

Testes de software e simulações como ferramentas para game design

Davi Sadao Araki

Bacharelado em Sistemas de Informação
Universidade Federal de Juiz de Fora
Juiz de Fora, Brasil
davi.araki@gmail.com

Igor de Oliveira Knop

Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora
Juiz de Fora, Brasil
igorknop@ice.ufjf.br

Abstract—O projeto de jogos modernos de tabuleiro é um processo iterativo complexo. Esse processo é custoso no uso de pessoas e também dependente da experiência do projetista. Este trabalho propõe um método amparado por algumas técnicas e ferramentas de engenharia de software para controlar o processo de desenvolvimento de um jogo de tabuleiro real e seus testes subsequentes. Um modelo digital é construído desde a idealização do jogo, com as regras regidas por testes unitários, e simulações sobre esse modelo aliviam em grande parte a necessidade dos testes com humanos nas etapas iniciais. Um jogo é criado e métricas sobre o modelo simulado são colhidas, permitindo identificar ajustes nas regras em minutos ao invés de se levar semanas de testes com pessoas reais.

Keywords—teste de software, modelagem e simulação, jogos de mesa

I. INTRODUÇÃO

Jogos são atividades voltadas para o entretenimento. Essas atividades não possuem propósito externo além do próprio ato voluntário de jogar [1]. Essa independência de objetivo externo cria uma realidade artificial, na qual os comportamentos, valores e objetivos, são definidos pelas regras impostas aos jogadores antes do início da atividade. Ressalta-se que as definições atribuídas ao conceito de jogos na literatura, tendem a variar, além de serem debatidas em diversos contextos e aplicações [2]. No presente trabalho, consideramos como jogo, um sistema no qual os jogadores se envolvem em um conflito artificial, restrito por um conjunto de regras, que tem um resultado final quantificável [3].

Em particular, um grupo de jogos de tabuleiro têm sido conhecido como *designer's games* ou jogos de tabuleiros modernos por trazerem o nome do autor na capa [4]. Esses são jogos desenvolvidos a partir de 1995, com o lançamento de Colonizadores de Catan [5] se tornaram populares na Europa e, na sequência, no mundo todo, seguindo uma linha editorial parecida com o mercado literário. O foco da maioria desses jogos está no perfil tático ou estratégico, envolvendo um grande conjunto de relações e dinâmicas ao longo da partida. Atualmente, o BoardGameGeek, site que atua como o maior banco de dados de jogos do mundo, possui em torno de 108 mil jogos cadastrados, no momento da escrita deste trabalho [6]. Os lançamentos anuais apresentam um crescimento contínuo nas últimas duas décadas, demonstrando a demanda por esse tipo de entretenimento.

Para se desenvolver um jogo, são utilizados três pilares principais, também chamados de arcabouço MDA (do inglês, *Mechanics, Dynamics and Aesthetics*, ou Mecânicas, Dinâmicas e Estética em uma tradução livre) [7]. Desse modo, os Mecanismos descrevem as ações disponíveis aos jogadores e são restritas pelo conjunto de regras do jogo. A Dinâmica são os comportamentos emergentes dos elementos componentes, do sistema ou das interações dos jogadores. Já a estética engloba aspectos temáticos, narrativos e respostas emocionais dos jogadores [8]. É importante dizer que existem diversos métodos para desenvolvimento de jogos. Alguns projetistas partem do tema (estética) para só depois experimentar com os mecanismos de jogo (mecânica). Outros já fazem o caminho inverso, criando primeiro os mecanismos e dinâmicas para só depois em uma fase de produção escolherem o tema junto aos editores. Existem 53 mecanismos cadastrados no BoardGameGeek. Eles são combinados e readaptados para criar jogos modernos. Esses mecanismos estão sendo atualizados e reorganizados após estudos mais amplos do sistema de categorização [9], [10].

Esses jogos são testados em sessões de *playtest* ou testes de jogabilidade. Esse tipo de teste é realizado por meio de sessões nas quais convidados jogam o protótipo. O autor do jogo também pode observar o comportamento dos jogadores a fim de averiguar tanto a funcionalidade e equilíbrio da partida quanto o entretenimento dos jogadores. Essas sessões podem levar horas, fazendo com que o processo seja cansativo e de difícil controle, principalmente nas fases iniciais do projeto. Além disso, os resultados obtidos por esse tipo de metodologia nem sempre são claros para o projetista.

