

## NÍVEL AVANÇADO



PROJETO 15

```
(CONTEÚDO DISPONÍVEL) {  
TETRIS;  
(end);  
})();
```

#PORTFÓLIOBOOSTPROGRAM

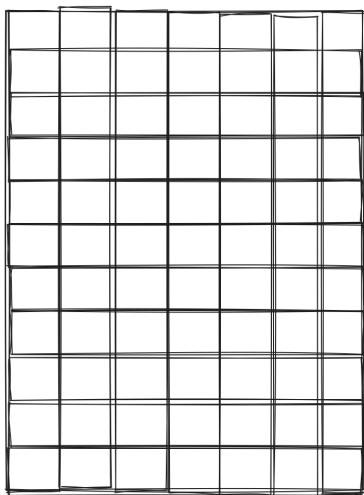
**CONHECIMENTOS REQUIRIDOS:**



**FRONT-END**

# WIREFRAME

Tetris



Pontuação

300

próxima peça



Novo Jogo

# TETRIS

Crie um jogo semelhante ao tetris que exija que alguém gire as formas para que se encaixem como um quebra-cabeça.

## TECH STACK

- ➔ React
- ➔ React-hooks
- ➔ TailwindCSS
- ➔ ViteJS
- ➔ Vercel (conhecimento básico de serverless)



## BRIEFING

Tetris é um jogo de quebra-cabeça que parece nunca sair de moda. Independentemente do tema, em sua essência, o tetris é composto por formas de blocos que você precisa usar para completar as linhas para evitar transbordar o espaço disponível.

Esse projeto prático é um excelente momento para você se tornar expert em animações e transições em CSS.



## NÍVEL 1

O principal controle do jogador em Tetris é que você pode mover um único objeto para a esquerda e para a direita conforme ele cai no chão. Isso ajuda a alinhar os blocos à medida que eles se empilham.

Adicione um único objeto composto de blocos que cai naturalmente com a capacidade de movê-lo para a esquerda e para a direita. Uma vez que o objeto cai, outro objeto deve cair depois.

## NÍVEL 2

Um controle adicional que você tem como jogador é a capacidade de girar cada forma 90 graus por vez. Adicione a capacidade de girar a forma conforme ela cai 90 graus.

## NÍVEL 3

O objetivo ao posicionar as formas é tentar completar uma linha. Isso remove essa linha da área de jogo, o que ajuda a evitar que os blocos transbordem. Remova uma linha da área de jogo depois de concluída.





## REQUISITOS DETALHADOS

### Crie uma área de jogo.

- ➔ A área de jogo é a área onde o jogo é jogado. Ela pode ser criada usando um **elemento div** e definindo seu estilo para ter um tamanho e uma posição específicos.
- ➔ Por exemplo, o seguinte código criará uma **área de jogo** com 1000 pixels de largura e 500 pixels de altura, localizada no centro da tela:

```
<div class="game-area" style="width: 1000px; height: 500px; position: center;"></div>
```

### Adicione blocos.

- ➔ Os **blocos** são as peças que são usadas para **construir o jogo**. Eles podem ser criados usando **elementos div** e definindo seu estilo para ter uma cor e um tamanho específicos.
- ➔ Por exemplo, o seguinte código criará quatro blocos, cada um com 20 pixels de largura e 20 pixels de altura, e os definirá como vermelho, verde, azul e amarelo:

```
<div class="block" style="width: 20px; height: 20px; background-color: red;"></div>  
<div class="block" style="width: 20px; height: 20px; background-color: green;"></div>  
<div class="block" style="width: 20px; height: 20px; background-color: blue;"></div>  
<div class="block" style="width: 20px; height: 20px; background-color: yellow;"></div>
```

## Adicione formas de bloco.



As formas de bloco são as diferentes formas que os blocos podem assumir. Existem **7 formas de bloco** diferentes no jogo Tetris: **I, J, L, O, S, T e Z**. Se você nunca jogou Tetris, veja esta página:

# TETRIS



Cada forma pode ser criada usando uma **matriz de blocos** e definindo o estilo de cada bloco para ter uma cor e um tamanho específicos. Por exemplo, o seguinte código criará a forma I:

```
const iShape = [
  <div class="block" style="width: 20px; height:
  20px; background-color: red;"></div>,
  <div class="block" style="width: 20px; height:
  20px; background-color: red;"></div>,
  <div class="block" style="width: 20px; height:
  20px; background-color: red;"></div>,
  <div class="block" style="width: 20px; height:
  20px; background-color: red;"></div>
```

### Adicione forma de queda.

- ➡ A forma de queda é a forma de bloco que o jogador controla. Ela começa no topo da área de jogo e cai para baixo a cada segundo.
- ➡ O jogador pode mover a forma de queda para a **esquerda**, para a **direita** e **girá-la** para evitar que ela bata no fundo da área de jogo.

### Adicione empilhamento de formas.

- ➡ Quando a forma de queda atingir o fundo da área de jogo, ela será empilhada em cima dos blocos existentes. Se uma linha da área de jogo estiver cheia de blocos, ela será **removida** e o jogador ganhará pontos.

### Adicione movimento de forma.

- ➡ O jogador pode mover a forma de queda para a esquerda, para a direita e girá-la. O jogador pode **mover a forma de queda** pressionando as teclas **esquerda**, **direita** e **cima** no teclado.
- ➡ O jogador pode girar a forma de queda pressionando a tecla **espaço** no teclado.

### Adicione rotação de forma.

- ➡ O jogador pode girar a forma de queda para evitar que ela bata no fundo da área de jogo. O jogador pode girar a forma de queda pressionando a tecla **espaço** no teclado.

### Adicione finalização de forma.

- ➡ Quando uma linha da área de jogo estiver cheia de blocos, ela **será removida e o jogador ganhará pontos**. O jogador ganhará 100 pontos por cada linha removida.
- ➡ Adicione **Game Over** quando estiver cheio. Quando a área de jogo estiver **cheia de blocos**, o jogo terminará e o jogador perderá.
- ➡ Boa parte do desenvolvimento desse app será dado a partir do **event.onKeyDown** e de **hooks**, então segue aqui uma colinha para você:
- ➡ segue a lista de todos os eventos na **DOM**, nessa lista você tem os **keyboard events** também:

## EVENT



- ➡ **ArrowLeft**: Move a forma de queda para a esquerda.  
keyCode = 37
- ➡ **ArrowRight**: Move a forma de queda para a direita.  
keyCode = 39
- ➡ **ArrowDown**: Move a forma de queda para baixo.  
keyCode = 40

➡ **Spacebar:** Gira a forma de queda.

Aqui estão alguns dos hooks React que podem ser usados no jogo Tetris:

➡ **useState:** É usado para gerenciar o estado do jogo, como a posição dos blocos, a pontuação do jogador e o estado do jogo.

➡ **useEffect:** É usado para executar código quando um componente é montado ou desmontado.

➡ **useRef:** É usado para criar referências a elementos do DOM.

➡ **useContext:** É usado para compartilhar dados entre componentes.

➡ Caso seja necessário **DYOR (Do your own Research):**

**DYOR**

