

Play-MVC: Desenvolvimento de um jogo de tabuleiro educacional para o ensino de arquitetura de software

Play-MVC: Development of an educational tabletop game to teach software architecture

Ágatha Alice Fabris Dias de Souza¹, Yasmim Ferreira¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Paraná
Londrina, PR – Brasil

agathalicefdsouza@gmail.com, yasmimferreira8000@gmail.com.br

Abstract. Introduction: Educational games have consolidated themselves as an efficient strategy for the teaching-learning of several notions of computing, including software architecture. **Objective:** This article introduces the development of the educational tabletop game Play-MVC to support the teaching of the software architecture pattern Model View Controller. **Steps:** A methodology inspired by Design Based Research was adopted, focused in the creational cycles, tests and updates of the game to adapt the tabletop game War to the concepts of the software architecture pattern. **Expected Results:** Thus, it is expected to amplify the validity of Play-MVC as a teaching supporting tool and make the teaching material available for the replication of instructors.

Keywords Educational game, Model View Controller, Software architecture, Gamification.

Resumo. Introdução: Os jogos educacionais têm se consolidado como uma estratégia eficaz para ensino-aprendizagem de diversos conceitos da computação, inclusive arquitetura de software. **Objetivo:** Este trabalho apresenta o desenvolvimento do jogo de tabuleiro educacional Play-MVC para apoiar o ensino do padrão de arquitetura de software Modelo Visão Controle. **Etapas:** Adotou-se uma metodologia inspirada em Design Based Research, focada nos ciclos de criação, teste e atualização do jogo para adaptar o jogo de tabuleiro War para os conceitos do padrão de arquitetura de software. **Resultados esperados:** Assim, espera-se ampliar a validade do Play-MVC como ferramenta de apoio ao ensino e disponibilizar o material para replicação por instrutores.

Palavras-Chave Jogo educacional, Model View Controller, Arquitetura de software, Gamificação.

1. Introdução

A utilização de Jogos Educacionais (JE) tem sido adotada como um método para apoiar o processo de ensino e aprendizagem dos estudantes, e pesquisas sobre a sua utilização estão sendo feitas constantemente [Santana e Oliveira 2019]. Dentro da computação, os JE têm se consolidado como uma estratégia eficaz para apresentar o conteúdo da área de Ciência da Computação de forma que encoraja bons princípios de aprendizagem [Gee 2003], pois os jogos auxiliam no processo de assimilação de informações, principalmente em ambientes escolares, além de aumentar a animação e participação dos estudantes [Santos et al. 2015].

Dentre os diferentes conteúdos apresentados na computação, a arquitetura Model-View-Controller (MVC) pode apresentar um desafio significativo no entendimento e assimilação de informações. Essa complexidade relacionada ao tema decorre principalmente do fato de que, mesmo que seja um assunto introdutório, ele também é maçante. Neste contexto, este trabalho tem como principal objetivo apresentar o desenvolvimento do jogo Play-MVC que objetiva familiarizar estudantes do Ensino Básico, Técnico e Tecnológico com conceitos do padrão arquitetural MVC a partir da representação desses conceitos em exemplos lúdicos.

A seguir, são abordados trabalhos relacionados (Seção 2), métodos (Seção 3), as definições do jogo (Seção 4), resultados parciais (Seção 5) e os resultados esperados (Seção 6).

2. Trabalhos Relacionados

Em Ludificação, os autores apresentam um jogo de tabuleiro educacional com o objetivo de ensinar e reforçar o aprendizado de Redes de Computadores por meio de uma série de perguntas e a jogabilidade funciona com base em respostas corretas e o avanço de peões [Silva et al. 2024] que não existem na jogabilidade do Play-MVC.

Em outro caminho, os autores apresentam a tradução e modificação do conhecido jogo Uno para Unno Binário e Unno Hexadecimal, que modificam os números e cores das cartas do jogo original por números binários e hexadecimais, respectivamente [Junior et al. 2025]. Assemelhando-se a Play-MVC na criação de um jogo educacional para a área de computação a partir da adaptação de um jogo já estabelecido. E se diferenciando na jogabilidade, visto que não utiliza um tabuleiro.

