



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

BUSCA DE CAMINHOS EM STORYTELLING INTERATIVO COM
INFLUÊNCIA DE EMOÇÕES

Felipe Guedes da Silveira

Orientador
Prof. Dr. Angelo Ernani Maia Ciarlini

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2014

BUSCA DE CAMINHOS EM STORYTELLING INTERATIVO COM
INFLUÊNCIA DE EMOÇÕES

Felipe Guedes da Silveira

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA
OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO
EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE
JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO
ASSINADA.

Aprovada por:

Angelo Ernani Maia Ciarlini, D.Sc. – UNIRIO

Sean Wolfgang Matsui Siqueira, D.Sc. – UNIRIO

Bruno Feijó, Ph.D. – PUC-RJ

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2014

< Página para Ficha Catalográfica >

Ao meu Avô Ribeiro, com carinho.

Agradecimentos

Ao Professor e Orientador Angelo Ciarlini, sempre flexível com as ideias, objetivo em suas realizações, incentivador nas horas difíceis e inspirador durante todo o processo. É um privilégio poder acompanhar seu brilhantismo como orientando.

À minha mãe, Fátima, por toda paciência e apoio nas horas mais difíceis – seu apoio e crença em mim foram indispensáveis para concluir o processo com sucesso.

Ao meu pai, Alpoim, por me fazer enxergar o mesmo problema por diferentes ângulos – as incontáveis horas de exercícios lógicos foram essenciais para o interesse e desenvolvimento desta pesquisa.

Ao meu irmão, Alex, por todas as nossas conversas e por ter me motivado a continuar – mesmo nos momentos de maior dificuldade.

Agradeço a meu amigo Julio Teixeira, pelas incontáveis horas de conversas e discussões matemáticas – mais de vinte anos de contato são difíceis de contabilizar.

Agradeço a meu amigo Arlindo Tavares por toda paciência e dedicação ao longo dos anos – este trabalho é fruto dos alicerces educacionais do passado.

Agradeço a todos os amigos que me acompanharam através do processo do mestrado – foi uma longa jornada, muito obrigado por todo apoio.

À CAPES pelo apoio financeiro e acesso aos bancos de dados de pesquisa.

SILVEIRA, Felipe Guedes da. **Busca de Caminhos em Storytelling Interativo com Influência de Emoções**. UNIRIO, 2014. 80 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Situada na área de *Storytelling* Interativo e inspirada por uma necessidade de enriquecer as cenas dramatizadas em ambientes 3D no sistema Logtell, esta dissertação propõe um modelo de busca de caminhos que acompanha um arco dramático emocional, fornecido pelo autor, durante a busca, permitindo que o resultado se aproxime a uma curva emocional modelada em função do tempo. Utilizando de um sistema de grafos que abstrai um ambiente (possivelmente gerado procedimentalmente) e adicionando informações de influência emocional aos nós deste grafo, é possível realizar uma busca que respeite as restrições emocionais da cena a ser apresentada e encontrar um caminho que faça sentido visualmente, emocionalmente e possa ser utilizada em aplicações que demandem resposta em tempo real.

Palavras-Chave: *Storytelling* Interativo; Jogos; Busca de Caminhos; Emoções; Geração Procedimental; Tempo Real

SILVEIRA, Felipe Guedes da. **Emotional Pathfinding in Interactive Storytelling.**

UNIRIO, 2014. 80 pages. Master's Thesis. Departamento de Informática Aplicada,

UNIRIO.

ABSTRACT

Located in the Interactive Storytelling research field and inspired by the need to improve the 3D dramatization in Logtell system, this dissertation proposes a model of pathfinding that follows a given emotional dramatic arc during the search, allowing the result to maximize an emotional curve modeled as a function of time. Using a graph system that abstracts an environment (possibly procedurally generated) and adding emotional influence information to the nodes of this graph, we perform a search that obeys the emotional constraints of the scene to be rendered and find a path that makes sense both visually and emotionally, also allowing this model to be used in applications that require real time responses.

Keywords: Interactive Storytelling; Games; Emotional Pathfinding; Emotion; Pathfinding; Procedural Generation; Real Time

1	INTRODUÇÃO.....	14
1.1	MOTIVAÇÃO	14
1.2	PROBLEMA.....	16
1.3	PROPOSTA.....	17
1.4	ESTRUTURA	19
2	FUNDAMENTAÇÃO BÁSICA.....	21
2.1	STORYTELLING INTERATIVO	21
2.2	GERAÇÃO PROCEDIMENTAL DE CONTEÚDO.....	26
2.3	BUSCA DE CAMINHOS	30
2.3.1	<i>Pseudocódigo do A*</i>	34
2.3.2	<i>Algoritmo de Busca A* em execução</i>	38
3	MODELO	42
3.1	DEFINIÇÃO DO PROBLEMA.....	42
3.1.1	<i>Dados de Entrada</i>	42
3.1.2	<i>Requisitos para o modelo</i>	46
3.2	MODELO PROPOSTO.....	47
3.3	IMPLEMENTAÇÃO.....	50
3.3.1	<i>Pseudocódigo</i>	50
3.3.2	<i>Interface</i>	57
4	VALIDAÇÃO.....	61
4.1	COMPATIBILIZAÇÃO ENTRE NATURALIDADE DE CAMINHO E VARIAÇÃO DE EMOÇÕES.	64
4.1.1	<i>Comparação com Soluções Ótimas em Cenários Pequenos</i>	65
4.1.2	<i>Caminhos Naturais e Curvas Emocionais em Cenários Grandes</i>	67
4.2	TEMPO	71

5	CONCLUSÃO.....	73
5.1	CONSIDERAÇÕES GERAIS.....	73
5.2	PRINCIPAIS CONTRIBUIÇÕES.....	73
5.3	TRABALHOS FUTUROS.....	74
6	REFERÊNCIAS.....	77

Lista De Figuras

Figura 1: Ordem de visitação dos nós no BFS	31
Figura 2: Ordem de visitação dos nós no DFS	32
Figura 3: Exemplo de grafo com Waypoints.....	39
Figura 4: O A* aplicado com três heurísticas diferentes.....	40
Figura 5: Terreno base e Grafo resultante sobreposto.....	43
Figura 6: Grafo resultante da extração	44
Figura 7: Grafo com informação emocional.....	45
Figura 8: Interface principal do utilitário utilizado no desenvolvimento.	58
Figura 9: Representação visual do grafo com as áreas de medo e dois possíveis caminhos.	59
Figura 10: Curvas emocionais ideais e reais.	60
Figura 11: Grafo com dimensão 125x125 utilizado.	62
Figura 12: Grafo com dimensão 10x10 utilizado.	63
Figura 13: Grafo com dimensão 5x5 utilizado.	63
Figura 14: Grafo com dimensão 8x8 utilizado.	64
Figura 15: Exemplo de dois caminhos encontrados pelo modelo com ponderações diferentes e do melhor caminho encontrado por força bruta, para três diferentes variações de curvas emocionais lineares.	67
Figura 16: Caminho encontrado e Análise de Curva para uma curva emocional linear com variação emocional de 20 a 60	69
Figura 17: Caminho encontrado e Análise de Curva para uma curva emocional que começa em 20, tem seu pico no meio do caminho em 60 e retorna a 20 no final.	69

Figura 18: Caminho encontrado e Análise de Curva para uma curva emocional mais complexa. A curva ideal é representada em preto e a do caminho emocional em azul. 70

Figura 19: Caminho e curva emocional efetiva encontrada ao utilizar o A^* 71

Lista de Tabelas

Tabela 1: Comparação de classificação de caminhos, para os pesos $w_1=0.2$ $w_2=0.8$..	66
Tabela 2: Comparação de classificação de caminhos, para os pesos $w_1=0.1$ $w_2=0.9$..	66
Tabela 3: Consolidação dos dados utilizados na visualização gráfica dos caminhos encontrados por força bruta e pelo modelo	67
Tabela 4: Tempo para encontrar todos os caminhos com força-bruta.....	71
Tabela 5: Tempo médio de execução do modelo proposto e do A*.....	72

Lista de Pseudocódigos

Pseudocódigo 1: Funções Auxiliares para o A^*	34
Pseudocódigo 2: Pseudocódigo do A^*	36
Pseudocódigo 3: Funções Auxiliares para o Modelo	51
Pseudocódigo 4: Pseudocódigo para o Modelo proposto.....	53

1 Introdução

1.1 Motivação

Esta pesquisa foca na área de *Storytelling* Interativo, uma forma de mídia digital que envolve o estudo da criação de histórias com múltiplos desenrolares, onde o público pode influenciar o enredo.

A área de *Storytelling* Interativo pode ser aplicada em diversos campos, como simulação, entretenimento e educação. Embora sejam domínios diferentes, esta área é constantemente correlacionada com a área de jogos e videogames – “*Storytelling Digital Interativo* deve emergir como uma forma de arte legítima nos próximos anos, e os videogames estão abrindo o caminho” (SQUIRE, 2003, tradução nossa).

Embora a maior parte do foco seja voltado para *Storytelling* Interativo, este modelo também pode ser livremente utilizado em Jogos que tenham interesse em controlar o caminho dos personagens, fazendo com que estes respeitem restrições emocionais do arco dramático proposto pela ambientação do jogo em si, visando aumentar a coerência das cenas e imersão do jogador no ambiente apresentado.

Assim como nos videogames, existe um esforço muito grande para criar cada parte que compõe uma história no *Storytelling* Interativo – e.g. enredo, cenário, iluminação, personagens entre outros.

Atualmente são utilizadas duas principais alternativas para gerar estas partes: o esforço humano e seu trabalho manual, criando individualmente cada uma delas; e a geração procedimental, onde softwares são responsáveis por gerar de maneira algorítmica várias destas.

A pesquisa na área de *Storytelling* Interativo vem crescendo, possuindo uma conferência especializada no assunto, a *International Conference on Interactive Digital Storytelling* (ICIDS), ativa desde 2008.

Este trabalho foi inspirado principalmente por uma necessidade de estender o sistema Logtell (CIARLINI, 1999), um sistema de *Storytelling* Interativo que visa cuidar de todas as etapas necessárias para gerar e dramatizar de histórias, cuidando de todas as etapas do processo.

Na última versão do Logtell, com os modelos propostos por GOTTIN (2013) e FERREIRA (2013), as histórias podem tratar de diversos efeitos emotivos em sua fase de planejamento.

No entanto, são necessários vários passos no meio do caminho para que esta visão lógica possa ser transformada e apresentada de forma rica ao público, em um ambiente tridimensional.

Alguns modelos foram propostos para auxiliar nessa transição (ARAUJO, 2011; CAMANHO et al., 2009; DÓRIA; CIARLINI; ANDREATTA, 2008), onde a representação da cena no Logtell culmina em uma série de micro-ações parametrizáveis passadas ao módulo gráfico, onde este fica responsável por executá-la da melhor maneira possível.

Desta forma, existe um interesse em que o módulo gráfico trate da melhor maneira possível estas micro-ações, respeitando a cena que está sendo dramatizada e as restrições

relativas às emoções impostas pelo enredo. Como os enredos são gerados automaticamente, de acordo com as interações com o usuário, e se deseja que sejam representados com flexibilidade em vários ambientes, possivelmente gerados procedimentalmente, é essencial prover mecanismos que adequem as cenas à emoção que precisa ser representada.

Existem incontáveis possibilidades de cenas e de aspectos relacionados a emoções nessas cenas, no entanto, a movimentação de atores é um acontecimento frequente e de fundamental importância para a coerência da história. As rotas têm implicação direta no fundo emocional a ser representado por uma cena – e.g. em um determinado momento que exista a necessidade de passar tensão ao público, a escolha de uma rota que passe por um beco escuro é mais apropriada do que um caminho iluminado, sem ameaças aparentes. De uma forma geral, a emoção durante uma cena pode ser representada por uma curva em função do tempo e a passagem dos personagens em cada ponto pode ter um impacto no valor da emoção corrente.

Isto despertou o interesse pelo foco deste trabalho, onde o caminho pelo qual os personagens vão de um ponto a outro no cenário é escolhido de acordo com a variação das emoções que precisam ser representadas. Tal escolha precisa ser feita de forma eficiente para que ocorra em tempo real em aplicações de *Storytelling* Interativo, onde enredos são gerados e dramatizados dinamicamente, em cenários criados de forma procedimental.

1.2 Problema

A movimentação de personagens em jogos e em *Storytelling* Interativo costuma ser tratada como a busca de caminhos em grafos, onde os nós são pontos fixos do cenário e

as arestas representam as possibilidades de movimentação entre esses pontos. A busca de caminhos é um problema bem definido que pode ser resolvido eficientemente por meio de algoritmos de busca heurística como o A*, que minimizam eficientemente o comprimento do caminho (HART; NILSSON; RAPHAEL, 1968). No entanto, o menor caminho pode ser completamente incompatível com a emoção a ser representada. Por outro lado, a escolha de caminhos que representam as emoções, mas que são demasiadamente longos, podem comprometer a coerência, não transmitindo a mensagem relativa à intenção de movimentação do personagem entre dois pontos específicos.

A combinação da busca de caminhos com a representação das emoções é portanto altamente desejável, mas essa combinação implica em um custo computacional mais alto que se torna um desafio em particular diante da necessidade de executar essas buscas em tempo real, com enredos e cenários gerados dinamicamente.

Desta forma, existe a necessidade de encontrar um caminho adequado à cena, que combine de forma balanceada três características:

- Vá no sentido do objetivo;
- Aproxime-se o máximo possível da emoção a ser representada na cena, definida como uma função do tempo;
- Possa ser gerado em tempo real, mesmo para áreas grandes.

1.3 Proposta

É proposto um modelo que procura resolver o problema de busca de caminhos em jogos e *Storytelling* Interativo, onde existe a necessidade de encontrar uma rota que faça sentido emocionalmente, acompanhando um determinado arco dramático – *i.e.* acompanhe uma

variação emocional especificada durante a movimentação – passando por áreas do mapa que modifiquem a emoção do personagem de acordo com as necessidades da cena.

Também é necessário que a rota sempre alcance o destino, se existir uma rota viável para isto. É importante lembrar que, dadas as características tanto dos jogos quanto do *Storytelling* Interativo, existe a necessidade do balanceamento entre alcançar o destino de forma rápida e respeitar as emoções necessárias para a rota.

No entanto, existe a necessidade de que as rotas não sejam redundantes em certas áreas do mapa – e.g. não pode acontecer de o personagem ficar andando em ziguezague dentro de um beco escuro para maximizar uma emoção de medo – a rota precisa fazer sentido visualmente, indo sempre que possível em direção ao destino.

Por último, é necessário que esta rota possa ser gerada por demanda, em tempo real, adaptando com rapidez a movimentação em cena, mesmo em cenários grandes. O público não pode ficar esperando que o software leve muito tempo para que a movimentação na cena aconteça.

Para solucionar estes problemas, é proposto um modelo estendido de busca de caminhos, que tem como base o algoritmo A* (HART; NILSSON; RAPHAEL, 1968). Assume-se para tal a existência de:

- Uma curva de emoção a ser cumprida durante a cena; e
- Um cenário com valores de modificação da emoção associados a cada ponto.

Na medida em que o personagem caminha ao longo do cenário, com velocidade constante, o valor da emoção pode se aproximar ou afastar da curva ideal. Faz-se uma alteração nas funções de custo do A* de modo a tentar aproximar a curva gerada pelo caminho escolhido da curva ideal. Para permitir o balanceamento entre os objetivo de minimização

do comprimento do caminho e de aproximação da curva ideal de emoção, bem como de eficiência na busca, são propostos:

- Pesos a considerar no balanceamento entre os objetivos;
- Comprimento de caminho máximo a ser considerado;
- Formas de normalização dos custos referentes as medidas de custo relacionadas a cada objetivo.

O enfoque adotado é avaliado comparando-se a eficiência e os resultados com métodos exaustivos de busca da solução ótima.

