

Modelagem e análise de níveis de jogos baseadas em Redes de Petri Coloridas

Modeling and analysis of game levels based on Colored Petri Nets

Lucas Emerenciano¹, Hugo Borges¹, Criscilla M. C. Rezende¹,
Joslaine C. Jeske de Freitas¹, Franciny M. Barreto¹

¹Universidade Federal de Jataí (UFJ)
Jataí – GO – Brasil

{lucas.ramos, hugo.borges}@discente.ufj.edu.br

{criscillarezende, joslaine, franciny}@ufj.edu.br

Abstract. Introduction: Despite the growth of the electronic games market, designers still lack formal tools in the early stages of design. Existing works using Petri Nets typically model single levels, ignoring the interconnection of multiple stages. **Objective:** To propose an approach for modeling and analyzing game levels using Hierarchical Colored Petri Nets, enabling the representation of multiple levels and their interconnections. **Methodology:** Game elements are mapped onto the net, where tokens represent the player, places represent states, and transitions represent actions. Hierarchy is used to structure the progression between levels. The models are analyzed through simulation and structural properties of Petri Nets. **Results:** To illustrate the proposed approach, a segment of the game Dark Souls I was used, demonstrating its applicability in the specification and verification of game levels during the design phase.

Keywords Game Design, Hierarchical Colored Petri Nets, Dark Souls I, Formal Modeling, CPN Tools.

Resumo. Introdução: Apesar do crescimento do mercado de jogos eletrônicos, designers carecem de ferramentas formais nas etapas iniciais de projeto. Trabalhos com Redes de Petri costumam modelar níveis únicos, ignorando a interconexão de múltiplas fases. **Objetivo:** Propor uma abordagem para modelagem e análise de níveis de jogos utilizando Redes de Petri Coloridas com hierarquia para representar múltiplos níveis e suas interconexões. **Metodologia:** Elementos do jogo são mapeados para a rede, na qual fichas representam o jogador, lugares representam estados e transições representam ações. A hierarquia é utilizada para estruturar a progressão entre os níveis. Os modelos são analisados por meio de simulação e de propriedades estruturais da rede de Petri. **Resultados:** Para exemplificar a abordagem proposta foi utilizado um recorte do jogo Dark Souls I e demonstrada a aplicabilidade da abordagem na especificação e verificação de níveis de jogo ainda em fase de projeto.

Palavras-Chave Design de Jogos, Redes de Petri Coloridas Hierárquicas, Dark Souls I, Modelagem Formal, CPN Tools.

1. Introdução

Jogos são aplicações multimídia complexas que requerem participação ativa do usuário [Kanode e Haddad 2009]. A criação de um jogo exige o conhecimento de muitas

disciplinas e isso cria barreiras que excluem milhões de criadores de jogos em potencial em uma indústria que, apenas em 2023, gerou uma receita de US\$ 384,9 bilhões, com projeções de crescimento anual de aproximadamente 12% até 2027 [Statista 2024], [Yang et al. 2025].

A literatura aponta que *designers* de jogos frequentemente não utilizam ferramentas para projetar, o que pode levar a atrasos no projeto e custos elevados [Kanode e Haddad 2009], [Neil 2012]. Embora a pesquisa em Engenharia de Software aplicada a jogos esteja amadurecendo, a construção de diagramas que documentem e validem os requisitos na fase de *design* é um desafio a ser superado [Montero-Reyno e Carsí-Cubel 2009], [Chueca et al. 2023].

[Kessing et al. 2012], por exemplo, destacam a falta de consistência entre as representações visuais nos jogos e seu comportamento semântico. Como solução prática, [Marín-Lora et al. 2020] apresentaram um sistema para modelagem da lógica de jogos 2D que reduziu o número de ações e condições durante o desenvolvimento. Já [Luhova et al. 2019] utilizaram a ferramenta Canva para representar a lógica do jogo. Entretanto, os autores apontam que esse tipo de ferramenta carece de algoritmos para realizar análises e garantir que a sequência de atividades seja viável. [Yang et al. 2025], para especificar e visualizar o jogo, utilizaram o Documento de Design de Jogo (*Game Design Document* - GDD) como metodologia principal.