Por sua último, o jogo LEARN 2.0 busca melhorar o jogo original, LEARN. Ambos os jogos tem como objetivo ensinar conceitos de arquitetura de software, que inclusive abrange a arquitetura MVC, foco do jogo Play-MVC. Learn 2.0 utiliza cartas e um tabuleiro, todavia, se difere na maneira de como se joga o jogo, por se tratar de perguntas e respostas, como também um tabuleiro que consta com o avanço de peões [Lelis e Marques 2024].

3. Método

O desenvolvimento do JE Play-MVC utilizou uma metodologia iterativa inspirada em Design Based Research (DBR) [Jobim e Giraffa 2026], focada nos ciclos de criação, teste e atualização do jogo. Atualmente o desenvolvimento se situa no primeiro ciclo, na porção de atualização.

A criação do jogo foi caracterizada por uma pesquisa do funcionamento do jogo de tabuleiro War, principal inspiração de jogabilidade de Play-MVC, e o teste foi realizado com uma aplicação a jogadores voluntários e uma aplicação do JE para estudantes de nível técnico, esse teste objetivou verificar do engajamento no jogo, as decisões de design e a compreensão das regras.

4. Definições do jogo

O Play-MVC se caracteriza como um jogo de estratégia competitivo por turnos, recompensando o planejamento antecipado para completar objetivos, a percepção de

tendências de comportamentos e controle de recursos próprios e dos outros jogadores. Com forte inspiração no jogo War, Play-MVC tem como ciclo de jogabilidade a conquista de cartas no tabuleiro e a sua junção para a conclusão de objetivos.

4.1. Componentes

Play-MVC tem a quantidade mínima de 2 jogadores e quantidade máxima de 10 jogadores e é composto por: um tabuleiro, um conjunto de 10 cartas de objetivo, três conjuntos de 30 cartas “M”, “V” e “C”, botões coloridos com cores diferentes que identificam cada jogador, papel e lápis com finalidade de controlar a quantidade de “programadores” que cada jogador possui, as ações de jogador e pontuação total. As cartas de objetivo são cartas com requisitos específicos que devem ser atingidos pelo jogador e que forçam a disputa por cartas MVC específicas.

Cartas MVC é o conjunto das cartas: Model, que possuem componentes como lógica de negócios, bancos de dados e linguagens de programação; View, que possuem componentes de interface de usuário, juntamente de um número representando a quantidades de "telas", que fazem parte de alguns objetivos; e Controller, que representam uma conexão. A separação entre os tipos de cartas funciona como uma forma de isolamento entre os componentes, para que os jogadores os entendam como as porções diferentes de um sistema que utiliza a arquitetura MVC. Ainda, cada carta tem uma representação visual do que ela equivale em um sistema real, reforçando essa separação.

4.2. Rodadas e turnos

Ao iniciar o jogo, cada jogador recebe: 1 carta de objetivo, 3 cartas MVC e 3 programadores. Não é possível colocar uma carta “C” sem o jogador ter ao menos uma carta “M” ou “V” já conquistada, portanto, caso um participante retire apenas cartas “C”, ele pode descartá-las e comprar novas cartas até ter ao menos uma carta “M” ou “V”.

Play-MVC é organizado por meio de rodadas, cada jogador tem seu turno, em que decide entre comprar uma nova carta MVC ou “contratar” cinco programadores. A compra de cartas ocorre no mesmo turno, mas a contratação de programadores só é efetivada no próximo turno, quando o jogador recebe os programadores. Como ação de turno opcional, o jogador pode distribuir seus programadores como quiser para conquistar cartas e pode colocar 1 carta MVC no tabuleiro, sendo que esta ficará disponível até se tornar completa. Uma carta se torna completa quando foi conquistada por um jogador e passa-se a vez de todos os outros jogadores sem que a carta seja interferida.