1.4 Estrutura

Esta dissertação apresenta seu conteúdo em cinco capítulos e uma seção adicional com referências bibliográficas que embasaram a pesquisa.

O segundo capítulo apresenta uma visão geral sobre *Storytelling* Interativo, como este evoluiu como mídia proeminente na apresentação de histórias e um histórico geral do sistema Logtell. Em seguida a Geração Procedimental é abordada, ressaltando como esta foi concebida, como alguns títulos comerciais (sobretudo jogos) a utilizam e como esta abordagem pode ser utilizada como uma ferramenta de autoria no *Storytelling* Interativo. Finalmente é tratado o tópico de busca de caminhos, dois métodos que tentam solucionar o problema com uma abordagem “força-bruta” e culmina com a apresentação do Algoritmo de Busca de Caminhos A*.

O terceiro capítulo trata de formalizar o problema, definindo as estruturas lógicas utilizadas no mesmo, os dados de entrada e a saída esperada. Em seguida são definidos os requisitos para o modelo e o modelo em si é proposto. O capítulo termina com a

apresentação do pseudocódigo do algoritmo desenvolvido, explicado passo a passo, e a interface da principal ferramenta desenvolvida durante este trabalho.

O quarto capítulo trata da validação dos dados gerados através da análise de tempo e de quão próximo do ideal emocional está o caminho encontrado em cada caso.

Por último, o quinto capítulo contém a conclusão da dissertação, mostrando as principais contribuições e possíveis trabalhos futuros.

2 Fundamentação Básica

Este trabalho trata da Busca de Caminhos com Influência de Emoções, com o intuito de ser utilizado em Jogos e Sistemas de *Storytelling* Interativo (SSI).

Com o objetivo de embasar os conceitos utilizados, serão abordados os temas de *Storytelling* Interativo (incluindo um breve histórico do desenvolvimento do sistema Logtell (CAMANHO et al., 2009; CIARLINI et al., 2005, 2009)), a Geração Prodecimental de Conteúdo (e alguns de seus usos em jogos comerciais) e finalmente será apresentada a Busca de Caminhos, com algumas alternativas e encerrando-se o embasamento com a apresentação do A*.

2.1 Storytelling Interativo

Muito antes da escrita, histórias eram usadas para transmitir conhecimento entre gerações e entreter os ouvintes. Com o advento de técnicas capazes de armazenar seu conteúdo – i.e. escrita, gravação, filmagem – as pessoas ao redor do mundo começaram a registrar suas histórias de diversas maneiras, formatos e línguas, permitindo que estas atravessassem as fronteiras de suas origens, e o seu conteúdo fosse divulgado, seja para educação ou deleite de seu público.

Contar histórias tornou-se uma forma de arte única, sendo feita de uma enorme variedade de maneiras, seja através de livros, peças teatrais, filmes ou até mesmo em videogames.

Quando olhamos para a história dos videogames, a grande maioria dos jogos lançados inicialmente não tinham um enredo responsável por guiar o jogador através do jogo. Os

primeiros jogos que possuíam uma trama eram jogos de aventura apresentados em formas textuais – como o Colossal Cave Adventure (CROWTHER; WOODS, 1976). Mais jogos do gênero foram lançados em seguida. Vários jogos de aventura e *Role Playing Games* (RPG) foram desenvolvidos, enfrentando os problemas de combinar apresentação gráfica com uma trama atraente, criando franquias famosas como Ultima (ORIGIN SYSTEMS, 1981), Dragon Quest (ENIX, 1986) e Final Fantasy (SQUARE, 1987).

Estes jogos foram a chave para criar um novo gênero de entretenimento. Primeiramente, eles tinham a história como sua componente chave, onde o desenvolvimento dos personagens e do mundo era importante durante o jogo. Além disso, a interatividade era outra componente chave: o jogador precisava explorar o ambiente apresentado a ele e avançar na história. Eles eram os ancestrais do *Storytelling* Interativo.

Storytelling Interativo é uma forma de entretenimento digital onde um contexto é apresentado e o público pode influenciar o enredo, através do controle de um protagonista ou agindo como um diretor dos eventos da narrativa. “O objetivo é desenvolver uma mídia interativa onde a apresentação da narrativa, e sua evolução, podem ser influenciadas em tempo real pelo usuário” (PORTEOUS; CAVAZZA; CHARLES, 2010, tradução nossa).

Embora *Storytelling* Interativo tenha várias similaridades com jogos, a principal diferença reside na geração de suas histórias, as quais devem ser criadas de forma que permitam interação, sejam interessantes para o público e coerentes. Além disto, graças a sua característica interativa, existe todo um cuidado para que a coerência e a consistência da história sejam mantidas enquanto a história é alterada para acomodar a intervenção do usuário.

No que diz respeito à influência do mundo, existem dois estilos primários de tramas em *Storytelling* Interativo: *Plot-Based* e *Character-Based*.

Em sistemas *Character-Based* os “usuários podem interagir fisicamente com os personagens e navegar através do ambiente usando os controles normais de um jogo” (CAVAZZA; CHARLES; MEAD, 2002, tradução nossa).

Já nos sistemas *Plot-Based*, “uma série de regras rígidas, construídas dentro da trama, guiam a narrativa, fazendo com que a intervenção do usuário seja bem mais limitada. Nestes sistemas, existe um controle mais forte da história sendo apresentada, prevenindo que o usuário desvie demais do contexto definido pelo autor. Geralmente, a história inclui uma estrutura composta por pontos fixados anteriormente pelo autor de começo, meio e fim e as interações do usuário afetam apenas a forma que a história irá alcançar estes pontos predefinidos.” (CIARLINI et al., 2008, tradução nossa).

Um dos principais elementos no *Storytelling* Interativo é a geração da história, onde a trama em si é criada, existindo a preocupação de se manter a coerência e, ao mesmo tempo, manter a história interessante para o público (KARLSSON et al., 2006).

Com isto em mente foi desenvolvido o software Logtell (CAMANHO et al., 2009; CIARLINI et al., 2005, 2009) – a ideia por trás deste software é expressar o contexto através de um modelo de lógica temporal e a subsequente verificação das histórias que podem ser geradas.

O Logtell teve como ponto de partida um gerador de enredos baseado em modelagem do contexto das histórias com lógica temporal, o *Interactive Plot Generator – IPG* (CIARLINI, 1999). Ao IPG foi integrado um módulo de dramatização com suporte à interação dos usuários (POZZER, 2005). Para aumentar a diversidade de possibilidades de dramatização das cenas, foi concebido um modelo que utiliza autômatos não-determinísticos (DÓRIA; CIARLINI; ANDREATTA, 2008). Ainda aprimorando esta ferramenta, foi concebido um modelo de dramatização e interação voltado para TV

Digital Interativa (CAMANHO et al., 2009) que possibilitou a geração e dramatização contínua das histórias, dando origem à segunda versão do sistema.

A segunda versão do Logtell já garantia a coerência da história em sua etapa de geração e uma boa diversidade na quantidade de enredos gerados, os quais eram dramatizados automaticamente em capítulos. Até este ponto, a interação com os usuários determinava apenas condições sobre o que deveria acontecer nos capítulos futuros.

De modo a incrementar as possibilidades de interação e a variedade dos enredos, um novo modelo de planejamento não determinístico foi incorporado ao sistema (SILVA, 2010), o qual passou a gerar, para cada capítulo, uma árvore de eventos não determinísticos, os quais podem terminar de diferentes formas. A dramatização do capítulo passou a ter várias alternativas e a seleção dessas alternativas passou a poder ser feita como uma nova opção de interação, aumentando ainda mais a diversidade dos enredos.

A coerência e a diversidade dos enredos, com múltiplas oportunidades de interação, no entanto, não garantem que as histórias sejam interessantes. Um aspecto importante a ser tratado é o atendimento a requisitos referentes às emoções representadas ao longo do enredo.

Visando evoluir o sistema e acrescentar mais detalhes à história, um sistema de matemático de emoções foi criado com base na Roda de Plutchik (PLUTCHIK, 2001), juntamente com um modelo de verificação de restrições temporais, expressas em *Computation Tree Logic* (CTL) (KONUR, 2010) sobre a variação das emoções nas possíveis diversas ramificações da história (ARAUJO, 2011). De início foi proposto um esquema de geração de capítulos dos enredos com subsequente verificação das

propriedades das possíveis ramificações. Quando capítulo não satisfazia restrições, esse era descartado e outra alternativa era gerada.

Este sistema de geração e teste pode acabar desperdiçando muito tempo com ramificações inviáveis para a história. Desta forma, foram criados dois modelos que trabalham em conjunto para garantir uma eficiência em geração de conteúdo em tempo real.

O primeiro modelo realiza “uma pré-verificação exaustiva das restrições sobre propriedades dramáticas em eventos não determinísticos”, utilizando “uma lógica de tempo contínuo ramificado proposta para o contexto de *Storytelling*” e “programação em lógica com restrições para verificar a satisfação de restrições sobre a variação linear de propriedades contínuas ao longo de eventos narrativos abstratos e não determinísticos” (GOTTIN, 2013). O pré-processamento proposto procura mapear as restrições, que podem ter que ser verificadas a partir de um evento não determinístico, nas condições que precisam ser verificadas após o evento. Este leva em conta a variação das emoções nas transições dos autômatos que representam os eventos e como cada evento pode terminar. É feito de forma simbólica sem se conhecer os valores reais das emoções que serão conhecidos apenas quando evento for incluído no enredo. Nas transições entre estados dos autômatos, assume-se a variação linear entre valores das emoções, de modo que a variação ao longo do evento é sempre uma curva linear aos pedaços.

O segundo modelo utiliza o modelo anterior, propondo planejamento com uma rede hierárquica de tarefas que “incorpora a verificação das formulas de lógica temporal durante o planejamento” (FERREIRA, 2013). Desta forma é possível escolher a ramificação certa que mantenha a coerência com um tempo de resposta que o permite ser executado em tempo real.

Depois que a trama principal é gerada, existe a necessidade de apresentar a história ao público, onde a forma de apresentação já é um grande desafio por si só. Como não se sabe a priori o valor das emoções a serem representadas em cada evento, ocorre que é necessária a criação de mecanismos capazes de representar em tempo real a variação de emoções ao longo dos eventos da história. Isso motivou a pesquisa descrita nesta dissertação. Embora tenha surgido em razão das demandas de dramatização no Logtell, a representação dinâmica das emoções é um problema geral em *Storytelling* interativo.

Além da necessidade de se adaptar a dramatização das emoções ao enredo, deve-se considerar que o cenário e demais elementos da dramatização podem ter que ser gerados dinamicamente. Ao considerar um método de apresentação que exhibe um mundo tridimensional renderizado por computador para o público, é necessário que sejam utilizados vários elementos funcionais para que a apresentação seja convincente e agradável, similar a uma criação cinematográfica – e.g. atores virtuais, posicionamento da câmera, modelagem de cenário, ajuste de luz.

Com o objetivo de reduzir o esforço humano necessário para a criação de cada um destes componentes e ao mesmo tempo ganhar uma certa flexibilidade autoral com custo menor, é possível utilizar soluções que automatizem a criação de boa parte deste conteúdo. Uma vez que esse conteúdo seja gerado dinamicamente, passa a ser necessária a adequação, em tempo real, das ações e da representação das emoções a ele.

2.2 Geração Procedimental de Conteúdo

Uma das técnicas que podem ser empregadas na automatização é a Geração Procedimental, onde um programa de computador é responsável por gerar conteúdo – e.g. terreno, ambiente, iluminação, texturas, entre outros – de forma algorítmica.

Os primeiros jogos de computador tinham sérias restrições em sua execução, em especial de memória e armazenamento. A solução passou a ser gerar o conteúdo de forma algorítmica – não era possível armazenar uma grande quantidade de níveis ou arte criados anteriormente: Até o final da década de 80, o maior tamanho comumente disponível de disquete era de 720kb.

Para possibilitar a variedade nos elementos gerados a partir do mesmo algoritmo são utilizados geradores de números pseudoaleatórios onde, a partir de um número inicial, chamado “semente”, uma sequência de outros números aparentemente aleatória é gerada.

Os primeiros jogos oficialmente a utilizar geração procedimental (GUINNESS WORLD RECORDS, 2014) são: Elite (BRABEN; BELL, 1984), utilizando geração procedimental do seu mundo, Frontier: Elite II (BRABEN, 1993), que gera sistemas estelares procedimentalmente e Frontier: First Encounters (FRONTIER DEVELOPMENTS, 1995), o primeiro jogo a conter terreno e texturas geradas procedimentalmente. Embora de forma não oficial, há mais jogos que utilizaram técnicas de geração procedimental antes destes citados, como Akalabeth: World of Doom (GARRIOTT, 1980), que começava pedindo ao jogador um “número de sorte” e utilizava este número para criar os atributos do personagem e os calabouços pelos quais o jogador se aventuraria.

Graças aos avanços tecnológicos, a capacidade de armazenar dados, atualmente, é bem superior, possibilitando que os jogos atuais possuam uma quantidade de modelos, texturas e mapas (entre outros) muito superior aos anteriores. No entanto, os jogadores estão cada vez mais críticos e, em um contexto onde o fotorrealismo se tornou um padrão da indústria, cada vez menos são aceitos elementos duplicados ou pouco detalhados. A utilização de muitos elementos repetidos pode ameaçar a Suspensão de Descrença, diminuindo a imersão e interesse na mídia apresentada.

A Suspensão de Descrença ocorre quando o “autor cria uma narrativa ficcional ou fantástica que é suficientemente envolvente para fazer com que o leitor acredite no que não está lá – ou seja, ter uma ‘fé poética’” (PERLA; MCGRADY, 2011, tradução nossa). Embora originalmente cunhado para a mídia escrita, este termo passou a ser utilizado para outras mídias, como filmes e entretenimento digital. A utilização de técnicas que possam auxiliar a manter esta característica e aumentar o potencial criativo são importantes.

“Técnicas de Geração Procedimental vêm ganhando muita popularidade” (NAREYEK, 2007, tradução nossa). Para atender à demanda de uma variedade de elementos em seus jogos, a indústria de jogos começou a empregar cada vez mais geração procedimental em seu fluxo de trabalho, tanto diminuindo os custos para gerar arte manualmente, quanto ganhando muito tempo para a criação de artefatos valiosos para seus jogos – inclusive criando mecânicas diferenciadas em certos títulos.

Alguns exemplos representativos de uso de geração procedimental ocorrem em alguns jogos. Em *Left 4 Dead* (VALVE, 2008), um módulo de software – chamado pela empresa de “Diretor” – dispõe os inimigos que o jogador irá enfrentar nos mapas de acordo com o desempenho dos jogadores que estão jogando o jogo no momento, graduando a dificuldade em tempo real. Os jogos *Minecraft* (MOJANG, 2011) e *Terraria* (RE-LOGIC, 2011) utilizam um módulo de software responsável por criar por completo o ambiente em que o jogador se aventurará, cada vez que um novo mundo é criado. Em *RoboBlitz* (NAKED SKY ENTERTAINMENT, 2006) é utilizada uma versão modificada do software ProFX (ALLEGORITHMIC, 2006) para gerar a maior parte das texturas utilizados em seu jogo em tempo real, permitindo uma maior riqueza visual e ao mesmo tempo mantendo o tamanho total do jogo pequeno.

Na elaboração de sistemas de *Storytelling* Interativo, existe um esforço grande para criar os componentes usados na apresentação das várias alternativas de histórias que

podem ser geradas. Com esta variedade, a quantidade de conteúdo necessário começa a crescer exponencialmente, demandando o uso de técnicas que automatizem ao máximo a geração do conteúdo utilizado. Como uma vantagem extra, permitir que este conteúdo seja gerado por computador permite uma maior experimentação e exploração das possibilidades da história ainda durante as fases iniciais de criação por parte dos autores da história.

