

Desenvolvendo Jogos por Meio de Linha de Produto Dinâmica

Diego Castro^{1,2}, Cláudia Werner¹

¹ Programa de Engenharia de Sistemas e Computação
COPPE (Universidade Federal do Rio de Janeiro)
Rio de Janeiro – RJ – Brazil

² Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET)
Rio de Janeiro – RJ – Brazil

{diegocbcastro,werner}@cos.ufrj.br

Resumo. *O desenvolvimento de jogos é um processo que pode envolver vários colaboradores e estágios, fazendo com que muitos títulos e atualizações demorem meses ou até anos para serem lançados. Essa demora no lançamento pode gerar ansiedade e descontentamento na comunidade de usuários, que devido a isso, acabam migrando para outros jogos ou até mesmo criando suas próprias versões dos jogos a partir do original. Algumas empresas já implementaram soluções para mitigar esse problema, permitindo ao usuário final realizar alterações no jogo por meio de kits de desenvolvimento e plataformas visuais. No entanto, essas ferramentas ainda são pouco exploradas permitindo apenas pequenas edições. Linha de Produtos de Software Dinâmica (LPSD) é extensão do conceito de linha de produto tradicional, utilizada para projetar sistemas em larga escala para auto adaptação em relação ao ambiente que está sendo executado, permitindo gerar várias versões de um sistema em tempo real a partir da combinação das características pré-produzidas de um sistema. Com base nisso, o objetivo inicial deste trabalho é utilizar o conceito de LPSD para dar mais autonomia ao usuário final, permitindo que o mesmo consiga criar variadas versões do mesmo jogo que acaba de adquirir, fazendo com que o mesmo tenha mais opções de jogos, diminuindo sua ansiedade. Um protótipo da proposta foi desenvolvido e avaliado, confirmando o vasto potencial de aplicação deste conceito.*

1. Introdução

Os jogos progrediram significativamente ao longo dos anos, tornando-se uma das formas de entretenimento mais populares da atualidade, com lucros e investimentos atingindo bilhões de dólares por ano [Lee et al. 2020]. Com esse aumento contínuo de investimentos, os games se tornaram uma vitrine do ambiente industrial, atraindo fãs de todos os gêneros, idades e gostos, tornando esta indústria ainda maior, com expectativa de arrecadar cerca de 260 bilhões de dólares em 2025 [Beattie 2020].

Apesar da quantidade de dinheiro movimentado / investido e do tamanho da comunidade de usuários, o processo atual de desenvolvimento de um jogo, desde o planejamento até o lançamento, pode ser altamente custoso, demorado e exigir diversos especialistas, fazendo com que alguns títulos ou atualizações demorem meses ou até mesmo anos para serem lançados [Bilińska et al. 2020]. Essa demora pode causar ansiedade e frustração na comunidade, contribuindo para a busca por novos jogos por

parte desses usuários [Unger 2012], o que pode acarretar na migração de usuários de um jogo para o outro ou até mesmo na criação de uma nova versão do jogo por algum jogador [Scacchi 2011]. Algumas empresas já estão utilizando estratégias para mitigar essa ansiedade, como a disponibilização de ferramentas para permitir que o jogador edite determinados elementos do jogo utilizando kits de desenvolvimento e interfaces de edição visual [Hofman-Kohlmeyer 2019, Scacchi 2011]. No entanto, esses métodos são bastante limitados, pois permitem que o usuário edite apenas mapas e imagens de personagens, sem possibilitar a edição de partes significativas dos jogos, como regras, comportamentos e mecânicas, como modificação do tipo de ataque, movimentação, opções de jogabilidade, entre outros.

Linha de Produtos de Software (LPS) é uma das abordagens propostas dentro da área de Reutilização de Software (RS), permitindo gerar várias versões de um sistema a partir da combinação de características pré-produzidas. Essas versões podem ser geradas por meio do agrupamento e edição de Pontos de Variação (PVs), que podem ser compreendidos como trechos do software que podem ser alterados [Krueger 1992]. De modo geral, quando um software possui uma série de características semelhantes e algumas novas que podem ser alterados ou adicionados, LPS, por meio de PVs, pode ser utilizada para gerar um novo produto [Krueger 1992].

