

Urso e Abelhas

Um Jogo casual de coleta



Orientador: Prof. BEATRIZ CAMPOS SANTANA

Autor(es):

Lucas de faria sodré

Walid abdel hadi nadi

Conceitos do Jogo

- Jogo 2D com elementos interativos simples.
- O jogador controla as (abelhas) para coletar mel nas flores.
- O mel é depositado em um pote. Quando cheio, o urso acorda para comer.
- O ciclo se repete até que o urso hiberne.

Mecânicas principais

- Clique em flores para enviar a abelha.
- Cada flor desaparece após o uso.
- 2 flores enchem 1 pote.
- O urso acorda, consome o mel e volta a dormir.
- Após 3 refeições, ele hiberna e o jogo termina.

linguagem & addons utilizados

- Linguagem: Java
- Interface gráfica: Swing
- Concepção de Threads: Thread para o urso
- Controle de concorrência: Semaphore no pote
- Imagens otimizadas para UI

Bibliotecas Utilizadas

`javax.swing.*` Utilizada para construir a interface gráfica (botões, painéis, janelas).

`java.awt.*` Responsável por componentes gráficos mais avançados, como posicionamento e fontes.

`java.util.concurrent.Semaphore` Usada para controlar o acesso ao recurso compartilhado (pote de mel).

`java.awt.event.*` Permite capturar eventos como cliques de botão.

Implementação das Threads

Uma **thread principal** controla a interface e o fluxo do jogo.

- A **ação da abelha** é controlada dentro de um botão que simula uma thread executando um processo demorado (tempo para buscar mel).

Se não tivesse threads, o que aconteceria?
Funciona normal.

Mas **sem animação**, o mel sobe **instantaneamente**

“Isso não seria o mesmo que a função ‘esperar terminar e depois fazer’?”.

Sim! Mas a diferença é que com thread, essa espera não para o resto do jogo.

Uso de Semáforos

- No jogo, o pote de mel é esse recurso compartilhado:
 - - A abelha quer colocar mel,
 - O urso quer consumir o mel.
 -
- Sem semáforo:
- Se a abelha tentar colocar mel enquanto o urso estiver comendo, pode dar erro, travamento ou comportamento inesperado.