Os modelos formais também são utilizados para especificar e representar os jogos. [Natkin et al. 2004] e [Araújo 2009], por exemplo, utilizaram as redes de Petri para modelar os aspectos lógicos do jogo. [Carvalho 2014b] propôs uma ferramenta para a geração automática de instâncias jogáveis, utilizando as redes de Petri Coloridas. [Barreto e Julia 2017] utilizaram dois tipos de redes de Petri para analisar tanto a sequência lógica das atividades, quanto a topologia do mundo virtual. Já [Nedopetalski et al. 2024] utilizaram uma ferramenta de simulação 3D para representar e simular modelos de cenários de jogos definidos em redes de Petri Coloridas. Tais abordagens, em sua maioria, tratam os jogos como uma única sequência lógica de atividades que precisam ser executadas pelo jogador. Quando se trata de modelos formais, especificamente no uso de redes de Petri, essa estratégia pode levar à construção de modelos excessivamente grandes e complexos, o que dificulta a visualização da especificação, a manipulação e análise do modelo. A modelagem de jogos pode se beneficiar ao considerar que os jogos são naturalmente estruturados em níveis, sendo cada nível composto por um conjunto de atividades que precisam ser executadas pelo jogador em regiões bem definidas do mundo virtual.

Diante deste cenário, o presente trabalho tem como objetivo propor uma abordagem para a modelagem e análise de jogos, baseada em redes de Petri Coloridas Hierárquicas. O jogo é decomposto em níveis, e cada nível é representado por um modelo individual. A abordagem também propõe a conexão entre esses modelos, de modo análogo ao funcionamento dos jogos, onde os níveis não são isolados, mas sim interligados, compondo o sistema como um todo. Com isso, é possível visualizar a especificação do jogo de forma modular e também integrada. A conexão dos níveis permite que seja realizada uma análise global do jogo, por meio de simulação e verificação de propriedades das redes de Petri. A partir disso, é possível identificar potenciais problemas de especificação e também garantir o bom funcionamento do jogo. Para a

criação, simulação e análise dos modelos foi utilizado o software CPN Tools. A aplicação prática dessa abordagem é ilustrada por meio de um estudo de caso do jogo *Dark Souls I*.

2. Literatura Correlata

A modelagem e simulação de cenários de jogos utilizando redes de Petri têm sido objeto de diversos estudos acadêmicos, principalmente devido ao potencial dessas ferramentas na fase de pré-produção dos jogos. Descritas inicialmente por [Petri 1962], as Redes de Petri (RPs) são ferramentas visuais e matemáticas para modelagem formal de sistemas, oferecendo um alto nível de abstração [Peterson 1981, Jensen e Kristensen 2009]. Uma RP é um grafo orientado bipartido composto por lugares (condições/estados) e transições (eventos/ações). As fichas (*tokens*), associadas aos lugares, determinam o estado atual do sistema (marcação) [Reis 2006]. O fluxo de controle é ditado pelos arcos direcionados e ponderados, que interligam lugares e transições e indicam a quantidade de fichas consumidas ou produzidas em um disparo [Reisig 2013].

O trabalho de [Araújo 2009] investiga as Redes de Petri como alternativa eficiente a linguagens tradicionais, como UML (*Unified Modeling Language*), para modelar sistemas e fluxos de jogo, destacando a viabilidade de verificação formal, apesar da complexidade visual em diagramas extensos. Ampliando esse contexto para jogos *multiplayer*, [Barreto et al. 2020] utilizaram Redes de Petri Coloridas (RPC) para representar atividades concorrentes, implementando funções de guarda e um modelo de temporização [Jensen e Kristensen 2009, Jensen et al. 2007].

Buscando tornar o uso das RPCs mais acessível, [Silva 2021] propôs sua integração com o software de simulação 3D SIMIO, permitindo visualizações mais intuitivas na pré-produção. Já em uma abordagem voltada à geração automatizada, [Carvalho 2014a] apresentou a ferramenta CPN Games, que interpreta modelos do CPN Tools para criar instâncias jogáveis multinível, permitindo a identificação prévia de problemas lógicos como *deadlocks*.

Os estudos mencionados ilustram o potencial e a aplicabilidade das Redes de Petri em diversas etapas do design e desenvolvimento de jogos, desde a modelagem de mecânicas e fluxos até a geração automática e simulação visual. Neste trabalho serão utilizadas as RPCs para a modelagem e análise de níveis de jogos. Em particular, as RPCs oferecem recursos avançados como hierarquia, permitindo decompor sistemas complexos em submodelos interconectados, facilitando a reutilização e a modularidade [Jensen e Kristensen 2009]. Além disso, para apoiar a modelagem e simulação, destaca-se a ferramenta gratuita CPN Tools, que oferece edição gráfica, simulação automática e análise formal pelo cálculo do espaço de estados [Jensen et al. 2007].