Alguns softwares específicos necessitam de adaptação em tempo real, e para isso o conceito de Linha de Produtos de Software Dinâmica (LPSD) pode ser introduzido, sendo uma extensão da LPS que permite a modificação em tempo de execução de certos elementos da LPS para auto adaptação em relação ao ambiente que está sendo executado [Ayala et al. 2021].

Com isso, o presente trabalho propõe a utilização de LPS, mais especificamente LPSD, para o desenvolvimento de jogos, visando dar mais autonomia aos usuários finais permitindo-lhes criar variadas versões do mesmo jogo que acaba de adquirir através da seleção dos PVs em tempo de execução, no intuito de diminuir a ansiedade criada pela demora no lançamento de novos títulos. O uso de LPSD foi escolhido para proporcionar uma melhor interação e dinamismo para o usuário final, possibilitando que o mesmo consiga ajustar as mecânicas, regras e opções visuais do jogo em tempo real.

Para avaliar este conceito foi desenvolvida uma LPSD, disponibilizada por meio de um protótipo de um jogo de plataforma no qual o próprio usuário final pode modificar partes da linha, selecionando as características desejadas através de PVs, permitindo assim a criação de múltiplos jogos. O jogo foi validado por meio de avaliação com profissionais e estudantes da área, demonstrando o potencial do uso de LPSD para expansão de jogos.

O restante deste artigo é apresentado da seguinte forma: a Seção 2 apresenta alguns conceitos principais para o bom entendimento do artigo, a Seção 3 descreve e explica como um protótipo foi criado, como prova de conceito da solução proposta, a Seção 4 apresenta uma avaliação inicial e o que foi observado com a criação do protótipo, e a Seção 5 conclui com algumas observações, limitações e trabalhos futuros.

2. Fundamentação Teórica

Os jogos podem conter inúmeras características, onde a complexidade do mesmo pode aumentar proporcionalmente ao número de seus elementos. Portanto, devido a

complexidade de um jogo, são necessárias técnicas para organizar esses elementos. Existem inúmeras maneiras de organizar as propriedades e elementos para o desenvolvimento de um jogo e uma das mais conhecidas é a Tétrade Elementar (TE) [Schell 2008] que visa classificar as propriedades de um jogo em quatro categorias principais: [1] **Mecânica:** pode ser interpretada como as regras e as ações que podem ocorrer durante o jogo. [1.1] **Mecânica de segundo nível:** embora as mecânicas de segundo nível não façam parte da TE, elas serão adicionadas neste trabalho para descrever mecânicas que são geradas a partir da combinação de mecânicas primárias. Essa adição foi feita para trazer mais dinamismo à proposta de alteração dos jogos a serem construídos. [2] **História:** descreve o aspecto narrativo do jogo. Neste trabalho a história será modificada através da modificação direta da estética. Neste trabalho, a história é consequência da estética e da seleção das características da LPS. [3] **Estética:** são as imagens do jogo, personagens, objetos e mapas. [4] **Tecnologia:** refere-se às ferramentas e aos sistemas usados para implementar a jogabilidade. Neste trabalho esta categoria não é modificada já que a tecnologia utilizada para todos os jogos é a mesma, sempre utilizando Unity [Koulaxidis e Xinogalos 2022] para o desenvolvimento de jogos mobile.

Conforme mencionado na introdução deste trabalho, o principal problema que este estudo visa abordar é reduzir a ansiedade da comunidade de usuários de jogos, que frequentemente precisam esperar meses ou até mesmo anos pelo lançamento de um novo título. Para resolver isso, foi proposta a LPSD para o desenvolvimento de N versões do mesmo jogo. Além disso, foi realizada uma busca da literatura para descrever as lacunas no estado da arte em relação às modificações de jogos [Castro e Werner 2023].