Visto o processo de desenvolvimento descrito, percebe-se que os projetos de jogos têm algumas características em comum com os processos de desenvolvimento ágil, como: a rápida mudança de requisitos; a prototipação rápida; a necessidade de se manter o processo leve e a adequação rápida a mudanças. Entende-se como desenvolvimento ágil como uma disciplina que estuda processos ou ferramentas práticas para criação de produtos. Essa metodologia pode ser aplicada inclusive, no desenvolvimento de softwares [11].

Nessa perspectiva, esse desenvolvimento pode ser dirigido por testes (do inglês Test Driven Development ou TDD), estendida como uma técnica que tem a premissa de escrever

primeiro os testes antes de implementar o sistema. Dessa forma, esse teste visa facilitar o entendimento do projeto. O processo deve ser realizado em três passos ordenados, a saber: (1º) escrever um caso de teste para um requisito e observar se o teste irá falhar; (2º) escrever o código que fez o teste passar e observar todos aqueles testes que foram feitos passarem; (3º) realizar refatoração de código. Os testes são aplicados a trechos de código determinísticos, diferentemente de uma grande parcela dos jogos, que possui alguma forma de aleatoriedade em seus mecanismos básicos. Isso impõe dificuldades no projeto dos testes fazendo com que técnicas especiais sejam desenvolvidas para contornar essas características [12].

Visto o cenário discutido, a presente pesquisa busca elucidar se as técnicas e ferramentas de desenvolvimento ágil de software, podem melhorar o processo de desenvolvimento de projetos e de testes dos jogos modernos. Para tal, um conjunto de técnicas e ferramentas oriundas da engenharia de software é aplicada na criação de um novo jogo de mesa, como forma de explorar comportamentos observados empiricamente pela comunidade de jogos. Nesse caso, O TDD foi utilizado como método para a modelagem do jogo de forma a facilitar a transposição das regras desejadas para as regras escritas.

II. MÉTODO

Esta pesquisa exploratória envolveu a criação de um jogo de tabuleiro moderno sobre um conjunto de regras simples e de curta duração. O projeto do jogo foi inteiro gerido e auxiliado pelo uso de ferramentas de software.

Cada regra do jogo era descrita inicialmente por testes unitários e posteriormente implementada. Com conjunto de regras implementado, agentes de software eram empregados para realizar testes de jogabilidade e o resultado das partidas registrado.

Três agentes de software foram implementados para servirem de jogadores de testes de estresse e fornecer dados para balanceamento. As decisões de balanceamento no projeto foram direcionadas para os valores de risco e recompensa das cartas individuais, bem como os limites de rodada e pontuação para condição de vitória.

A cada iteração com as regras e valores das cartas, 500 partidas simuladas era realizadas e uma análise pelo projetista permite observar o comportamento do jogo. Ao observar os resultados dos testes automatizados de regras e dos resultados dos testes de jogabilidade, o projetista pode realizar novas mudanças nas regras e componentes. O que gera um novo ciclo fechado e iterativo de projeto de jogo, simulação e análise.

III. DESENVOLVIMENTO

A. Estética, Mecanismos e Dinâmica

O protótipo de jogo foi desenvolvido com a estética e um tema popular em mente: representar uma interação em redes sociais entre pessoas. O objetivo do jogador é se tornar o maior influenciador digital, escolhendo tópicos para fazer seus envios de textos e vídeo. Entretanto, cada tópico tem um risco de ser denunciado à administração da rede social, que é diretamente

proporcional a chance de receber envolvimento, representado por “curtidas”.

A mecânica da etapa testada é a escolha de uma carta de tópico pelo jogador, após a escolha o jogador abre um número de cartas indicado na carta e contabiliza os resultados. Se um número denúncias ultrapassa um limiar, o tópico é eliminado.

A dinâmica é idealizada como uma corrida baseada em gerenciamento de risco e recompensa: ganha aquele jogador que gerenciar melhor escolher a hora certa para se arriscar em função da proximidade da vitória e desempenho dos oponentes.

