

Show do Milhão

Paola Campos da Silva, Kaio Campos Tadeu, João Pedro Huppes Arenales

*Universidade Federal de Mato Grosso do Sul – Campo Grande – Mato Grosso do Sul –
Brasil*

`paola.campos@ufms.br, c.kaio@ufms.br, joao_huppes@ufms.br`

1 Introdução

O "Show do Milhão" trata-se de um programa de televisão, tendo sua era de ouro apresentada por Silvio Santos, no qual um jogo de perguntas e respostas é realizado e o maior prêmio oferecido é de um milhão (daí o nome).

Dito isso, aqui será destrinchado o desenvolvimento do código, implementado pelos participantes do presente grupo referente ao "Show do Milhão".

2 Ambiente de Desenvolvimento

Para a criação do trabalho, o editor de código escolhido foi o *Visual Studio Code*, por sua facilidade de utilização e aparência de simples compreensão, e, alguns compiladores *online* (*Programiz C++ Compiler* e *GDB Online Compiler* também para C++), para testes imediatos à parte como funções e impressões.

As extensões do *VS Code* utilizadas foram: *C/C++* para *Visual Studio Code* para programar com essa linguagem, e *Live Share* para que todos os participantes pudessem codar simultaneamente. Ambas são disponibilizadas pela Microsoft.

3 Implementações

3.1 Ideais Iniciais

Seguindo as instruções oferecidas pelo professor, a ideia inicial era apenas a confecção do jogo imitando o funcionamento do real programa de televisão.

3.1.1 Funcionalidades Não Implementadas

Durante a execução do projeto, os membros do grupo tiveram algumas ideias, mas que, para evitar complicações com a ideia principal, foram descartadas.

Um desses objetivos que não foi alcançado foi a possibilidade de adicionar um *timer* à ficha do jogador. Dessa forma, essa nova informação (quanto tempo o usuário demorou para terminar sua tentativa) poderia ser utilizada como um novo critério de desempate no *ranking*.

3.2 Problemas Enfrentados

Inicialmente, utilizar a extensão *Live Share* parecia interessante, mas, frequentemente, o que era desenvolvido por um membro, era apagado por outro já que ela não suportava atualizações simultâneas.

Continuamente, foram enfrentados problemas na verificação de resposta: o código sempre lia que a alternativa inserida estava errada. A preocupação principal foi o risco da resposta nem estar sendo lida, mas após algumas horas de análise, a situação foi resolvida.

E, finalmente, foi especialmente desafiador desenvolver os algoritmos com a lógica necessária para as ajudas presentes no jogo; e o trabalho em equipe como um todo, já que, muitas vezes, os alunos não concordavam entre si, apesar de isso tornar o projeto menos estressante do que normalmente seria.

3.3 Projeto Final

Visando facilitar o entendimento, a explicação detalhada do código será dada em partes.

3.3.1 Interação com o Jogador

Para criar um ambiente divertido, cada ação realizada na progressão do jogo (como troca de pergunta, avanço de bloco, etc.) é comunicada de forma lúdica ao usuário. Continuamente, buscando trazer a nostalgia do verdadeiro "Show do Milhão", surgiu a ideia de criar um *leaderboard* semelhante aos vistos em jogos de *arcade* (local onde máquinas de jogo ficavam) de antigamente.

3.3.2 Aleatorização das Ajudas

Neste tópico será explicado como a *randomização* utilizada durante o jogo foi feita.

Temos que, grande parte das funções de ajuda tem como base a aleatorização de valores "sentinelas". Ou seja, a implementação dos métodos de aleatorização de valores por meio da função *rand()* foi de extrema importância para simular a "chance" de algo acontecer.

Assim, foi utilizado o intervalo dos dez primeiros números inteiros positivos (1 a 10) para simular porcentagens de um evento ocorrer. Por exemplo, ao solicitar ajuda aos universitários, caso números do intervalo de um a sete (1 a 7) fossem sorteados, o universitário escolheria a alternativa correta, enquanto valores superiores (8, 9, 10), o fariam retornar uma resposta aleatória (evitando a certa). A mesma lógica foi utilizada para quando fosse solicitada a ajuda da plateia, apenas os valores foram alterados.

Além disso, há um segundo método chamado "embaralhar" utilizado quando o usuário solicita ajuda das cartas. Nesse caso, foram usados os números aleatorizados, em um intervalo de zero a três (0 a 3), para embaralhar as cartas que podem ser escolhidas

pelo jogador. Logo, é impossível deduzir em qual ordem elas ficaram.

Em suma, é possível notar como fatores que demandam "sorte" são cruciais para vencer o jogo, sem descartar que, após alguns testes, foi possível notar que nem sempre o que é previsto matematicamente realmente acontece.

3.3.3 Funcionalidades

Em busca de tornar a implementação mais fácil, praticamente todo o código foi subdividido em funções, sendo estas:

Função	Descrição
main()	Aqui ocorre as chamadas das demais funções e a manipulação dos arquivos
menu()	Tela de apoio ao usuário, focado nas escolhas gerais
menu_pergunta()	Exibe a pergunta, as alternativas e as ajudas disponíveis
embaralha_cartas()	Referente a ajuda das cartas, onde elas são embaralhadas para melhor aleatorização
mostre_carta()	As cartas são mostradas, com uma breve explicação de seu significado (com a exceção do Rei de Copas, que fica presente na função <i>ajuda_cartas()</i>)
jogo()	O jogo de fato, onde o status do usuário é continuamente atualizado e avisado
ajuda()	Manipulação de cada tipo de ajuda oferecida
ajuda_plateia()	Aleatorização das escolhas da plateia
ajuda_universitarios()	Aleatorização das escolhas dos universitários
ajuda_cartas()	Interação com o usuário referente à ajuda das cartas e a sua aleatorização
verifique_alternativa_ajuda()	Verificação da entrada do usuário (se será uma ajuda ou uma alternativa)
itere_ajuda()	Tratamento da quantidade, uso e, se necessário, <i>loop</i> das ajudas
pergunte()	Aleatorização das perguntas e verificação para evitar a repetição destas, além de averiguar a veracidade da resposta
edite_leaderboard()	Onde o <i>score</i> das tentativas é feito e salvo num arquivo binário (.dat)
mostre_leaderboard()	Exposição dos <i>scores</i> salvos em formato de pódio

4 Considerações Finais

Esse trabalho foi uma oportunidade de explorar e aplicar os conhecimentos adquiridos no decorrer da matéria de **Algoritmos e Programação 2** de uma forma única, promovendo o aprendizado e a recriação dos estudantes de maneira simples, mas educativa.

Por conseguinte, sua confecção também incentiva e aprimora as habilidades necessárias para o trabalho em equipe, as chamadas *soft skills*, que atualmente são tão essenciais no mercado de trabalho, além de promover um desafio para os estudantes por ser um projeto relativamente extenso.