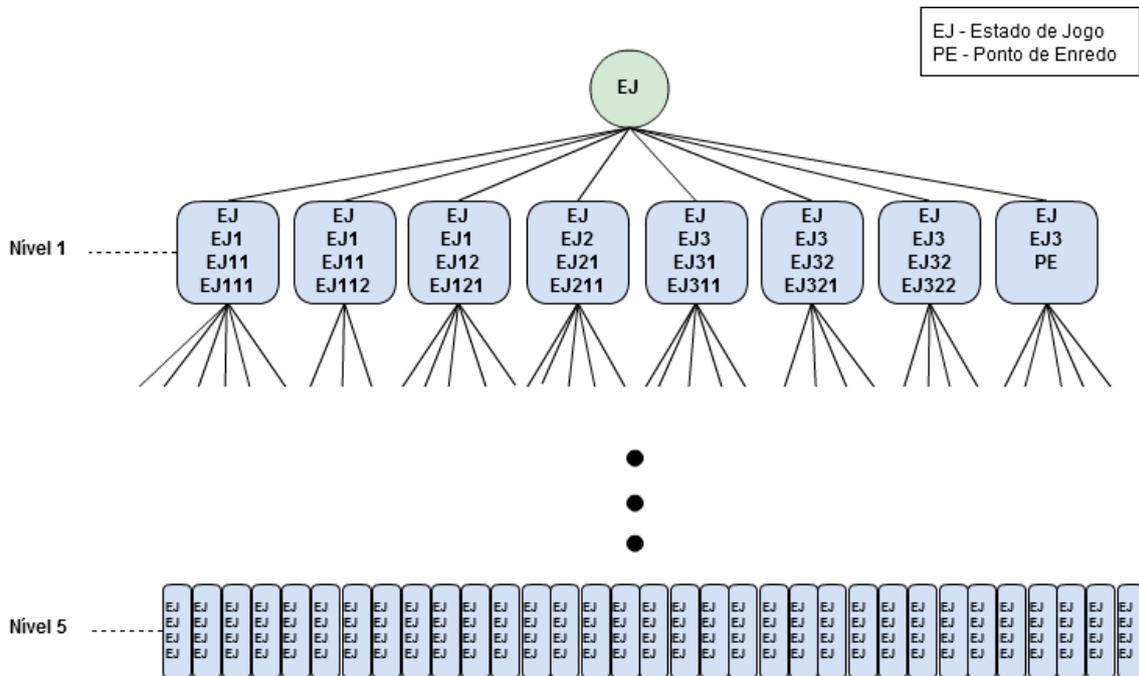


Figura 23 – Segunda fase da criação das subárvores

Durante a criação da subárvore, a cada um destes nós, é associado um valor, esse valor é calculado tendo em conta o número de estados presentes no nó e o número de itens que o jogador tem nos vários estados. Por exemplo, o valor do nó correspondente ao caminho EJ, EJ1, EJ11, EJ111 é a soma do valor destes estados, onde cada estado vale 1 e por cada item que o jogador tiver consigo nesse estado soma-se mais 1.

Todos os estados valem 1 inicialmente (antes de se verificar se têm itens), excluindo a raiz da árvore e os seus sucessores directos (EJ e EJ'), por exemplo, o estado inicial EJ vale 0, mas o seu sucessor EJ1 vale 1, mais o valor do estado anterior que neste caso é 0, mais 1 por cada item que o jogador tiver consigo. Neste caso, se o jogador não tiver itens consigo em nenhum dos estados EJ, EJ1, EJ11 e EJ111, então o valor deste nó vai ser $0+1+1+1=3$. Se por acaso o jogador tivesse 1 item no estado EJ111, o valor do nó seria $0+1+1+(1+1)=4$. No caso de encontrar um ponto de enredo (como na Figura 22), vai somar 1 ponto extra, ou seja, no caso do caminho EJ, EJ3, PE, o valor do nó vai ser $0+1+2 = 3$.

O ideal seria deixar o Gestor de Narrativas analisar todos os estados de jogo desde o ponto em que o jogador se encontra até ao final do jogo, nesse caso seria possível determinar com maior exactidão que caminhos seriam os melhores para garantir uma boa experiência de jogo. Mas infelizmente isso é impraticável, quanto mais profunda for a pesquisa, mais tempo vai consumir.

Quando estas duas fases terminam ficamos com uma árvore semelhante à da Figura 24, pronta para ser usada pelo médiamax.

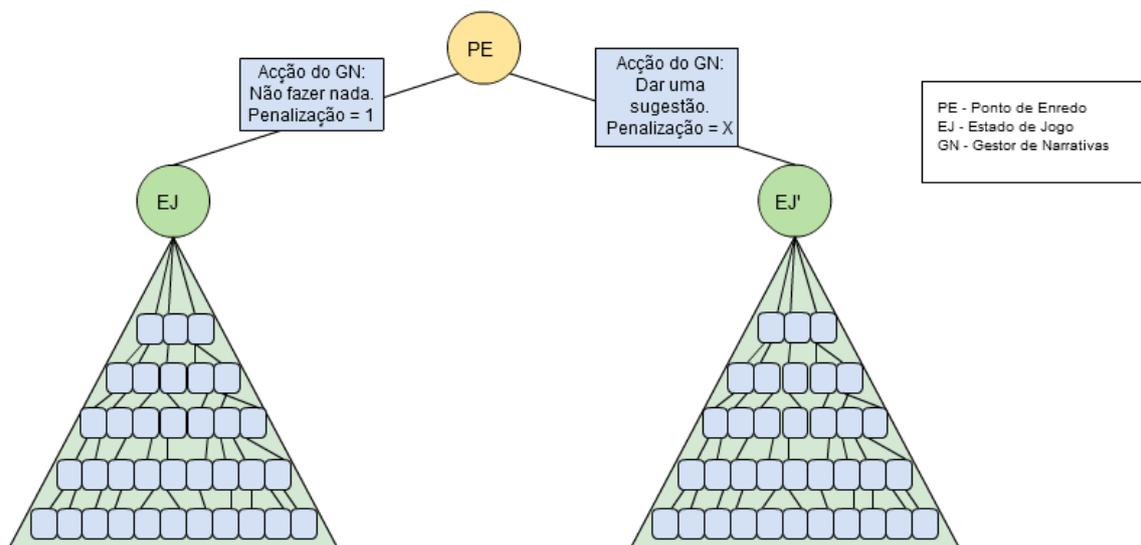


Figura 24 – Árvore de pesquisa pronta para aplicação do médiamax

Quando se aplica o *médiamax*, ele vai percorrer os nós da árvore, vai avaliar até chegar o momento onde o nó EJ e EJ' têm o valor correspondente ao melhor caminho que o jogador pode tomar e está prestes a mandar esse caminho para o nó raiz. Antes desta importante decisão, os valores são multiplicados por uma penalidade, o nó EJ que corresponde ao resultado da acção “Não fazer nada.” é multiplicado por 1, enquanto o nó EJ' que corresponde à acção “Dar uma sugestão.” é multiplicado por X (como se pode ver na Figura 24). Este X é definido pelo autor durante a criação do ponto de enredo no separador Plot Points do editor do FIGN.

Esta penalidade existe porque é importante que o gestor de narrativas participe e ajude o jogador, mas é ainda mais importante que este não se torne demasiado manipulador e comece a restringir a liberdade do jogador.

