

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Lucas Carvalho Batista Canhadas Genvigir

PROJETO 1: JOGO DA VELHA

CORNÉLIO PROCÓPIO

2023

INTRODUÇÃO

Este documento explica as principais funcionalidades de um jogo (Jogo da Velha), desenvolvido com a plataforma Kodular. O desenvolvimento deste aplicativo utilizou o conteúdo da disciplina Desenvolvimento sem Código, ministrada pelo Professor Matheus Mollon.

Todo o material de aula, disponível no Moodle da disciplina, foi utilizado como base de consulta. Parte de alguns blocos foram adaptados para o jogo.

1 DESENVOLVIMENTO DO APP DO JOGO DA VELHA

Após a inicialização da tela do jogo (*quando o bloco Screen 1 for inicializado*¹), os blocos *Habilitar Botões*² e *Inicializar Botões*³ são chamados na sequência.

O bloco da Figura 1 é *passa_a_vez*. Este bloco é usado para passar a vez entre os jogadores, usando a comparação de texto da variável *global_vez*. Se ela for igual a X alterna para a vez do O, caso contrário, é a vez do X jogar. Há incluso nesta função uma chamada para a função *troca_de_cor*, como pode ser visto na Figura 2. A função *troca_de_cor*, alterna a cor do nome do jogador, e a o bloco descrito na Figura 4, alterna a cor do botão, da seguinte forma: sempre que é a vez do jogador 1 realizar uma jogada, tanto seu nome (função *troca_de_cor*) quanto o X (bloco da Figura 4) fica na cor azul, e sempre que o jogador 2 for realizar uma jogada, seu nome e o O fica na cor vermelho. O bloco da Figura 4 representa apenas o botão 1, mas há 9 blocos semelhantes, um para cada botão do tabuleiro de jogo.

Também foi pensado, durante o desenvolvimento do jogo, em criar uma função para que, quando o jogo terminasse por meio de empate ou vitória, as cores dos nomes retornassem a preto, deixando apenas a mensagem indicando o vencedor colorida. Desta forma, criou-se a variável *fim_do_jogo*, que pode ser vista na Figura 2, que é inicializada em *false*, para que, quando chegar o fim do jogo ela se torne *true*, fazendo com que os nomes fiquem na cor preta. Esta função é mostrada na Figura 3.

¹ O bloco Screen 1 não está neste texto, deve ser visualizado no Kodular.

² O bloco Habilitar Botões não está neste texto, deve ser visualizado no Kodular.

³ O bloco Inicializar Botões não está neste texto, deve ser visualizado no Kodular.



