

Universidade Paulista - UNIP

Bruno de Paula Silva - C992534

Daniel Sousa David De Oliveira - D137GC0

Gustavo Felipe De Santana Marques - C993AH8

Marcelo Bueno Silva - N805CA0

Wesley Luiz Carvalho Silva - C993077

Biblioteca de Tomada de Decisão com Lógica Paraconsistente para jogos de cartas

São Paulo

2019

Bruno de Paula Silva - C992534
Daniel Sousa David De Oliveira - D137GC0
Gustavo Felipe De Santana Marques - C993AH8
Marcelo Bueno Silva - N805CA0
Wesley Luiz Carvalho Silva - C993077

Biblioteca de Tomada de Decisão com Lógica Paraconsistente para jogos de cartas

Trabalho apresentado para aproveitamento da
disciplina Trabalho de Curoso II , do curoso
de Ciência da Computação, Da Universidade
Paulista - UNIP Campus Cidade Universitária.

Orientadora: Prof^a. Dr.a Amanda Luiza S. Pereira

São Paulo

2019

LISTA DE ILUSTRAÇÕES

Figura 1 – Valores	7
Figura 2 – Maximização	7

LISTA DE TABELAS

Tabela 1 – Visualização das Cartas	6
Tabela 2 – Relação entre o status e parecer analítico	8

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Justificativa	6
1.2	Validação Empírica	6
1.3	Objetivos	8
1.4	Objetivos Específicos	8
2	REFERÊNCIA TEÓRICA	10
2.1	Lógica Paraconsistente	10
2.2	Teoria dos jogos	10
2.3	Biblioteca	10
2.4	Engenharia de Software	11
2.4.1	Metodologia	11
2.4.2	UML	11
2.4.3	RUP	11
3	MATERIAIS E MÉTODOS	12
3.1	Linguagem C#	12
3.2	Plataforma Unity	12
3.3	Visual Studio	12
	REFERÊNCIAS	13

1 INTRODUÇÃO

Este trabalho acadêmico demonstra a aplicação da Lógica Paraconsistente Anotada (LPA), em jogos do gênero *Trading Card Games* (Jogos de Cartas Colecionáveis – TCG), através da criação de uma biblioteca de Tomada de Decisão que implementa a Lógica Paraconsistente Anotada Evidencial (LPA $E\tau$), além disso será criado um jogo de cartas que utiliza a biblioteca para demonstrar as suas funcionalidades.

Com parte do senso comum, as pessoas acreditam que os jogos têm como única finalidade entreter ignorando as diversas opções que um jogo eletrônico pode trazer para auxiliar o desenvolvimento humano, de modo que a utilização dos jogos de forma educacional ou para resolver problemas usando raciocínio lógico pode trazer benefícios à saúde (LOPES et al., 2011).

A LPA é uma lógica não clássica que admite contradições e incertezas, é uma boa solução para fazer tratamento de situações reais, no qual a Lógica Clássica, por ser binária, se mostra ineficaz ou impossibilitada de ser aplicada (SILVA FILHO, 2006). Assim possibilita as mais variadas aplicações em áreas tais como computação, robótica, tráfego aéreo e de trens, distribuição de energia em grandes usinas, programação, redes neurais, pesquisa operacional entre outras (CARVALHO; ABE, 2011).

Uma biblioteca é uma coleção de subprogramas ou um programa que facilita o desenvolvimento de sistemas, no núcleo da biblioteca desenvolvida será utilizado a LPA. Dessa forma, a biblioteca implementada no jogo será responsável por tomar as decisões dos resultados de batalha, sendo o intuito de criar um software que pode ser reutilizável por outros, iniciando um estudo da aplicação da LPA em jogos TGC.

Será explicado como a biblioteca foi desenvolvida e implementada no jogo, juntamente com a sua documentação para utilização. Também será relatado como o jogo foi desenvolvido, quais ferramentas e metodologias foram utilizadas e quais resultados que foram obtidos em vantagem com a utilização da LPA.

Este documento está estruturado nos seguintes tópicos, *1 - Introdução* apresenta o projeto, os objetivos e as justificativas. No capítulo *2 - Referência Teórica* é exposta a base conceitual do projeto. Na seção seguinte *3 - Materiais e Métodos* é retratada a metodologia utilizada para desenvolvimento da prototipagem além das ferramentas utilizadas no processo.

1.1 Justificativa

No desenvolvimento de jogos de cartas, são encontrado diversas bibliotecas disponibilizado na *Unity Asset Store*, com uso de LC, que uma proposição é classificada como verdadeira ou falsa. Não há qualquer outra possível alternativa, ou algo é Verdadeiro ou exclusivamente Falso (ABE, 2013).