Durante a criação das subárvores pode ocorrer encontrar pontos de enredo, como está na figura 22, nesse caso em vez da função continuar a procurar os estados que estão ligados ao ponto de enredo, é aplicada a função que cria a árvore de pesquisa, mas com uma diferença, em vez de ser aplicada com uma profundidade 5N, é aplicada com base na profundidade onde foi encontrado o ponto de enredo, ou seja, se foi encontrado no nível 1 da subárvore, como está representado na Figura 23, a árvore de pesquisa do ponto de enredo encontrado só vai ter 4N de profundidade, se tivesse sido encontrado no nível 2 da subárvore, então a sua profundidade só seria 3N.

Como foi possível observar, na criação da árvore de pesquisa para a aplicação da função de avaliação *médiamax*, os nós que sucedem directamente o nó raiz vão corresponder aos estados em que o jogo se encontra após a acção do gestor de narrativas, quer seja a acção de actuar ou a de se abster. No caso da acção de actuar corresponder a dar uma sugestão, o estado resultante dele actuar é exactamente igual ao de se abster, ou seja, o facto de ele dar uma sugestão não altera em nada o estado do jogo, apenas a maneira como o jogador vê as tarefas que tinha em mente para fazer. O mesmo não se aplica às acções de negar uma zona ou adicionar um item. No caso de dar um item, o estado resultante da acção vai ser um estado com esse item, logo a parte da árvore de pesquisa que corresponde a essa acção vai ser diferente. Na negação de uma zona também, já que durante a procura essa zona vai estar bloqueada.

Num exemplo concreto na história do nosso jogo, no ponto de enredo que dá uma sugestão ao jogador caso jogador caso tenha apanhado o papel que está dentro do livro na Igreja e ainda não tenha visitado o Cemitério (é possível ver a edição deste ponto de enredo na secção 3.3, Figura 38, pág. 55). Quando o gestor de narrativas reconhece este ponto e cria a árvore de pesquisa e aplica o médiamax, vai chegar à altura onde tem o valor de não actuar multiplicado por 1 e o valor de actuar, neste caso, multiplicado por 0,6 (ver Figura 25).

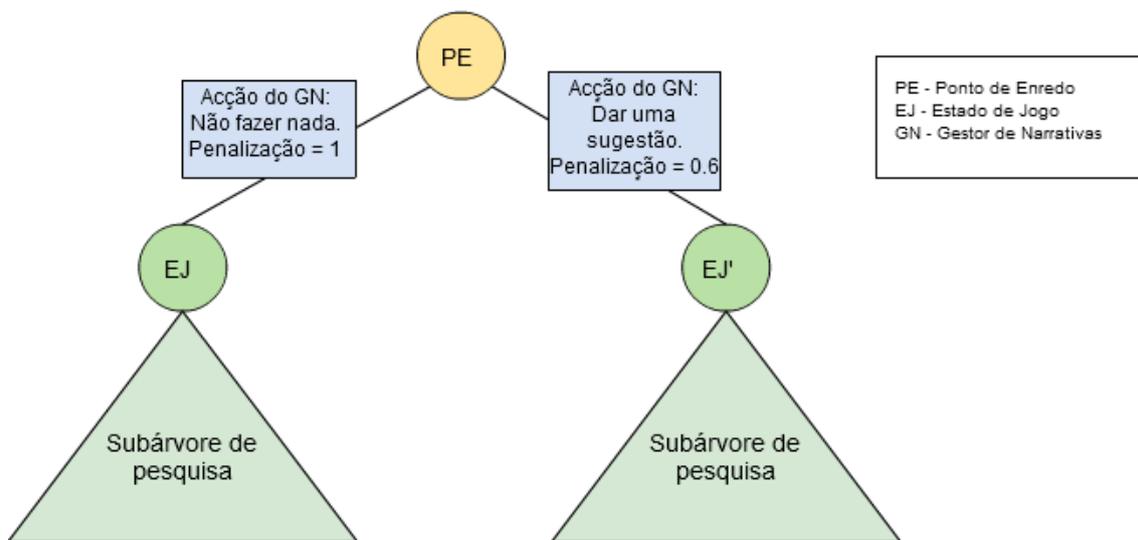


Figura 25 – Decisão do Gestor de Narrativas

Como já foi referido, a acção de dar uma sugestão não altera o estado do jogo, ambas as acções vão ter o mesmo valor (isto é, ambas as subárvores de pesquisa vão ser iguais). Sendo X um número qualquer pertencente ao conjunto dos números reais, $X*1 > X*0.6$, ou seja, neste momento a opção do gestor se abster de qualquer tipo de interferência no jogo tem mais peso. Mas também é preciso ter em conta o modelo do jogador, caso o jogador tenha um progresso fraco, por exemplo de 20%, é somado à penalidade da acção os 80% de progresso em falta, ficando $0.6+0.8 = 1.4$. Neste caso o gestor de narrativas ao fazer a acção para ajudar o jogador já não está a ser punido mas sim beneficiado (pois $X*1 < X*1.4$), optando por realizar a acção. Se o modelo do jogador não existisse, o gestor de narrativas tomava sempre a mesma decisão, assim, conforme o progresso do jogador, este verifica se é necessário ajudar ou deixa-lo seguir o seu rumo sozinho.

Na Figura 26 é possível visualizar a acção do gestor de narrativas face a este ponto de enredo.

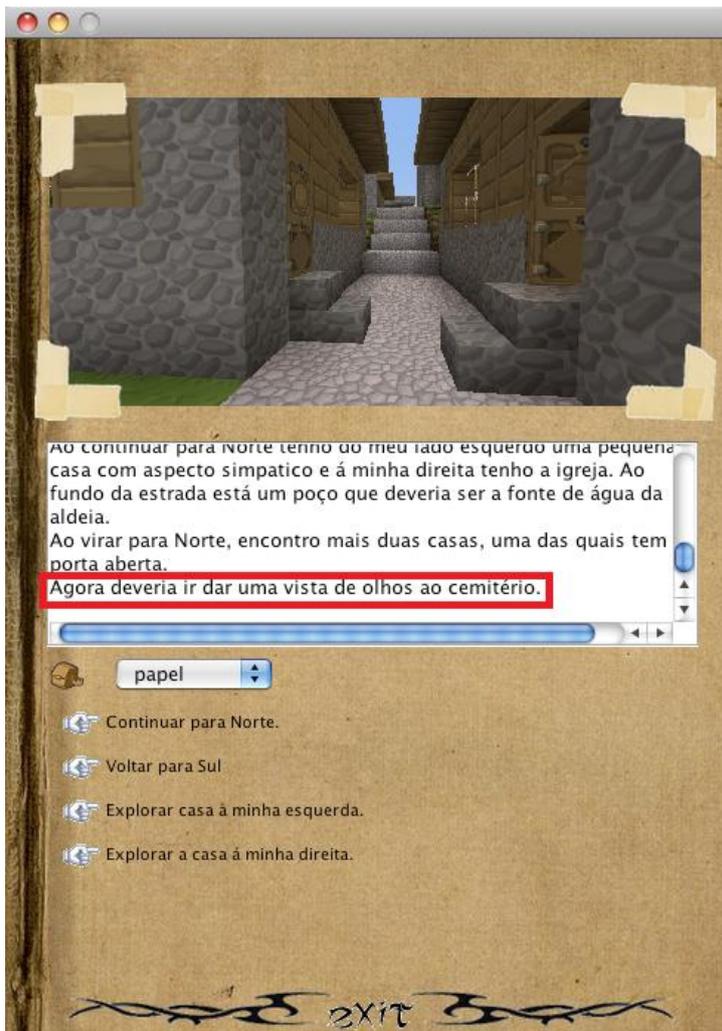


Figura 26 – Acção do Gestor de Narrativas a vermelho

Como se pode ver na Figura 26, a interface do interpretador é muito simples, é constituída por uma zona onde são apresentadas imagens ao jogador, uma caixa de texto onde pode ser lida a história e o resultado das acções do jogador, uma *combobox* onde é possível consultar o inventário, as opções de que o jogador dispõe e um botão para sair (o exit).