Figura 1 – passa_a_vez Fonte: Autoria própria

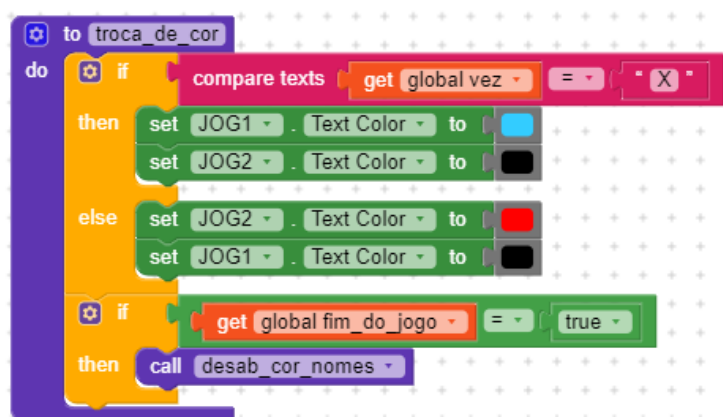


Figura 2 – troca_de_cor Fonte: Autoria própria

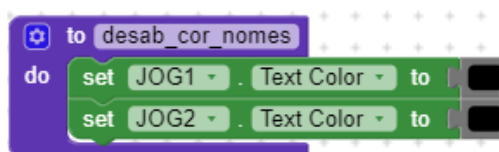


Figura 3 – desb_cor_nomes Fonte: Autoria própria

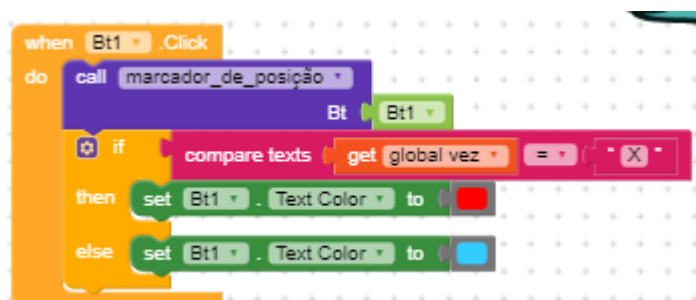


Figura 4 – When Bt1 click (chama marcador_de_posição) Fonte: Autoria própria

O bloco *marcador_de_posição*, na Figura 5, foi criado para jogadores não poderem jogar em casas já ocupadas. Ele checa se o botão está com algum texto, se já estiver ele mostra uma notificação, falando que a posição já está ocupada, dando a oportunidade de o jogador fazer uma jogada válida, se não tiver ele permite a jogada. Depois ele chama a função *verifica_velha*, como mostra a Figura 6, que tem como base analisar os botões na vertical, horizontal e diagonal, para ver se o jogo acabou. Havendo um vencedor ele executa o processo *verificar_vencedor*, mostrado na Figura

7. O próximo passo é chamar a função *desabilitar_botões*, esta chamada pode ser vista na Figura 6, desabilitando os botões do tabuleiro de jogo. Caso contrário, ele irá verificar todos os botões novamente, final do bloco na Figura 6, para assim afirmar que houve um empate, exibindo uma mensagem na tela.

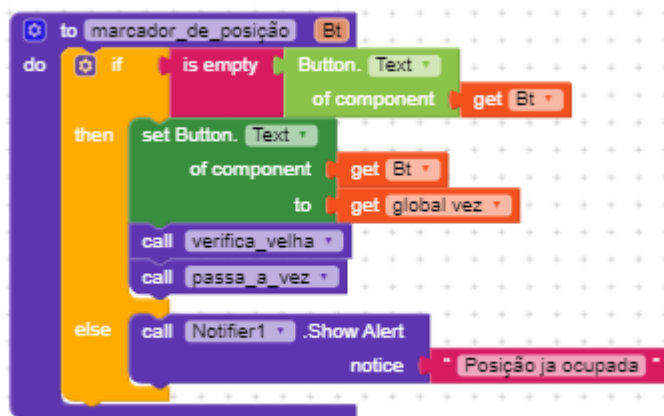


Figura 5 – marcador_de_posição Fonte: Autoria própria

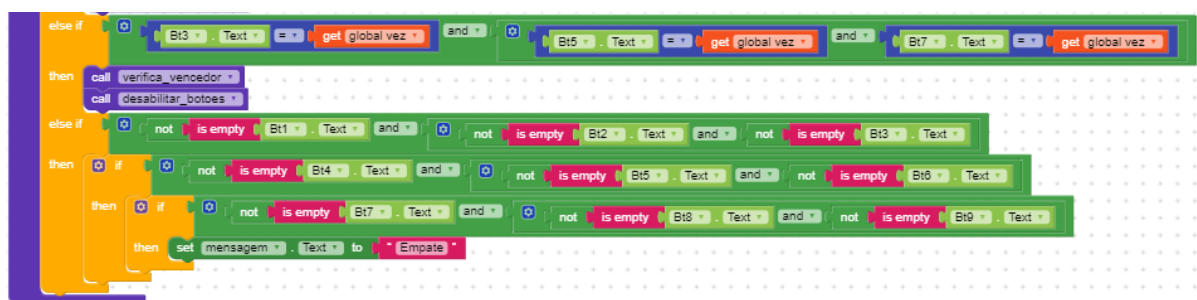


Figura 6 – Parte final do Bloco verifica_velha – Fonte: Autoria própria

A função *verifica_vencedor*, da Figura 7, funciona em conjunto com a função *verifica_velha*, da Figura 6⁴. Ela utiliza como base a informação de qual jogador foi o vencedor, analisando o conteúdo da variável *global_vez*, verificando com uma estrutura de seleção: se for igual a X ele exibe a mensagem de vencedor é com o nome atribuído ao jogador 1 e faz a incrementação do ponto para o jogador 1 no placar, e armazena no banco de dados, para que quando o botão reset for clicado, não apague estes valores. O mesmo processo é feito caso o jogador 2 seja o vencedor, isto é variável *global_vez* não é igual a X. Quando o jogo acaba altera o valor da variável *fim_do_jogo* para *true*, desabilitando as cores dos nomes, como explicado anteriormente.

