

Parte Práctica: Pacman

En ISW-Games siguen incursionando en la creación de juegos y quieren hacer un prototipo simplificado del **Pacman** para ver cuánto costaría hacerlo con **TDD** y si tendría sentido empezar a comercializarlo.

Para interfacear con el juego, ya existe un objeto que se encarga de recibir una colección de strings (a partir de ahora formato “string”) como representación de un tablero y verificar si es válido. Además permite visitarlo para que puedan obtener toda la información necesaria de este y crear la representación del tablero que consideren necesaria para el modelo que se les pide construir. Este es un ejemplo de cómo se especifica el tablero con este objeto, utilizando la colección de strings:

```
'#####'  
'*-----*'  
'#-B==C-#'  
'#--p---#'  
'*---==-*'  
'#####'
```

Este tablero tiene 6 filas y 8 columnas. La posición 1@1 (x@y) es la que se encuentra en la primera fila (arriba) primera columna (izquierda). Cada letra tiene el siguiente significado:

#: Representa una pared

*: Representa una Big Pill (píldora grande)

-: Representa una Small Pill (píldora chica)

B: Representa al fantasma (ghost) llamado Blinky

C: Representa al fantasma (ghost) llamado Clyde

p: Representa al Pacman

=: Representa un espacio (que queda después de que el pacman come píldoras)

Como es un prototipo del juego, solo se implementará con 2 fantasmas en vez de 4. La clase **StringPacmanBoard** se encarga de validar que la representación string del tablero sea correcta. Provee el mensaje *#visitElementsWith: aStringPacmanBoardVisitor* que permite visitar todos los elementos del tablero con un visitor que recibirá los mensajes *#visitWallAt: aPosition*, *#visitSmallPillAt: aPosition*, *#visitBigPillAt: aPosition*, etc. (ver implementación de *#visit: anElement at: position with: aPacmanBoardVisitor*) lo que les permitirá crear el tablero del juego que ustedes consideren necesario y conveniente hacer.

LOS STRINGS NO SON EL TABLERO, SON UNA REPRESENTACIÓN DEL TABLERO. Tienen por objetivo facilitar la escritura de los tests, tanto para ingresarlos, como para chequear que el juego haya evolucionado correctamente.

Se espera que el juego del **Pacman** genere una representación string del tablero luego de estar jugando para poder verificar en los tests que todo anduvo según lo esperado. Por ejemplo:

test01PacmanStartsMovingLeft

```
| board game |

board := StringPacmanBoard representedAs:(
'#####'
'#-BC-#'
'#-p--#'
'#####').
"Crea el juego y hace que avance un solo tick"
game := ...
self
    assert: game stringRepresentationOfBoard
    equals: #(
'#####'
'#-BC-#'
'#p--#'
'#####').
```

StringPacmanBoard tiene un conjunto de tests que verifican su funcionamiento. El test *#test10BoardTranslationExample* falla para que vean un ejemplo de cómo funciona la traducción de la representación String a un Dictionary. Para ello hagan lo necesario para hacerlo pasar (ver comentario del test).

El juego del **Pacman** debe permitir las siguientes **acciones**:

- Que el juego avance. Para ello debe haber un mensaje, por ejemplo *#tick*, que le indique al juego que debe mover los personajes según las reglas correspondientes.
- Indicarle al juego que el personaje Pacman debe empezar a moverse a la izquierda, derecha, arriba o abajo.

Las **reglas** que el juego debe implementar en este prototipo son:

- Se empieza con 0 puntos y 3 vidas disponibles.
- El Pacman empieza moviéndose hacia la izquierda.
- El Pacman se mueve de a un casillero dependiendo de hacia donde está yendo.
 - Si está yendo a **izquierda** se mueve con un desplazamiento de **-1@0**
 - Si se mueve a **derecha** su desplazamiento será **1@0**
 - Si se mueve para **arriba** su desplazamiento es **0@-1**
 - Si se mueve para **abajo** su desplazamiento es **0@1**.

ACLARACIÓN: El movimiento del Pacman **NO ES** como el del MarsRover. En el MarsRover se puede ir hacia adelante, atrás, girar a izquierda o derecha. En este juego se debe indicar hacia dónde debe ir directamente el pacman, de manera absoluta y no relativa.

- Si el Pacman come una Small Pill, suma un punto y deja un espacio en el lugar que estaba la Small Pill.
- Si el Pacman come una Big Pill, suma dos puntos y deja un espacio en el lugar que estaba la Big Pill.