Esta interface foi inspirada no gestor de narrativas C-DraGer que mostrava todas as opções que o jogador podia fazer na altura, bastava escrever o número da opção para a executar. Com intuito de dar um passo em frente, perante as novas tecnologias de écran táctil surgiu a ideia de colocar as opções do jogador de modo a poderem ser clicadas em vez de escrever o seu número.

Perante esta interface o jogador não é obrigado a memorizar um conjunto de palavras para comunicar com o jogo. Já não é necessário tentar vários comandos

escritos e ficar frustrado por nenhum ajudar a atingir os objectivos. Deste modo é muito mais simples navegar pela história. Uma interface destas é facilmente adaptada para telemóveis e tablets, permitindo o jogador usufruir destes jogos, no conforto do seu lar, nos transportes públicos ou em qualquer lado.

3.1.7 Representação da Estrutura do Jogo

Para representar a estrutura do jogo foi escolhido o formato Xml, além de ser independente do sistema operativo, é muito útil para manter a informação organizada com a ajuda das etiquetas. Deste modo os locais, os objectos, os personagens, os pontos de enredo, as acções do gestor de narrativas são representados por etiquetas. O nosso sistema de ficção interactiva com gestor de narrativas incorporado (FIGN) vai ter o papel de interpretar o Xml.

```
1 <?xml version="1.0" encoding="ISO-8859-1" ?>
2 <document>
3 <interface>
4 <interfaceWasChanged>>false</interfaceWasChanged>
5 </interface>
6 <room>
7 <name>Entrada</name>
8 <imagePath>./imagens/Entrada.png</imagePath>
9 <description>Depois dos achados arqueológicos de Figueira de Castelo Rodrigo
10 deram-me a oportunidade de explorar também a aldeia fantasma
11 de Colmeal. Agora que cheguei estou entusiasmado para ver o
12 estado em que a aldeia se encontra.</description>
13 <option>
14 <optionID>room_1_opt_1</optionID>
15 <optionName>Ir para Norte.</optionName>
16 <optionImagePath></optionImagePath>
17 <optionDisplayText></optionDisplayText>
18 <optionCode>0</optionCode>
19 <optionTargetIndex>1</optionTargetIndex>
20 <optionTarget>Ponto_1</optionTarget>
21 <isHiddenOption>>false</isHiddenOption>
22 </option>
```

Figura 28 – Exemplo da estrutura do jogo

Na Figura 28 temos um exemplo da estrutura do jogo. O *encoding* do Xml é do tipo ISO-8859-1 para poder suportar os caracteres do alfabeto latino. De seguida encontra-se a etiqueta <document> que define onde começa o jogo e o seu par </document> encontra-se no final do ficheiro.

Na etiqueta <interface> encontra-se tudo o que o autor alterou na interface, neste caso não existem alterações, logo a etiqueta <interfaceWasChanged> tem o valor *false*. Caso existissem alterações, o seu valor seria *true* e existiriam mais etiquetas correspondente às zonas do interface que teriam novas imagens associadas.

Neste exemplo também é possível ver os campos que constituem um local, representado pela etiqueta <room>, como por exemplo o nome do local, a imagem que vai ter, a descrição que vai ser apresentada e as opções que o jogador vai ter nesse local.

```
586 <item>
587 <itemName>papel</itemName>
588 <itemDescription>O papel tem escrito 'Ninguém pode fugir ao mundo que vem do
589 solo'.</itemDescription>
590 <itemDuration>-2</itemDuration>
591 </item>
592 <item>
593 <itemName>relógio</itemName>
594 <itemDescription>Um relógio de prata bonito.</itemDescription>
595 <itemDuration>-1</itemDuration>
596 </item>
```

Figura 28 – Exemplo da estrutura do jogo em relação aos itens

Na Figura 28 podemos observar um exemplo da estrutura dos itens que o jogador pode ter e na Figura 29 é possível ver como os pontos de enredo e as ações do gestor de narrativas ficam codificadas no ficheiro do jogo.

```
614 <DramaManagerSearchDepth>4</DramaManagerSearchDepth>
615 <plotPoint>
616 <plotPointZoneIndex>6</plotPointZoneIndex>
617 <plotPointVisitedZones>Cemiterio</plotPointVisitedZones>
618 <plotPointNotVisitedBool>true</plotPointNotVisitedBool>
619 <plotPointZone>Ponto_6</plotPointZone>
620 <plotPointPlayerInventory>papel</plotPointPlayerInventory>
621 <plotPointPenalty>0.6</plotPointPenalty>
622 <dmAction>
623 <dmActionName>Avisa jogador</dmActionName>
624 <dmActionImagePath></dmActionImagePath>
625 <dmActionImagePathDM></dmActionImagePathDM>
626 <dmActionDisplayText>Agora deveria ir dar uma vista de olhos ao cemitério.</dmActionDisplayText>
627 <dmActionCode>0</dmActionCode>
628 <dmActionItemName></dmActionItemName>
629 <dmActionTextForOldAction></dmActionTextForOldAction>
630 </dmAction>
631 </plotPoint>
```

Figura 29 – Exemplo da estrutura do jogo em relação aos pontos de enredo e ações do gestor de narrativas

Os campos que aparecem nestas imagens vão ser explicados com maior detalhe na secção 3.2.

3.2 Editor do FIGN

3.2.1 Preocupação na interface do editor

Para a criação do FIGN o IDE escolhido foi o Netbeans [30, 31, 32, 33, 34] (ver Figura 30), este permite desenvolver múltiplas aplicações para múltiplas linguagens e proporciona um rápido desenvolvimento devido às suas funcionalidades, das quais se podem destacar:

- Facilidade na criação de Interfaces gráficas, que foi algo importante para a realização deste trabalho.
- Correção automática, permitindo detectar erros mais rapidamente.
- Auto-conclusão, auxiliando o programador com sugestões para completar as palavras que este escreve.
- Utiliza cores diferentes para separar os diferentes elementos do código, facilitando assim a sua percepção.