Em *Storytelling* Interativo, vários elementos importantes podem ser gerados de forma procedimental, desde os atores, através da combinação de características visual destes – i.e. vestimentas, brincos, chapéus, armas entre outros – até cenários que podem ter tamanhos, relevo, vegetação, ambientação e caminhos variados.

Com isto em mente, é necessário criar ferramentas que garantam a produtividade, permitindo uma geração de conteúdo mais ágil, desde os estágios iniciais de planejamento da história até os passos finais para apresentação. Também é importante conceber componentes reutilizáveis o tanto quanto possível, reduzindo o esforço autoral nas iterações de melhoria da história e de histórias futuras.

Ao revisitar o caso dos cenários gerados procedimentalmente, nota-se que é possível criar uma grande quantidade de ambientes para apresentação em tempo muito menor do que se cada um destes ambientes fosse criado manualmente, um a um, por um artista. O ganho efetivo é grande, permitindo tanto explorar várias novas opções para uma mesma cena, assim como ter uma variedade grande no momento de apresentação para o público, com um custo muito menor.

No entanto, isto cria um novo desafio para a apresentação de histórias de *Storytelling* Interativo – A movimentação de atores virtuais dentro destes novos cenários. Entre outros fatores, o ambiente pode ser completamente desconhecido até a hora da dramatização,

dependendo da estratégia utilizada. Além disto, pode ser que apenas a movimentação entre dois pontos do mapa não seja o suficiente para uma cena, existindo a necessidade de que os caminhos passem por pontos chaves neste ambiente desconhecido que causem um impacto emocional na cena. Para solucionar isto é possível explorar estratégias de busca de caminhos mais sofisticadas porém eficientes. De modo a se complementar a fundamentação teórica deste trabalho, é feita uma revisão de busca de caminhos a seguir.

2.3 Busca de Caminhos

A busca de caminhos, ou *pathfinding*, pode ser definida como uma solução que procura encontrar o melhor caminho entre dois pontos. Geralmente, especialmente em jogos, o melhor caminho é associado ao caminho espacialmente mais curto, embora outros critérios possam ser definidos em outros cenários.

Esta busca é realizada em cima de um grafo definido no domínio do problema, onde é dado um nó inicial para a busca e um nó final que pretende ser alcançado. “Um grafo G é definido como sendo um conjunto $\{n_i\}$ de elementos chamados de nós e um conjunto $\{e_{ij}\}$ de segmentos de reta chamados de arcos que ligam nós. Se e_{pq} é um elemento do conjunto $\{e_{ij}\}$ então dizemos que há um arco do nó n_p para o nó n_q .” (HART; NILSSON; RAPHAEL, 1968, tradução nossa). Grafos podem ser direcionados ou não direcionados. No primeiro caso, há uma direção explícita de n_p para n_q e n_q é dito sucessor de n_p . No segundo caso, assume-se que os arcos podem ser percorridos em ambas as direções.

Existem várias formas de solucionar um problema de busca de caminho. Uma solução simples seria tentar encontrar o caminho por força bruta, como, por exemplo, ir visitando todos os nós de forma sistemática até encontrar um caminho. As soluções clássicas mais

conhecidas são a *busca em largura* (*breadth-first search* – BFS) e *em profundidade* (*depth-first search* – DFS).

A estratégia utilizada na busca em largura é simples: começa-se visitando um nó e acrescentando todos os seus vizinhos em uma fila para visita futura. O processo é repetido até ser encontrado o nó alvo ou se exaurirem as opções de busca. Na Figura 1 é possível ver a ordem de visitação em uma árvore simples.

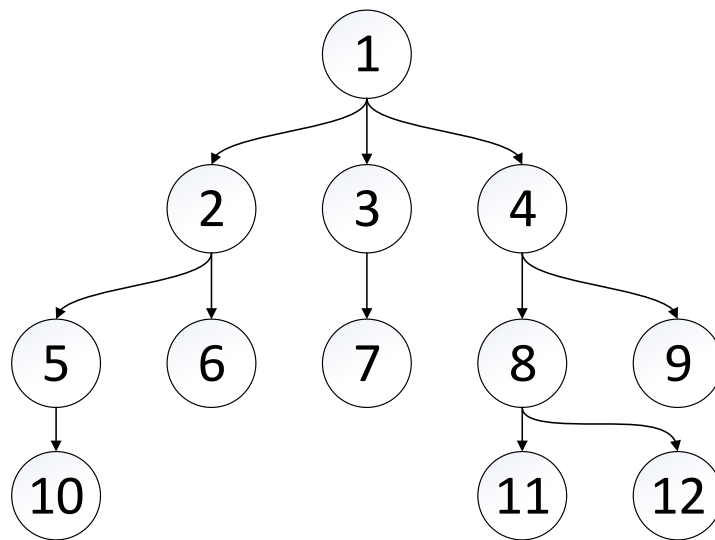


Figura 1: Ordem de visitação dos nós no BFS

A busca BFS garante retornar o melhor caminho para buscas espaciais e de custo fixo. No entanto, a BFS sofre sérios problemas de armazenamento em memória, podendo esgotar rapidamente este limite.

Uma solução alternativa é utilizar a busca em profundidade, na qual a estratégia utilizada é gulosa: ao encontrar vizinhos, ir visitando a partir do primeiro encontrado até o limite do grafo, retornando quando não houver mais vizinhos a serem visitados naquela ramificação. Tomando os devidos cuidados para evitar *loops*, pode conseguir achar um caminho sem ter que armazenar todos os nós em memória, embora não garanta a qualidade do caminho retornado. Na Figura 2 é possível ver a ordem de visitação do DFS.

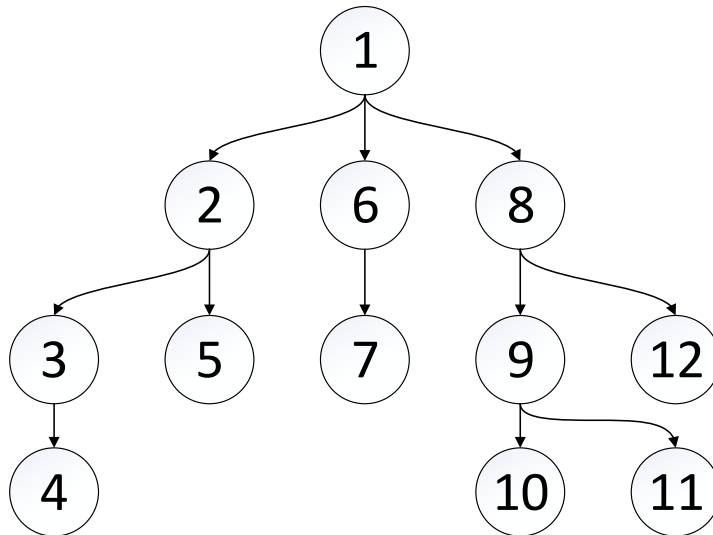


Figura 2: Ordem de visitação dos nós no DFS

Ambas as estratégias de força bruta tornam-se rapidamente inviáveis tanto no domínio de jogos quanto de *Storytelling*, BFS alcança rapidamente os limites de memória e DFS não fornece garantia sobre a qualidade do caminho encontrado.

Através dos anos, várias soluções foram propostas para problemas de buscas em grafo, no entanto uma em especial se tornou muito popular, o A* (HART; NILSSON; RAPHAEL, 1968).

O algoritmo A* caracteriza-se por retornar o caminho de menor custo dado um nó inicial e um nó *goal* e, enquanto o A* navega pelo grafo, ele mantém uma fila de prioridades de caminhos alternativos promissores.

A principal peculiaridade do A* é a função que ele utiliza para calcular qual é o próximo caminho a ser visitado. Comumente denotada como $f(x) = g(x) + h(x)$, a função $f(x)$ é a soma de:

- Função de custo do caminho do passado ou $g(x)$, que é a distância conhecida do nó inicial até o nó x .

- Função de custo do caminho futuro ou $h(x)$, que é uma estimativa heurística da distância de x para o *goal*.

Para garantir que a solução encontrada seja necessariamente o caminho de menor custo, $h(x)$ deve ser admissível, ou seja, nunca deve superestimar a distância até o *goal*.

Além disto, se a heurística h satisfizer a condição adicional $h(x) \leq d(x, y) + h(y)$ para cada nó do grafo, onde d é a distância entre dois nós – ou seja, que a estimativa dada pela heurística aplicada a um nó não ultrapassa a estimativa da heurística aplicada ao vizinho deste nó somada ao custo do passo para alcançar este vizinho – esta heurística h é consistente e a busca pode ser implementada de forma ainda mais eficiente, evitando que cada nó seja processado mais de uma vez.

Estas características fizeram com que o A* se tornasse muito popular, permitindo uma implementação eficiente que utiliza um baixo custo computacional, passando a ser utilizada amplamente em jogos.

Assim como nos jogos, é necessário que no *Storytelling* Interativo atores virtuais se movimentem de um ponto a outro da cena. Uma solução simples seria utilizar o A*, assim como nos jogos. No entanto isto nos leva à questão: quando um ator se movimenta em cena, ele deve ter como principal objetivo encontrar o caminho mais curto entre dois pontos?

A movimentação em filmes e peças teatrais geralmente está associada à continuidade da história, ou seja, quando um personagem envolvido na trama se locomove, existe um propósito, raramente acontecendo a esmo.

Além de se mover por um motivo, existe um fundo emocional que pode estar associado à movimentação em cena – e.g. um personagem que passa por um beco escuro tende a ficar mais assustado. Como muitas vezes a tensão da cena é transmitida ao público, o

autor da história pode querer causar este efeito emocional em momentos específicos, onde a escolha do caminho deve depender disto. A escolha do caminho complexo que atende à intenção do autor é um instrumento narrativo que pode alavancar os mais variados tipos de cena.

A* por si só não resolve a complexidade da movimentação esperada em cena, apenas buscando o caminho mais curto entre dois pontos. Este trabalho trata exatamente da influência de emoções na busca de caminhos. Antes de se tratar o problema específico, cabe no entanto explicar o funcionamento do A*.

2.3.1 Pseudocódigo do A*

O A* pode ser implementado de diversas formas, de acordo com o domínio em que será utilizado. No domínio deste problema, será utilizada a técnica em que os nós já visitados são armazenados em um conjunto, evitando que estes sejam expandidos mais de uma vez.

Para apresentar o pseudocódigo do A*, serão definidas uma estrutura e três funções auxiliares:

```
1. struct Path
2. {
3.     Path PreviousSteps
4.     Node LastStep
5.     float CurrentCost
6. }
7.
8. function AddStep( Path oldPath, Node node, float Cost ) : return Path
9. {
10.     return new Path( oldPath, node, Cost )
11. }
12.
13. function EuclideanDistance( Node n1, Node n2 ) : returns the distance between
14.                                                     two given nodes
15. function StepCost( Node n1, Node n2 ) : returns the cost to move between two
16.                                         neighboring given nodes
```

Pseudocódigo 1: Funções Auxiliares para o A*

Na linha 1, a estrutura **Path** (caminho) é definida, a qual armazena a sequência dos passos anteriores, o último passo dado e o custo espacial atual do caminho, ou seja, o caminho será uma lista de passos e o custo agregado do caminho.

Na linha 8 é definida uma função que anexa um novo passo ao caminho, retornando um novo caminho com o valor de custo atualizado.

Nas linhas 13 e 15 são definidas mais duas funções auxiliares. A primeira, **EuclideanDistance**, define a distância euclidiana entre dois nós dados e a segunda, **StepCost**, define o custo real de movimentação entre dois nós vizinhos. Ambas não possuem um corpo de código por serem específicas do domínio.

É importante notar que, conforme vai ser detalhado mais à frente, a **EuclideanDistance** é utilizada como função heurística. No entanto, esta não é a única heurística utilizada no A*, embora isto esteja fora do escopo deste trabalho.

Com isto definido, é apresentado o pseudocódigo do A*, cuja funcionalidade é explicada a seguir, linha a linha, para esta possível implementação:

```

1. function AStar( Node start, Node goal ) : return path on success; null on failure
2. {
3.   closedSet = Empty Set of Node
4.   pathQueue<Path, Cost> = { ( start, 0 ) } // A Priority Queue
5.
6.   while ( pathQueue not empty )
7.   {
8.     path = Dequeue cheapest path in ( pathQueue )
9.
10.    if ( path.LastStep is in closedSet ) continue
11.
12.    if ( path.LastStep == goal ) return path
13.
14.    Add path.LastStep to closedSet
15.
16.    for each ( neighbor of path.LastStep )
17.    {
18.      gx = path.CurrentCost + StepCost( Path.LastStep, neighbor )
19.      hx = EuclideanDistance( neighbor, goal )
20.      newPath = AddStep( path, neighbor, gx )
21.      pathQueue.Enqueue( newPath, gx + hx )
22.    }
23.  }
24.
25.  return null
26.}

```

Pseudocódigo 2: Pseudocódigo do A*

Na primeira linha é definido que a função **AStar** recebe um nó inicial e um nó final e retorna o caminho encontrado ou nulo, caso não encontre um caminho válido.

Nas linhas 3 e 4, são declaradas as variáveis internas que auxiliarão na navegação do grafo, onde **closedSet** armazena conjunto dos nós já visitados, evitando uma possível repetição no caso da heurística não ser consistente e **pathQueue** é uma fila de prioridades que ordena os caminhos mais promissores, os colocando em ordem crescente de acordo com o seu custo. Inicialmente, **pathQueue** contém apenas o nó inicial, sem custo no caminho.

Na linha 6, começa um *loop* que vai até a linha 23, este *loop* é executado enquanto ainda existam nós a serem processados na fila de prioridades **pathQueue**.

Na linha 8, o caminho com o custo mais barato armazenado em **pathQueue** é removido e armazenado na variável **path**, onde este será trabalhado internamente pelo *loop*.

Na linha 10 é feita uma verificação se o último passo deste caminho já foi visitado antes. Caso tenha sido visitado, o *loop* recomeça, descartando este candidato.

Na linha 12 é verificado se o último passo neste caminho chega ao *goal*, completando a busca. Caso isto tenha acontecido, é retornado o caminho encontrado e o algoritmo encerra sua execução.

Na linha 14, o último passo é acrescentado ao conjunto **closedSet**, para evitar que as etapas seguintes sejam executadas mais de uma vez no mesmo nó.

Na linha 16 começa um novo *loop*, que é executado para cada vizinho do último passo do caminho atual – i.e. para cada vizinho, as linhas 18 a 21 são executadas.

Na linha 18 é calculado o $g(x)$, o qual é a soma do custo real dos passos do nó inicial até o nó atual, somado com o custo do passo para o vizinho atual.

Na linha 19 é calculado $h(x)$, que é a função heurística para o A*. Neste caso, é a distância euclidiana entre o vizinho e o *goal*.

Na linha 20, a função auxiliar **AddStep** é executada, criando um novo caminho que é constituído anexando o vizinho atual ao caminho. Este novo caminho então é enfileirado na linha 21, de acordo com a função $f(x) = g(x) + h(x)$.

Isto é repetido até o caminho ser encontrado. Caso não seja encontrado, a linha 25 retorna nulo indicando que o caminho não pode ser encontrado.

2.3.2 Algoritmo de Busca A* em execução

Como já detalhado anteriormente, o A* precisa de um Grafo G para realizar sua busca, um nó inicial e um nó *goal* para o qual deve procurar um caminho e trabalha com uma função $f(x) = g(x) + h(x)$, a qual utiliza o $g(x)$ como o custo de travessia até determinado nó e $h(x)$ como uma heurística que estima um custo de quanto falta para se chegar até o *goal*.

O A* foi inicialmente concebido como um algoritmo de busca em grafos. Isto permite que sejam utilizados grafos com um forte vínculo a informação espacial. No ambiente de Jogos e de *Storytelling* Interativo, existe a necessidade de formar o grafo de acordo com o ambiente em que os atores virtuais irão caminhar. Em razão da geração procedimental de ambientes, é necessária uma maneira de extrair o grafo levando em consideração os obstáculos do terreno, inclinação entre outros fatores. A extração destas informações e a geração do grafo em si serão tratadas em maior detalhe na seção 3.1.1. Por hora, pode-se assumir que grafo já está definido. O grafo convertido a partir de um ambiente/terreno bidimensional ou tridimensional, geralmente toma a forma de um “mapa quadriculado” (e.g. Figuras 5 e 6 da seção 3.1.1, onde este processo é melhor detalhado).