3. Modelagem dos Níveis de Jogo

O jogo *Dark Souls I* será usado para ilustrar a abordagem apresentada neste trabalho. Trata-se de um jogo de ação e aventura ambientado em um mundo assolado por uma maldição, no qual o jogador assume o papel de um personagem portador de uma alma poderosa. Sua missão consiste em enfrentar inimigos, explorar áreas interconectadas, resolver enigmas e, por fim, impedir que o mundo sucumba às trevas. Para a aplicação da abordagem, foi selecionado o cenário *Mundo Pintado de Ariamis*. Nele, o jogador

precisa percorrer diferentes áreas, coletando itens especiais que serão usados nos outros níveis que compõem esse cenário.

3.1. Modelo Hierárquico

Um jogo pode ser estruturado em níveis [Barreto e Julia 2021], [Sugiarti et al. 2023]. Um nível corresponde a uma etapa do jogo que guia a progressão do jogador. Essas etapas também são chamadas de fases, capítulos, missões, entre outros. Para ir de um nível a outro, o jogador deve completar uma série de atividades. A atividade é um elemento atômico que consiste em uma ação realizada pelo jogador, como, por exemplo, recuperar um determinado item. Os itens do jogo podem assumir diferentes funções. Alguns servem apenas como acessórios, enquanto outros são necessários para liberar áreas ou outras atividades. Um item adquirido em um nível pode ser usado imediatamente ou não. Em alguns casos, eles são coletados em um nível do jogo e utilizados para executar atividades em um outro nível. Portanto, esses elementos são importantes para que o fluxo do jogo ocorra corretamente.

Para modelar a estrutura lógica do nível, cada elemento da RPC é associado a um elemento de jogo. A ficha (*token*) representa o jogador e seu inventário. A ficha possui então diferentes variáveis que armazenam as características do jogador e os itens coletados durante o jogo. Os lugares da rede representam os estados do nível, como zonas de exploração, áreas de combate ou pontos de transição entre níveis. As transições correspondem às atividades, como, por exemplo, coletar itens ou derrotar inimigos. Uma das características da RPC é que ela permite a implementação de funções por meio da linguagem de programação funcional Standard ML. Essas funções podem ser associadas tanto aos arcos quanto às transições da rede. Quando associadas aos arcos, as funções têm como objetivo modificar as fichas, seja em termos de valor ou em quantidade. Já as funções de guarda constituem um tipo específico de função lógica associada às transições, responsável por permitir ou impedir seu disparo, de acordo com os valores das fichas que as habilitam.

A RPC possui o recurso da hierarquia, que permite a criação de sub-redes que podem detalhar seções específicas do jogo, como níveis inteiros. A principal vantagem é a capacidade de conectar essas sub-redes a uma rede principal, representando todos os níveis do jogo. Assim, uma transição principal atua como “caixa preta” (super-página), enquanto a sub-rede detalha o nível, preservando a integridade da ficha do jogador.

A Figura 1 ilustra o primeiro nível do cenário *Mundo Pintado de Ariamis* do jogo *Dark Souls I*. No modelo, os lugares são representados graficamente por elipses e correspondem ao estado atual do jogador. As transições (representadas como retângulos) correspondem às atividades que o jogador precisará executar, e a ficha representa o jogador. Na RPC as fichas possuem tipos de dados associados. Na Figura 1, a ficha do jogador possui como valor o identificador do jogador e um item, representados pela inscrição 1'(*player*, *vazio*) no lugar *Santuário*. Especificamente, a ficha é do tipo *PLAYER*, um tipo formado pelo produto de duas *strings*. A primeira *string* corresponde ao nome do jogador, que neste exemplo é “*player*”. A segunda corresponde a um item que pode ser coletado ao longo do jogo. Inicialmente, o jogador não possui esse item, então esse valor é definido como “*vazio*”.

O fluxo de atividades no modelo na Figura 1 procede da seguinte forma:

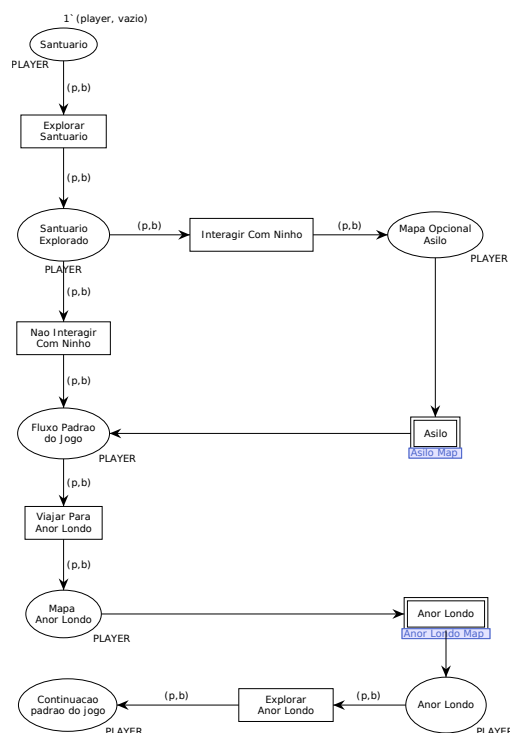


Figura 1. O primeiro nível da missão Mundo Pintado de Ariamis, do jogo *Dark Souls I*.