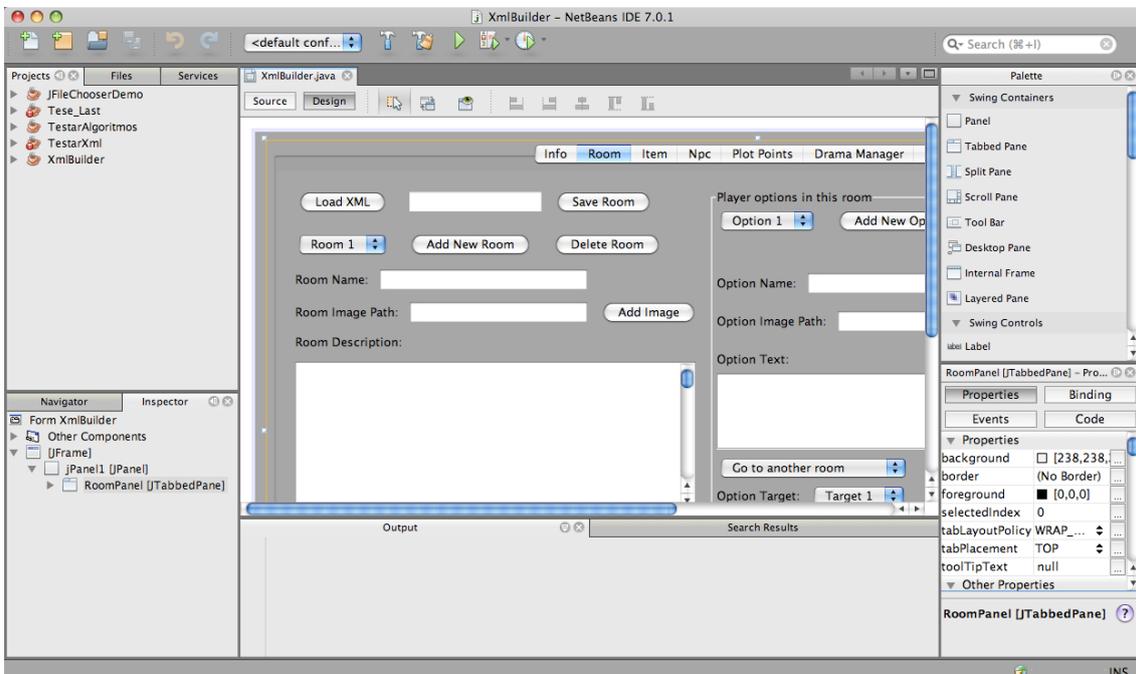


Figura 30 – Screenshot do NetBean

Na criação do editor do FIGN tivemos cuidado para obter uma interface que pudesse abstrair o autor da programação, aumentando assim o leque de público-alvo para esta ferramenta e facilitando a criação de jogos. No entanto é preciso ter em conta que é extremamente difícil criar uma interface simples e intuitiva e conseguir ao mesmo tempo fornecer o controlo total ao utilizador sobre todos os aspectos do jogo.

Para isso foram implementadas certas características para o apoiar durante a criação de jogos, como por exemplo:

- *File chooser* – Quando o utilizador quer abrir um jogo existente, para continuar a trabalhar sobre ele, ou pretende adicionar uma imagem ao jogo, pode contar com o *file chooser* para abrir uma janela que permite percorrer as pastas do computador até encontrar o ficheiro pretendido. Na Figura 31 é possível ver um exemplo de quando o utilizador clica no botão Add Image. Ao *file chooser* também foi adicionado um filtro para evitar que o utilizador cometa erros, como por exemplo, adicionar em vez de uma imagem outro tipo de ficheiro.

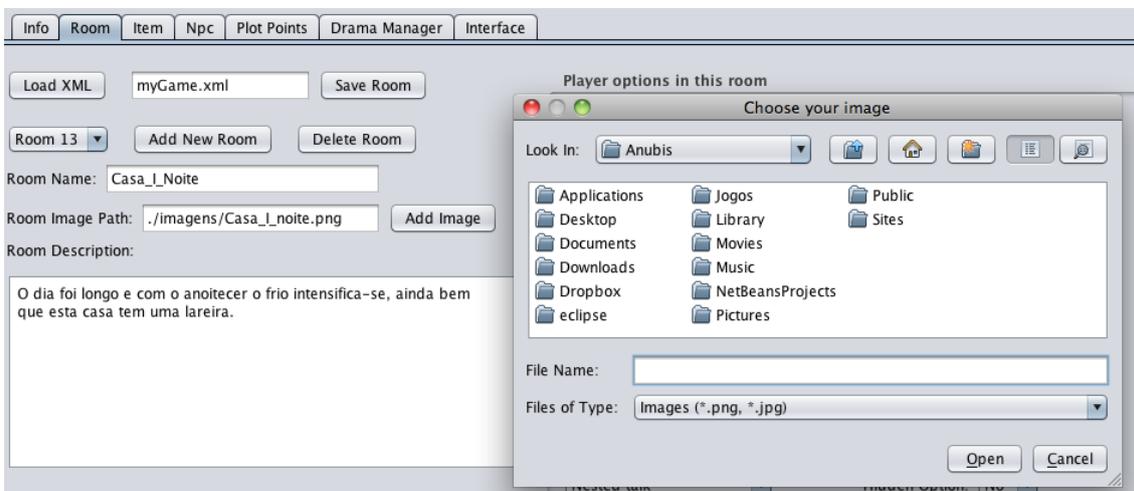


Figura 31 – Exemplo do *File chooser*

- Redimensionamento de imagens – sempre que o utilizador adiciona uma imagem ao jogo, é criada uma cópia da imagem já redimensionada para ser apresentada correctamente quando o jogo for executado.

- Sugestões – quando o utilizador coloca o cursor sobre o nome de um campo a preencher recebe uma indicação do propósito desse campo, como podemos ver na Figura 32.

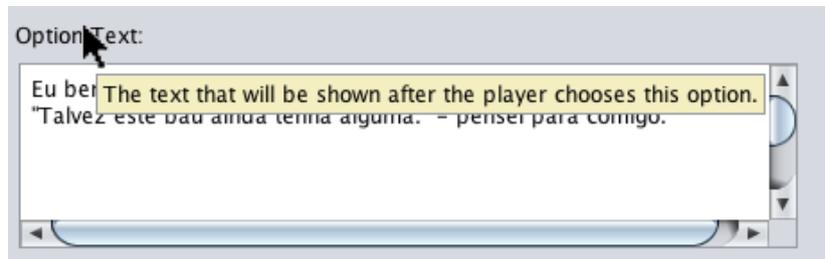


Figura 32 – Sugestões de ajuda

Para a construção do jogo ser mais intuitiva, a interface do editor está dividida em vários separadores, onde em cada um é possível trabalhar as várias componentes do jogo. Um jogo é composto por 3 componentes principais: zonas, itens e personagens.

3.2.2 Separador de informação

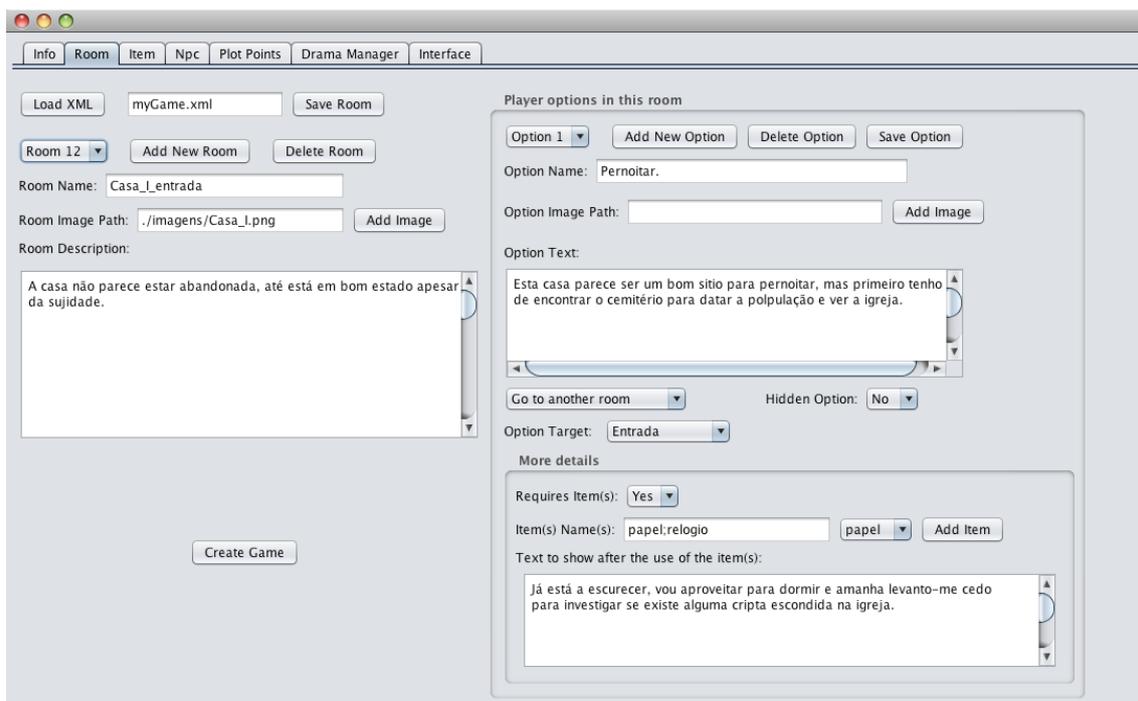
O primeiro separador apenas contém algumas sugestões para o autor saber como deve criar as zonas da sua história, criar itens, *Npcs*, para o ambientar ao editor.