Com a finalidade de diminuir o número de alternativas para os caminhos, podem ser utilizados grafos criados manualmente com *waypoints* – i.e. grafos onde os artistas definem manualmente os lugares do mapa por onde os personagens deverão caminhar.

Na Figura 3 é possível ver um exemplo com uma configuração por *waypoints*. Os nós do grafo são os pontos principais do ambiente e as arestas são os caminhos que ligam estes nós.

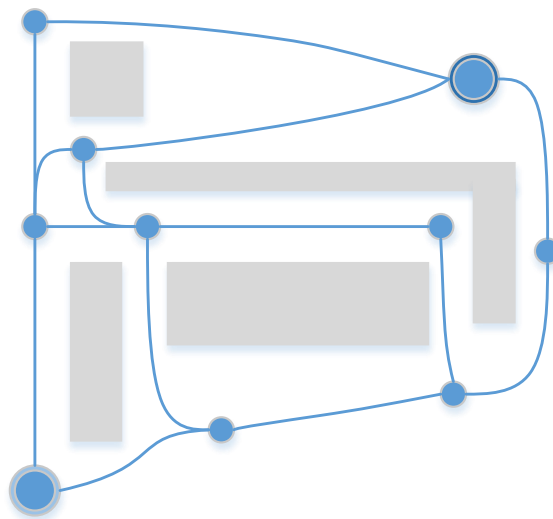


Figura 3: Exemplo de grafo com Waypoints.

Embora o uso de waypoints possa aumentar a eficiência dos algoritmos, o seu maior problema é que estes precisam de trabalho manual para serem criados e conectados, ou seja, os artistas responsáveis por criar os cenários, além do esforço gasto para criar o cenário, também têm que criar os caminhos possíveis para movimentação dos personagens. Embora isto possa ser desejável em casos específicos, o acúmulo de carga nos artistas apenas causa mais demora na criação de conteúdo.

Dessa forma, torna-se desejável não se depender da definição de *waypoints* pelos usuários. Em especial, com a geração procedimental dos ambientes, é necessário extrair as informações diretamente do ambiente gerado e executar a busca em cima deste grafo gerado, como é possível ver na Figura 4.

É importante notar o papel que a heurística tem na busca. Na Figura 4 é possível ver três heurísticas diferentes sendo utilizadas sob um grafo de pontos equidistantes na horizontal e na vertical, onde:

- Em (a) é utilizada uma heurística que sempre retorna zero, o que faz com que os nós sejam expandidos de forma circular, uniformemente, até que seja encontrado o *goal*. No total, são expandidos 773 nós até encontrar a solução;
- Em (b) é utilizada a distância euclidiana, direcionando a busca ao *goal*. 379 são expandidos até encontrar o caminho;
- Em (c) é utilizada a distância euclidiana multiplicada por 5, tomando um rumo muito mais direto ao *goal*. São expandidos apenas 71 nós até encontrar a solução, no entanto, esta solução possui um comprimento maior do que a solução ótima (pois a heurística não é admissível).

Ainda na Figura 4, os círculos preenchidos com cores que variam entre vermelho e verde representam a distância do *goal*, os círculos azuis vazados são candidatos para o caminho que não foram expandidos e os círculos azuis preenchidos são possíveis candidatos que não foram expandidos por terem o caminho obstruído por obstáculos. O caminho encontrado em cada caso é representado com a cor preta.

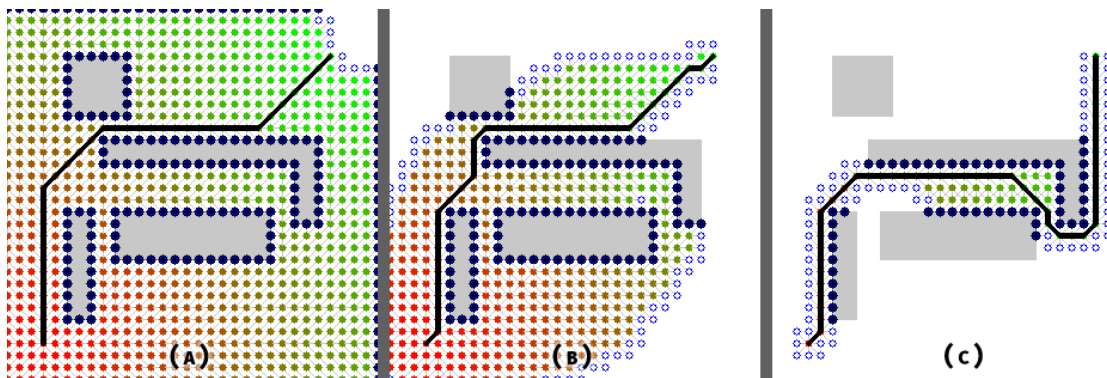


Figura 4: O A* aplicado com três heurísticas diferentes.

Neste trabalho, procura-se uma busca de caminhos capaz de tratar as variações de emoção do personagem para atender às expectativas do autor. No entanto, dois outros modelos foram propostos anteriormente para lidar com emoções durante uma busca.

O primeiro modelo (DONALDSON; PARK; LIN, 2004) lida com um sistema de prioridades de emoções – onde este trata estados fisiológicos (fome e sede) como emoções – procurando atender o estado prioritário. Não há um controle de como o personagem irá se comportar, apenas modificando o *goal* para atender a estado emocional que possui maior prioridade.

O segundo modelo (JOHANSSON; DELL'ACQUA, 2012) propõe um sistema de buscas de caminho que atribui informações emocionais ao grafo e tenta encontrar um bom caminho. Este modelo se importa em fazer com que o personagem prefira áreas de alegria e evite áreas de tristeza e de medo. Conformes eventos aconteçam durante a execução, uma nova área é marcada de acordo com a emoção deste evento, fazendo com que a próxima vez que seja necessário procurar uma nova rota, os eventos anteriores alterem a escolha do caminho.

Estas soluções, no entanto, não permitem tratar o objetivo de gerar uma determinada variação emocional de acordo com a vontade do autor, não sendo adequadas para o *Storytelling* Interativo.

O próximo capítulo apresenta uma nova proposta de modelo para tratar a variação de emoções na escolha de caminhos em um contexto de *Storytelling* Interativo.

3 Modelo

Este capítulo é dividido em 3 seções. A primeira seção é responsável por definir o problema de buscas de caminhos com influência de emoções, quais dados compõem este problema e quais são os requisitos para a solução deste problema.

A segunda seção descreve o modelo proposto para solucionar o problema.

A terceira seção trata da implementação do modelo proposto, exibindo o pseudocódigo para a solução proposta, sua explicação e a interface das ferramentas implementadas com base neste modelo.

3.1 Definição do Problema

De um ponto de vista de alto nível, o problema de tratamento da variação de emoções na busca de caminhos pode ser definido como: dado um cenário, um ponto inicial e ponto final (*goal*), encontrar um caminho que faça sentido emocionalmente e visualmente.

3.1.1 Dados de Entrada

A especificação do problema contém os seguintes dados (que serão abordados detalhadamente a seguir):

- um cenário;
- o grafo que representa os caminhos possíveis neste cenário;
- informações de variação de emoção dadas pelo cenário;
- um ponto inicial;
- um ponto final (*goal*);

- uma curva emocional de referência (a curva resultante da movimentação deve ser tão próxima quanto possível desta curva de referência).

O cenário pode ser gerado proceduralmente ou manualmente, o único requisito é que seja possível extrair duas informações deste: as informações de caminhos – ou seja, um grafo que determina em quais lugares o ator virtual pode caminhar e onde não pode – e as informações que indicam como a emoção é alterada ao longo do caminho.

O grafo que determina os lugares onde a movimentação pode ocorrer pode ser extraído utilizando várias técnicas que vão além do escopo desta dissertação. O grafo resultante precisa ser conexo, onde os nós determinam os pontos do mapa por onde o ator virtual pode ficar de pé – i.e. existe uma estrutura (e.g. chão) para dar suporte ao ator – e os arcos devem marcar a possibilidade de deslocamento entre cada um dos nós do grafo, com o devido suporte necessário – i.e. o terreno é contíguo, sem rupturas entre um ponto e outro quando ligados por um arco. Na Figura 5 é possível ver o terreno base à esquerda e o grafo resultante de sua extração sobreposto ao terreno original. Na Figura 6 é possível ver o grafo isoladamente.

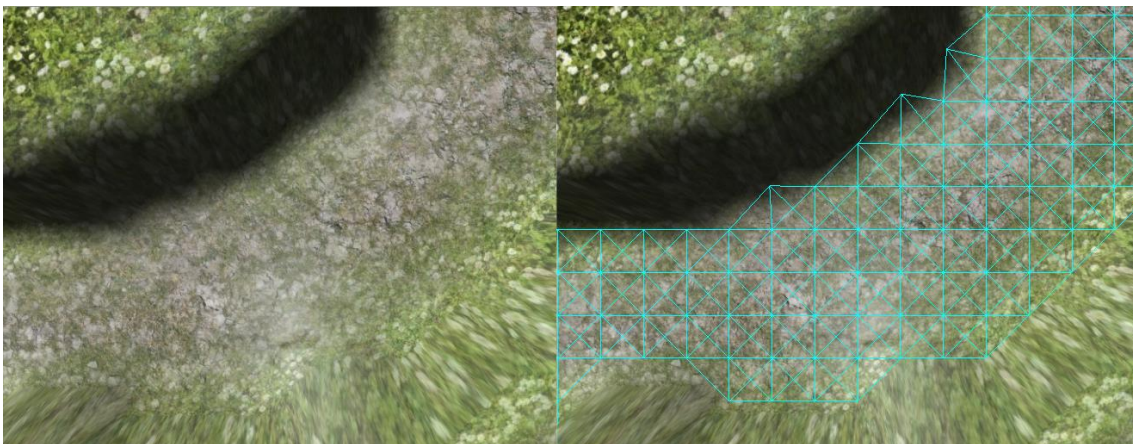


Figura 5: Terreno base e Grafo resultante sobreposto

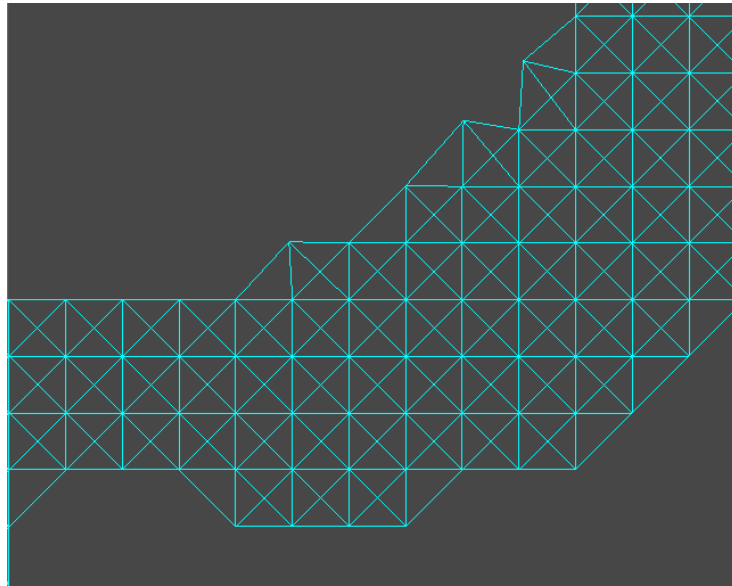


Figura 6: Grafo resultante da extração

As informações de emoção atribuem um “custo” de emoções por utilizar o nó no caminho, ou seja, quando o ator passar por aquele nó, sua emoção será modificada de acordo com o valor atribuído ao mesmo. Idealmente, estes valores são pequenas variações emocionais atribuídas a uma pequena área de pontos no grafo, onde é possível existir várias áreas diferentes com variações diferentes pelo grafo. Na Figura 7, é possível ver um grafo onde as informações de emoção são representadas por quadrados em seus nós: os quadrados pretos aumentam, por exemplo, o medo quando o ator passa por eles, os cinzas não alteram a emoção e os brancos diminuem o medo.

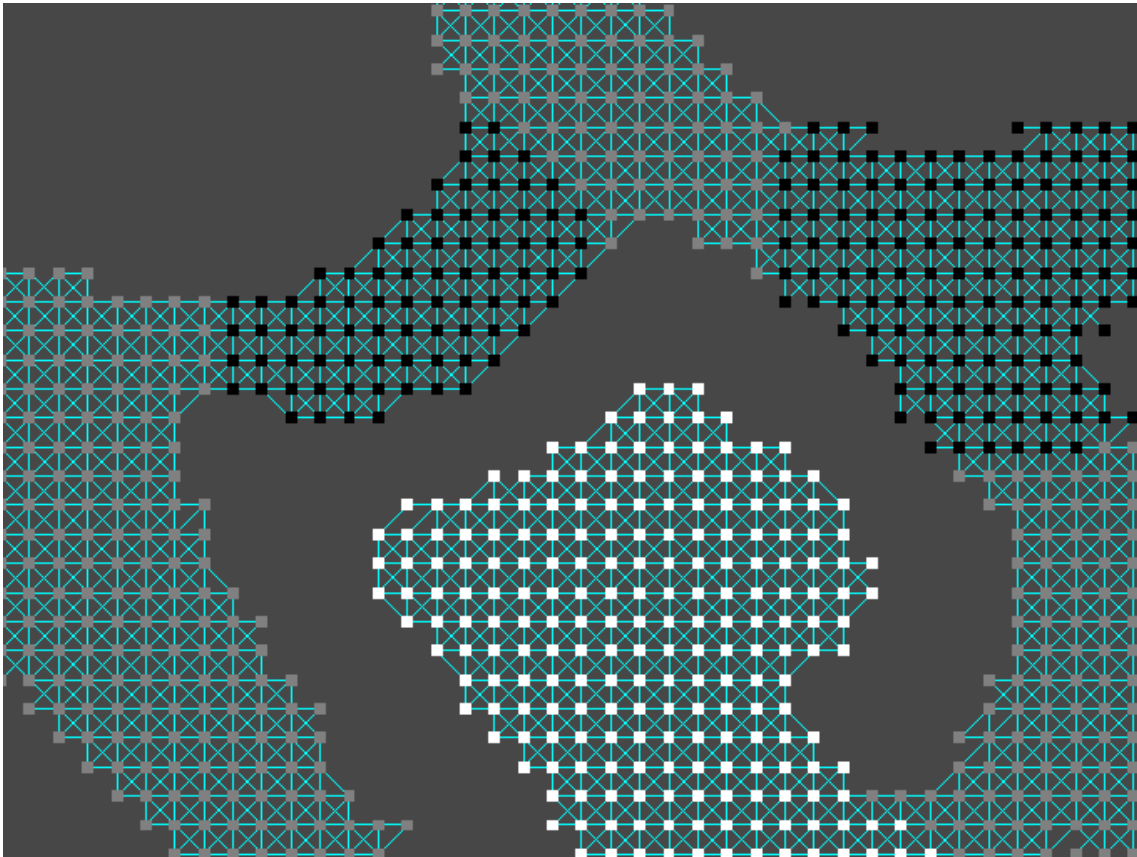


Figura 7: Grafo com informação emocional.

A curva emocional de referência é uma maneira de descrever a variação emocional que o autor da história espera que aconteça na cena. Esta curva é utilizada pelo algoritmo para definir o estado emocional inicial e avaliar o quão próximo da variação de emoção pretendida pelo autor está o caminho. A curva é tipicamente linear aos pedaços. O algoritmo reavalia a proximidade da curva emocional de referência a cada passo da busca, utilizando uma estimativa do comprimento total da trajetória e assumindo uma velocidade de movimentação constante.