- **Início da Missão:** o jogador começa a missão no lugar *Santuário* e, após executar a atividade *Explorar Santuário*, atinge o estado *Santuário Explorado*. Neste estado, o jogador encontra um ninho gigante e deve escolher entre *Interagir Com Ninho*, que leva a um nível opcional do jogo ou *Não Interagir Com Ninho*, seguindo o fluxo padrão.
- **Mapa Asilo dos Mortos-Vivos:** caso o jogador opte por executar a atividade de *Interagir Com Ninho*, então ele será conduzido para um novo nível do jogo. No modelo da Figura 1, a transição *Asilo* corresponde a esse novo nível, denominado no jogo como *Asilo dos Mortos-Vivos*. Assim, ao disparar a transição *Asilo*, a ficha do jogador seguirá para o modelo correspondente do nível, ilustrado na Figura 2. O fluxo de atividades deste nível envolve as atividades de derrotar ou de evitar o combate com o chefe. Ambas as alternativas levam ao estado *Caminho Liberado*. Neste ponto, o jogador poderá coletar o item *boneca* ao executar a atividade *Pegar Boneca*. Ao recuperar o item, a ficha do jogador precisa ser alterada, indicando que agora o item está disponível em seu inventário. Assim, o valor correspondente ao item é passado por meio do arco de saída da transição *Pegar Boneca*. Em seguida, o jogador poderá retornar ao fluxo padrão do jogo, representado pelo lugar *Fluxo Padrão do Jogo*. Note que este lugar está marcado com o rótulo *out*. Quando a ficha alcança esse lugar, significa que o jogador concluiu o nível e seguirá para o próximo.
- **Anor Londo:** na Figura 1, a transição *Anor Londo* representa um novo nível, ilustrado pela Figura 3. Ao encontrar a *Catedral de Anor Londo*, o jogador pode interagir com uma pintura. Uma função de guarda controla o fluxo: possuir o item *boneca* habilita a transição *Ariamis*, direcionando-o a um novo nível;

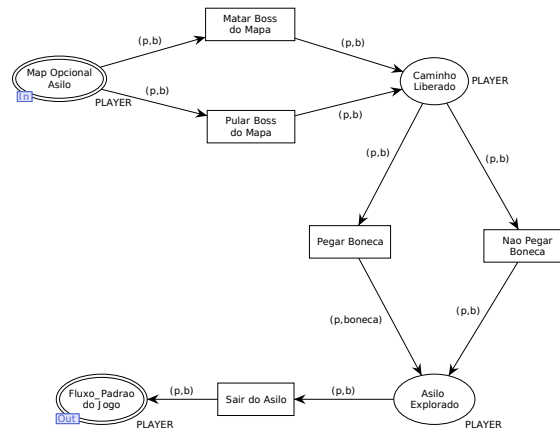


Figura 2. O modelo do nível Asilo dos Mortos-Vivos.

caso contrário, a transição habilitada o retorna ao estado anterior no modelo da Figura 3.

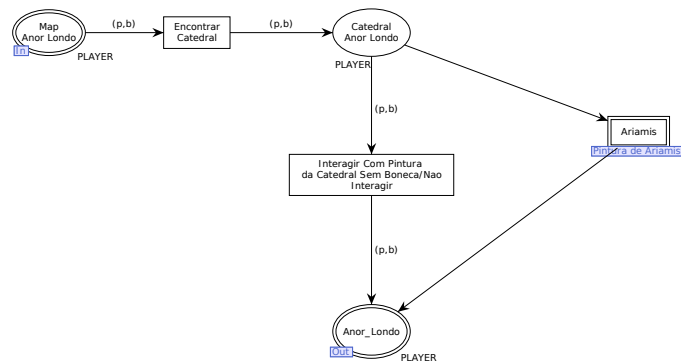


Figura 3. O modelo do nível Anor Londo.