B. Ferramental Computacional

A abordagem foi desenvolvida utilizando JavaScript na plataforma Node.js¹. O núcleo do jogo foi modelado utilizando os pacotes *boardgame.io*², feito especificamente para jogos discretos, como é o caso da maioria dos jogos de tabuleiro.

Durante a modelagem, O TDD foi aplicado pelo uso do pacote de testes *Jest.js*³. Através dele, as regras do jogo foram descritas como testes de software e depois implementadas na plataforma de simulação.

Como agentes para o jogo, os três perfis de jogador foram criados na forma de jogadores automatizados: o *jogador seguro* (P0), o *jogador equilibrado* (P1) e o *jogador corajoso* (P2). Programados de forma a adotar uma postura mais ou menos arriscada nas tomadas de decisão. Os agentes foram implementados com uma heurística simples de observar o estado final do jogo e tomar sua decisão.

IV. RESULTADOS E DISCUSSÃO

As simulações para os três perfis foram realizadas com P0, P1 e P2. O objetivo dos testes de jogabilidade foi identificar as condições de término do jogo e como elas afetam no resultado.

A. Cenário 1: Término do jogo com 15 Curtidas e eliminação com 5 Denúncias

No cenário inicial é observado nas Fig. 1a e 1c uma grande quantidade de vitórias do P0, em virtude do número de curtidas necessárias ser alta e os outros perfis acabam sendo eliminados por denúncias (Fig. 1b).

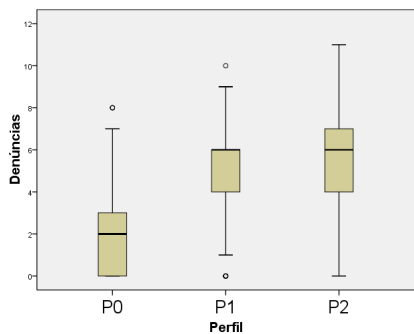
B. Cenário 2: Término do jogo com 10 Curtidas e eliminação com 5 Denúncias

Em um segundo cenário, foi realizada a diminuição do número de Curtidas necessárias para a vitória. O número de vitórias de P0 cai, mas ainda se mantém a estratégia com maior número de vitórias em relação às demais, já que ela mantém a média de denúncias abaixo do limite.

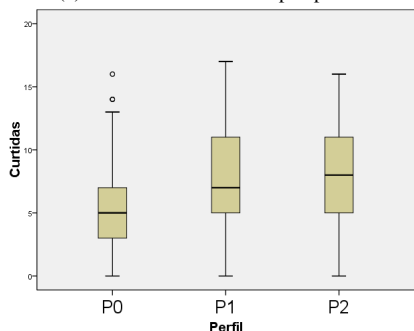
¹<https://nodejs.org>

²<https://boardgame.io>

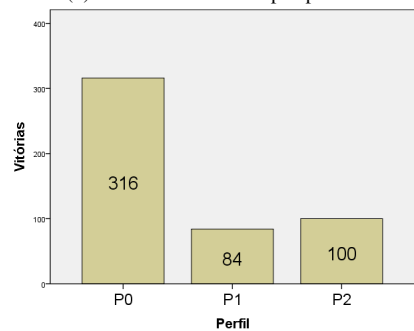
³<https://jestjs.io>



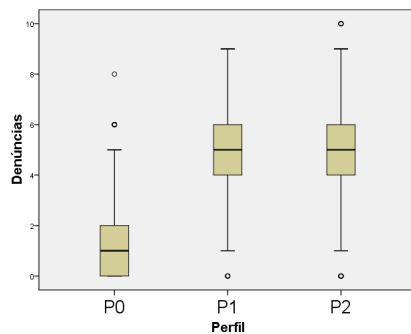
(a) Cenário 1: Denúncias por perfil.



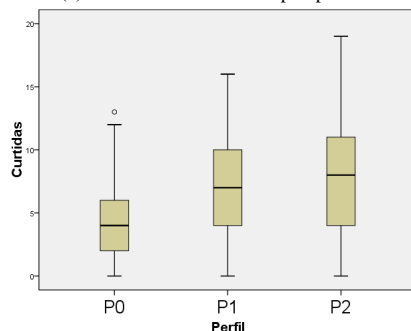
(b) Cenário 1: Curtidas por perfil.