⁴ A Figura 6 mostra apenas a parte final do bloco *verifica_velha*. O bloco completo, que compara todas as possibilidades de existir um vencedor, pode ser visto no Kodular.

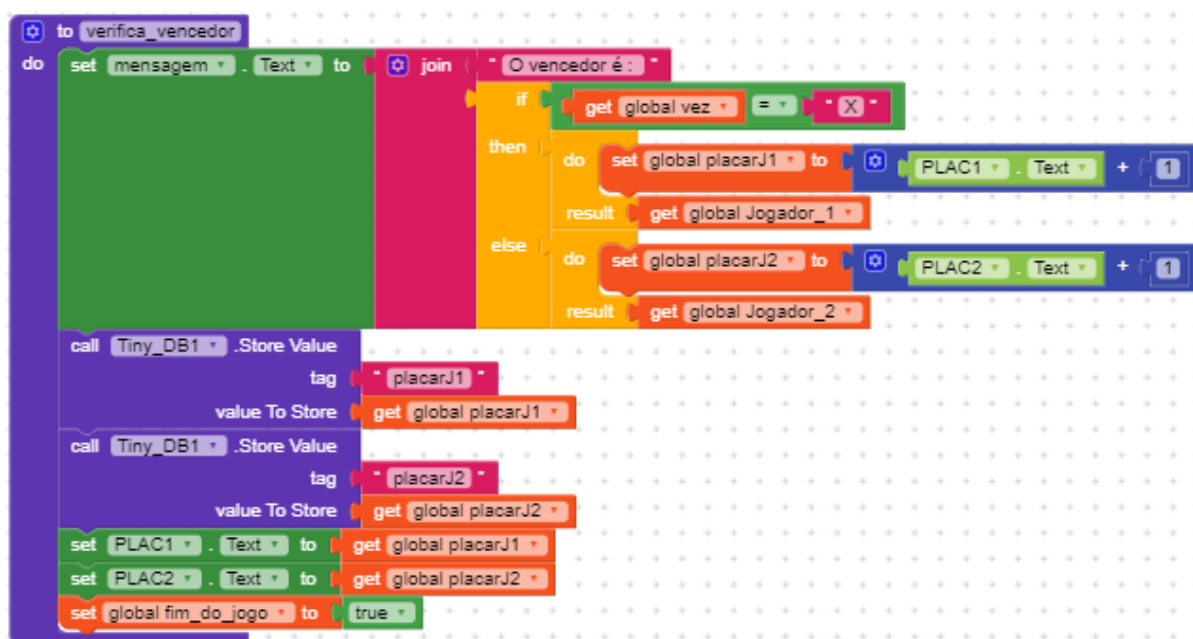


Figura 7 – verifica_vencedor – Fonte: Adaptado de material de aula

No bloco *verifica_vencedor*, Figura 7, a chamada da função *Tiny_DB1.Store Value* irá armazenar o valor que está na variável *global placarJ1*, nomeando-a com o tag *placarJ1*. Isso é executado na sequência para a variável *global placarJ2*. Logo após, o comando *set* insere na caixa de texto *PLAC1* o conteúdo de *global placarJ1*, fazendo o mesmo para *PLAC2* e *placarJ2*, na sequência.

O bloco *Bt_Reset*⁵ limpa o banco de dados e volta para a tela de cadastro, e o bloco *Bt_Restart*⁶ limpa o tabuleiro, habilita os botões para um novo jogo. Altera também o valor da variável *global fim_do_jogo* para *false*, indicando que não há um vencedor, como explicado anteriormente, e chama a função *troca_de_cor*.

Os jogadores podem optar por continuar com diversas partidas, somando as suas vitórias no placar ou reiniciando um novo jogo.

⁵ O bloco *Bt_Reset* não está neste texto, deve ser visualizado no Kodular.

⁶ O bloco *Bt_Restart* não está neste texto, deve ser visualizado no Kodular.

REFERÊNCIAS

Os materiais de consulta e apoio utilizados foram as transparências disponíveis no Moodle da disciplina.