Para simplificar tanto o desenvolvimento quanto a usabilidade da ferramenta, os valores emocionais foram modelados para tipicamente variar de 0 a 100 – e.g. o autor pode querer que uma determinada emoção de um personagem aumente de 20 para 40 na primeira metade da cena e depois retorne a 20 na segunda metade.

3.1.2 Requisitos para o modelo

Na definição em alto nível do problema foi dito que é necessário “encontrar um caminho que faça sentido emocionalmente e visualmente”. O melhor caminho em *Storytelling* Interativo, onde são consideradas mudanças emocionais durante a trajetória do personagem, é aquele que permita ao público perceber a variação emocional determinada pelo enredo e que não cause estranheza.

Portanto, o caminho precisa encontrar um balanceamento que respeite os três requisitos básicos:

- vá no sentido do objetivo, nunca se distanciando por demais na tentativa de cumprir apenas a emoção;
- aproxime-se o máximo possível da curva emocional a ser cumprida;
- seja viável gerar em tempo real com áreas grandes.

O requisito referente à variação emocional demanda que se procure uma movimentação que se aproxime da curva emocional pretendida pelo enredo – e.g. o personagem começa a cena com o medo em 20 e este medo sobe gradativamente visando atingir 60 no final da cena. Nem sempre será possível atingir este valor por diversos fatores, como cenário não possuindo áreas de variação emocional suficientes, cenário pequeno demais para uma mudança grande demais, entre outros.

Visualmente, o caminho precisa fazer sentido, ou seja, existe a necessidade de parecer que o personagem está sempre indo em direção ao *goal*, mesmo que não tenha escolhido a rota mais curta para alcançar o mesmo. Ainda assim, existe a necessidade de evitar que voltas desnecessárias ocorram em uma área que provoca variação emocional – e.g. visando aumentar o “medo”, não faz sentido que o personagem fique andando em círculos

em frente a um beco escuro. Caso isto aconteça, pode haver uma quebra na Suspensão de Descrença e prejudicar a imersão do público na trama.

O balanceamento com o caminho mais curto visa orientar o personagem para seguir em uma direção que caminhe para o *goal*, o tanto quanto possível – e.g. geralmente não faz sentido um personagem começar andando na direção contrária do *goal* para atender a uma variação emocional, quando a mesma pode ser alcançada através de outros caminhos. Com o objetivo de encontrar um bom equilíbrio, assume-se que a solução está encontrada entre o comprimento mínimo do A^* e duas vezes este comprimento.

Existe mais um critério que precisa ser cumprido enquanto se procura por um caminho adequado: o tempo de busca do caminho ideal. A procura precisa ser realizada dentro de um tempo curto, para que um bom caminho seja encontrado assim que uma nova cena for especificada pelo enredo, com uma determinada variação emocional, e tiver que ser dramatizada. É importante lembrar que o cenário pode ter sido gerado procedimentalmente e, nesse caso, pode acontecer de nem o ambiente de movimentação ser conhecido previamente.

3.2 Modelo Proposto

Com os dados de entrada em mãos e sabendo como deve ser o formato da saída esperada, é hora de definir o modelo de como transformar estes dados em uma solução.

Inicialmente é executado o A^* em cima do grafo, retornando o caminho mais curto possível entre o ponto inicial e o *goal*. Desta forma é possível saber que o caminho emocional nunca será menor que este.

Conforme definido anteriormente, o A^* utiliza a fórmula $f(x) = g(x) + h(x)$ para atribuir um custo aos candidatos e os enfileirar em um *priority queue*.

O cerne do modelo é constituído pela modificação dos componentes $g(x)$ e $h(x)$ desta fórmula, onde o componente $g(x)$ passa a normalizar o custo espacial e agrega o valor emocional normalizado, enquanto o $h(x)$ passa a ser a normalização da heurística. A necessidade de cada um dos componentes utilizados na formula será explicada após a apresentação da mesma.

Para realizar esta alteração, precisaremos dos seguintes elementos:

- $d(x)$ é o custo do nó inicial s até o nó x ;
- Γ é o custo do caminho encontrado pelo A^* ;
- w_1 é a constante definida para estabelecer o peso do comprimento do caminho na escolha das alternativas;
- Δe_x é a diferença, para cada ponto ao longo do caminho, entre o estado emocional ideal, de acordo com a curva emocional passada, e o estado emocional atual para o caminho já percorrido, para o nó candidato x ;
- Δm_x é a diferença emocional, para cada ponto ao longo do caminho, no cenário de pior caso (ou seja, para a pior curva emocional de referência que poderia ter sido estabelecida) – para esse cálculo, assume-se que a emoção é mapeada com valores entre 0 e 100; para valores menores que 50, adota-se $(100 - valor)$, para valores maiores que 50, adota-se o próprio $valor$;
- w_2 é a constante usada para estabelecer o peso que a proximidade da curva emocional de referência deve ter na escolha das alternativas.
- h_0 é uma função heurística para estimar o comprimento do caminho a partir de um ponto – neste trabalho é utilizada a distância euclidiana.

Com estes elementos definidos, a formula para as funções $g(x)$ e $h(x)$ podem ser definidas como:

$$g(x) = \left[\left(\frac{d(x)}{2 * \Gamma} \right) * w_1 \right] + \left[\frac{\sqrt{\sum(\Delta e_x)^2}}{\sqrt{\sum(\Delta m_x)^2}} * w_2 \right]$$

$$h(x) = \frac{h_0(x)}{2 * \Gamma} * w_1$$

Com os componentes devidamente enumerados, é possível explicar algumas peculiaridades:

- Existe a necessidade de normalizar os valores emocionais e espaciais para que seja possível somar grandezas de tipos diferentes.
 - Com a abstração que a solução está entre o comprimento do caminho encontrado pelo A* (referenciado como Γ) e duas vezes este comprimento, a normalização espacial é realizada utilizando $(2 * \Gamma)$.
 - Para as emoções, a normalização é realizada utilizando o cenário de pior caso para uma curva emocional que pudesse ser estabelecida.
- A ponderação dos valores espaciais e emocionais é utilizado como um mecanismo de ajuste entre cada uma destas partes, balanceando seu uso na formula.
- $g(x)$ é modificado para considerar a parte emocional e a parte espacial. Como a estimativa da emoção desejada em $g(x)$ depende da estimativa do comprimento do caminho, este é reavaliado a cada passo da busca. Isto será abordado com mais detalhes na seção 3.3.1.
- $h(x)$ leva em consideração apenas a parte referente ao custo de comprimento do caminho, pois não foi possível encontrar uma heurística para a parte

emocional com o custo computacional baixo – essa escolha é equivalente a assumir de forma otimista que a aproximação da curva emocional no futuro é sempre perfeita.

3.3 Implementação

Com a definição do problema e um modelo proposto de como solucionar este problema, foi definido um algoritmo correspondente a este modelo. A implementação será abordada na forma de um pseudocódigo explicado linha a linha.

Em seguida, é apresentada a interface do programa criado para auxiliar na depuração do algoritmo juntamente com um analisador de curvas de emoção.

3.3.1 Pseudocódigo

Assim como no A*, é necessário definir funções que irão auxiliar na busca em si. Abaixo serão definidas a estrutura e as funções auxiliares utilizadas. Algumas funções são idênticas às apresentadas na seção 2.3.1.

```

1. struct Path
2. {
3.     Path    PreviousSteps
4.     Node    LastStep
5.     float   NormalizedSpatialCost
6. }
7.
8. function AddStep( Path oldPath, Node node, float Cost ) : Returns Path
9. {
10.     return new Path( oldPath, node, Cost )
11. }
12.
13. function EuclideanDistance( Node n1, Node n2 ) : Returns the distance between
                                                two given nodes
14.
15. function StepCost( Node n1, Node n2 ) : Returns the cost to move between
                                                two neighboring given nodes
16.
17. function EmotionalChange( Node n1, Node n2 ) : Returns the emotional change between
                                                two neighboring nodes
18.
19. function NextStep ( Node n, Path p ) : Return the next node in an already
                                                existing path
20.
21. function sqrt ( float f ) : Returns the square root of a given number
22.
23. function LargestDifference( float value ) : Returns the maximum value for the
                                                emotional difference of a given value
24. {
25.     if ( value < ( 100.0f / 2 ) )
26.         return 100.0f - value
27.     else
28.         return value;
29. }

```

Pseudocódigo 3: Funções Auxiliares para o Modelo

Na linha 1, a estrutura **Path**, responsável por armazenar o caminho, é definida. Nesta estrutura são armazenados a sequência dos passos anteriores, o último passo dado e o custo espacial atual do caminho até o momento, normalizado – esta normalização é a principal diferença entre esta estrutura e a estrutura apresentada na seção 2.3.1.

Na linha 8 é definida uma função que anexa um novo passo ao caminho, retornando um novo caminho com o valor de custo normalizado atualizado.

Na linha 13 é definida a função **EuclideanDistance**, onde dados dois nós, é retornada distância euclidiana entre estes nós. Esta função será utilizada como a função heurística neste trabalho.

Na linha 15 é definida a função **StepCost** que determina o custo de movimentação entre dois nós vizinhos dados.

Na linha 19 é definida a função auxiliar **NextStep**, onde dado um nó pertencente a um caminho e o caminho, retorna o próximo passo do caminho.

Na linha 21 é definida a função **sqrt**, que retorna a raiz quadrada do número passado.

Por último, na linha 23 é definida a função **LargestDifference**, onde, dado um valor, é retornada a maior margem de erro. Esta função é utilizada na normalização do custo emocional, representando o Δm_x na seção 3.3.

Com estas funções auxiliares definidas, é apresentado o pseudocódigo para o modelo proposto:

```

1. function EmotionalPath( Node    start,
2.                        Node    goal,
3.                        int     aStarLength,
4.                        EmotionalCurve    emotionalCurve,
5.                        float    spatialWeight,
6.                        float    emotionalWeight ) : return path on success;
7.                                                    null on failure
8. {
9.     closedSet = Empty Set of Node
10.    pathQueue<Path, Cost> = { ( start, 0 ) } // A Priority Queue
11.
12.    while ( pathQueue not empty )
13.    {
14.        path = Dequeue cheapest path in ( pathQueue )
15.
16.        if ( path.LastStep is in closedSet ) continue
17.
18.        if ( path.LastStep == goal ) return path
19.
20.        Add path.LastStep to closedSet
21.
22.        for each ( neighbor of path.LastStep )
23.        {
24.            // Handle Spatial Part for g(x)
25.            normalizedStepCost = StepCost( Path.LastStep, neighbor ) /
26.                                   ( 2 * aStarLength )
27.
28.            gx1 = path.NormalizedSpatialCost + normalizedStepCost
29.
30.            // Handle Emotional Part for g(x)
31.            emotionalState    = emotionalCurve.InitialEmotion
32.            numerator         = 0
33.            denominator       = 0
34.
35.            for each ( node in path )
36.            {
37.                Update walkedPercentage for this node
38.                Reevaluate targetEmotion from emotionalCurve using walkedPercentage
39.
40.                emotionalState += EmotionalChange( node, NextStep( node, path ) )
41.                numerator      += ( targetEmotion - emotionalState ) ^ 2
42.                denominator    += LargestDifference( emotionalState ) ^ 2
43.            }
44.
45.            gx2                = sqrt( numerator ) / sqrt( denominator )
46.
47.            gx                 = gx1 * spatialWeight + gx2 * emotionalWeight
48.
49.            // Handle h(x) (Spatial only)
50.            hx                 = ( EuclideanDistance( neighbor, goal ) /
51.                                   ( 2 * aStarLength ) ) * spatialWeight
52.
53.            // Enqueue candidates
54.            newPath            = AddStep( path, neighbor, gx1 )
55.            pathQueue.Enqueue( newPath, gx + hx )
56.        }
57.    }
58.
59.    return null
60.}

```

Pseudocódigo 4: Pseudocódigo para o Modelo proposto.

Na primeira linha é definida a função **EmotionalPath**, onde são passados o nó inicial (*start*), nó objetivo (*goal*), o comprimento da busca utilizando o A* (*aStarLength*), a Curva Emocional para a busca (*emotionalCurve*), o multiplicador espacial para ponderação dos valores espaciais (*spatialWeight*) e o multiplicador emocional de ponderação dos valores emocionais (*emotionalWeight*). A função retorna o caminho encontrado ou retorna nulo se não encontrar um caminho válido.

As linhas 9 e 10 declaram as duas variáveis auxiliares responsáveis por auxiliar na navegação do grafo.

A variável *closedSet*, declarada na linha 9, armazena o conjunto de nós já visitados, evitando uma possível repetição de candidatos. Este conjunto é necessário pois, com a adição do valor da emoção em $g(x)$, é possível que caminhos diferentes mudem o valor emotivo e o façam possuir um custo menor. Como o objetivo do modelo é sempre “caminhar em direção ao *goal*”, é importante que exista este controle. Este controle também é responsável por evitar que o *pathfinder* busque caminhos que ficam dando voltas na mesma área até encontrar o valor emocional ideal.

A variável *pathQueue* é uma fila de prioridades responsável por enfileirar em ordem crescente de custos os candidatos para o caminho. Inicialmente possui o nó inicial, sem custo de caminho.

Na linha 12 começa um *loop* que vai até a linha 54, que é executado enquanto houver nós enfileirados em *pathQueue*.

Na linha 14, o nó com custo mais baixo em *pathQueue* é removido e atribuído a *path*.

Na linha 16 verifica-se se este nó já foi processado, ou seja, se está em *closedSet*. Se estiver, o nó é desconsiderado e o *loop* recomeça.

A linha 18 verifica se o último passo de *path* é o *goal*. Se for, retorna o caminho e o algoritmo encerra.

A linha 20 adiciona o último passo de *path* a *closedSet*, evitando que este nó seja expandido novamente.

Na linha 22 começa um bloco de instruções que é executado para cada vizinho do último nó em *path*.

Na linha 25 é calculado o custo do passo e normalizado utilizando $(2 * \Gamma)$, ou seja, duas vezes o comprimento do A*. Como explicado no modelo proposto (seção 3.2), assume-se que um caminho que cumpra os três requisitos terá comprimento igual ou maior ao comprimento obtido com o A* e menor que duas vezes o tamanho deste.

Na linha 26 é calculada a primeira parte de $g(x)$, a parte espacial. Esta é a soma do custo espacial normalizado do caminho até então com o custo normalizado do passo até o vizinho em questão. Esta primeira parte é armazenada como *gx1*.

As linhas 28 a 41 são responsáveis por calcular a segunda parte de $g(x)$, que trata das emoções no processo. Para que seja feito de forma adequada, é necessário recalculá-las as estimativas emocionais do caminho a cada passo.

Na linha 29, a variável *emotionalState* recebe o valor inicial da curva emocional, para que possa ser recalculado para cada passo do caminho.

Nas linhas 30 e 31 são declaradas suas variáveis auxiliares (*numerator* e *denominator*)

que serão utilizadas para calcular a segunda parte da formula de $g(x)$ – i.e. $\frac{\sqrt{\sum(\Delta e_x)^2}}{\sqrt{\sum(\Delta m_x)^2}}$.

Das linhas 33 a 41, o valor do caminho é reestimado, passo a passo. Para cada nó do caminho, as seguintes instruções são executadas:

- Linha 35: A estimativa de comprimento do caminho para atingir o *goal* é recalculada e armazenada em *walkedPercentage*. Para tal, é calculado o comprimento até o ponto do caminho atual que está sendo processado, dividido pelo comprimento do caminho atual + heurística de caminho até o *goal* (deslocamento em linha reta).
- Linha 36: A emoção alvo para este momento na curva emocional é estimado novamente, de acordo com o valor de *walkedPercentage*.
- Linha 38: A mudança emocional real é calculada e armazenada em *emotionalState* de acordo com a variação associada ao próximo passo do caminho
- Linha 39: A diferença emocional entre o estado ideal na curva (*targetEmotion*) e o estado atual no caminho (*emotionalState*) é elevado ao quadrado e acumulado em *numerator*, representando o numerador da fórmula: $\sum(\Delta e_x)^2$.
- Linha 40: A diferença para o pior caso é acumulada em *denominator*, representando o denominador da fórmula: $\sum(\Delta m_x)^2$.