3.2.3 Separador da criação de zonas

Um jogo pode ser constituído por um número aleatório de zonas, onde em cada zona o autor define que acções o jogador pode executar.

Como se pode ver na Figura 33, no lado esquerdo define-se o nome da zona, escolhe-se a imagem que vai aparecer quando o jogador chegar a esse local e a descrição que lhe vai ser apresentada. No lado direito o autor pode definir que opções é que o jogador vai ter à sua disposição nessa zona. Nas opções o autor define o nome da opção, pode alterar a imagem que está a ser exibida ao jogador ou pode deixar o campo em branco para manter a imagem da zona onde se encontra e pode escrever o texto que vai aparecer quando o jogador escolher essa opção.

Figura 33 – Exemplo da criação de uma zona e suas opções



Dentro das várias opções que o jogador pode executar temos:

- Mudar de zona.
- Adicionar um item ao inventário.
- Conversas encadeadas.
- Conversa com Npc.
- Examinar algo.

Mudar de zona

Se o autor escolher a opção mudar de zona, então no campo *Option Target* vão aparecer todas as zonas (já criadas) para onde o jogador pode ir.

Adicionar um item ao inventário

Caso escolha adicionar um item ao inventário, no campo *Option Target* vão aparecer todos os itens (já criados) que podem ser adicionados.

Conversas encadeadas

A opção conversas encadeadas, corresponde a acções que têm como resultado uma nova acção para o jogador executar. No campo *Option Target* vão aparecer todas as opções já criadas. Por exemplo, posso ter uma acção “abrir baú.” que tem como alvo a acção “retirar tesouro.” que por sua vez tem como alvo a opção “fugir antes que isto tudo desmorone!”. Neste caso teríamos as acções “abrir baú” e “retirar tesouro” como conversas encadeadas e a acção para mudar de zona “fugir antes que isto tudo desmorone!”.

Falar com NPCs

Falar com *NPCs* vai ser semelhante às opções anteriores, no campo *Option Target* vai ser possível escolher o *NPC* pretendido para a conversa, dentro dos *NPCs* criados.

Examinar algo

A opção examinar algo apenas contribui para o enriquecimento da narrativa, como por exemplo “espreitar pela janela” ou “olhar para o quadro”. Este tipo de opção também pode ser usado para terminar uma conversa encadeada. Não tem o campo *Option Target*.

Após escolher a acção que o jogador pode tomar, temos o campo *Hidden Option* que serve para ocultar a acção, este campo deve ser usado nas acções encadeadas, onde apenas a primeira acção deve ser visível, mas as acções para as quais ela aponta devem

ser invisíveis. No caso do exemplo anterior as acções “retirar tesouro” e “fugir antes que isto tudo desmorone!” seriam opções escondidas, ou seja, não estariam visíveis ao jogador. A opção “retirar tesouro” apareceria após a opção “abrir baú” ser utilizada e esta ficaria escondida por sua vez (ver Figura 34).



Figura 34 – Visibilidade das opções disponíveis ao jogador

Na área de mais detalhes encontramos a opção *Requires Item(s)*. Caso esta opção seja verdadeira é possível definir que itens são necessários, através da lista de itens que se encontra à frente do campo *Item(s) Name(s)*. Para adiciona-los basta escolher o item desejado na lista e clicar em adicionar item. O último campo serve para o autor escrever o que quer que apareça no écran após o jogador executar a acção (que requer item), por exemplo, a acção “abrir baú” podia precisar do item chave e se o jogador não tivesse esse item, no momento em que executava a acção iria aparecer o texto da opção “o baú não está a abrir, deve estar trancado.” e quando finalmente ele tivesse na posse da chave, ao executar a mesma acção apareceria o texto deste campo “experimentei a chave e o baú abriu-se.”. Caso a opção não necessite de itens, os campos da área de mais detalhes ficam ocultos (ver Figura 35).

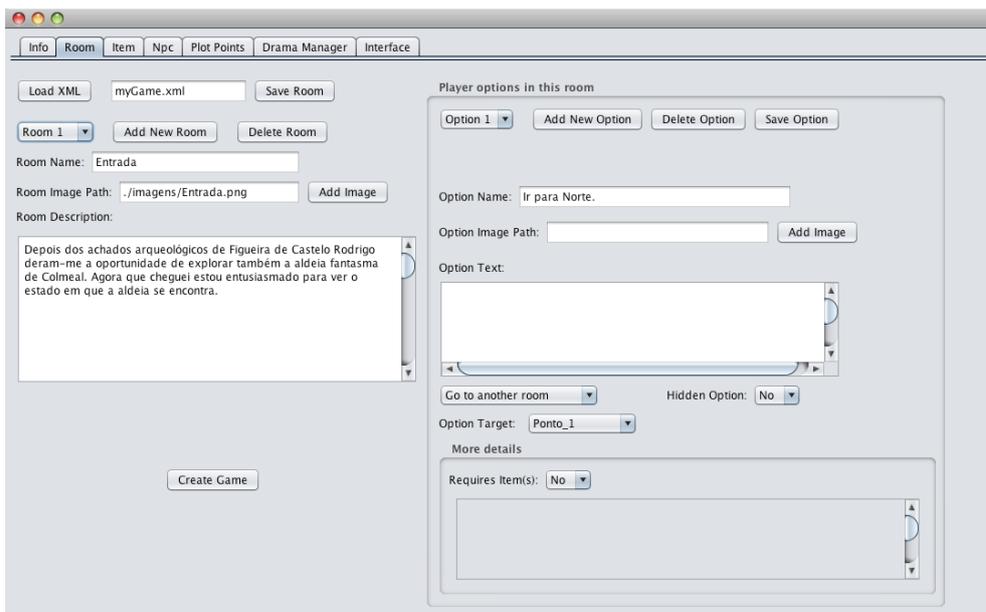


Figura 35 – Exemplo de opção que não requer itens

3.2.4 Separador de Itens

A criação de itens é muito simples, o autor indica o nome que o item vai ter, uma descrição para o jogador ler quando clicar no item e decide se o item pode ser consumido ou não (ver Figura 36). Os itens só podem ser consumidos nas acções de mudança de zona, nas outras acções, este vai permanecer no inventário do jogador. Na imagem anterior no capítulo 3.1.7 (página 44) é possível observar a diferença entre os valores de um objecto que pode ser consumido, o papel (que tem valor -2) e o relógio (que tem valor -1) que não pode ser consumido.