3. Protótipo

Esta seção busca apresentar o protótipo de um jogo que visou disponibilizar uma LPS, mais especificamente, uma LPSD para jogos de plataforma, permitindo aos usuários finais editar ou adicionar novos recursos à linha de jogos disponibilizada em tempo real. O protótipo foi desenvolvido como um jogo devido à sua simplicidade e para dar mais dinamismo à solução proposta, sem a necessidade de instalação de novas ferramentas. Assim, o próprio jogo geraria novos jogos para o usuário com base nas funcionalidades escolhidas.

Como mencionado na Seção 2, os jogos podem possuir muitas características e, devido a isso, existem algumas maneiras de demonstrá-las. Nesse trabalho foi utilizado método da Tétrade Elementar, a seguir o protótipo criado é descrito de acordo com a mesma:

O protótipo proposto visa disponibilizar uma LPSD de um jogo de plataforma onde o protagonista deve completar três tarefas em cada fase. No início do jogo o jogador deve selecionar as características que deseja incorporar ao jogo, através das três árvores de características, uma para cada nível da TE (mecânica, mecânica de segundo nível e estética). A Figura 1 apresenta alguns níveis gerados pelo jogo, a partir dos quais também é possível visualizar os objetivos que foram selecionados. A Figura 2 mostra a LPSD do jogo, onde o usuário pode selecionar as funcionalidades desejadas para o jogo que quer criar. Nela, de cima para baixo, é possível ver: um ícone de chave inglesa que representa a mecânica, um haltere representando a mecânica do segundo nível e um lápis

representando a estética do jogo. Por fim, também exibe o menu de regras do jogo, que descreve como jogar.



Figura 1. Jogos gerados pela jogo.



Figura 2. Árvore de seleção de características.

O jogo funciona como uma LPSD onde o jogador pode escolher cada elemento do jogo como: sua mecânica, seus objetivos, seus inimigos e outros elementos. Em cada jogo, o jogador escolherá as características de sua preferência e a própria inteligência da linha criará um jogo com as características selecionadas. A seguir, é descrito cada um dos elementos que podem ser selecionados ou modificados pelo jogo.

Mecânicas: [1] **Drops:** na linguagem dos jogos essa palavra pode ser compreendida como o ato de um inimigo proporcionar um item ao usuário ao morrer. Portanto ao selecionar umas dessas características na LPS, o jogador poderá coletar um dos itens listados a seguir. [1.1] **Moedas:** permite o jogador coletar moedas ao matar um inimigo; [1.2] **Vidas:** permite o jogador coletar vidas ao matar determinados inimigos, que ao serem coletadas incrementam a barra de vida do jogador; [1.3] **Poderes:** permite o jogador trocar de arma ao longo do jogo. [2] **Obstáculos:** tarefas a serem superadas pelo jogador ao longo do jogo. [2.1] **Chefe:** escolha se o jogo terá um ou vários chefes (inimigos finais) ao longo da partida. [2.2] **Check Point:** obstáculo onde o jogador deve encontrar determinados itens para poder completar a fase. [3] **Objetivo:** objetivo

principal do jogo. **[3.1] Salvar cachorro:** o jogador deve resgatar o cachorro que foi preso em uma gaiola pelo vilão principal do jogo. **[3.2] Timer:** escolha se o jogo deve ser vencido antes que o tempo acabe. **[3.3] Encontrar objetos:** selecione os itens que o jogador deve encontrar para resolver um quebra-cabeças.

Mecânica de segundo nível: **[1] Mover:** escolha a mecânica de segundo nível do jogo. **[1.1] Andar:** o personagem só pode andar. **[1.2] Correr:** o personagem poderá ter mais velocidade ao longo do jogo para fugir dos inimigos. **[2] Ataque:** escolha a mecânica de ataque. **[2.1] Shoot:** escolha entre as duas possibilidades de armas, com um tiro de cada vez ou vários tiros. **[2.2] Corte:** escolha uma espada como opção de arma.

Estética: **[1] Ecosistema:** escolha o mapa do jogo, tendo quatro opções de mapa. Os inimigos e imagens do jogo mudam de acordo com o mapa (deserto e floresta) e tempo (dia e noite).