Após o *loop*, a segunda parte da fórmula é calculada e armazenada em *gx2*, na linha 43.

Na linha 45, $g(x)$ é calculado e armazenado em *gx*, multiplicando *gx1* por *spatialWeight* e *gx2* por *emotionalWeight*, completando a fórmula.

Na linha 48, $h(x)$ é calculado e armazenado em *hx*, utilizando a distância euclidiana como heurística, normalizando a, ao dividir por $2 * \Gamma$. O valor é finalmente multiplicado por *spatialWeight*.

Na linha 51, é criado um novo caminho candidato, anexando o vizinho candidato ao caminho e atribuindo o custo espacial armazenado em ***gx1***. Este valor é armazenado explicitamente para evitar desperdício de tempo computacional na estimativa realizada na linha 35.

Na linha 52, o novo caminho é enfileirado, com o custo total equivalente a soma de ***gx*** e ***hx***.

Na linha 56, é retornado nulo, caso o *loop* tenha acabado e não tenha sido encontrado um caminho válido no *loop*.

3.3.2 Interface

Durante o desenvolvimento do algoritmo, foram criadas algumas ferramentas para auxiliar na análise e validação dos dados. A principal ferramenta gráfica utilizada pode ser vista na Figura 8, onde do lado esquerdo temos uma representação visual do grafo e do lado direito a representação de curvas emocionais – a de referência, resultante do A*, a resultante do algoritmo proposto e outras correspondentes a caminhos alternativos que o usuário queira considerar para efeito de comparação. Na explicação da interface e na validação apresentada no próximo capítulo assume-se que a emoção modelada que é afetada pelo caminho é o medo.

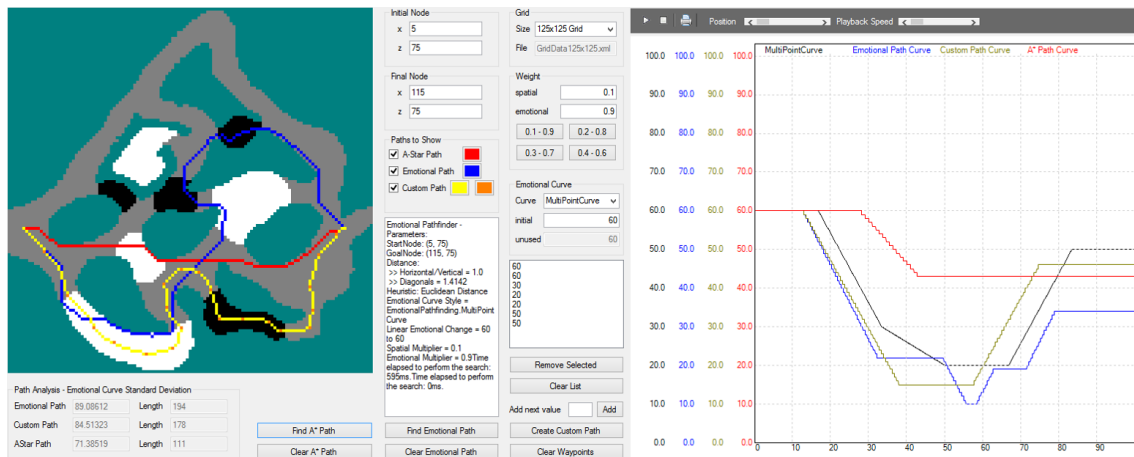


Figura 8: Interface principal do utilitário utilizado no desenvolvimento.

Para entender melhor, na Figura 9 é possível ver o mesmo grafo da Figura 8, só que ampliado, com o caminho encontrado pelo A* traçado em vermelho e o caminho encontrado pelo algoritmo de busca emocional proposto em azul. As cores utilizadas para representar o grafo são branco, cinza e preto, onde cinza é o caminho sem alteração emocional, preto aumenta o medo e branco diminui o medo. A cor verde-azulada (*teal*) representa as áreas onde o grafo não é transitável.

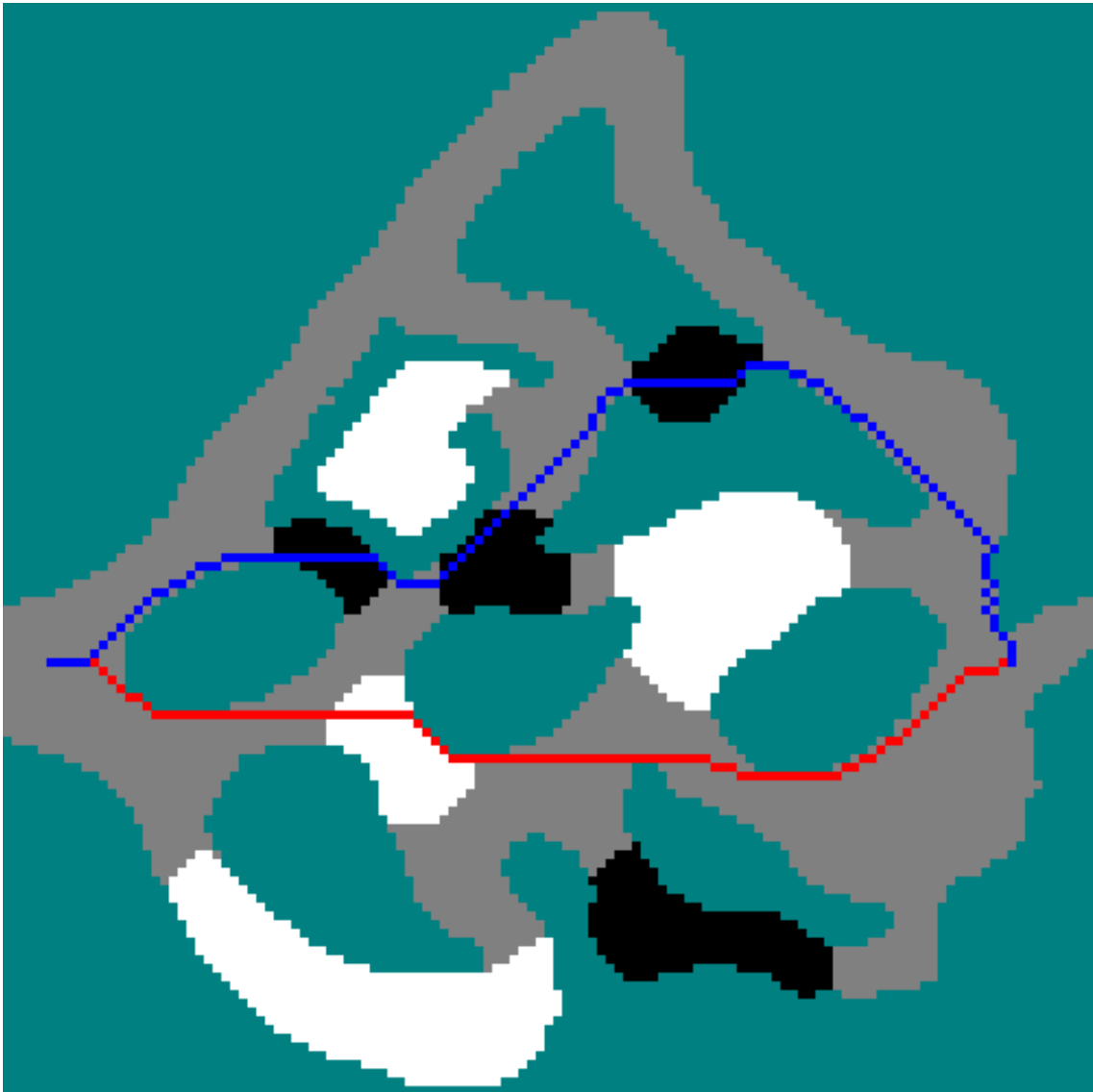


Figura 9: Representação visual do grafo com as áreas de medo e dois possíveis caminhos.

Na Figura 10 é possível ver a curva emocional para os mesmos caminhos da Figura 9. Os gráficos são diferenciados pelas cores, onde o gráfico em preto é a linha emocional ideal para o caminho, o gráfico em azul é a curva emocional efetiva para o caminho encontrado usando a busca emocional proposta e o gráfico vermelho é a curva emocional efetiva para o caminho encontrado com o A^* .

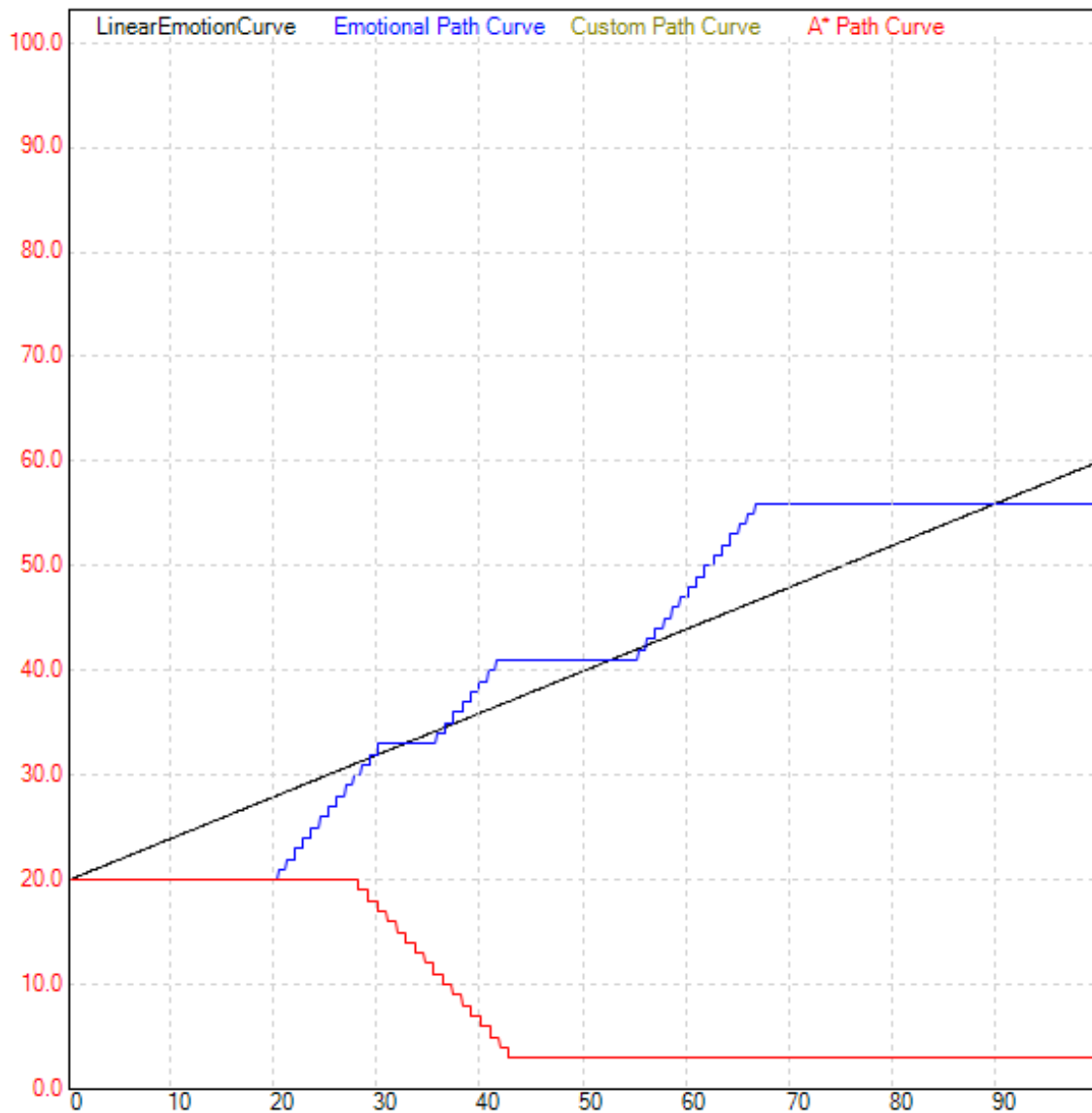


Figura 10: Curvas emocionais ideais e reais.

Além desta ferramenta, foram criadas ferramentas de saída textual que processam em massa vários caminhos para auxiliar na validação do algoritmo. Estas saídas serão tratadas no Capítulo 4.

4 Validação

Com o intuito de ter uma base ampla para comparação, foram tentados alguns métodos para encontrar outros caminhos.

Para realizar estes experimentos foi utilizado um computador com um Intel Core i7 4770k com 32gb de RAM, rodando o Windows 8.1 como sistema operacional.

Para realizar a comparação do modelo desenvolvido, foram exploradas alternativas que tentassem encontrar todos os caminhos entre o ponto de partida e o *goal*, utilizando uma abordagem de força bruta. Como não foi encontrada uma outra abordagem mais rápida na literatura, esta técnica será utilizada, existindo a garantia de achar o melhor caminho.

A mais promissora destas abordagens foi utilizar uma versão modificada do A*, que não utiliza o conjunto *closed* e evita repetição de nós no caminho. Esta abordagem armazena todos os caminhos encontrados que possuam até o dobro do comprimento do caminho mais curto (este encontrado pelo A* sem modificação). Os resultados obtidos foram:

- Ao tentar calcular todos os caminhos possíveis em um grafo com a dimensão de 125x125 (15625 nós), conforme a Figura 11, foi executada uma busca por aproximadamente 36 horas e foram encontrados 11.207.723 caminhos até busca ser manualmente interrompida. Não é viável realizar esta busca, portanto o domínio foi reduzido;



Figura 11: Grafo com dimensão 125x125 utilizado.

- Ao tentar calcular todos os caminhos possíveis em um grafo com a dimensão de 10x10 (100 nós), conforme a Figura 12, onde cada nó está ligado a todos os vizinhos adjacentes – com o nó inicial no topo à esquerda (1,1) e o *goal* no fundo à direita (10,10), a busca foi executada por aproximadamente 36 horas e foram encontrados 74.260.001 caminhos até a busca ser manualmente interrompida;

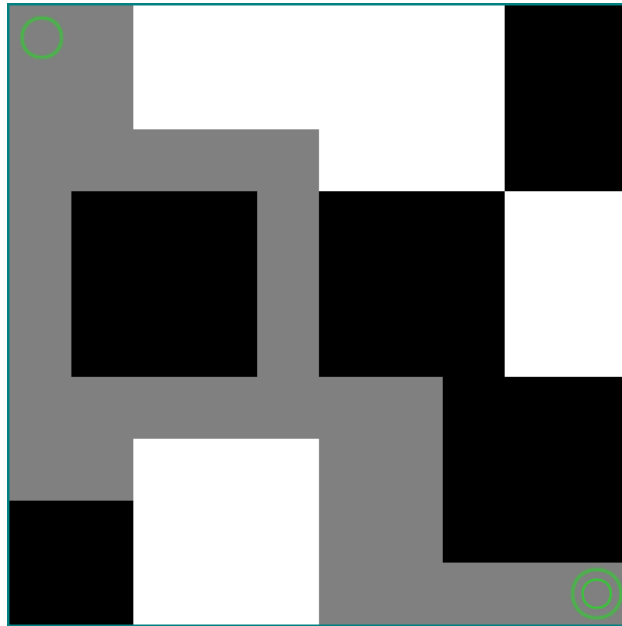


Figura 12: Grafo com dimensão 10x10 utilizado.