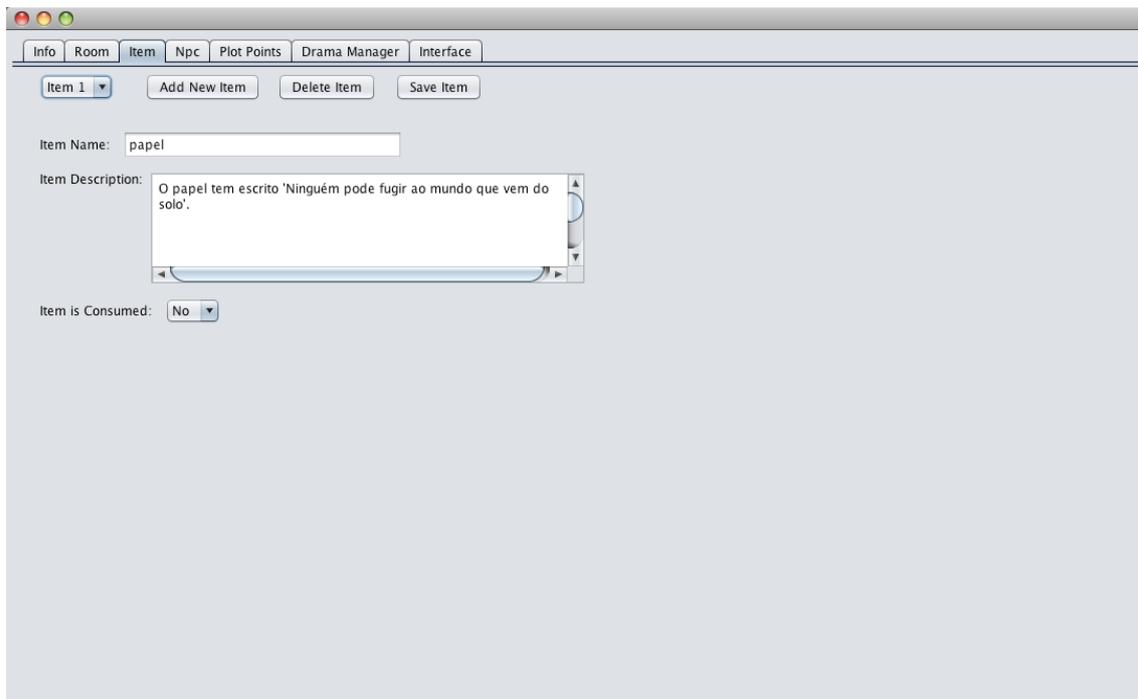


Figura 36 – Exemplo de criação de Item

3.2.5 Separador de *NPCs*

Na criação de *NPCs*, o autor atribui-lhes um nome e escolhe a zona do jogo onde eles se vão encontrar. Após a sua criação, só falta definir os diálogos dos *NPCs* e as respostas que o jogador pode dar. A comunicação entre o jogador e os *NPCs* vai começar no jogador com o tipo de acção falar com *NPC*, a resposta que o jogador vai receber vai ser o texto que está no campo *NPC talk to player* (nos diálogos, ver Figura 37). Quando o jogador falar com o *NPC* vai aparecer na sua lista de respostas a opção que o autor tiver definido na zona *Player Answers*. A criação das respostas é igual à criação das opções já referida na secção 3.2.3. Os *NPCs*[28] foram inicialmente criados com o propósito de se poderem deslocar e terem alguma autonomia na narrativa, mas isso acabou por se tornar trabalho futuro. Tendo isto em conta os *NPCs* também podem ser substituídos por conversas encadeadas que simulem conversas com personagens da narrativa.

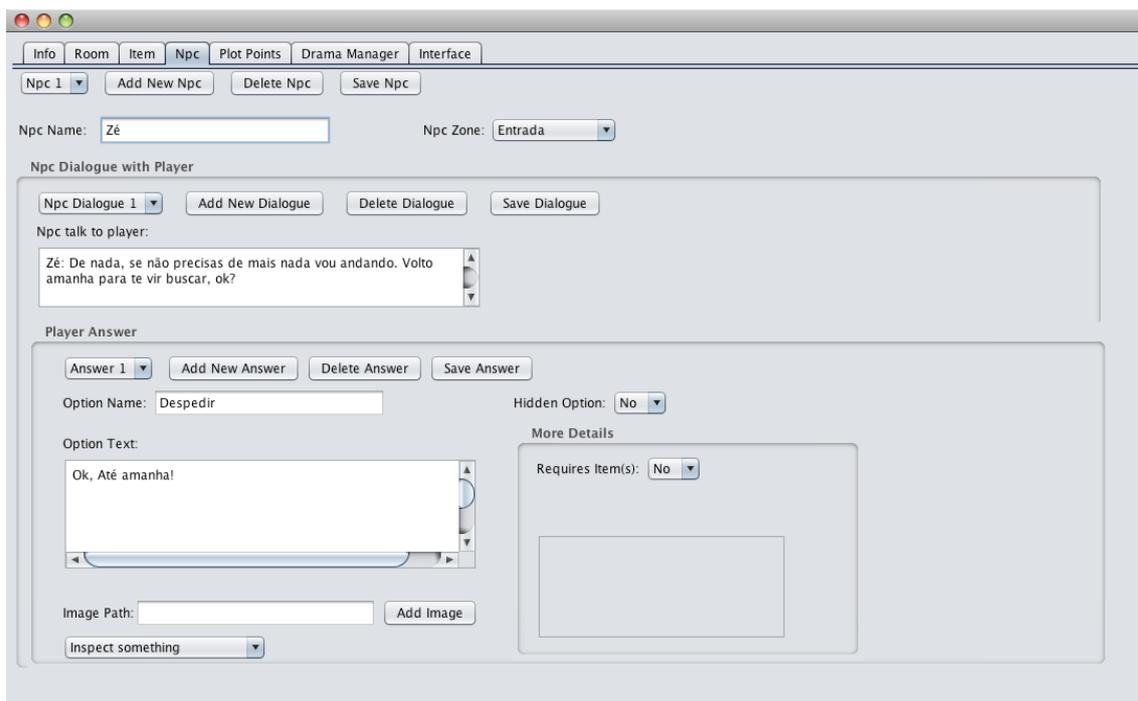


Figura 37 – Exemplo de criação de *NPC*

3.2.6 Separador de Pontos de Enredo

A criação de pontos de enredo permite o autor escolher em que zona é que o ponto de enredo se vai encontrar, as zonas que o jogador já visitou ou ainda não visitou, os itens que o jogador tem que ter na altura e a penalidade da acção do gestor de narrativas nesse ponto de enredo.

Como é possível ver na Figura 38, os pontos de enredo são constituídos pela zona onde são detectados (Ponto_6), pelos itens que o jogador tem que ter na altura (papel), pelas zonas onde o jogador já passou (ou pelas zonas por onde ainda não passou, neste caso o Cemitério) e pela penalidade aplicada ao gestor de narrativas por actuar (0,6). Em cada ponto de enredo o autor define que acções o gestor de narrativas vai ter, neste caso a acção escolhida foi dar uma sugestão. Caso todas as condições do ponto de enredo sejam satisfeitas, ou seja, o jogador chegue ao Ponto_6, ainda não tenha visitado o Cemitério e o único item no seu inventário seja só o papel, vai ser apresentado o texto "Agora deveria ir dar uma vista de olhos ao cemitério."

