

**Concurrencia, Dibujo Básico en 2D y Persistencia****Objetivos****Unidad 3: Concurrencia, Dibujo Básico en 2D y Persistencia**

Al finalizar esta unidad, el estudiante estará en capacidad de:

- OE3.1 Desarrollar un programa que maneje concurrencia, de manera que sea posible que ejecute más de una parte del programa de manera simultánea, utilizando hilos de ejecución (threads).
- OE3.2 Construir interfaces de usuario que incluyan gráficas en 2 dimensiones como una alternativa en la presentación de información al usuario.
- OE3.3 Hacer persistir el estado del modelo de solución del problema durante la ejecución de un programa y restaurarlo cuando se requiera.
- OE3.4 Manipular archivos de texto y utilizarlos para implementar requerimientos del cliente relacionados con persistencia.

**Entregables. Unidad 1.**

1. Requerimientos Funcionales.
2. Diagrama de clases de modelo y control de la interfaz (no generado automáticamente)
3. Implementación completa de todos los requerimientos en Java.
4. Tabla de trazabilidad de requerimientos vs métodos (tabla con una columna de los requerimientos, tal que, por cada requerimiento se indica en la columna siguiente todos los métodos que contribuyen a resolverlo).

**1. Requerimientos funcionales.**

Nombre	R1: mostrar varios PacMan en la pantalla, los cuales deben moverse unos vertical y otros horizontalmente.
Resumen	Cuando el jugador inicia el juego se deben mostrar varios PacMan moviéndose vertical y horizontalmente.
Entradas	Archivo de texto.
Resultado	Se mueven varios PacMan en la dirección ingresada.

Nombre	R2: rebotar cada vez que un PacMan toque alguno de los extremos de la pantalla o toque algún otro PacMan, al rebotar el PacMan deberá moverse en sentido contrario.
Resumen	Cada que un PacMan toque alguno de los extremos de la pantalla o algún otro PacMan, este debe moverse en sentido contrario.
Entradas	Ninguna.
Resultado	Cada que el PacMan toca algún otro PacMan o un extremo de la pantalla, este rebota en sentido contrario.

Nombre	R3: detener el PacMan cada que el jugador haga clic sobre este.
Resumen	Una de las opciones de que el jugador obtenga más puntos es que se pueda detener el Pacman en movimiento, por ello el programa debe estar en capacidad de detener el Pacman cada vez que el usuario haga clic sobre el.
Entradas	Ninguna
Resultado	Se detiene un Pacman cada vez que un usuario haga clic sobre el.

Nombre	R4: almacenar la cantidad de rebotes que tenga cada PacMan y la cantidad de veces que el jugador detenga el PacMan antes de que rebote.
--------	---

**Concurrencia, Dibujo Básico en 2D y Persistencia**

Resumen	Se debe almacenar la cantidad de rebotes de todos los PacMan y la cantidad de veces que el jugador detenga al PacMan para así contar el puntaje del jugador y agregarlo al Hall de la fama.
Entradas	Cantidad de rebotes. Cantidad de veces que el jugador detuvo al PacMan.
Resultado	Se almacenan las veces que el jugador detiene al PacMan y también la cantidad de rebotes del PacMan.

Nombre	R5:cargar desde un archivo de texto la opción de selección del menú.
Resumen	se debe leer desde un archivo las opciones del menú, para “crear” el juego con las indicaciones que llegan a través del texto
Entradas	Archivo de texto.
Resultado	Se crea un archivo con las indicaciones que llegan a través del archivo de texto.

Nombre	R6: guardar el estado actual del juego y el estado actual del programa.
Resumen	Se debe poder almacenar el estado actual del juego para así mantener los mejores puntajes visibles al jugador.
Entradas	
Resultado	Se almacena el estado del juego.