- A busca em um grafo 5x5 (25 nós), demonstrado na Figura 13, com o nó inicial no topo à esquerda (1,1), e o nó *goal* no fundo à direita (5,5), a busca foi executada e terminou em aproximadamente 2 minutos, encontrando 7.992 caminhos diferentes;

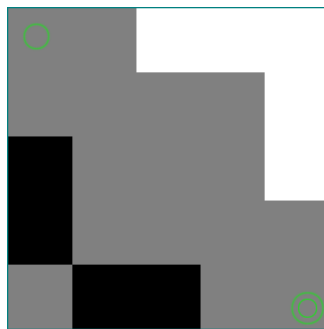


Figura 13: Grafo com dimensão 5x5 utilizado.

- Para encontrar um grafo um pouco mais complexo, foi utilizado um grafo de tamanho 8x8 (64 nós), com o nó inicial nas coordenadas (2,2) e *goal* em (7,7), a busca durou cerca de 6 minutos e encontrou 216.964 caminhos diferentes.

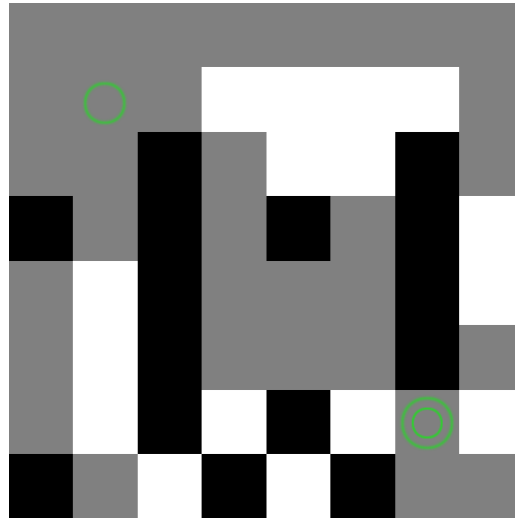


Figura 14: Grafo com dimensão 8x8 utilizado.

Para a análise dos dados será utilizado o grafo de tamanho 8x8 citado acima, já que foi um resultado viável de ser obtido computacionalmente com força bruta e que sirva de comparação para a qualidade dos caminhos encontrados pelo algoritmo proposto, bem como para a eficiência do algoritmo. Os exemplos anteriores maiores são descartados por não poderem ser resolvidos em um tempo viável graças a explosão combinatória.

A análise dos resultados é feita tendo como medida os requisitos explicitados no capítulo 3 para o modelo. Inicialmente, comparam-se os resultados obtidos em termos de compatibilização da naturalidade do caminho com a aproximação da variação de emoção em relação a curva de referência. Em seguida, analisa-se o tempo para a obtenção da solução.

4.1 Compatibilização entre Naturalidade de Caminho e Variação de Emoções.

Devido ao tempo para gerar a solução com força bruta, a comparação de compatibilização foi feita em duas etapas. A primeira em cenários pequenos, onde a força bruta pode gerar a solução ótima para o problema e então pode-se comparar esta solução com a solução

encontrada pelo modelo proposto. A segunda etapa é realizada em cenários grandes, onde diferentes tipos de curvas foram exploradas.

4.1.1 Comparação com Soluções Ótimas em Cenários Pequenos

Para analisar a curva emocional, as saídas do método força-bruta foram salvas e processadas em sequência, sendo comparadas posteriormente ao caminho encontrado utilizando o modelo proposto.

A diferença entre o valor real e o valor ideal das emoções foi calculada utilizando a raiz quadrada da média das diferenças quadradas, desta forma:

$$\sqrt{\frac{\sum(\text{valor_emocional_ideal_por_passo} - \text{valor_emocional_real_por_passo})^2}{\text{comprimento_do_caminho}}}$$

É possível classificar de forma abrangente muitos caminhos ao mesmo tempo para cada variação possível da curva emocional linear, automaticamente. Foi utilizada a curva emocional linear, onde a variação emocional foi testada para 10 variações possíveis, entre 0 e cada limite superior entre 1 e 10. Foram classificados 414594 caminhos entre as posições de origem e destino que receberam um número de *ranking* para cada uma das variações emocionais testadas. A posição no ranking da solução fornecida pelo algoritmo proposto foi então localizada em cada um dos testes. Os dados também foram classificados para dois pares de multiplicadores diferentes, referentes aos pesos de ponderação entre comprimento do caminho (w_1) e aproximação da variação emocional desejada (w_2). Os resultados podem ser vistos na Tabela 1 e 2.

Para os Pesos: $w_1=0.2$ $w_2=0.8$				
Varição emocional	Ranks Possíveis	Ranking encontrado pelo modelo	Valor encontrado pelo modelo	Melhor valor encontrado pela força bruta
1	3181	65	0,489898	0,285714
2	3213	141	0,600000	0,282843
3	3411	582	1,166190	0,272166
4	4257	114	0,544331	0,282843
5	4016	606	1,144633	0,307653
6	5077	791	1,779513	0,314918
7	6051	1162	2,421929	0,363636
8	6189	1329	3,067150	0,453609
9	5919	1381	3,713713	0,333333
10	4832	1198	4,361022	0,707107

Tabela 1: Comparação de classificação de caminhos, para os pesos $w_1=0.2$ $w_2=0.8$

Para os Pesos: $w_1=0.1$ $w_2=0.9$				
Varição emocional	Ranks Possíveis	Ranking encontrado pelo modelo	Valor encontrado pelo modelo	Melhor valor encontrado pela força bruta
1	3181	5	0,296586	0,285714
2	3213	141	0,600000	0,282843
3	3411	19	0,353553	0,272166
4	4257	114	0,544331	0,282843
5	4016	606	1,144633	0,307653
6	5077	30	0,473804	0,314918
7	6051	173	1,000000	0,363636
8	6189	225	1,609918	0,453609
9	5919	269	2,236068	0,333333
10	4832	226	2,867837	0,707107

Tabela 2: Comparação de classificação de caminhos, para os pesos $w_1=0.1$ $w_2=0.9$

No entanto, estas métricas analisam apenas o lado emocional da curva. Para auxiliar na visualização, a Figura 15 mostra três caminhos possíveis no grafo usado para estes exemplos, onde a cor cinza representa partes neutras que não alteram a emoção, enquanto a cor branca representa a diminuição do medo e a cor preta o aumento do medo, ambas em uma unidade. O caminho em azul é o melhor caminho encontrado pela força bruta, o vermelho é o caminho encontrado pelo modelo proposto com $w_2 = 0.8$ e o verde é o caminho encontrado pelo modelo proposto com $w_2 = 0.9$. Nesta figura são utilizadas as variações emocionais 1, 5 e 10 para as imagens a esquerda, centro e direita respectivamente. A Tabela 3 consolida os dados das Tabelas 1 e 2 utilizados na Figura 15.

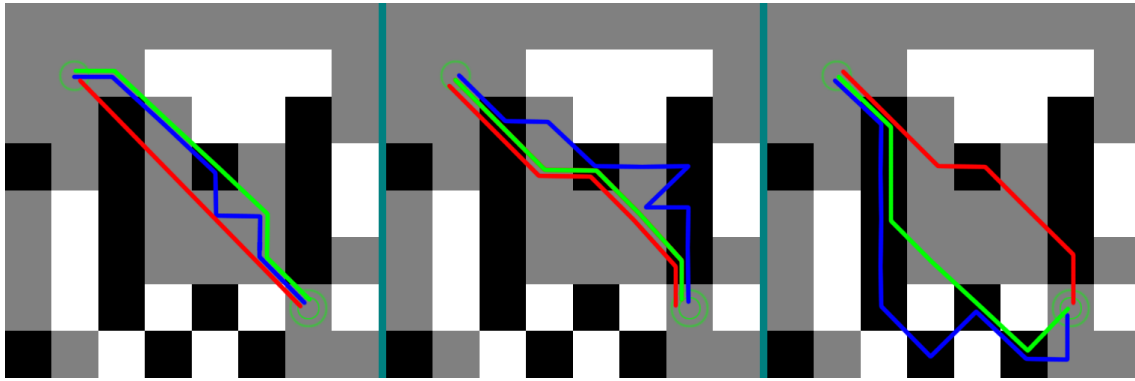


Figura 15: Exemplo de dois caminhos encontrados pelo modelo com ponderações diferentes e do melhor caminho encontrado por força bruta, para três diferentes variações de curvas emocionais lineares.

			$w_1=0.2 \ w_2=0.8$		$w_1=0.1 \ w_2=0.9$	
Variação Emocional	Ranks Possíveis	Dif. Emocional Força Bruta	Rank – Modelo	Dif. Emocional Modelo	Rank – Modelo	Dif. Emocional Modelo
1	3181	0,285714	5	0,296586	65	0,489898
5	4016	0,307653	606	1,144633	606	1,144633
10	4832	0,707107	226	2,867837	1198	4,361022

Tabela 3: Consolidação dos dados utilizados na visualização gráfica dos caminhos encontrados por força bruta e pelo modelo

Visualmente, é fácil de verificar a diferença: Ao comparar os caminhos encontrados pelo modelo com os encontrados pelo método de força bruta, estes são muito mais diretos, sendo mais apropriados para um ambiente de *Storytelling* Interativo.

4.1.2 Caminhos Naturais e Curvas Emocionais em Cenários Grandes

É importante lembrar que, durante a apresentação de história em um Sistema de *Storytelling* Interativo, são utilizados ambientes muito maiores do que geralmente podem

ser representados com grafos que possuem um tamanho de 8x8 – existindo a necessidade de tratar ambientes bem maiores que estes – assim como também existe a possibilidade de uma variação emocional muito mais complexa que pode ser representada por uma curva de emoção com variação linear, mas sem relaxar a restrição de tempo – i.e. é necessário que o caminho possa ser encontrado em tempo real.

Nas Figura 16, Figura 17 e Figura 18 é possível ver caminhos encontrados para três tipos de curvas emocionais diferentes:

- Na Figura 16 é utilizada uma curva com variação linear, começando em 20 e terminando em 60.
- Na Figura 17 é utilizada uma curva que começa em 20, vai linearmente até 60 na metade do caminho e retorna linearmente para 20.
- Na Figura 18 é utilizada uma curva emocional mais complexa, começando em 20, subindo linearmente até 30, permanecendo em 30, retornando a 20 de forma linear, onde permanece por mais uma seção, sobe até 40 e permanece em 40 até o fim do caminho.

É importante lembrar que a curva azul representa o valor emocional efetivo encontrado pelo caminho e que esta só pode ser alterada conforme as áreas do mapa demarcadas para mudanças emocionais.

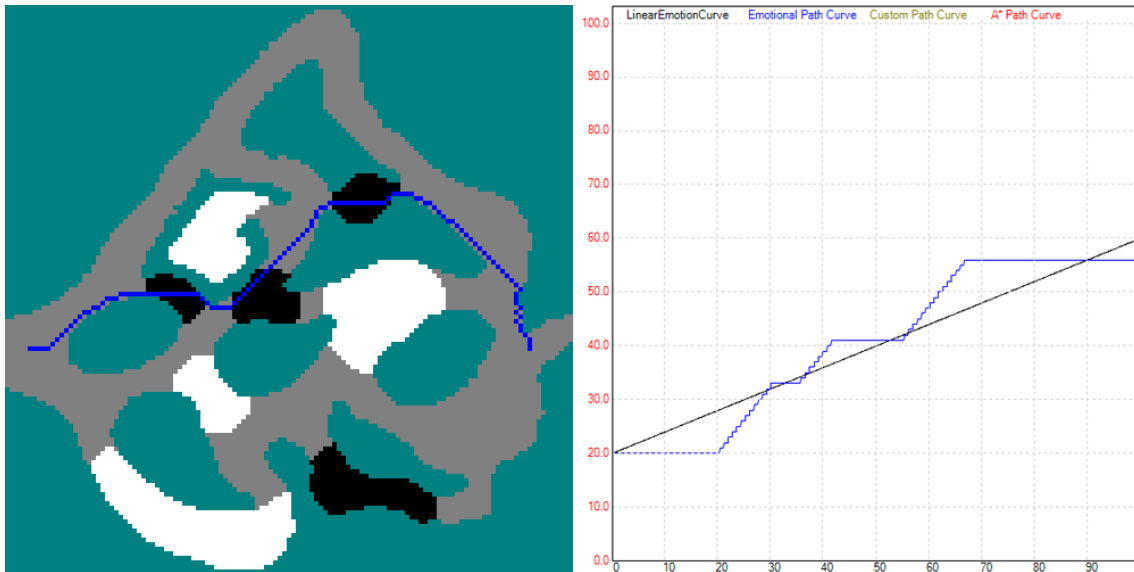


Figura 16: Caminho encontrado e Análise de Curva para uma curva emocional linear com variação emocional de 20 a 60

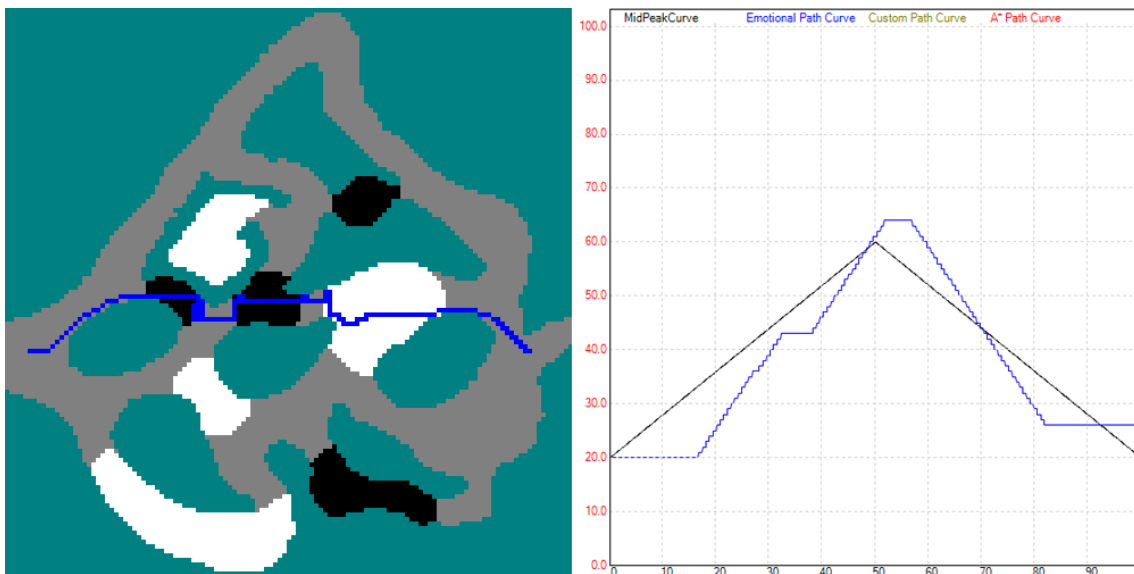


Figura 17: Caminho encontrado e Análise de Curva para uma curva emocional que começa em 20, tem seu pico no meio do caminho em 60 e retorna a 20 no final.

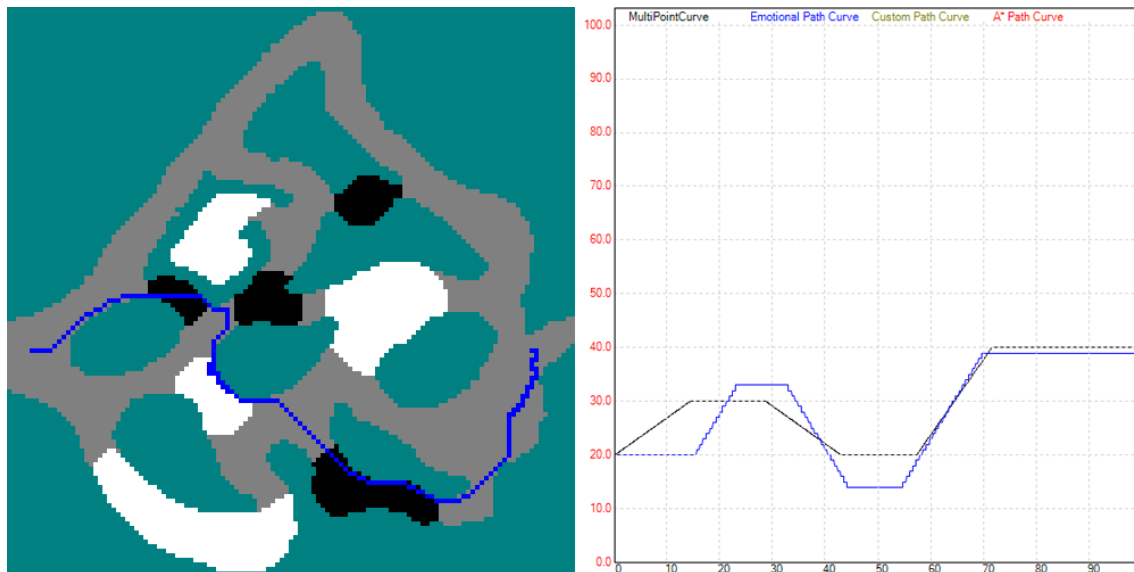


Figura 18: Caminho encontrado e Análise de Curva para uma curva emocional mais complexa. A curva ideal é representada em preto e a do caminho emocional em azul.