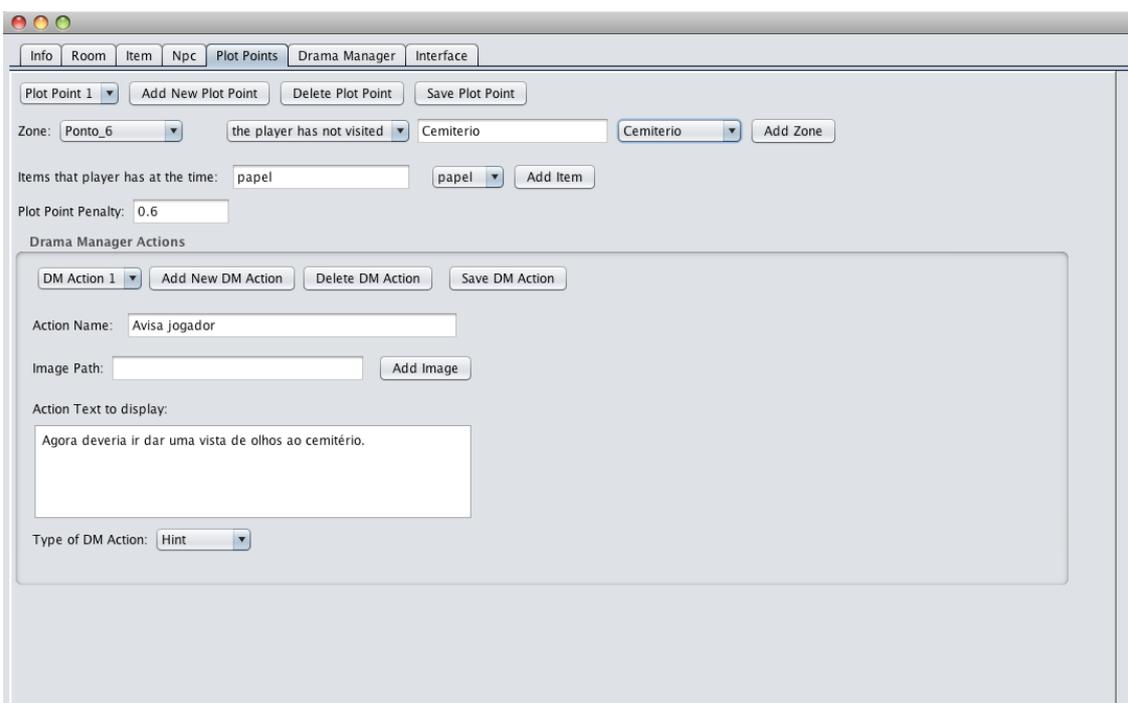


Figura 38 – Exemplo de criação de pontos de enredo e respectivas acções do gestor de narrativas

3.2.7 Separador de Gestor de Narrativas

No separador *Drama Manager* (ver Figura 39), é possível escolher a profundidade da árvore de pesquisa que o gestor de narrativas vai usar para simular os possíveis futuros do jogador. Na escolha deste valor, é preciso no entanto ter noção de que quanto maior for a profundidade de procura, mais tempo vai levar a calcular o melhor caminho para o jogador. Neste separador também se encontram algumas informações para explicar o funcionamento dos pontos de enredo e das acções do gestor de narrativas.

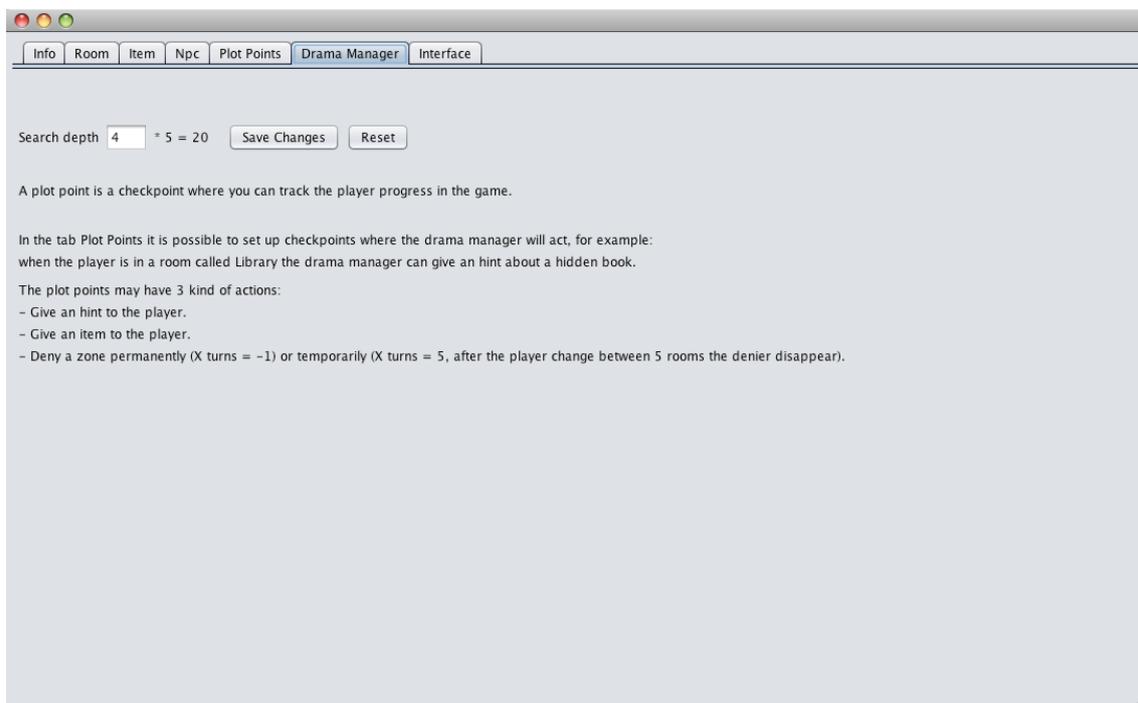


Figura 39 – Exemplo da escolha de profundidade da pesquisa

3.2.8 Separador de Interface

No separador *Interface* o autor pode personalizar vários aspectos da interface do jogo. Pode alterar a imagem de fundo, a moldura das imagens que vão ser exibidas durante o jogo, o ícone do inventário, as marcas antes das opções do jogador, o botão para sair do jogo, a imagem que, neste caso está em tribal à esquerda e à direita do botão para sair do jogo (ver Figura 40). Deste modo é possível personalizar a interface de acordo com a história do jogo criado para aumentar um pouco mais o sentido de imersão.



Figura 40 – Exemplo de personalização da interface do interpretador

Capítulo 4

Discussão e Trabalho Futuro

4.1 Conclusões

Nesta dissertação apresentámos o protótipo FIGN, constituído pelo par editor de jogos de ficção interactiva e interpretador com gestor de narrativas incorporado. O editor é composto por uma interface amigável, é compatível com os sistemas operativos mais usados (Windows, Linux, Mac), tem condições para abrir o caminho a novos autores no mundo da ficção interactiva.

No editor, o autor pode contar com uma série de características especialmente criadas para o apoiar, como *filechoosers*, redimensionamento automático das imagens que pretender usar no seu jogo e sugestões sobre o propósito dos vários campos do editor.

No interpretador, dado um conjunto de pontos de enredo, um conjunto de acções do gestor de narrativas e uma função de avaliação da qualidade da história, o gestor de narrativas vai guiar o jogador pelo enredo de modo a melhorar a sua experiência de jogo. Com o FIGN os jogadores “recém chegados” aos jogos de ficção interactiva já não vão sentir a frustração de se perderem, de tentarem vários comandos e nenhum os levar a uma saída óbvia ou de não conseguirem lembrar-se de todos os comandos que podem utilizar para interagir com o jogo. O FIGN incentiva-os a chegar ao fim do jogo, proporcionando uma sensação de realização pessoal ao jogador.