- **Ariamis:** a Figura 4 ilustra a sub-rede do nível *Pintura de Ariamis*. Após explorar a área e derrotar inimigos, o jogador retorna ao nível anterior para seguir com o fluxo padrão do jogo. O lugar *Continuacao padrao do jogo* na Figura 1 indica o fim da missão.

3.2. Análise do modelo

Uma vez que o modelo do jogo está definido, é possível realizar a análise formal para verificar o seu funcionamento e identificar possíveis falhas. O CPN Tools permite dois tipos principais de análise automática: a simulação e o cálculo do espaço de estados. A simulação pode ser executada de forma automática, por meio de replicações, ou de forma interativa, passo a passo. Em particular, a simulação passo a passo permite acompanhar detalhadamente os disparos das transições e a progressão da ficha entre os diferentes níveis interconectados.

A Figura 5 ilustra o modelo do jogo sendo simulado no CPN Tools. O jogador inicia no lugar *Santuario*, e apenas a transição *Explorar Santuario* está habilitada para disparo.

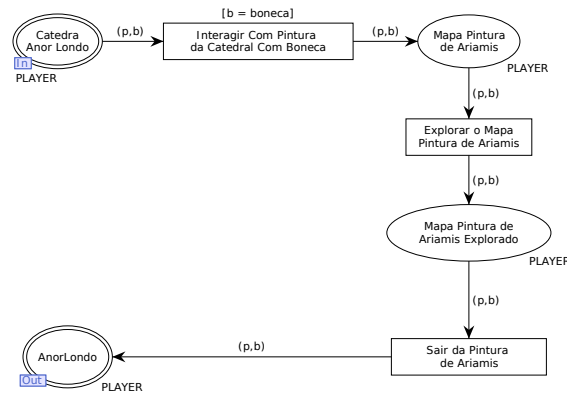


Figura 4. O modelo do nível secreto da missão Mundo Pintado de Ariamis.

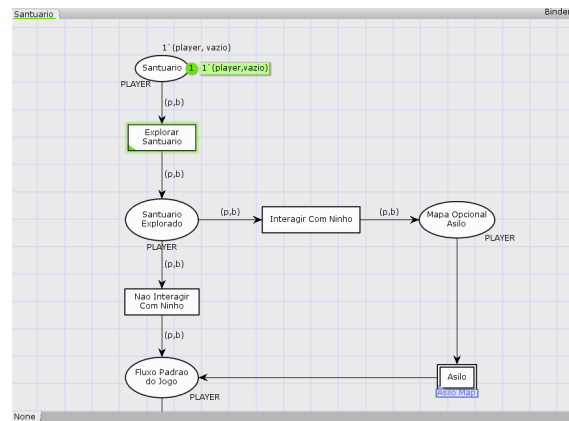


Figura 5. Primeiro passo na simulação da missão Mundo Pintado de Ariamis.

Após o disparo da transição, a ficha do jogador é depositada no lugar *Santuário Explorado*, como mostra a Figura 6. O retângulo amarelo com os valores das variáveis da ficha que aparece junto à transição habilitada demonstra a atribuição de variáveis antes do disparo. Neste estado, o jogador pode escolher entre duas atividades. Para fins de exemplificação, foi escolhido disparar a transição *Interagir Com Ninho*.

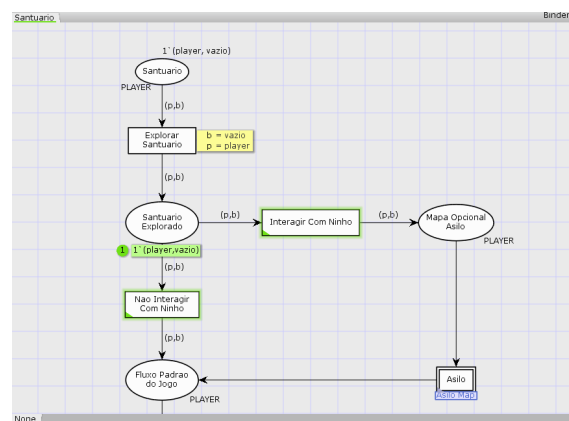


Figura 6. Primeira escolha de transições que a ficha do jogador realiza.

A Figura 7 ilustra como o modelo se comporta com a hierarquia. O lugar *Mapa*

Opcional Asilo, com o rótulo *in*, atua como o ponto de entrada para a sub-rede no nível *Asilo*. Após disparar as transições do nível, a ficha alcança o lugar *Continuação Padrão do Jogo*, que representa o ponto de saída da sub-rede e o retorno para o modelo principal, como ilustrado na Figura 8. É possível observar que o valor da ficha foi atualizado de acordo com as atividades executadas no nível.