Nestes casos, as curvas dos caminhos encontrados pelo modelo acompanharam o tanto quanto possível as curvas ideais, dada a estrutura do grafo que representa a topologia do ambiente e as modificações emocionais destes, mostrando resultados promissores.

Para comparar, é possível ver o caminho encontrado pela busca utilizando o algoritmo A* na Figura 19. Como o A* se preocupa em achar o caminho mais curto, a curva emocional gerada para um mesmo grafo sempre será a mesma e dificilmente respeitará as necessidades para o *Storytelling* Interativo.

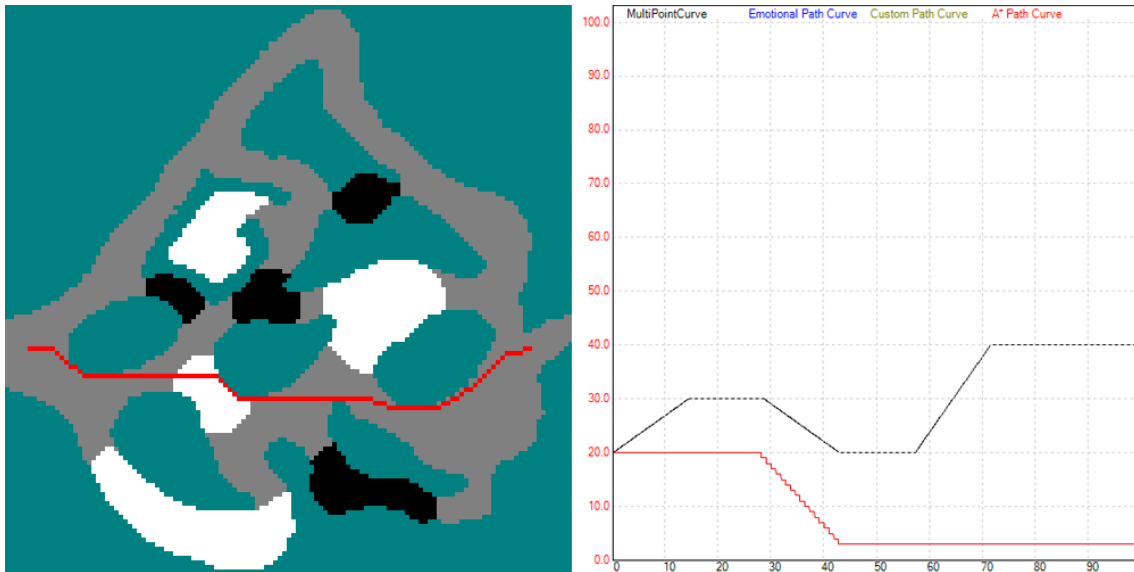


Figura 19: Caminho e curva emocional efetiva encontrada ao utilizar o A*

4.2 Tempo

É necessário que o código execute em ambientes grandes em um tempo curto, preferencialmente que seja possível executar o algoritmo em tempo real. Na Tabela 4 é mostrado o tempo médio de execução para encontrar todos os caminhos utilizando o método força bruta mencionado anteriormente, assim como o tempo para classificar todos os caminhos encontrados utilizando a formula de classificação apresentada na seção 4.1.1.

Tempo médio para Encontrar todos os caminhos	
125x125	36 horas+
10x10	36 horas+
8x8	6 minutos
5x5	2 minutos

Tabela 4: Tempo para encontrar todos os caminhos com força-bruta

Esta solução se torna inviável para uma execução em tempo real. A Tabela 5 mostra o tempo médio de busca para o algoritmo proposto em cada caso.

Tempo médio para execução do:	
Modelo	A*

125x125	250ms	25ms
10x10	5ms	1ms
8x8	5ms	1ms
5x5	5ms	1ms

Tabela 5: Tempo médio de execução do modelo proposto e do A*.

Embora aumente o tempo de procura em relação ao A*, o resultado é bem promissor para atender ao requisito de tempo para uso em aplicações que demandam respostas em tempo real.

Para calcular o tempo médio, foram executadas as buscas no grafo de 125x125, com dados de entrada aleatório para as curvas de emoção, em todos os casos da Tabela 4.

5 Conclusão

5.1 Considerações Gerais

Esta pesquisa apresentou um novo modelo de busca de caminhos, capaz de encontrar um balanceamento entre um caminho curto e emocionalmente compatível com uma curva emocional desejada e que possa ser gerado em tempo real. Desta forma, é possível melhorar a qualidade de dramatização das cenas apresentadas sem que haja um impacto no tempo de geração do desenrolar a cena.

Embora esta pesquisa tenha sido inspirada por um módulo que se viu necessário no sistema Logtell, o resultado desta pesquisa é *independente de contexto*, ou seja, pode ser utilizado em qualquer tipo de história que precise de uma movimentação que seja influenciada emocionalmente, podendo ser aplicado em qualquer sistema de *Storytelling* Interativo ou até mesmo em jogos onde haja a necessidade de controlar a emoção de personagens através de um determinado percurso.

A validação do modelo foi realizada através da criação de um protótipo em que é possível se experimentar diversos tipos de grafos e que aceita como entrada diversos tipos de curvas emocionais a serem seguidas.

5.2 Principais Contribuições

Como principal contribuição desta pesquisa, tem-se o modelo, que permite atender às expectativas emocionais do autor para a cena, permite que este caminho seja encontrado em um ambiente desconhecido – possivelmente gerado procedimentalmente – e que seja

visualmente agradável para o público, sempre dando a impressão que está caminhando em direção ao destino.

Os resultados obtidos indicam uma alta viabilidade para uso em tempo real, possuindo um custo computacional baixo, resultando em um tempo de busca bem curto, em especial quando comparado a uma solução de força bruta que se torna inviável em cenários grandes.

Também foi implementado um protótipo que permite um estudo mais aprofundado do algoritmo, o qual suporta a busca de caminhos utilizando o A*, a busca de caminhos com o modelo proposto e o traçado manual de caminhos. Além disto, este protótipo fornece uma representação visual do grafo utilizado para a busca, assim como das curvas emocionais geradas em cada caso, superpostas com a curva ideal, facilitando ainda mais na análise.

5.3 Trabalhos Futuros

Um primeiro trabalho futuro refere-se ao uso efetivo do modelo proposto no sistema Logtell. Para tal, existe a necessidade de integrar as estruturas internas que regem o grafo de movimentação com a forma como a movimentação dos atores virtuais ocorre atualmente no Logtell.

Além disto, existe a necessidade de testar o modelo com cenários ainda mais complexos, preferencialmente utilizando geração procedimental. Com o desenvolvimento de um módulo de geração de cenários que seja consistente e adequado para as necessidades do Logtell, a avaliação deste modelo com este módulo é imprescindível, visando garantir uma maior riqueza na dramatização.

O valor de duas vezes o comprimento do A* foi escolhido de forma *ad hoc*, onde é proposto que mais do que isto poderia fazer com que o caminho fique muito longo. É desejável encontrar o ponto ideal em cima de experimentos, como um trabalho futuro.

É possível expandir o modelo permitindo que este suporte outros elementos durante a busca de caminho, como tratar do nível de iluminação. Para tal, uma solução possível seria tratar da aproximação com curvas, levando em conta valores absolutos em cada ponto e não o efeito que passar pelo ponto causa no personagem.

Também é necessário expandir o modelo para que possa tratar múltiplas emoções durante a busca de caminhos. Esta expansão permitirá uma busca de caminhos ainda mais complexa e apropriada para cenas complexas.

Outra possível expansão é permitir que exista a variação de emoções com a movimentação de múltiplos personagens, existindo a necessidade de cuidar da movimentação onde existem elementos concorrentes, evitando colisão entre os personagens e a modificação de emoções em grupo.

A eficiência em cenários muito maiores ainda não foi determinada, podendo-se estudar possibilidades de aprofundamento iterativo na busca e particionamento do cenário para adequação da curva emocional, visando melhorar a eficiência da busca como um todo.

Com o objetivo de aproximar ainda mais a emoção obtida na solução ao arco dramático, é possível experimentar com o oposto, métodos de adequação do cenário para que este colabore com o ajuste da curva emocional para cumprir o arco originalmente pretendido pelo autor.

Após isto, também é possível considerar a possibilidade de um planejamento estocástico, que consiga tratar cenários que mudem dinamicamente. Este planejamento estaria “consciente” que determinada área do mapa pode causar medo, mas sem a total

certeza que isto irá ocorrer. Além disto, o cenário não é totalmente conhecido, não sendo possível afirmar com certeza o que pode acontecer em determinada área. Em um cenário deste tipo, tentar maximizar as chances de acompanhar o arco dramático de acordo com as probabilidades do cenário. Considerar a utilização de processos de Markov para tal.

Por último, é necessário modelar a variação das emoções para outros tipos de evento além da movimentação, tratando de ações como diálogos, lutas e outras mais complexas, oferecendo um sistema completo que trate da dramatização dinâmica das emoções em *Storytelling* Interativo.

6 Referências

ALLEGORITHMIC. *ProFX*, 2006.

ARAÚJO, E. T. *Verificação de Restrições com Tempo Contínuo em Storytelling Interativo Não-Determinístico*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), 2011.

BRABEN, D. *Frontier: Elite II*, 1993.

BRABEN, D.; BELL, I. *Elite*, 1984.

CAMANHO, M. M.; CIARLINI, A. E. M.; FURTADO, A. L.; POZZER, C. T.; FEIJÓ, B. “A Model for Interactive TV Storytelling”. In: *VIII Brazilian Symposium on Games and Digital Entertainment*, pp.197-206. Rio de Janeiro, Brasil. Oct. 2009.

CAVAZZA, M.; CHARLES, F.; MEAD, S. “Character-based interactive storytelling”. *IEEE Intelligent Systems*, v. 17, n. 4, pp.17-24, Jul/Ago 2002.

CIARLINI, A. E. M. *Geração Interativa de Enredos*. Tese de PhD. Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO), 1999.

CIARLINI, A. E. M.; POZZER, C. T.; FURTADO, A. L.; FEIJÓ, B. “A logic-based tool for interactive generation and dramatization of stories”. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology - ACE '05*, pp. 133–140. New York, New York, USA: ACM Press, 2005.
doi:10.1145/1178477.1178495

CIARLINI, A. E. M.; CAMANHO, M. M.; DÓRIA, T. R.; FURTADO, A. L.; POZZER, C. T.; FEIJÓ, B. “Planning and Interaction Levels for TV Storytelling”. In *U.*

Spierling & N. Szilas (Eds.), First Joint International Conference on Interactive Digital Storytelling, ICIDS 2008 Erfurt, Germany, November 26-29, 2008 Proceedings, Vol. 5334, pp 198–209. Erfurt, Germany: Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-89454-4

CIARLINI, A. E. M.; CASANOVA, M. A.; FURTADO, A. L.; VELOSO, P. A. S. “Modeling interactive storytelling genres as application domains”, *Journal of Intelligent Information Systems* v. 35, pp. 347–381, 2009. doi:10.1007/s10844-009-0108-5

CROWTHER, W.; WOODS, D. *Colossal Cave Adventure*, 1976.

DONALDSON, T.; PARK, A.; LIN, I. “Emotional pathfinding”. In *Advances in Artificial Intelligence*, pp. 31–43. Springer Berlin Heidelberg, 2004. doi:10.1007/978-3-540-24840-8_3

DÓRIA, T. R., CIARLINI, A. E. M., & ANDREATTA, A. “A nondeterministic model for controlling the dramatization of interactive stories”. In *Proceeding of the 2nd ACM international workshop on Story representation, mechanism and context - SRMC '08*, pp. 21. New York, New York, USA: ACM Press, 2008. doi:10.1145/1462014.1462020

ENIX. *Dragon Quest*, 1986.

FERREIRA, P. A. *Modelo de Planejamento Temporal Não Determinístico Considerando Propriedades Dramáticas em Mudança Contínua para Storytelling Interativo*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), 2013.

FRONTIER DEVELOPMENTS. **Frontier: First Encounters**, 1995.

- GARRIOTT, R. **Akalabeth: World of Doom**, 1980.
- GOTTIN, V. M. *Verificação Abstrata de Propriedades Dramáticas Contínuas em Eventos Não Determinísticos*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), 2013.
- GUINNESS WORLD RECORDS, *Guinness World Records 2014*. London, UK: Guinness World Records, 2013.
- HART, P., NILSSON, N., & RAPHAEL, B. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. *IEEE Transactions on Systems Science and Cybernetics*, v. 4, n. 2, pp.100–107, 1968. doi:10.1109/TSSC.1968.300136
- JOHANSSON, A., & DELL’ACQUA, P. “Pathfinding with Emotion Maps”. *Intelligent Computer Graphics 2011*, v. 374, pp. 139–155, 2012. doi:10.1007/978-3-642-22907-7_8
- KARLSSON, B., CIARLINI, A. E. M., FEIJÓ, B., & FURTADO, A. L. “Applying a plan-recognition/plan-generation paradigm to interactive storytelling”. In *ICAPS06*. English Lake District, 2006. doi:10.1.1.67.4061
- KONUR, S. *Real-time and Probabilistic Temporal Logics: An Overview*, pp. 370–403. *Logic in Computer Science*, 2010. Retrieved from <http://arxiv.org/abs/1005.3200>
- MOJANG. *Minecraft*, 2011.
- NAKED SKY ENTERTAINMENT. *RoboBlitz*, 2006.
- NAREYEK, A. “Game AI is Dead. Long Live Game AI!”. *IEEE Intelligent Systems*, v.22, n. 1, pp. 9–11, 2007. doi:10.1109/MIS.2007.10
- ORIGIN SYSTEMS. *Ultima*, 1981.

- PERLA, P., & MCGRADY, E. “Why wargaming works”. *Naval War College Review*, v. 64, n. 3, 2011.
- PLUTCHIK, R. “The Nature of Emotions”. *American Scientist*, v. 89, n. 4, 344, 2001.
doi:10.1511/2001.4.344
- PORTEOUS, J.; CAVAZZA, M.; CHARLES, F. “Applying planning to interactive storytelling”. *ACM Transactions on Intelligent Systems and Technology*, v. 1, n. 2, pp. 1–21, Nov. 2010. doi:10.1145/1869397.1869399
- POZZER, C. T. *Um Sistema para Geração, Interação e Vizualização 3D de Histórias para TV Interativa*. Tese de PhD. Pontifícia Universidade Católica, Rio de Janeiro (PUC-RIO), 2005.
- RE-LOGIC. *Terraria*, 2011.
- SILVA, F. A. G. *Geração de Enredos com Planejamento Não-Determinístico em Storytelling para TV Interativa*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), 2010.
- SQUARE. *Final Fantasy*, 1987.
- SQUIRE, K. “Video Games in Education”. *International Journal of Intelligent Simulations and Gaming*, v. 2, n.1, pp. 49–62, 2003.
- VALVE. *Left 4 Dead*, 2008.