Ao conquistar uma carta de cada tipo, um jogador pode arranjá-las no seu turno para formar uma conexão, chamada de “conexão MVC”, como ocorre na Figura 1a. Isso ensina fisicamente a conexão, dependência e o fluxo de dados entre os componentes de um sistema que adota a arquitetura MVC, que é reforçado constantemente durante o jogo.

4.3. Conquista de cartas e vitória

Cartas MVC podem estar em dois estados: disponível ou completa. Cartas completas não podem ser alteradas e ficam na horizontal, enquanto as cartas disponíveis alteram seu estado conforme os jogadores fazem suas ações e ficam na vertical, como na Figura 1b.

As cartas MVC disponíveis podem variar entre os estados: neutra, em que não há nenhum programador nela e portanto não pertence a nenhum jogador; conquistada, em

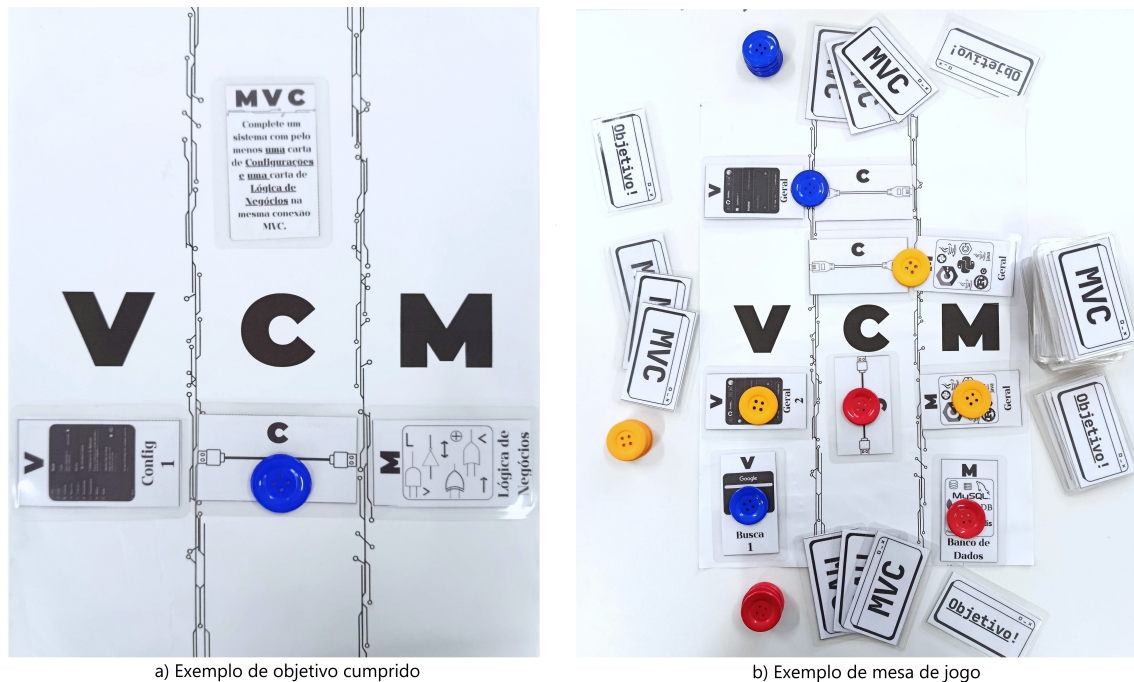


Figura 1. Fotos do jogo de tabuleiro Play-MVC em duas situações de jogo

que há programadores de um jogador; e interferida que é a carta que altera seu estado por ter se tornado neutra ou conquistada por um jogador diferente do anterior.

As cartas são conquistadas utilizando subtração e soma de programadores. Assim, para neutralizar uma carta que esteja sendo conquistada com 3 programadores, é necessário que um jogador diferente do que está conquistando utilize também três programadores ($3 - 3 = 0$), agora, para conquistar essa mesma carta neutralizada é necessário que outro jogador use qualquer quantidade de programadores.

É definido como vencedor o jogador que: possui no mínimo uma conexão “MVC”, completou dois ou mais objetivos de uma carta de objetivo e tem a maior quantidade de pontos dentre os jogadores que também completaram os requisitos anteriores. A Tabela 1 demonstra a quantidade de pontos para cada ação dos jogadores, essas ações devem ser anotadas ao longo do jogo para apurar os pontos totais de um jogador.