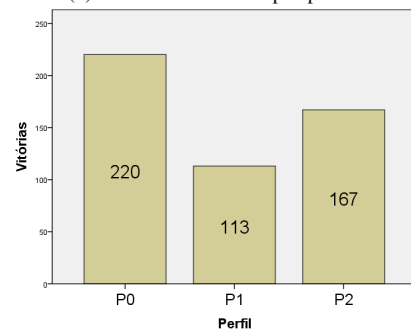
(c) Cenário 1: Vitórias por perfil.



(a) Cenário 2: Denúncias por perfil.



(b) Cenário 2: Curtidas por perfil.



(c) Cenário 2: Vitórias por perfil.

Fig. 1. Primeiro cenário com término do jogo com 15 Curtidas e eliminação com 5 Denúncias.

Fig. 2. Segundo cenário com o término do jogo com 10 Curtidas e eliminação com 5 Denúncias.

C. Cenário 3: 10 Curtidas necessárias e 6 Denúncias

Num terceiro cenário, com o aumento da tolerância a denúncias, a quantidade de vitórias de P0 e P2 ficam muito próximas, se tornando o cenário mais interessante. Dessa forma os jogadores ainda possuem uma maior liberdade de escolha durante seu jogo em resposta aos oponentes se preocupar com uma tendência do sistema *a priori*.

D. Cenário 4: 10 Curtidas necessárias e 7 Denúncias

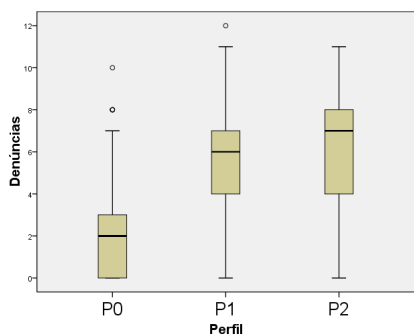
Com um novo aumento no limite de denúncias, observa-se um aumento na porcentagem de vitórias de P2 e a redução de vitórias de P0, obrigando-o ficar mais em jogo. Ainda próximo do Cenário 3, mas que permite ao projetista do jogo ter opções dentro da experiência desejada, forçando os jogadores a se arriscarem mais.

V. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

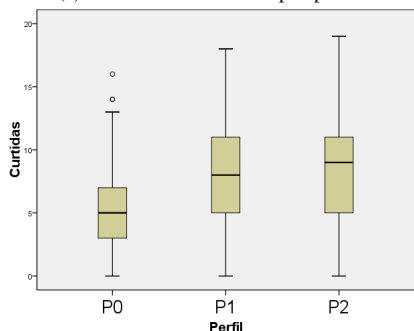
O processo de desenvolvimento de jogos modernos de tabuleiro é complexo pois seu grande atrativo são as decisões tomadas durante a partida. Para tanto, o projetista toma o cuidado para que as decisões sejam interessantes entre si e haja espaço para o jogador se adaptar dentro da partida.

Este trabalho explora o uso de técnicas de software e simulações durante a fase de projeto de um jogo moderno de tabuleiro. As regras são regidas testes de software e sua implementação conduz a os partidas simuladas por agentes para gerar testes de estresse. Estes testes são utilizados para gerar modificações e um novo ciclo de testes é realizado.

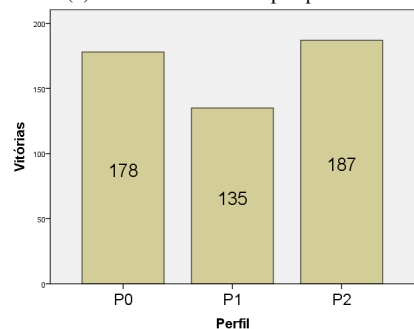
Ao realizar o projeto e balanceamento amparado por testes de software permitiu ao projetista ter um modelo do jogo simulável e tomar decisões sobre dados de partidas em paralelo com os testes de jogabilidade com pessoas reais.



(a) Cenário 3: Denúncias por perfil.