A lógica paraconsistente introduz duas novas categorias além do Verdadeiro e do Falso. Podemos ter proposições classificadas como Verdadeiras, Falsas, Inconsistentes ou Paracompletas. Para uma proposição ser classificada com Inconsistente tem haver uma evidência sugere que ela seja Verdadeira e outra evidência aponte que ela é Falsa, Agora quando não tem evidência Verdadeira nem tampouco que ela seja Falsa a proposição é classificada como Paracompleta (ABE, 2013).

A proposta do projeto é desenvolver uma biblioteca aplicando TD com a LPA ao invés do LC, para auxiliar no processo decisório aceitando valores contraditórios, possibilitando a criação de novas dinâmicas nos jogos.

1.2 Validação Empírica

A LPA é uma lógica não clássica que aceita contradições. A partir disso criar um cenário de jogo de cartas para demonstrar o uso da paraconsistente. Para exemplificar, foram criadas quatro cartas com atributos de força e velocidade com valores favoráveis e desfavoráveis de acordo com arma e idade conforme a tabela 1.

Tabela 1 – Visualização das Cartas

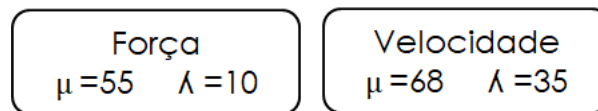
Carta	Atributos	Favorável	Desfavorável	Detalhes	Valor
Arqueiro	<i>Força</i>	20	10	<i>Arma</i>	Arco e Flecha
	<i>Velocidade</i>	70	35	<i>Idade</i>	25
Espadachim	<i>Força</i>	55	20	<i>Arma</i>	Espada
	<i>Velocidade</i>	40	20	<i>Idade</i>	19
Lanceiro	<i>Força</i>	50	10	<i>Arma</i>	Lança
	<i>Velocidade</i>	68	35	<i>Idade</i>	30
Bárbaro	<i>Força</i>	70	50	<i>Arma</i>	Martelo
	<i>Velocidade</i>	60	80	<i>Idade</i>	40

Fonte: Produzido pelos autores.

O primeiro passo é realizar o processo de maximização, a partir do qual se obtém os maiores valores das evidências favoráveis e os menores das evidências desfavoráveis, entre as cartas *Arqueiro* e *Espadachim*, repetindo o processo em relação às cartas *Lanceiro* e *Bárbaro*. Na sequência, realiza-se o processo de minimização, o qual consiste na obtenção dos menores valores das evidências favoráveis e dos maiores valores das evidências desfavoráveis, as quais

foram maximizadas anteriormente. Após realizar o processos de maximização e minimização nos dois atributos das cartas, obteve-se os seguintes valores:

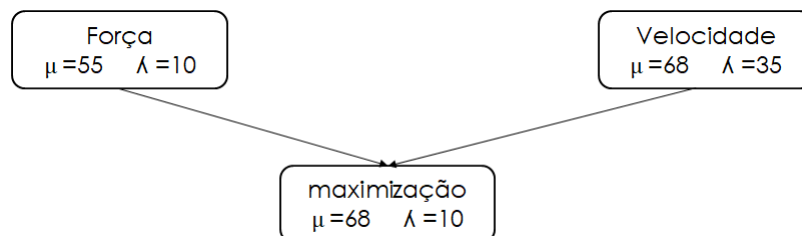
Figura 1 – Valores



Fonte: Produzido pelos autores.

Após a realização da maximização entre esses valores chegou-se ao seguinte resultado:

Figura 2 – Maximização



Fonte: Produzido pelos autores.

Aplicando o grau de certeza e incerteza sobre esses valores a saída foi o seguinte estado lógico:

$$Gi = 0.68 + 0.1 - 1 = -0.22$$

$$Gc = 0.68 - 0.1 = 0.58$$

Através do estado lógico foi produzido o parecer analítico com uma tabela pré definida cujo resultado refere-se a porcentagem que as 4 cartas conseguem tirar de vida do adversário.

Tabela 2 – Relação entre o status e parecer analítico

Status	Parecer Analítico
V	100%
F	0%
T	0%
\perp	0%
$T \rightarrow V$	20%
$T \rightarrow F$	10%
$V \rightarrow \perp$	16%
$F \rightarrow \perp$	8%
$Qv \rightarrow T$	50%
$Qv \rightarrow \perp$	40%
$Qf \rightarrow T$	6%
$Qf \rightarrow \perp$	2%

Fonte: Produzido pelos autores.