Por fim, para tornar o jogo mais dinâmico e desafiador, uma série de recursos foram adicionados a inteligência da LPSD, incluindo inimigos com níveis de vida variados, diferentes tipos de ataques, um algoritmo A* usado para fazer os inimigos perseguirem o personagem principal, contador de danos e itens que incrementam a vida do personagem quando encontrado.

4. Avaliação e Discussão

4.1. Planejamento

A avaliação buscou verificar a aprovação do desenvolvimento da solução proposta por parte da comunidade, e para isso foram utilizadas algumas questões do modelo de *Technology Acceptance Model* (TAM) [Davis 1993], que coleta dados principalmente sobre a utilidade e usabilidade da solução apresentada, permitindo aos usuários determinar se ela será útil para a comunidade. Porém, como a solução proposta foi implementada através de um jogo, o questionário MEEGA [Petri et al. 2016] também foi utilizado. Este questionário é utilizado principalmente para avaliar a usabilidade e experiência proporcionada por um jogo, porém, neste estudo, foi utilizado para determinar se a usabilidade e experiência do jogo influenciaram ou não na aceitação da solução proposta. Portanto, o questionário MEEGA foi utilizado para avaliar a usabilidade e experiência do jogo, enquanto as questões do TAM foram utilizadas para avaliar a utilidade da ferramenta a ser construída.

O estudo foi composto por duas fases: piloto e estudo final. Para o estudo piloto, apenas um aluno de graduação avaliou o jogo e respondeu o questionário a fim de identificar possíveis problemas. Três grupos de participantes foram escolhidos para o estudo final. Esta divisão de grupos foi criada de forma a dividir os diferentes níveis de experiência dos participantes, envolvendo 46 avaliadores, que se dividiram em: especialistas, que eram profissionais com experiência na área de jogos e Reutilização de Software (RS); alunos do curso de informática que já haviam cursado disciplinas de RS; e a comunidade de entusiastas de jogos que foi composta por usuários de foruns sobre jogos e pesquisadores da área.

O estudo foi realizado remotamente mediante disponibilização dos materiais necessários à execução do mesmo, sendo realizado por meio de uma chamada do *Google Meeting*, fazendo uso do protocolo *think-aloud* [Jääskeläinen 2010]. Esse

protocolo foi utilizado para coletar informações adicionais, como se o jogador estava tendo dificuldades e se gostou da solução proposta, entre outras coisas. As principais etapas deste estudo foram: **Descrição do estudo e do jogo:** distribuição de formulário contendo as principais informações necessárias para a realização da avaliação. **Caracterização dos participantes:** questionário utilizado para saber as experiências do avaliador. **Questionário qualitativo:** os questionários TAM [Davis 1993] e MEEGA [Petri et al. 2016] adaptados ao contexto foram disponibilizados a cada um dos participantes. O questionário utilizado pode ser visto com mais detalhes a seguir. **Usabilidade do jogo - MEEGA [Petri et al. 2016]:** [1] O *design* do jogo é atrativo (*sprites*, interfaces, gráficos, etc). [2] Textos, cores e fontes combinam e são consistentes. [3] Aprender a jogar este jogo foi fácil para mim. [4] O jogo é fácil de jogar. [5] As regras do jogo são claras e fáceis de entender. [6] As fontes (tamanho e estilo) utilizadas no jogo são fáceis de ler. **Experiência do jogo - MEEGA [Petri et al. 2016]:** [7] As cores usadas no jogo são atrativas e são fáceis de compreender. [8] Este jogo é apropriadamente desafiador para mim. [9] O jogo oferece novos desafios (obstáculos, situações ou variações) em um ritmo adequado. [10] O jogo não se torna monótono à medida que avança. [11] Eu recomendaria este jogo aos meus colegas. [12] Eu me diverti jogando o jogo. [13] Houve algo interessante no jogo que chamou minha atenção. [14] Eu estava tão envolvido em minha tarefa de jogo que perdi a noção do tempo. [15] Esqueci-me do ambiente ao meu redor enquanto jogava este jogo. [16] Os conteúdos do jogo são relevantes para os meus interesses. **Utilidade da ferramenta - TAM [Davis 1993]:** [17] Entendi facilmente como usar a abordagem LPS? [18] Apliquei a estratégia corretamente? Eu consegui projetar os jogos que queria jogar. [19] Entendi o que aconteceu na interação com a ferramenta? [20] Percebi como é simples criar um novo jogo usando LPS? [21] Eu usaria essa ferramenta para criar / expandir jogos se ela proposta?