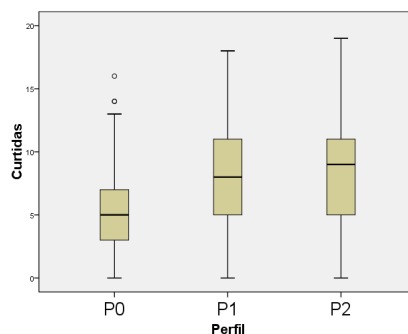


(b) Cenário 3: Curtidas por perfil.

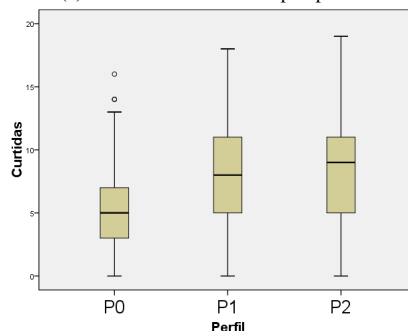


(c) Cenário 3: Vitórias por perfil.

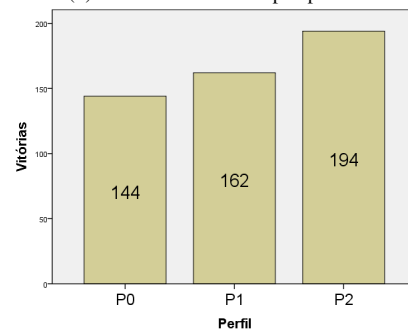
Fig. 3. Terceiro cenário com 10 Curtidas necessárias e 6 Denúncias.



(a) Cenário 4: Denúncias por perfil.



(b) Cenário 4: Curtidas por perfil.



(c) Cenário 4: Vitórias por perfil

Fig. 4. Quarto e último cenário de iteração com 10 Curtidas necessárias e 7 Denúncias.

Espera-se realizar uma abordagem mista para permitir que um agente humano interaja com os agentes bem como agentes mais complexos, aptos a se adaptarem estrategicamente durante uma série de partidas. O método ainda não permite avaliar a estética e usabilidade dos componentes e este é um ponto ainda em aberto.

Por fim, os autores agradecem a Pró-Reitoria de Pesquisa da UFJF pelo financiamento do projeto.

REFERÊNCIAS

[1] J. Huizinga, *Homo ludens: o jogo como elemento da cultura*. Editora da Universidade de S. Paulo, Editora Perspectiva, 1971, vol. 4.
 [2] J. Ehrmann, C. Lewis, and P. Lewis, "Homo ludens revisited," *Yale French Studies*, no. 41, pp. 31–57, 1968.
 [3] K. Salen, K. S. Tekinbaş, and E. Zimmerman, *Rules of play: Game design fundamentals*. MIT press, 2004.
 [4] S. Woods, *Eurogames: The design, culture and play of modern European board games*. McFarland, 2012.

[5] K. Teuber, "Die siedler von catan: Regelheft," *Kosmos Verlag, Stuttgart*, 1995.
 [6] BoardGameGeek, "Game Database," <https://boardgamegeek.com/browse/boardgame>, 2000.
 [7] R. Hunicke, M. LeBlanc, and R. Zubek, "Mda: A formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI*, vol. 4, no. 1, 2004, p. 1722.
 [8] K. Hullett, S. Kurniawan, and N. Wardrip-Fruin, "Better game studies education the carassonne way," in *DiGRA Conference*, 2009.
 [9] J. Kritz, E. Mangeli, and G. Xexéo, "Building an ontology of boardgame mechanics based on the boardgamegeek database and the mda framework," *Proceedings of SBGames*, vol. 16, 2017.
 [10] G. Engelstein and I. Shalev, *Building Blocks of Tabletop Game Design: An Encyclopedia of Mechanisms*. CRC Press, 2019.
 [11] R. K. Gupta, H. Prajapati, and H. Singh, *Test-Driven JavaScript Development*. Packt Publishing Ltd, 2015.
 [12] A. A. Ivo, E. M. Guerra, S. M. Porto, J. Choma, and M. G. Quiles, "An approach for applying test-driven development (tdd) in the development of randomized algorithms," *Journal of Software Engineering Research and Development*, vol. 6, no. 1, p. 9, 2018.