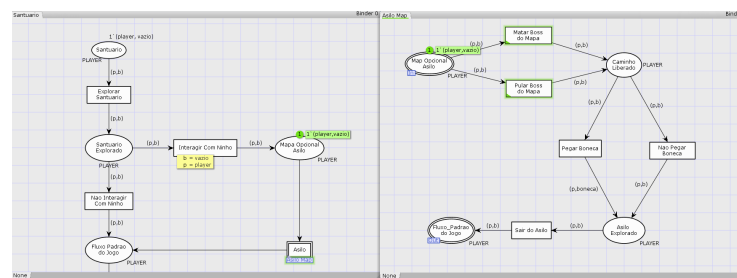


Figura 7. Ficha do jogador na sub-rede *Asilo*.

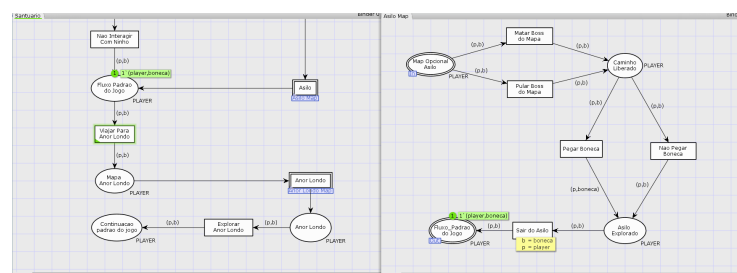


Figura 8. Ficha do jogador de volta a rede principal.

Ao alcançar o nível *Anor Londo*, e após disparar a transição *Encontrar Catedral*, o jogador pode seguir dois caminhos. A lógica de disparo é controlada pela função na transição *Interagir Com Pintura da Catedral Com Boneca* que possui a guarda $[b = boneca]$, permitindo o acesso ao próximo nível somente se o item foi recuperado no nível anterior. Na Figura 9, como o item consta na ficha, a transição *Ariamis* fica habilitada, liberando o acesso ao nível.

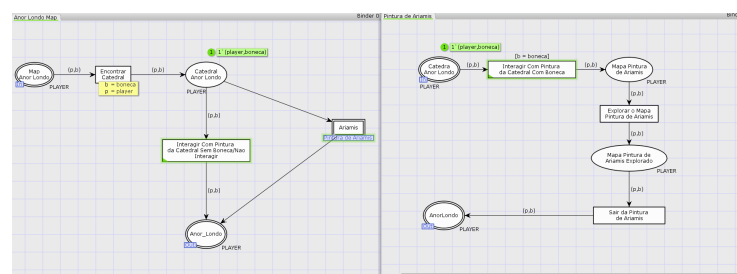


Figura 9. Sub-rede do nível *Ariamis* com o item *boneca* na ficha.

Ao disparar a transição *Interagir Com Pintura da Catedral Com Boneca*, a ficha percorrerá todas as transições do nível e em seguida retornará para a rede principal.

A simulação passo a passo permite observar diretamente o comportamento dinâmico do modelo. É possível identificar, em cada modelo de nível, quais transições estão habilitadas e como ocorrem seus disparos, além de acompanhar a evolução da ficha.

Esse tipo de simulação também favorece a identificação de problemas, como transições que nunca disparam, estados inalcançáveis ou situações de impasse causadas pela lógica incorreta das atividades projetadas.

Outra forma possível de analisar o modelo hierárquico é por meio do cálculo do espaço de estados, usando a ferramenta *State Space* do CPN Tools. Por meio dela é possível investigar propriedades estruturais da rede, como a limitabilidade e a vivacidade [Murata 1989].

A Figura 10 ilustra o trecho do relatório referente à limitabilidade. As colunas *Upper* e *Lower* indicam, respectivamente, o número máximo e mínimo de fichas acumuladas nos lugares. Como observado, nenhum lugar contém mais de uma ficha, o que corresponde ao comportamento esperado com um único jogador atuando no cenário, caracterizando o modelo como 1-limitado.

Boundedness Properties		
Best Integer Bounds		
	Upper	Lower
Anor_Londo_Map'Catedral_Anor_Londo 1	1	0
Asilo_Map'Asilo_Explorado 1	1	0
Asilo_Map'Caminho_Liberado 1	1	0
Pintura_de_Ariamis'Mapa_Pintura_de_Ariamis 1	1	0
Pintura_de_Ariamis'Mapa_Pintura_de_Ariamis_Explorado 1	1	0
Santuario'Anor_Londo 1	1	0
Santuario'Continuacao_padrao_do_jogo 1	1	0
Santuario'Fluxo_Padrao_do_Jogo 1	1	0
Santuario'Mapa_Anor_Londo 1	1	0
Santuario'Mapa_Opcional_Asilo 1	1	0
Santuario'Santuario 1	1	0
Santuario'Santuario_Explorado 1	1	0

Figura 10. Resultado da propriedade de Limitabilidade do modelo da missão Mundo Pintado de Ariamis.

