



ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES

<i>Nombre:</i>	<i>RF1 – Guardar partida</i>
<i>Resumen:</i>	<i>El sistema permite guardar una partida, cualquier partida en curso (donde no se haya ganado aún) se podrá guardar para ser continuada en cualquier otro momento</i>
<i>Entrada:</i>	<i>La ruta donde se guardará la partida</i>
<i>Salida:</i>	<i>La partida ha sido guardada satisfactoriamente</i>

<i>Nombre:</i>	<i>RF2 – Cargar partida</i>
<i>Resumen:</i>	<i>El sistema permite cargar una partida, cualquier partida que haya sido guardada previamente se podrá cargar y reanudar</i>
<i>Entrada:</i>	<i>La ruta de la partida</i>
<i>Salida:</i>	<i>La partida ha sido cargada satisfactoriamente</i>

<i>Nombre:</i>	<i>RF3 – Mover una pelota</i>
<i>Resumen:</i>	<i>El sistema permite dar movimiento a una pelota, cada pelota se moverá en una dirección específica, cada vez que ocurra una colisión (que una pelota llegue a algún extremo, ya sea superior, inferior, izquierdo o derecho) la pelota se moverá en la dirección contraria a la que se estaba moviendo antes de la colisión</i>

Entrada:	<i>Ninguna</i>
Salida:	<i>La pelota se mueve según corresponda</i>

Nombre:	<i>RF4 – Detener una pelota</i>
Resumen:	<i>El sistema permite detener una pelota que se encuentre en movimiento</i>
Entrada:	<i>Ninguna</i>
Salida:	<i>La pelota se ha detenido satisfactoriamente</i>

Nombre:	<i>RF5 – Contar rebotes</i>
Resumen:	<i>El sistema permite llevar una cuenta de cuantos rebotes han ocurrido. Un rebote ocurre cuando una pelota alcanza algún extremo, ya sea superior, inferior, derecho o izquierdo</i>
Entrada:	<i>Ninguna</i>
Salida:	<i>Se cuentan cuantos rebotes han ocurridos</i>

Nombre:	<i>RF6 – Comprobar si gana</i>
Resumen:	<i>El sistema permite comprobar si se ganó el juego, se gana el juego cuando todas las pelotas han sido detenidas</i>
Entrada:	<i>Ninguna</i>
Salida:	<i>Se comprueba si se ganó</i>

Nombre:	<i>RF7 – Guardar puntajes más altos</i>
Resumen:	<i>El sistema permite guardar los 10 puntajes más altos de cada dificultad (0, 1 y 2)</i>
Entrada:	<i>El nombre del jugador que logró el puntaje</i>
Salida:	<i>El puntaje se ha guardado satisfactoriamente</i>

Nombre:	<i>RF8 – Cargar los puntajes más altos</i>
Resumen:	<i>El sistema permite cargar los 10 puntajes más altos de cada dificultad (0, 1 y 2)</i>
Entrada:	<i>Ninguna.</i>
Salida:	<i>Los puntajes se han cargado satisfactoriamente</i>

ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES

<i>Nombre:</i>	<i>RNF1 – Cargar desde archivos planos</i>
<i>Resumen:</i>	<i>Las partidas se deben guardar desde un archivo plano</i>

<i>Nombre:</i>	<i>RNF2 – Serialización</i>
<i>Resumen:</i>	<i>Los puntajes deben ser guardados con serialización</i>

<i>Nombre:</i>	<i>RNF3 – Guardar en archivos planos</i>
<i>Resumen:</i>	<i>Las partidas se deben guardar desde un archivo plano</i>

<i>Nombre:</i>	<i>RNF4 – Interfaz gráfica</i>
<i>Resumen:</i>	<i>El programa debe contar con interfaz gráfica</i>

TRAZABILIDAD

<i>RF1 – Guardar partida</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	saveGame()
<i>Game</i>	saveGame()
<i>Ball</i>	toString()

<i>RF2 – Cargar partida</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	loadScores()
<i>Game</i>	initGame()
<i>Game</i>	createBall()

<i>RF3 – Mover una pelota</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Ball</i>	move()
<i>Ball</i>	move()
<i>Ball</i>	detectCollisionBorder()

RF4 – Detener una pelota

<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	stopBalls()
<i>Ball</i>	stop()

RF5 – Contar rebotes

<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	calculateScore()
<i>Ball</i>	getBounces()

RF6 – Comprobar si gano

<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	win()
<i>Ball</i>	isMoving()

RF7 – Guardar puntajes más altos

<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	addScore()

<i>Game</i>	checkFreeSlot()
<i>Game</i>	addScoreFreeSlot()
<i>Game</i>	addScoreSorted()
<i>Game</i>	sortScores()
<i>Score</i>	compareTo()

<i>RF8 – Cargar los puntajes más altos</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Game</i>	loadScores()

CASOS DE PRUEBA

Caso de prueba 1

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	move()	Una pelota que se mueve hacia la derecha	Las dimensiones de la pantalla	Verdadero, la pelota rebota cuando alcanza los límites de las dimensiones

Caso de prueba 2

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	move()	Una pelota que se mueve hacia arriba	Las dimensiones de la pantalla	Verdadero, la pelota rebota cuando alcanza los límites de las dimensiones

Caso de prueba 3

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	move()	Una pelota que se mueve hacia la izquierda	Las dimensiones de la pantalla	Verdadero, la pelota rebota cuando alcanza los límites de las dimensiones

Caso de prueba 4

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	move()	Una pelota que se mueve hacia abajo	Las dimensiones de la pantalla	Verdadero, la pelota rebota cuando alcanza los límites de las dimensiones

Caso de prueba 5

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	stop()	Una pelota cuya posición es 70 en x y 90 en y	La posición del clic	Verdadero, la pelota se detiene cuando la posición del clic está dentro de la pelota

Caso de prueba 6

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	stop()	Una pelota cuya posición es 70 en x y 51 en y	La posición del clic	Verdadero, la pelota se detiene cuando la posición del clic está dentro de la pelota

Caso de prueba 7

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	stop()	Una pelota cuya posición es 51 en x y 90 en y	La posición del clic	Verdadero, la pelota se detiene cuando la posición del clic está dentro de la pelota

Caso de prueba 8

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	stop()	Una pelota cuya posición es 430 en x y 890 en y	La posición del clic	Verdadero, la pelota se detiene cuando la posición del clic está dentro de la pelota

Caso de prueba 9

Clase	Método	Escenario	Valores de entrada	Resultado
Game	game()	Creo un juego con un archivo que no existe	La ruta de archivo	Verdadero, al intentar crear el juego se lanza la excepción esperada

Caso de prueba 10

Clase	Método	Escenario	Valores de entrada	Resultado
Game	load()	Creo un juego con un archivo que existe, pero que no cumple con el formato esperado	La ruta de archivo	Verdadero, al intentar crear el juego se lanza la excepción esperada

Caso de prueba 11

Clase	Método	Escenario	Valores de entrada	Resultado
Game	sortScores	Una matriz de puntajes vacía a la cual le voy agregando puntajes y comprobando si están ordenados	Los puntos y nombres del puntaje	Verdadero, los puntajes están ordenados