- Si el Pacman quiere moverse a un casillero donde hay una pared, no puede. Se quedará en el casillero que está hasta que se le indique que empiece a moverse en otra dirección. No se debe informar ninguna excepción/error ya que es un comportamiento esperado.
- Los fantasmas se mueven de acuerdo a un desplazamiento al igual que el Pacman.
 - Blinky empieza moviéndose a la izquierda
 - Clyde empieza moviéndose a la derecha.
- Los fantasmas empiezan a moverse luego de cierta cantidad de ticks, a saber:
 - Blinky empieza a moverse a partir del 4to tick.
 - Clyde empieza a moverse a partir del 5to tick.
- Cuando un fantasma pasa por un casillero, debe dejar lo que había en él al salir del casillero. En el casillero inicial debe dejar un espacio cuando sale del mismo.
- Cuando un fantasma llega a una pared se debe decidir de manera random el próximo desplazamiento teniendo en cuenta los siguientes valores:
 - 1: Sigue usando el movimiento que tenía (o sea, no se movería porque está golpeando la pared).
 - 2: Empieza a mover el fantasma hacia la izquierda.
 - 3: Empieza a mover el fantasma hacia arriba.
 - 4: Empieza a mover el fantasma hacia abajo.
 - 5: Empieza a mover el fantasma hacia la derecha.
- Cuando el fantasma “come” al Pacman, este pierde una vida, vuelve a su posición inicial para empezar a moverse a izquierda, los fantasmas quedan donde estaban moviéndose cuando les toque como se estaban moviendo, y se mantienen las píldoras que fueron comidas. El contador de ticks debe volver a 0 para que luego de 4 ticks se empiecen a mover los fantasmas.
 Cuando el pacman vuelve a su posición inicial al ser comido, se debe asumir que no habrá un fantasma en esa posición.
- El juego se acaba y no se puede seguir jugando si el pacman llegó a 0 vidas o si no hay más píldoras para comer.

Debido a que es un prototipo, no es necesario implementar lo siguiente:

- Que el Pacman pueda comer fantasmas.
- La existencia de la fruta.
- Que se puede pasar de un lado a otro del tablero.
- Un fantasma se choque con otro fantasma. O sea, no se tiene que implementar que dos fantasmas puedan estar en el mismo casillero.
- Que se pase de nivel cuando el pacman come todas las píldoras.

Tips:

- Para obtener un número random se debe utilizar el mensaje `#nextIntInteger`: que está implementado en **Random**
- Recordar que la font default de Cuis no es proporcional, puede suceder que la representación string del tablero no se vea bien alineada. Si quieren usar fonts mono-espaciadas vayan al menú del desktop, elijan “Preferences > Set System Font ... > DejaVu > DejaVuSansMono” (o cualquier otra font Mono, de mono-espaciada). En algunos casos puede no quedar seleccionada esa font y solo la instalará, en ese caso se debe

seleccionarla como fuente a usar, para ello hagan “Preferences > Set System Font ... > DejaVu Sans Mono” de la parte del menú que dice “Installed Fonts”

Trabajo a realizar

Hacer filein de **2024-2C-Parcial-2.st** e implementar el prototipo del juego del Pacman pedido usando TDD y las heurísticas de diseño vistas en la materia, **sacando código repetido, reemplazando ifs por polimorfismo donde haga falta, etc**

Entrega:

1. Entregar el fileout de la categoría de clase **2024-2C-Parcial-2** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **2024-2C-Parcial-2.st**
2. Entregar también el archivo que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon. Usen otra instalación de CuisUniversity/imagen si es necesario) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Deben entregar usando el siguiente form: <https://tinyurl.com/inge1-2p-2c-2024>
5. De forma alternativa si no pudiste entregar con el form, realizar la entrega enviando mail a: **entregas@isw2.com.ar** con el **Subject: LU nnn-aa - Solución 1er parcial 2c2024.**
En caso de rebotar el envío, reintentar comprimiendo los adjuntos.
6. **RECOMENDACIÓN IMPORTANTE: Salvar la imagen de manera frecuente o con el autosave**
7. Se asume que a esta altura de la cursada saben trabajar con la imagen, recuperarla, recuperar código fuente, revertir cambios y demás incidencias que pudieran ocurrir durante el examen.

Revisen bien los puntos de arriba. Cualquier error en los nombres o formato podrían ser penalizados en la nota.

IMPORTANTE: No retirarse sin tener el ok de los docentes de haber recibido la resolución por algún medio.

CERRAR EL TRABAJO A LAS 21:50.

LAS ENTREGAS RECIBIDAS DESPUÉS DE LAS 22:00 HRS NO SERÁN TENIDAS EN CUENTA