Tabela 1. Equivalência de pontos por ação de jogador do Play-MVC

Ação do jogador	Pontos
Completar uma carta de objetivo	2500
Colocar uma carta MVC no tabuleiro	300
Completar uma conexão MVC	500
Roubar uma carta	200
Neutralizar uma carta	200

5. Resultados parciais

Quanto à criação, a utilização do jogo War como base diminuiu a carga cognitiva no entendimento das regras do jogo e também auxiliou a criar um ciclo interativo familiar e

funcional, utilizando decisões já testadas de design competitivo. No entanto, o design de Play-MVC enfrentou desafios próprios para demonstrar fisicamente as conexões e o fluxo de dados em um sistema real que utiliza a arquitetura MVC e eliminar parte do elemento de sorte, valorizando a ação estratégica dos jogadores.

Durante os testes, foi observado que o jogo manteve os participantes entretidos e suas regras têm fácil compreensão, contudo, os resultados foram inconclusivos para o aprendizado do padrão arquitetural MVC, visto que não adotou métodos sistemáticos para medir os conhecimentos prévios e posteriores à aplicação do JE.

Ainda, observou-se que o controle de programadores no papel pode ter impactado negativamente a experiência de jogo e que a elaboração de um manual de consulta poderia ser benéfico para eventuais dúvidas dos jogadores. Dessa maneira, são necessárias atualizações e mais aplicações do JE, com o intuito de testar a validade para o ensino e aprendizagem da arquitetura MVC e melhoria de regras e materiais.

6. Resultados esperados

Pretende-se concluir o ciclo atual e reiniciá-lo, com o intuito de melhorar o JE para que ele seja mais entendível e engajante aos jogadores e possa ser disponibilizado através de repositório online para instrutores e educadores.

Na porção de atualização, é esperada a inclusão de algum tipo de peça para representar os programadores e a produção de materiais complementares, como manuais de jogo, manuais de impressão, fichas guia e exemplos.

E na porção de teste do próximo ciclo são esperadas aplicações com metodologia avaliativa sistemática com a finalidade de validar o JE para familiarização e ensino-aprendizagem de conceitos do padrão de arquitetura MVC e não apenas jogabilidade, gerando conclusões mais confiáveis da utilização da ferramenta em contexto de ensino.

Referências

- Gee, J. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1:20.
- Jobim, A. e Giraffa, L. (2026). Pesquisa baseada em design (dbr): reposicionando estudos interdisciplinares para uma perspectiva metodológica integrada. *Educação e Pesquisa*, 52(00):e291162.
- Junior, M. C., Track, M., Assumpção, M., e Gama, G. (2025). Unno binário e unno hexadecimal: jogos educativos para o ensino de computação. In *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 1366–1376, Porto Alegre, RS, Brasil. SBC.
- Lelis, M. e Marques, A. (2024). Learn 2.0: Evolução de um jogo de tabuleiro para o ensino de arquitetura de software. In *Anais do XXIII Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 1397–1408, Porto Alegre, RS, Brasil. SBC.
- Santana, S. e Oliveira, W. (2019). Jogos educacionais como ferramenta de auxílio ao processo de alfabetização. In *Anais do XXV Workshop de Informática na Escola*, pages 148–157, Porto Alegre, RS, Brasil. SBC.

- Santos, W., Neto, S. R. S., Junior, C. S., e Bittencourt, I. (2015). Avaliação de jogos educativos: Uma abordagem no ensino de matemática. In *Anais do XXVI Simpósio Brasileiro de Informática na Educação*, pages 657–666, Porto Alegre, RS, Brasil. SBC.
- Silva, C., Rodrigues, F., Junior, R. V., e Alcantara, A. (2024). Ludificação: Um jogo híbrido para o ensino e aprendizagem da disciplina de rede de computadores em cursos de ciência da computação. In *Anais Estendidos do XXIII Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 147–152, Porto Alegre, RS, Brasil. SBC.