

DESENVOLVIMENTO DE UM COMPONENTE GERADOR DE DESAFIOS PARA UM JOGO SÉRIO NA MATEMÁTICA DISCRETA ¹

Vinícius Tomé Vieira², Adilson Vahldick³, Paolo Moser⁴.

¹ Vinculado ao projeto “Jogo Sério na Matemática Discreta”

² Acadêmico (a) do Curso de Engenharia de Software – CEAVI – Bolsista PROIP/UDESC

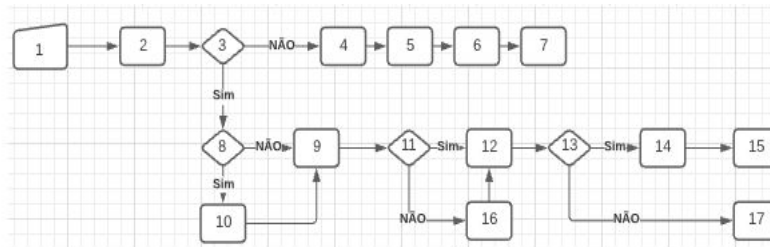
³ Orientador, Departamento de Engenharia de Software – CEAVI – adilson.vahldick@udesc.br

⁴ Orientador, Departamento de Engenharia de Software – CEAVI – paolo.moser@udesc.br

O objetivo do componente foi fazer com que as fases (desafios) do jogo sério fossem construídas de maneira rápida, eficiente e otimizada. Tendo isso em vista, o sistema desenvolvido dentro do jogo sério, através de códigos na linguagem *Blueprints* do motor de jogos *Unreal Engine*, fez com que o jogo ficasse mais leve, dinâmico e com uma fácil manutenibilidade. Um dos motivos da criação do sistema foi o fato de que os arquivos gerados pelo motor de jogos *Unreal Engine* se mostravam pesados, em questão de processamento, e quando carregados na memória acabavam fazendo com que o jogo se tornasse lento, o que apresentava uma experiência desagradável para o jogador. Porém, o principal motivo foi em relação à aumentar a produtividade e facilitar a manutenção. Com esse componente, é possível especificar os problemas através de parametrização e ao executar o jogo, o componente cria as instâncias dos desafios.

Primeiramente, foi feita uma refatoração no projeto, que anteriormente possuía um arquivo para cada desafio, para que todas as fases fossem centralizadas em poucos arquivos, utilizando práticas da POO (Programação Orientação a Objetos). Em seguida, foram criados quatro modelos, cada um correspondente a um tipo de desafio no jogo, que acontecem com todos os personagens. Os desafios são: (Tipo 1) clicar e arrastar itens para caixas formando combinações; (Tipo 2) apresenta uma fórmula parcialmente resolvida que o jogador deve resolver; (Tipo 3) apresenta uma fórmula para ser resolvida completamente; (Tipo 4) o aluno deve somente colocar a resposta do desafio. Caso não fosse aplicado, seriam necessários cerca de 40 *scripts em Blueprint* para fazer os desafios do jogo. Com o componente, são necessários somente 4 *scripts*, um para cada tipo de desafio. Ao longo do projeto foram feitos testes e mudanças, conforme a necessidade do jogo ao longo do tempo, chegando finalmente em um modelo definitivo. O diagrama do gerador de desafios pode ser visto na Figura 1.

A validação aconteceu em alguns momentos durante o projeto, como por exemplo na migração do projeto para dispositivos móveis, em que o componente funcionou em ambas as plataformas (computador e celular - Figura 2). O jogo contava com 10 personagens com 4 desafios cada: dois para o Tipo 1, um para o Tipo 3 e outro para o Tipo 4. Durante o projeto, a validação apresentou a necessidade de um novo tipo de desafio (Tipo 2). Foi adicionada essa funcionalidade no componente. A alteração no jogo foi realizada sem afetar o código já existente, demonstrando as características de flexibilidade e manutenibilidade do componente. Outro membro da equipe ficou responsável por desenvolver esses desafios e, por isso, ele era o usuário do componente permitindo que mais de um programador estivesse envolvido nos testes.



1	Input. Fase.	10	Remove a janela correspondente à fase seguinte.
2	Gera uma variável interna para controle da fase.	11	Na fase atual do vetor de janelas, existe uma janela criada?
3	Na fase atual há uma janela criada?	12	Libera a visualização da tela correspondente a fase atual para o usuário.
4	Carrega dinamicamente a referência de classe com base na fase atual.	13	Há uma fase anterior a atual já criada?
5	Cria uma nova janela com base nas informações contidas dentro da referência de classe.	14	Remove a janela correspondente à fase anterior.
6	Cria a navegação básica dentro da janela na instância do gerenciador (controle) atual.	15	Retorna a janela correspondente ao vetor de janelas no índice da fase atual.
7	Adiciona uma variável temporária na janela criada.	16	Adiciona no vetor de janelas a janela criada.
8	A próxima fase já foi criada?	17	Retorna a variável temporária com a janela criada
9	Adiciona uma variável temporária na janela correspondente ao índice do vetor de janelas na fase atual.		

Figura 1. Diagrama de fluxo do gerador de desafios

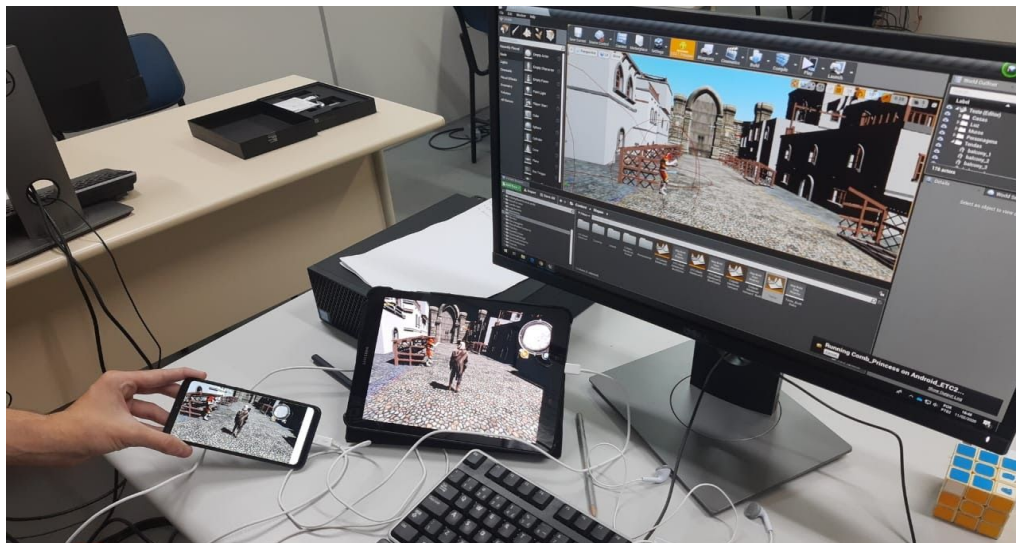


Figura 2.. Imagem do jogo em processo de adaptação para dispositivos móveis.

Palavras-chave: Jogos sérios. Matemática. Programação.