O status obtido na maximização foi $T \rightarrow F$, conforme a tabela 2 a porcentagem seria de 10%, com essa validação possui a lógica que será aplicada para Tomada de Decisão no jogo, assim iniciar a criação da dinâmica do jogo.

1.3 Objetivos

O objetivo geral é desenvolver uma biblioteca de Tomada de Decisão que utilize a LPA com foco em jogos TCG com objetivo de, protótipo final, isso é, biblioteca com manual de utilização e jogo de demonstração, com o intuito de ser utilizada por outros desenvolvedores de jogos de cartas

1.4 Objetivos Específicos

- Criar o modelo de paraconsistente a ser utilizado.
- Criar a biblioteca aplicando a LPA $E\tau$.
- Criar uma versão da biblioteca com valores fixo para uma implementação no jogo sem muitos problemas.
- Desenvolver uma segunda versão da biblioteca, permitindo que ela seja genérica o suficiente para atender diferentes regras de negócio em jogos TCG.
- Desenvolvimento da biblioteca.
- Construir um manual de utilização e documentação da biblioteca.
- Criar um jogo do gênero TCG que demonstre as funcionalidades da biblioteca.

- Gerar *Asset* e disponibilizar na plataforma *Unity Asset Store*.
- Criar a presente documentação descrevendo os materiais, métodos e referências utilizadas para a construção do projeto.

Finalizada a apresentação da estrutura do trabalho, prossegue para *capítulo 2 - Referência Teórica*.

2 REFERÊNCIA TEÓRICA

Neste capítulo serão apontados quais as referências que incentivaram a escolha do tema em questão.

2.1 Lógica Paraconsistente

A LPA teve como precursores o lógico russo N. A. Vasiliev e o lógico polonês J.Lukasiewicz. Os dois em 1910, publicaram trabalhos independentes, porém se restringiam a lógica aristotélica tradicional. Entre 1948 e 1954 o lógico polonês S.Jaskowski e o lógico brasileiro Newton C.A. da Costa, independentes construíram a LPA (CARVALHO; ABE, 2011, p. 27).

Segundo Silva Filho (2010) dentre as várias ideias no âmbito das Lógicas não-Clássicas criou-se uma família de lógicas que teve como fundamento principal a revogação do princípio da Não Contradição, a qual foi nomeada de Lógica Paraconsistente. Portanto, a LPA é uma Lógica não-Clássica que revoga o princípio da Não Contradição e admite o tratamento de informações contraditórias na sua estrutura teórica.

2.2 Teoria dos jogos

2.3 Biblioteca

A *Librar* ou Biblioteca que faz uso da Lógica Paraconsistente será criada, com intuito de ajudar ou facilitar a criação de jogos de carta mais especificamente para jogos TCG, a *Library* vem com o objetivo de entregar funções já desenvolvida sem ter a necessidade de ter elaborar as funcionalidades do zero. Com relação *Application Programming Interface* (Interface de programação de aplicações - API) e um *Framework* a uma *Library*, elas podem ter uma certa similaridade, mas tem conceitos distintos.

Biblioteca é uma conjunto de implementações de ações escritos em uma linguagem e importadas no seu código. Nesse caso, há uma interface bem definida para cada comportamento invocado.

A API tem um conceito diferente da Biblioteca, que basicamente contém um conjunto de instruções, rotina e padrões do código que contem acesso de um aplicativo específico via conexão. Com tudo ele vem pra interpretar os dados e integra-los com outras plataformas e softwares, gerando instruções recém-lançadas que vem com intuito de executar por esses softwares (RIBEIRO, 2016).

O *Framework* é a base sólida e padronizada de uma aplicação resumidamente é a unificação de Bibliotecas e API's de forma a oferecer uma estrutura ideal para desenvolver um aplicativo mobile (FINZI, 2016).

2.4 Engenharia de Software

Visando melhorar a qualidade dos produtos de software e aumentar a produtividade no processo de desenvolvimento, surgiu a Engenharia de Software. A Engenharia de Software trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software (FALBO, 2014, p. 2).

2.4.1 Metodologia

2.4.2 UML

Particularmente no tocante à Engenharia de Software, a *Unified Modeling Language* (Linguagem de Modelagem Unificada - UML) pode ser utilizada para modelar todas as etapas do processo de desenvolvimento de software, bem como produzir todos os artefatos de Software necessários à documentação dessas etapas (GUDWIN, 2015, p. 12).