4.2. Resultados

Como primeiro ponto de análise dos resultados, é importante destacar que, embora o questionário MEEGA [Petri et al. 2016] tenha sido empregado para coletar informações sobre a usabilidade e a experiência do jogo, essa informação foi obtida apenas para verificar se algum aspecto do jogo, como gráficos e cores, poderia influenciar o resultado final da avaliação. Quanto à usabilidade e experiência do jogo, foi possível observar que não teve efeito direto nos resultados do estudo; entretanto, foram identificadas algumas melhorias que poderiam ser feitas, entre elas: melhorar a movimentação do personagem usando o *joystick*, modificar o nível de dificuldade do jogo e aumentá-lo à medida que os jogos eram criados e adicionar *zoom in* e *zoom out* na tela da árvore de características.

É possível observar um número significativo de avaliadores com respostas positivas, indicando que os usuários conseguiram construir jogos baseados no conceito de LPSD e compreender como o conceito de VPs seria utilizado dentro da árvore de características. Na disponibilização de uma pergunta para respostas abertas, foi possível capturar respostas, como: A possibilidade de derivar jogos a partir da seleção de elementos em uma linha de produto, parece ser uma solução benéfica para a comunidade com aplicação no mundo real; Conceito muito interessante; Consegui criar três jogos totalmente diferentes; Conceito interessante que seria de grande benefício para a comunidade de desenvolvedores de jogos.

