Aluno: João Paulo Veloso
Matrícula: 1351812549
Disciplinas UCs:
Estrutura de dados e análise de algoritmos
Modelos, métodos e técnicas da engenharia de software

Documentação do Projeto Snake

1.Introdução

O projeto é um jogo clássico da cobrinha desenvolvido como parte de um trabalho acadêmico. Utiliza a linguagem TypeScript e a biblioteca React para criar uma experiência interativa e envolvente.
O jogo consiste em uma cobra que se move em um tabuleiro, coletando comida para crescer e ganhar pontos. O objetivo é evitar colisões com as bordas do tabuleiro ou com o próprio corpo da cobra.

2.Pré-requisitos

2.1 Linguagem e Bibliotecas:

- O código está escrito em JavaScript/TypeScript.
- Utiliza a biblioteca React para a construção da interface do usuário.
- Usa a biblioteca react-router-dom para navegação

2.2 Estilo e Layout:

- O layout do jogo é estilizado com CSS. O estilo está sendo importado a partir de um arquivo chamado App.css.
- A estrutura do jogo é dividida em células (representadas pelo componente Cell), formando um tabuleiro.

2.3 Controles do Jogo:

- A direção da cobra é controlada pelas teclas de seta (Arrow keys).
- O jogo responde aos eventos de tecla usando o hook useEffect para adicionar e remover um ouvinte de evento de tecla.

2.4 Lógica do Jogo:

- O tabuleiro do jogo tem dimensões de 20x20 células.
- A cobra é representada como uma série de segmentos, e a comida é gerada aleatoriamente no tabuleiro.
- A pontuação é incrementada quando a cobra come a comida.
- O jogo termina se a cobra colidir com as bordas do tabuleiro ou consigo mesma.

- Existe um sistema de aumento de velocidade da cobra a cada 5 pontos e exibe uma notificação usando a biblioteca react-hot-toast.

2.5 Notificações:
- O componente usa a biblioteca react-hot-toast para exibir notificações/toasts.
- Um toast é exibido quando a velocidade da cobra aumenta.

2.6 Persistência de Estado:
- O estado do jogo (posição da cobra, posição da comida, direção, etc.) é mantido usando o hook useState.
- O estado do jogo é reiniciado quando o botão "Restart" é clicado após o jogo terminar.

2.7 Navegação:
- Usa o hook useNavigate para navegar para uma página inicial quando o jogo é reiniciado.

2.8 Responsividade:
- Não há evidências diretas no código fornecido, mas considerando que é um jogo simples, a responsividade pode não ser o principal foco.

2.9 Referências e Efeitos Colaterais:
- Usa refs (useRef) para manter referências aos estados atuais dentro dos efeitos (useEffect).
- Usa efeitos (useEffect) para lidar com a entrada do teclado, mover a cobra no tabuleiro e manipular o reinício do jogo.

2.10 Dependências:
- Certifique-se de que as dependências do projeto, como React e react-router-dom, estão instaladas corretamente.

3.Instalação

```
git clone https://github.com/joaopauloev/Projeto-A2.git
cd Projeto-A2
npm install
```

4.Configuração

Estrutura do Código

O código é estruturado em componentes e utiliza hooks do React para gerenciar o estado e os efeitos colaterais. A seguir, são destacados os principais elementos do código:

Componentes

Cell:

- Componente otimizado usando React.memo para renderizar cada célula do tabuleiro.
- Aceita as propriedades isFood e isSnake para determinar se a célula é comida ou parte da cobra.

Snake:

- Componente principal do jogo.
- Gerência estados como posição da cobra, posição da comida, direção, estado de jogo (se acabou ou não), pontuação e velocidade da cobra.
- Utiliza a biblioteca react-hot-toast para notificações visuais.
- Contém efeitos para lidar com a entrada do teclado, movimento da cobra e atualização da pontuação.

Funções Auxiliares

getRandomFood:

- Função para gerar uma posição aleatória para a comida, garantindo que não coincida com a posição da cobra.

Estados

Posição da Cobra (snake):

- Array de objetos representando os segmentos da cobra no tabuleiro.

Posição da Comida (food):

- Objeto representando a posição da comida no tabuleiro.

Direção (dir):

- String representando a direção atual da cobra.

Fim de Jogo (GameOver):

- Booleano indicando se o jogo terminou.

Pontuação (Score):

- Número representando a pontuação do jogador.

Velocidade da Cobra (snakeSpeed):

- Número representando a velocidade de movimento da cobra.

Efeitos

Entrada do Teclado:

- Efeito para capturar as teclas pressionadas e atualizar a direção da cobra.

Movimento da Cobra:

- Efeito para mover a cobra no tabuleiro.
- Atualiza a posição da cobra de acordo com a direção e verifica colisões.
- Aumenta a pontuação, tamanho da cobra e velocidade conforme a *comida é consumida.*

5.Fluxo do Jogo

Inicialização:

- O jogo inicia com a cobra em uma posição específica, direção inicial, e a comida em uma posição aleatória.

Movimento:

- A cobra se move automaticamente na direção definida pelo jogador.

Colisões:

- O jogo verifica colisões com as bordas do tabuleiro ou com o próprio corpo da cobra.

Comer a Comida:

- Se a cobra alcançar a comida, seu tamanho aumenta, a comida é reposicionada, e a pontuação é incrementada.

Fim de Jogo:

- O jogo termina se a cobra colidir com as bordas ou consigo mesma.
- Exibe a pontuação final e permite reiniciar o jogo.

6.Conclusão

O projeto Snake oferece uma implementação interativa e divertida do clássico jogo da cobrinha. Utiliza conceitos avançados do React, como hooks e efeitos, para criar uma experiência de jogo dinâmica. O código é estruturado de maneira modular e fácil de entender, tornando-o adequado para fins educacionais e de aprendizado.