



Descripción de tareas a realizar

Se espera que la solución propuesta mejore el código del proyecto contemplando las diferentes direcciones de movimiento de los personajes y contenido del tablero. Para ello, se debe:

a) Desarrollar tipos enumerativos para representar las direcciones de los personajes (Pac-Man y fantasmas) y el **contenido** de cada celda del tablero.

Contenido de cada celda del tablero:

- WALL: Pared
- FOOD: Comida
- EMPTY: Vacio

Direcciones de movimiento de los personajes en el tablero: cuando Pac-Man o los fantasmas se mueven en el tablero, se actualizan sus posiciones sumando o restando un valor delta (o, 1, -1) a sus coordenadas actuales.



cambiar la coordenada x	Delta Y: -1
Mover hacia ABAJO	Delta X: 0
Incrementa la coordenada y del tablero sin cambiar la coordenada x	Delta Y: 1
	Delta X: -1
	Delta Y: 0
Mover hacia la DERECHA Incrementa la coordenada x del tablero sin cambiar la coordenada y .	Delta X: 1
	Delta Y: 0

Para mejorar la solución, incorpore de manera adecuada los tipos enumerativos desarrollados. Por ejemplo, usar el enumerativo Direcciones en los métodos: randomMove(), outOfRange() y setMoveUp().

- b) Eliminar el uso del operador "instanceof" en la clase Character que determina el tipo de personaje (fantasma o Pac-Man). Aproveche la jerarquia de clases, haciendo uso de la herencia y el polimorfismo. Identifique las clases de dicha jerarquia que son abstractas y modifique la definición de dichas clases.
- c) Incorporar al juego las siguientes estrategias de movimiento, contemplando el valor de la variable mustChangeDirection, que indica si el fantasma ha chocado una pared. Las estrategias a implementar son:
 - Al chocar una pared, el fantasma cambia de dirección a una aleatoria.

Facultad de Informática - UNLP

Cursada 2024

- Al chocar una pared, el fantasma cambia de dirección hacia la opuesta.
- El fantasma cambia su dirección de manera aleatoria, sin considerar el choque o no con una pared (Movimiento Zombie).

Las estrategias deben coultar su implementación al resto de las clases y ser enchufables, usando: interfaces, clases anidadas y el patrón singleton. La interface utilizada por las estrategias debe definir la siguiente firma del método:

- void changeDirection(Ghost ghost, boolean mustChangeDirection) //cambia la orientación del fantasma en el tablero

🧠 Temas a evaluar

- Especificadores de acceso
- Interface
- Clases abstractas
- Herencia y polimorfismo
- Clases anidadas e internas
- Tipos enumerativos