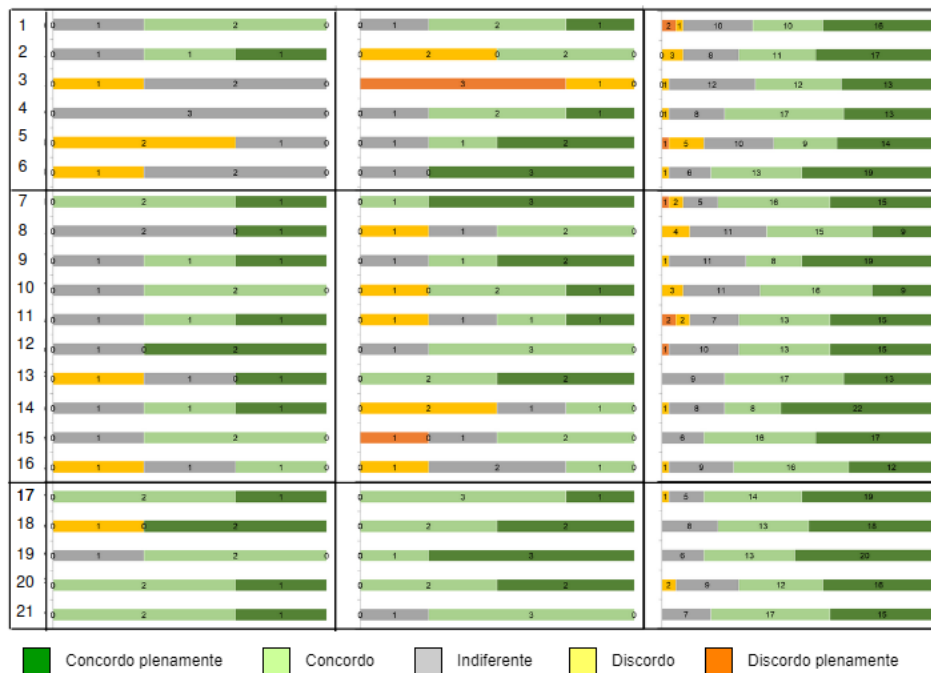


Figura 3. Respostas dos questionários Meega / TAM

Olhando para as respostas de uma forma geral, é possível perceber que mais de 70% das respostas obtiveram resultados positivos (Concordo e Concordo Plenamente) e que aproximadamente 95% dos respondentes deram respostas maiores ou iguais a Indiferente, com apenas 5% das respostas obtiveram nota negativa (Discordo e Discordo Plenamente), num total de 46 participantes. Esse percentual é ainda maior quando se consideram apenas as respostas que discutem a utilidade da ferramenta, com mais de 81% de respostas positivas, mais de 98% de respostas acima de Indiferente e menos de 2% de respostas negativas. A Figura 3 demonstra as respostas quanto à usabilidade, experiência e utilidade da solução proposta proposta. Com isso, é possível demonstrar que a solução tem potencial de utilidade e tem a possibilidade de auxiliar a comunidade de jogos na expansão de seus jogos com maior flexibilidade ao permitir a edição de múltiplos elementos do jogo.

5. Considerações Finais e Trabalhos Futuros

Este trabalho buscou utilizar o conceito de LPSD para dar mais autonomia ao usuário final, permitindo que o mesmo consiga criar variadas versões de um mesmo jogo em tempo real. Para testar a viabilidade da solução proposta, foi desenvolvido um protótipo que visa disponibilizar uma LPSD por meio de um jogo no qual o usuário final pode escolher as características que desejasse, selecionando-as na árvore de características. O protótipo foi avaliado por 46 participantes, onde foi determinado que a solução é interessante e útil para a comunidade de jogos.

A partir da avaliação foi possível perceber que a comunidade acredita no potencial da solução, abrindo a possibilidade de uma expansão da mesma para algo um pouco mais elaborado. Como trabalho futuro, pretende-se ampliar o conceito apresentado pelo protótipo para que seja possível desenvolver uma solução de desenvolvimento de jogos

a partir dos conceitos de LPS. A proposta visa o desenvolvimento de duas ferramentas, uma para ser utilizada pela própria empresa desenvolvedora do jogo, para entendimento do domínio do jogo e criação da LPS e dos PVs e outra onde o usuário final consiga editar a LPS do jogo adquirido, permitindo assim que ele consiga criar variadas versões do jogo por meio dos PVs.

Referências

- Ayala, I., Papadopoulos, A. V., Amor, M., e Fuentes, L. (2021). Prodspl: Proactive self-adaptation based on dynamic software product lines. *Journal of Systems and Software*, 175:110909.
- Beattie, A. (2020). How the video game industry is changing. <https://www.investopedia.com/articles/investing/053115/how-video-game-industry-changing.asp>. Online; accessed 29 May 2022.
- Bilińska, K., Dewalska-Opitek, A., e Hofman-Kohlmeyer, M. (2020). To mod or not to mod—an empirical study on game modding as customer value co-creation. *Sustainability*, 12(21):9014.
- Castro, D. e Werner, C. (2023). A multivocal review on derivation games. *IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, 1:144 to 149.
- Davis, F. D. (1993). User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *International journal of man-machine studies*, 38(3):475–487.
- Hofman-Kohlmeyer, M. M. (2019). Players as content creators. the benefits of game modding according to polish users. *International Scientific Journal News*, 2:8–26.
- Jääskeläinen, R. (2010). Think-aloud protocol. *Handbook of translation studies*, 1:371–374.
- Koulaxidis, G. e Xinogalos, S. (2022). Improving mobile game performance with basic optimization techniques in unity. *Modelling*, 3(2):201–223.
- Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys (CSUR)*, 24(2):131–183.
- Lee, D., Lin, D., Bezemer, C.-P., e Hassan, A. E. (2020). Building the perfect game—an empirical study of game modifications. *Empirical Software Engineering*, pages 1–34.
- Petri, G., von Wangenheim, C. G., e Borgatto, A. F. (2016). Meega+: an evolution of a model for the evaluation of educational games. *INCoD/GQS*, 3:1–40.
- Scacchi, W. (2011). Modding as an open source approach to extending computer game systems. *Proceedings of IFIP International Conference on Open Source Systems*, pages 62–74.
- Schell, J. (2008). *The Art of Game Design: A book of lenses*. CRC press.
- Unger, A. (2012). Modding as part of game culture. *Computer Games and New Media Cultures*, pages 509–523.