

Refuerzo Semanas 7 y 8

Actividad de refuerzo

A lo largo de la cursada, hemos trabajado en una primera aproximación a la codificación de un juego de Pacman. Hemos llegado a la última clase práctica habiendo implementado los siguientes diagramas de clase:

Mapa
<<Atributos de clase>> posiciones: list[] <<Atributos de instancia>> posiciones: list[] mapa: list
<<Constructores>> Mapa()
<<Comandos>> <<Consultas>> obtenerPosiciones(): list[] esBloqueNavegable(pos: list[2]): boolean sonPosicionesAdyacentes(pos1: list[2], pos2: list[2]): boolean

Pacman
<<Atributos de clase>> posicionInicial: list[2] <<Atributos de instancia>> posicion: list[2] mapa: Mapa
<<Constructores>> Pacman(mapa: Mapa)
<<Comandos>> mover(posicionDestino: list[2]) establecerPosicion(posicion: list[2]) establecerMapa(mapa: Mapa) <<Consultas>> puedeMover(posicionDestino: list[2]) obtenerPosicion(): list[2] obtenerMapa(): Mapa

Fantasma
<<Atributos de clase>> posicionInicial: list[2] <<Atributos de instancia>> posicion: list[2] mapa: Mapa color: str
<<Constructores>> Fantasma(mapa: Mapa, color: str)
<<Comandos>> mover(posicionDestino: list[2]) comer(pacman: Pacman) establecerPosicion(posicion: list[2]) establecerMapa(mapa: Mapa) establecerColor(color: str) <<Consultas>> puedeMover(posicionDestino: list[2]) obtenerPosicion(): list[2] obtenerMapa(): Mapa obtenerColor(): str

Dichas implementaciones pueden encontrarse en la siguiente carpeta en el campus:

<https://campus.uner.edu.ar/fcad/mod/folder/view.php?id=39206>

Actividad

Hasta aquí las posiciones en el mapa las representamos como una lista o arreglo de longitud 2 (list[2]). Dicho esto, deberá realizar la implementación de la clase presentada a continuación:

Posicion
<<Atributos de clase>> <<Atributos de instancia>> posicionX: int posicionY: int
<<Constructores>> Posicion(posX: int, posY: int)
<<Comandos>> establecerPosX(posX: int) establecerPosY(posY: int) <<Consultas>> obtenerPosicionX(): int obtenerPosicionY(): int esAdyacente(pos: Posicion): boolean

Las posiciones en el mapa de los objetos de las clases Pacman y Fantasma estarán representadas por objetos de la clase Posición, y no como listas de longitud 2:

- El atributo de instancia posicionX tomará el lugar y propósito del primer elemento de las listas que solían representar la posición de los objetos de las clases Pacman y Fantasma.
- El atributo de instancia posicionY tomará el lugar y propósito del segundo elemento de las listas que solían representar la posición de los objetos de las clases Pacman y Fantasma.
- El cálculo de adyacencia de dos bloques podrá realizarse enviando los mensajes **esAdyacente** a los objetos de tipo Posición, en lugar de hacerlo a través de la consulta **sonPosicionesAdyacentes** de la clase Mapa.
- Deberá actualizar, además, el resto de las clases como se indica en los diagramas debajo:

Mapa
<<Atributos de clase>> posiciones: list[] <<Atributos de instancia>> posiciones: list[] mapa: list
<<Constructores>> Mapa()
<<Comandos>> <<Consultas>> obtenerPosiciones(): list[] esBloqueNavegable(pos: Posicion): boolean

Pacman
<<Atributos de clase>> posicionInicial: Posicion <<Atributos de instancia>> posicion: Posicion mapa: Mapa
<<Constructores>> Pacman(mapa: Mapa)
<<Comandos>> mover(posicionDestino: Posicion) establecerPosicion(posicion: Posicion) establecerMapa(mapa: Mapa) <<Consultas>> puedeMover(posicionDestino: Posicion) obtenerPosicion(): Posicion obtenerMapa(): Mapa

Fantasma
<<Atributos de clase>> posicionInicial: Posicion <<Atributos de instancia>> posicion: Posicion mapa: Mapa color: str
<<Constructores>> Fantasma(mapa: Mapa, color: str)
<<Comandos>> mover(posicionDestino: Posicion) comer(pacman: Pacman) establecerPosicion(posicion: Posicion) establecerMapa(mapa: Mapa) establecerColor(color: str) <<Consultas>> puedeMover(posicionDestino: Posicion) obtenerPosicion(): Posicion obtenerMapa(): Mapa obtenerColor(): str