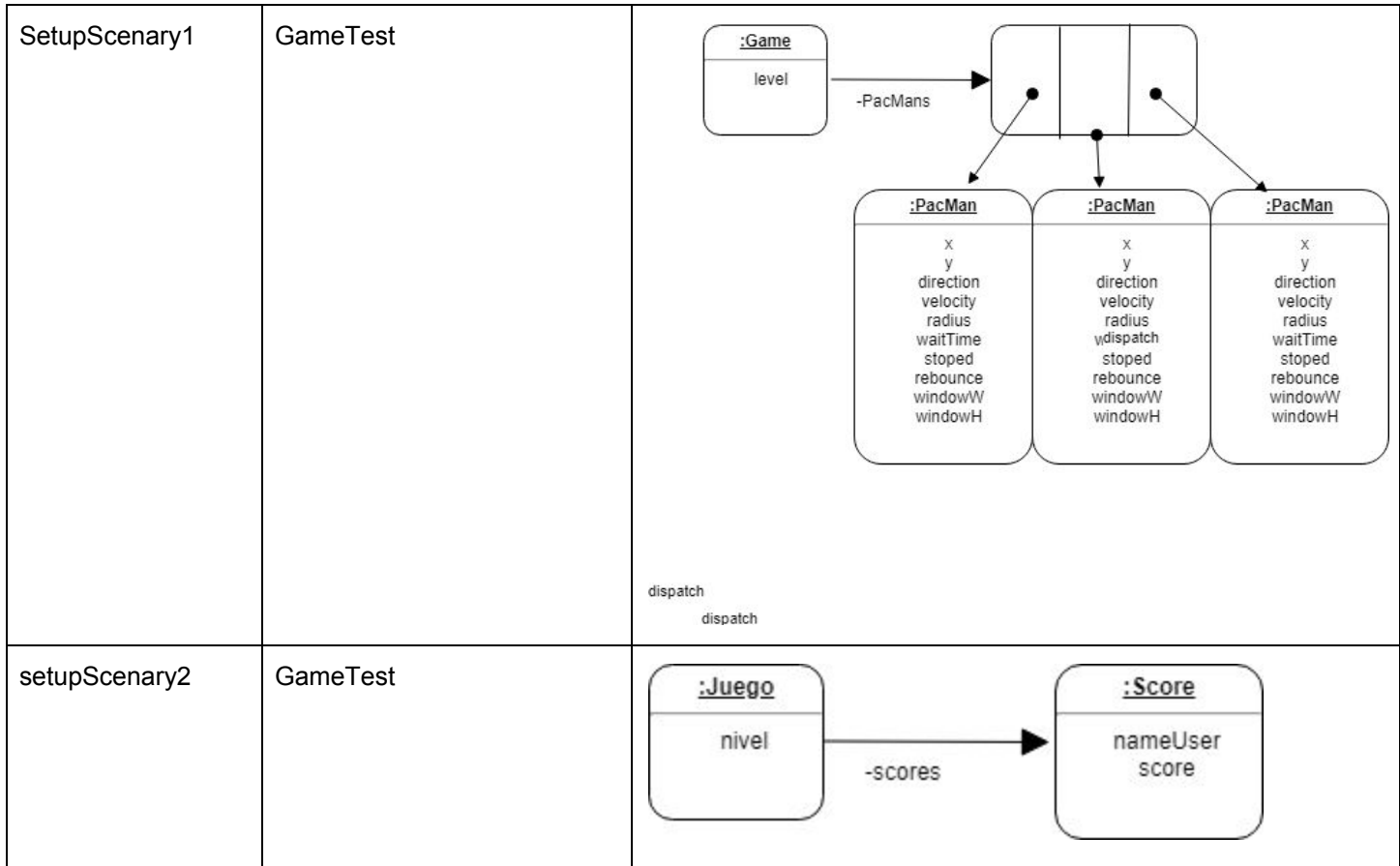
Tabla de trazabilidad

R#	Método	Clase
R6.	-saveGame()	Game
R1	-movePacMan()	PacMan
R2	-movePacMan()	PacMan
R3	-Stop()	PacMan
R4	-countRebounds()	PacMan
R5	-loadGameFile(String path, String sep)	Game

**Configuración de los Escenarios**

Nombre	Clase	Escenario
--------	-------	-----------

**Concurrencia, Dibujo Básico en 2D y Persistencia**



**Diseño de Casos de Prueba:**

Objetivo de la prueba: Al iniciar el juego, aparecen unos PacMan moviéndose vertical y horizontalmente.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Game		setupScenary1	Archivo texto	Aparecen unos PacMan moviéndose vertical y horizontalmente.

Objetivo de la prueba: El PacMan se detiene cuando un jugador hace clic sobre el.				
Clase	Método	Escenario	Valores de Entrada	Resultado
PacmanThread	stop()	setupScenary1	ninguno	se detiene el PacMan cuando el usuario hace clic sobre el.

**Objetivo de la Prueba:** Los PacMan rebotan cada que tocan los extremos de la ventana o algún otro PacMan

**Concurrencia, Dibujo Básico en 2D y Persistencia**

Clase	Método	Escenario	Valores de Entrada	Resultado
PacMan	movePacMan	setupScenary1	ninguno	los Pacman rebotan cuando se encuentran otro Pacman o llegan a los extremos.

Objetivo de la prueba: Se almacena la cantidad de rebotes que tenga cada PacMan.

Clase	Método	Escenario	Valores de Entrada	Resultado
Game		setupScenary1	rebound	Se obtiene un registro de todos los rebotes que realizó cada PacMan.

Objetivo de la prueba: Se calcula un puntaje por cada PacMan detenido en la menor cantidad de rebotes posibles.

Clase	Método	Escenario	Valores de Entrada	Resultado
Game		setupScenary2		Se obtiene un puntaje del jugador, según la cantidad de PacMan detenidos y el número de rebotes.

Objetivo de la prueba: Se muestra al usuario los mejores puntajes obtenidos en el juego (Hall de la fama).

Clase	Método	Escenario	Valores de Entrada	Resultado
Game		setupScenary2	puntaje	el usuario puede ver los mejores puntajes

Objetivo de la prueba: Se debe guardar el estado actual del juego.

Clase	Método	Escenario	Valores de Entrada	Resultado
Game	saveGame	SetupScenary2		Se guarda el estado del juego