A Figura 11 apresenta os resultados referentes à vivacidade. O relatório indica a ausência de *dead markings* e *dead transitions*, comprovando que a ficha flui sem interrupções e que todas as transições podem ser disparadas. No contexto do jogo, isso mostra que o jogador não ficará preso em nenhum nível e conseguirá progredir sem encontrar falhas de *design* lógico ou narrativo que bloqueiem a jornada.

Liveness Properties	
Dead Markings	None
Dead Transition Instances	None
Live Transition Instances	All

Figura 11. Resultado da propriedade de Vivacidade do modelo da missão Mundo Pintado de Ariamis.

Uma forma de verificar essas propriedades consiste na análise da propriedade de *soundness*, conforme definida em [van der Aalst 1997]. Para isso, cria-se uma rede estendida a partir do modelo original, adicionando uma transição *t* que conecta o lugar final de volta ao lugar inicial. Essa modificação permite transformar o modelo em uma

estrutura cíclica, adequada à análise por meio do espaço de estados. Se a rede estendida for viva e limitada, conclui-se que a rede original satisfaz a propriedade de *soundness* [van der Aalst 1997].

A ferramenta *State Space* do CPN Tools realiza automaticamente o cálculo do espaço de estados e gera um relatório com as propriedades estruturais verificadas. Ao aplicar esse método de análise ao modelo, verificou-se que a rede satisfaz a propriedade de *soundness*. Em termos do funcionamento do jogo, isso implica que todas as atividades podem ser executadas, que todos os estados, em todos os níveis, são alcançáveis e que o jogador poderá percorrer todos os níveis planejados do início ao fim.

4. Conclusão

A presente pesquisa propôs uma abordagem para a modelagem de níveis de jogos por meio do uso de Redes de Petri Coloridas Hierárquicas. Cada nível é representado como um modelo individual, composto por estados e atividades que o jogador deve executar. A ficha da rede representa o jogador, incorporando suas características e as informações necessárias ao progresso no jogo. Embora cada nível seja modelado de forma independente, eles seguem uma sequência bem definida. Nesse sentido, o uso de hierarquia em Redes de Petri Coloridas possibilita a conexão entre os diferentes modelos por meio de estados de entrada e saída, garantindo que a ficha percorra os níveis e seja atualizada ao longo de sua progressão.

Também foram apresentados dois métodos de análise do modelo. O primeiro, baseado em simulação, permite verificar a dinâmica do sistema passo a passo, possibilitando a identificação de caminhos percorridos e eventuais inconsistências semânticas na especificação do jogo. O segundo método é baseado na análise de propriedades estruturais da rede, realizada de forma automática por meio do CPN Tools. Essa análise permite verificar a correção estrutural do modelo e, consequentemente, avaliar se os níveis projetados podem ser executados conforme especificado.

A abordagem transforma a especificação do jogo em um modelo visual e formal, facilitando a representação e verificação ainda na fase de projeto, servindo como suporte para o *game* e *level design*. A utilização de um recorte do jogo *Dark Souls I* demonstrou de forma clara e realista a aplicabilidade dessa técnica. Cabe destacar que a análise do modelo foi realizada considerando as variáveis da ficha que foram apresentadas neste trabalho. Conforme a quantidade de variáveis na ficha aumenta, como em uma eventual adição de parâmetros de combate dinâmico, por exemplo, o cálculo do *state space* fica exponencialmente mais complexo, devido às diferentes instâncias geradas. Entretanto, o uso das Redes de Petri permite a aplicação combinada de diferentes estratégias para análise como, por exemplo, a simulação, que permite avaliar o comportamento dinâmico do modelo sem depender exclusivamente do cálculo de todo o *state space*.

Trabalhos futuros podem considerar a interação entre múltiplos jogadores, o compartilhamento de recursos e a incorporação de tempo nas atividades, ampliando a análise do impacto temporal na experiência do jogador. Por fim, informa-se que a Inteligência Artificial generativa (Gemini) foi utilizada exclusivamente para revisão gramatical e tradução do *abstract*, com todos os resultados integralmente validados pelos autores.

