**SPECIFICA 1: GHOST**

per procedere con la **Spec. 1 del Ghost**, analizziamo i requisiti e strutturiamo i passi:

**Obiettivi della Spec. 1: Ghost AI**

1. Implementare **Blinky**, un fantasma controllato da AI con due modalità di movimento:
   * **Chase Mode**: Insegue Pac-Man utilizzando un algoritmo di pathfinding (es. A\*).
   * **Frightened Mode**: Attivata quando Pac-Man mangia una Power Pill. Blinky fugge e può essere "mangiato" da Pac-Man.
2. Se il fantasma viene mangiato in modalità "Frightened":
   * Respawna nella posizione iniziale dopo 3 secondi.
3. La velocità del fantasma aumenta con il progredire del gioco.

**Strategia di Implementazione**

1. **Dichiarare e inizializzare il fantasma (Blinky):**
   * Creare una struttura dati per tracciare:
     + Posizione corrente.
     + Direzione.
     + Modalità attuale (**Chase** o **Frightened**).
   * Inizializzare la posizione di partenza del fantasma nella "central spawning point" del labirinto.
2. **Pathfinding e movimento:**
   * Implementare l'algoritmo A\* per calcolare il percorso ottimale verso Pac-Man in modalità **Chase**.
   * In modalità **Frightened**, calcolare il percorso verso un punto lontano da Pac-Man.
3. **Collisioni e stato del gioco:**
   * Se il fantasma tocca Pac-Man in modalità **Chase**, Pac-Man perde una vita.
   * Se Pac-Man tocca il fantasma in modalità **Frightened**, il fantasma viene "mangiato".
4. **Respawn e gestione del tempo:**
   * Usare un timer per gestire i 3 secondi di respawn.
5. **Disegno e animazioni:**
   * Aggiungere una funzione per disegnare il fantasma sul display.

**Step per il Movimento di Blinky**

1. **Aggiungere una funzione ghost\_update in Ghost.c:**
   * Calcola la prossima posizione di Blinky.
   * Usa una logica semplice per il movimento iniziale (es. scegli una direzione valida casuale o muovi in linea retta).
2. **Aggiornare la posizione di Blinky:**
   * Implementare un controllo per evitare che Blinky attraversi muri.
   * Integrare il movimento nel ciclo principale (ad esempio, chiamando ghost\_update periodicamente).
3. **Implementare il cambio di direzione:**
   * Aggiungere logica per cambiare direzione quando Blinky incontra un ostacolo.
   * Usa una funzione per verificare le direzioni disponibili.
4. **Gestire il disegno durante il movimento:**
   * Prima di muovere Blinky, chiama ghost\_clear per cancellare la posizione precedente.
   * Dopo aver aggiornato la posizione, chiama ghost\_draw per ridisegnarlo.
5. **Integrare il movimento nel gioco:**
   * Collegare ghost\_update al timer periodico (es. RIT\_IRQHandler o un altro timer).
6. **Testare il movimento:**
   * Compila ed esegui per verificare che Blinky si muova correttamente senza attraversare muri.
7. **Ottimizzare e aggiungere comportamenti:**
   * Aggiungere la logica per inseguire Pac-Man (modalità **CHASE**).
   * Aggiungere la logica per scappare da Pac-Man (modalità **FRIGHTENED**).