Jogo Threes!

Deve ser implementado o jogo Threes!, cujo objetivo é agrupar ao máximo os números que se encontram na matriz. Os valores numéricos são adendos e múltiplos de 3. O agrupamento ocorre juntando os números 1 e 2 ou números iguais.

É aconselhado que o aluno familiarize-se com os jogos existentes para melhor compreender seu funcionamento. O jogo pode ser experimentado em: http://play.threesgame.com/

Descrição do jogo

No jogo Threes!, o jogador deve tentar agrupar ao máximo os números que vão sendo gerados em um tabuleiro 4x4. Os valores numéricos são adendos e múltiplos de 3 (exceto os números 1 e 2). O agrupamento ocorre juntando os números 1 e 2 (jogada básica para formar o 3) e números iguais. Os valores iniciais devem ser gerados aleatoriamente e controlados para que as primeiras jogadas sejam possíveis.

Funcionamento do jogo

O jogo possui as seguintes características:

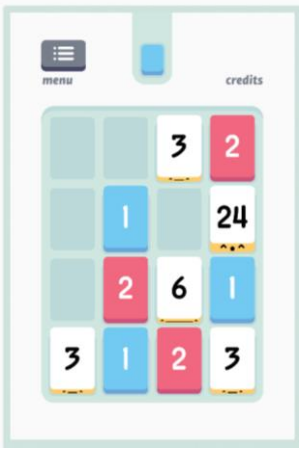
- Movimentação do número para cima, baixo, esquerda e direita,utilizando o teclado e/ou mouse;
- O jogador deve juntar os números 1 e 2 no início da jogada;
- O próximo número para juntar deve ser o mesmo ou o 1 com o 2;
- O resultado é a soma dos valores conectados;
- O jogo termina quando não for mais possível juntar dois números.

O programa a implementar

Você deve fazer um programa que implemente um **jogo com as características descritas acima e com as funcionalidades listadas a seguir**:

Tabuleiro:

O jogo original considera um tabuleiro 4x4, conforme exibe a figura a seguir.



Entretanto, no jogo a ser implementado, o **tabuleiro deve ser de 5x5**. Além disso, o tabuleiro a ser gerado deve conter uma casa (escolhida aleatoriamente) bloqueada (barreira). Neste sentido, as movimentações dentro do tabuleiro devem obedecer esta restrição. A cada execução do jogo, deve ser escolhida aleatoriamente a posição da casa bloqueada.

			3	2
		1		24
	6	2		1
	3		6	2
	3	1	2	3

Movimentações no tabuleiro do jogo:

O jogador deve ter a opção de utilizar o mouse ou o teclado (teclas de setas) para a movimentação dentro do tabuleiro. Cada movimentação ocasiona:

- a soma de todas as casas vizinhas que obedecerem à regra do jogo (1 somado com 2, demais casas somadas com mesmo valor, desde que alcancem o limite - borda ou casa ocupada);
- o reposicionamento de todas as casas uma posição adiante da atual no sentido do movimento (exceto quando alcançar o fim do tabuleiro ou a barreira);
- a inserção de um novo valor (gerado aleatoriamente) na borda oposta ao sentido do movimento, em posição aleatória.

Observe que nem toda movimentação implicará na soma de 2 valores , já que o tabuleiro pode conter lacunas e/ou casas que não podem ser somadas.

Valores:

Todos os valores devem ser adendos e múltiplos de 3 (exceto os números 1 e 2), sendo gerados de forma aleatória e cada valor deve ter uma cor específica. No jogo original são utilizadas as cores azul (número 1), rosa (número 2) e branco (demais números). Utilize bom senso na escolha das cores, de modo a não interferir no contraste visual do jogo.

O jogo deve controlar a aleatoriedade dos valores gerados, de modo que, por exemplo, no início do jogo, não sejam gerados valores altos (os quais não poderão ser usados em somas tão cedo). Da mesma forma, em uma partida em estágio avançado (com somas de valores altas), o número de valores baixos gerados deve ser controlado.

A cada movimentação é possível que ocorra mais de um agrupamento. Quando isso ocorrer, os pontos da jogada devem ser multiplicados por 2.

Informações na tela:

No topo da tela deve ser exibido o **próximo valor a ser gerado**.

Durante toda a execução do jogo, a **pontuação** do jogador deve ser exibida na tela do jogo. O cálculo da pontuação é obtido através da soma dos valores "juntados" (obs.: no jogo original a pontuação é gerada de forma diferente).

Além do próximo valor e da pontuação, a tela do jogo deve exibir ainda um **timer**, indicando há quanto tempo o jogador está jogando aquela "partida".

Botão Pause:

Um botão com a função de pausar o jogo deve estar disponível. O botão trará um menu de contexto com um botão de resumo da partida e outro para reiniciar a mesma.

Fim de jogo:

O jogo termina quando nenhuma jogada for possível. Neste caso, o jogador deve ser avisado.

Help:

O jogo deve possuir uma opção de ajuda, a qual explica ao usuário como jogar.

Recordes:

Deve ser implementado o armazenamento e a recuperação das 10 maiores pontuações anteriores, bem como a posição da pontuação do jogo atual nessa lista. Estas informações devem ser exibidas, em ordem decrescente, ao final de cada jogo. Cada pontuação deve estar acompanhada do nome do jogador, informação que deve ser solicitada no início do jogo (obs.: no jogo original não existe esta funcionalidade).

No caso de pontuações com valores empatados, preserva-se a ordem de alcance da pontuação, ou seja, o primeiro jogador que obteve a pontuação deve ocupar a posição superior.

As pontuações nomes dos jogadores devem ser mantidas em um arquivo.

Informações Complementares:

Além das informações anteriores, que descrevem o programa a ser implementado, deve-se considerar:

- O jogo deve ser implementado de forma modular, ou seja, com o uso de funções.
- Deve ser usada passagem de parâmetros (no lugar de variáveis globais).
- Não devem ser inseridos comentários no código.
- O jogo deve estar devidamente identado.
- O uso de bitmaps, sons,… não agregam valor ao jogo, a menos que todas as demais funcionalidades solicitadas tiverem sido atendidas.
- Trabalho individual.
- Data de entrega: **30/11/19 (o prazo não será estendido!!!)**
    - Não serão aceitos trabalhos fora do prazo de entrega.
- Cada trabalho deverá apresentado, em horário a combinar com a professora. Trabalhos não apresentados serão desconsiderados.