Referências

- Araújo, M. (2009). Modeling games with petri nets. In *Proceedings of DiGRA 2009 Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*, London. Digital Games Research Association (DiGRA).
- Barreto, F. M. et al. (2020). Uma abordagem baseada em redes de petri para modelagem, análise e simulação de cenários de vídeo games singleplayer e multiplayer.
- Barreto, F. M. e Julia, S. (2017). Modeling of video games using workflow nets and state graphs. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*, pages 261–266, Fortaleza, CE. SBC.
- Barreto, F. M. e Julia, S. (2021). Formal approach based on petri nets for modeling and verification of video games. *Computing and Informatics*, 40(1):216–248.
- Carvalho, V. V. d. S. (2014a). Geração de jogos multiníveis e com múltiplos usuários por meio de modelagem em redes de petri coloridas hierárquicas.
- Carvalho, V. V. S. (2014b). Geração de jogos multiníveis e com múltiplos usuários por meio de modelagem em redes de petri coloridas hierárquicas. Dissertação (mestrado em engenharia de teleinformática), Universidade Federal do Ceará, Centro de Tecnologia.
- Chueca, J. et al. (2023). The consolidation of game software engineering: A systematic literature review of software engineering for industry-scale computer games. *Information and Software Technology*, page 107330.
- Jensen, K. e Kristensen, L. M. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer, Berlin; Heidelberg.
- Jensen, K., Kristensen, L. M., e Wells, D. (2007). Introduction to cpn tools. Technical report, Aarhus University, Aarhus, Denmark.
- Kanode, C. M. e Haddad, H. M. (2009). Software engineering challenges in game development. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 260–265, Las Vegas, NV. IEEE.
- Kessing, J., Tutenel, T., e Bidarra, R. (2012). Designing semantic game worlds. In *Proceedings of the The third workshop on Procedural Content Generation in Games*, pages 1–9, Raleigh, NC. AAAI Press.
- Luhova, T., Blazhko, O., Troianovska, Y., e Riashchenko, O. (2019). The canvas-oriented formalization of the game design processes. In *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pages 1254–1259, Lviv, Ukraine. IEEE.
- Marín-Lora, C., Chover, M., e Sotoca, J. M. (2020). A game logic specification proposal for 2d video games. In *Trends and Innovations in Information Systems and Technologies: Volume 1 8*, pages 494–504, Cham, Switzerland. Springer.
- Montero-Reyno, E. e Carsí-Cubel, J. Á. (2009). A platform-independent model for videogame gameplay specification. In *Proceedings of DiGRA 2009 Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*, London. Digital Games Research Association (DiGRA).

- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.
- Natkin, S., Vega, L., e Grünvogel, S. (2004). A new methodology for spatiotemporal game design. In *Proceedings of CGAIDE 2004, 5th Game-On International Conference on Computer Games: Artificial Intelligence, Design and Education*, pages 109–113, Reading, UK. Eurosis (European Organisation for Simulation and Allied Sciences).
- Nedopetalski, F., Silva, E. F., Barreto, F. M., e Julia, S. (2024). 3d video game simulation based on colored petri net and simio. In *International Conference on Information Technology-New Generations*, pages 521–530. Springer.
- Neil, K. (2012). Game design tools: Time to evaluate. In *Proceedings of Nordic DiGRA 2012 Conference*.
- Peterson, J. L. (1981). *Petri Nets: An Introduction*. Prentice-Hall, Englewood Cliffs, NJ.
- Petri, C. A. (1962). *Kommunikation mit Automaten*. Phd thesis, Institut für Instrumentelle Mathematik, Bonn.
- Reis, J. (2006). *Modelagem, análise e implementação de sistemas de eventos discretos usando Redes de Petri*. Editora da Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC.
- Reisig, W. (2013). *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer, Berlin; Heidelberg.
- Silva, E. F. (2021). Simulação 3d de cenários de vídeo games modelados por redes de petri coloridas no software simio.
- Statista (2024). Video games: Market data & analysis. Market Insights Report.
- Sugiarti, Y., Zulfikar, M. N., Khalisha, N., Anwas, E. O. M., Ifitah, K. N., e Permatasari, A. D. (2023). Component level design in learning-based video games. In *2023 11th International Conference on Cyber and IT Service Management (CITSM)*, pages 1–4. IEEE.
- van der Aalst, W. M. P. (1997). Verification of workflow nets. In *Application and Theory of Petri Nets 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426, Berlin, Heidelberg. Springer-Verlag.
- Yang, R., Wan, Y., Li, S., e Lyu, M. R. (2025). 90% faster, 100% code-free: Mllm-driven zero-code 3d game development. *arXiv preprint arXiv:2509.26161*.