Segundo o mesmo autor (2015), a linguagem UML, por meio de seus diagramas, permite a definição e design de *threads* (tarefas) e processos, que permitem o desenvolvimento de sistemas distribuídos ou de programação concorrente. Da mesma maneira, permite a utilização dos chamados *patterns* são, a grosso modo, soluções de programação utilizadas devido ao seu bom desempenho e a descrição de colaborações esquemas de interação entre objetos que resultam em um comportamento do sistema.

2.4.3 RUP

3 MATERIAIS E MÉTODOS

Para a elaboração do presente trabalho foram adotadas técnicas e metodologias que serão mostrados nos próximos tópicos.

3.1 Linguagem C#

C# ou *C Sharp* é uma linguagem de programação desenvolvido pela Microsoft, que faz parte de sua plataforma *.Net Framework*. Tem como aspecto o ser uma linguagem fortemente tipada que utiliza paradigma de orientação a objeto com sintaxe semelhante C, C++ ou JAVA.

Segundo os autores Edwin e Eugenio (2002) Algumas das características essenciais do C# que podem ser mencionadas:

- Simplicidade: facilidade de codificação com alta performance;
- Completamente orientada a objetos: Tudo é um objeto em C#;
- Fortemente tipada: atribuição de tipos para evitar manipulação imprópria ou incorreta;
- Flexibilidade: caso necessário C# permitir o uso de ponteiros, ms com custo de desenvolver um código não gerenciado, chamado de *unsafe*;
- Linguagem gerenciada: os programa desenvolvido em C# é executado em ambiente gerenciado, na qual, o gerenciamento de memória é feito pelo *Garbage Collector* (Coletor de lixo - GC);

A linguagem C# é utilizada nesse projeto devido sua integração com a plataforma *Unity*, na qual, será mais detalhada no próximo tópico.

3.2 Plataforma Unity

A plataforma Unity é conhecido como uma das melhores plataforma de desenvolvimento de jogos do mundo, justamente porque ela é potencializada em serviços e ferramentas sendo elas 2D e 3D. Segundo Dias (2018) “atualmente a plataforma domina 45% do mercado global de desenvolvimentos de games, segundo a própria empresa; 34% dos 1.000 maiores jogos mobile disponibilizado gratuitamente são feitos com a própria Unity.”

3.3 Visual Studio

REFERÊNCIAS

ABE, J. M. *Aspectos de Computação Inteligente Paraconsistente*. [S.l.: s.n.], 2013. Citado na página 6.

CARVALHO, F. R. de; ABE, J. M. *Tomadas de Decisão com Ferramentas de Lógica Paraconsistente Anotada*. [S.l.: s.n.], 2011. Citado 2 vezes nas páginas 5 e 10.

DIAS, R. *Unity – Guia Completo sobre a Game Engine*. 2018. Disponível em: <<https://producaodejogos.com/unity>>. Citado na página 12.

EDWIN, L.; EUGENIO, R. *C# E .NET Guia do Desenvolvedor*. [S.l.: s.n.], 2002. Citado na página 12.

FALBO, R. de A. *Engenharia de Software*. 2014. Disponível em: <http://www.inf.ufes.br/~falbo/files/ES/Notas_Aula_Engenharia_Software.pdf>. Citado na página 11.

FINZI, E. *API, biblioteca e Frameworks: entenda a diferença entre eles*. 2016. Disponível em: <<https://blog.cedrotech.com/api-bibliotecas-e-frameworks-entenda-diferenca-entre-eles/>>. Citado na página 11.

GUDWIN, R. R. *Engenharia de Software Uma Visão Prática*. 2015. Disponível em: <<http://faculty.dca.fee.unicamp.br/gudwin/sites/faculty.dca.fee.unicamp.br/gudwin/files/ea975/ESUVP2.pdf>>. Citado na página 11.

LOPES, D. D. et al. **Jogos de gestão e estratégia voltados para educação e apoiados em lógica paraconsistente anota evidencial E_{τ}** . *XL IGIP*, p. 1, 2011. Citado na página 5.

RIBEIRO, M. *O que é API e como ele aumenta a produtividade nas empresas*. 2016. Disponível em: <<https://pluga.co/blog/api/o-que-e-api/>>. Citado na página 10.

SILVA FILHO, J. I. da. **Introdução ao conceito de estado Lógico Paraconsistente ϵ_{τ}** . *seleção documental do GLPA*, p. 20–24, 2010. Citado na página 10.

SILVA FILHO, J. Inácio da. **Métodos de Aplicações da Lógica Paraconsistente Anotada de anotação com dois valores-LPA2v**. *seleção documental do GLPA*, p. 20–24, 2006. Citado na página 5.