Tendo em conta a rápida evolução dos jogos, sistemas como o FIGN podem fazer a diferença entre aumentar o número de fans na comunidade da ficção interactiva ou perde-los para outros tipos de jogos.

Embora apenas eu tenha testado o jogo, fiquei satisfeito com os resultados, sempre que me desviava do caminho pretendido o gestor de narrativas intervinha para me guiar para o caminho certo. Apesar dos testes terem sido realizados num jogo

pequeno, acredito que o FIGN possa ter um impacto maior num jogo de maiores dimensões.

4.2 Trabalho futuro

Para trabalho futuro era muito interessante implementar certas características, como por exemplo, o som, efeitos sonoros e vídeos. Os NPCs actuais são dotados de pouca autonomia e inteligência, pretendemos melhorar estes aspectos para os tornar mais imprevisíveis.

O editor já é composto por uma interface amigável, mas ainda assim tem aspectos a melhorar, um exemplo é dar hipóteses ao autor de escolher a língua em que toda a documentação e texto aparecem.

Também seria de grande interesse implementar esta ferramenta online, de modo a facilitar a vida aos jogadores, não teriam de se preocupar em ter o interpretador no seu computador. Com esta ferramenta online seria possível utilizar outras abordagens para melhorar o gestor de narrativas, como por exemplo o modelo de jogador do C-DraGer com raciocínio baseado em casos. Seria possível reunir a informação dos jogadores que utilizassem o interpretador online para melhorar o modelo de jogador.

Cada vez mais os idosos têm tendência para aprender a utilizar computadores, isso ajuda-os a combater a solidão e os jogos servem para manter o cérebro activo. Ao longo de um ano fui professor de informática, voluntário na Academia Sénior, na Junta de Freguesia de Carnide. A oportunidade de os ensinar e perceber as suas dificuldades, levou-me a compreender que este tipo de jogos seria muito estimulante para eles, já que não implica destreza física e jogar estes jogos é como ler um livro, mas de uma forma interactiva. Tendo em conta os problemas que referi, estes jogos seriam impensáveis, mas com o FIGN, eles conseguiriam jogar sem problemas, já que o gestor de narrativas estaria presente para os apoiar. Seria interessante testar a adequação desta ferramenta à população sénior.

Outro aspecto a abordar pode passar pela criação de uma versão do interpretador para telemóveis, tablets, para permitir que os jogadores possam disfrutar desta ferramenta no conforto do seu lar, nos transportes públicos ou em qualquer lado.

NOTA: Esta dissertação foi escrita ao abrigo do antigo acordo ortográfico.

Capítulo 5

Bibliografia

- [1] Josiah Lebowitz, Chris Klug, Interactive Storytelling for Video Games
- [2] Ficção Interactiva, http://en.wikipedia.org/wiki/Interactive_fiction, consultado em 12/09/2012
- [3] Ficção Interactiva, <http://brasslantern.org/beginners/introif.html>, consultado em 12/09/2012
- [4] História da ficção interactiva, <http://www.digitalhumanities.org/companion/view?docId=blackwell/9781405148641/9781405148641.xml&chunk.id=ss1-5-8>, consultado em 12/09/2012
- [5] Adrift, <http://www.adrift.co/cgi/adrift.cgi>, consultado em 12/09/2012
- [6] Alan, <http://welcome.to/alan-if>, consultado em 12/09/2012
- [7] Hugo, http://www.generalcoffee.com/index_noflash.php, consultado em 12/09/2012
- [8] Inform 7, <http://inform7.com/>, consultado em 12/09/2012
- [9] Quest, <http://www.textadventures.co.uk/quest/>, consultado em 12/09/2012
- [10] TADS, <http://www.tads.org/>, consultado em 12/09/2012
- [11] Future Boy, <http://www.generalcoffee.com/futureboy/index.html>, consultado em 12/09/2012
- [12] Zork I, http://en.wikipedia.org/wiki/Zork_I, consultado em 12/09/2012
- [13] Mapa de Zork I, <http://www.ironrealms.com/image/original-zork-i-map-page-two>, consultado em 12/09/2012
- [14] Myst, <http://en.wikipedia.org/wiki/Myst>, consultado em 12/09/2012
- [15] Monkey Island, http://en.wikipedia.org/wiki/Monkey_Island_%28series%29, consultado em 12/09/2012

- [16] Façade, http://en.wikipedia.org/wiki/Fa%C3%A7ade_%28interactive_story%29, consultado em 12/09/2012
- [17] Cuts scene, <http://pt.wikipedia.org/wiki/Cuts scene>, consultado em 12/09/2012
- [18] Bates, J. 1992. Virtual Reality, Art, and Entertainment. *Presence: The Journal of Teleoperators and Virtual Environments* Vol. 2 No. 1: 133-138.
- [19] Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Technical Report CMU-CS-97-109.
- [20] Ari Lamstein, Michael Mateas - Search-Based Drama Management - The Georgia Institute of Technology College of Computing, Technology Square Research Building 85 5th Street, NW Atlanta GA 30332-0760
- [21] Mark J. Nelson, Michael Mateas, Search-Based Drama Management in the Interactive Fiction Anchorhead, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA
- [22] Mark J. Nelson, David L. Roberts, Charles L. Isbell, Jr., Michael Mateas, Reinforcement Learning for Declarative Optimization-Based Drama Management, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA
- [23] Manu Sharma, Santiago Ontañón, Manish Mehta, Ashwin Ram, Drama Management and Player Modeling for Interactive Fiction Games - In Computational Intelligence Journal, Volume 26 Issue 2, pp.183-211
- [24] Manu Sharma, Santiago Ontañón, Manish Mehta, Ashwin Ram, Drama Management Evaluation for Interactive Fiction Games, Cognitive Computing Lab (CCL) College of Computing, Georgia Institute of Technology Atlanta, Georgia, USA
- [25] Chinmay Barve, Devika Karnik, Kai Wang, Search Based Drama Manager, Georgia Institute of Technology
- [26] Andrea Corradini¹, Manish Mehta², Santiago Ontañón², Evaluation of a Drama Manager Agent for an Interactive Story-based Game, ¹University of Southern Denmark, IFKI 6000 Kolding, Denmark, ²Georgia Institute of Technology, Cognitive Computing Lab Atlanta 30332-0760, Georgia, USA
- [27] Anne Sullivan, Sherol Chen, Michael Mateas, From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience, Expressive Intelligence Studio University of California, Santa Cruz

- [28] NPCs aut3nomos, <http://brasslantern.org/writers/ifttheory/autonomousnpcs.html>, consultado em 12/09/2012
- [29] MinMax, <http://en.wikipedia.org/wiki/Minimax>, consultado em 12/09/2012
- [30] Site do NetBeans, <http://netbeans.org/>, consultado em 12/09/2012
- [31] Cria3o de interface, <http://netbeans.org/kb/docs/java/gui-functionality.html>, consultado em 12/09/2012
- [32] Manipula3o de imagens, http://netbeans.org/kb/docs/java/gui-image-display_pt_BR.html, consultado em 12/09/2012
- [33] Redimensionar imagens, <http://www.mkyong.com/java/how-to-resize-an-image-in-java/>, consultado em 12/09/2012
- [34] File chooser, <http://netbeans.org/kb/docs/java/gui-filechooser.html>, consultado em 12/08/2012
- [35] Import3ncia das Cutscenes, <http://www.gametrailers.com/full-episodes/1mk3gs/bonus-round-the-future-of-storytelling>, consultado em 12/08/2012
- [36] Hansel e Gretel, http://pt.wikipedia.org/wiki/H%C3%A4nsel_und_Gretel, consultado em 12/08/2012