

Requerimientos Funcionales

Requerimiento	1
Resumen	Implementacion de GUI
Entradas	
Salidas	Interfaz GUI
Requerimiento	2
Resumen	creacion de pac-mans
Entradas	
Salidas	Arcos en la escena formando los pacmans
Requerimiento	3
Resumen	rebote en paredes de pacmans
Entradas	
Salidas	Al rebotar se invierte la direccion del pacman
Requerimiento	4
Resumen	colisiones entre pacmans
Entradas	indice del pacman, indice del segundo pacman
Salidas	Booleano evaluando si colisionaron o no
Requerimiento	5
Resumen	Muerte de pacmans
Entradas	indice del pacman, posicion en x, posicion en y
Salidas	booleano evaluando si colisionaron o no
Requerimiento	6
Resumen	serializacion de puntaje
Entradas	
Salidas	archivo
Requerimiento	7
Resumen	guardado de estado de los pacmans en archivo de txt
Entradas	archivo
Salidas	actualizacion de archivo
Requerimiento	8
Resumen	cargado de estado de pacmans en archivo txt
Entradas	archivo
Salidas	carga el estado de los pacman al modelo
Requerimiento	9
Resumen	contar la cantidad de rebotes de todos los pacman
Entradas	
Salidas	actualiza la cantidad de rebotes en el modelo
Requerimiento	10
Resumen	Mostrar puntajes
Entradas	
Salidas	Muestra un mensaje con los puntajes

Trazabilidad

RQ	paquete	clase	metodo
1	application	Main	launch()
		PlaygroundController	initialize()
		Playground.fxml	
	model	Level	
		PacMan	
		Score	
	Thread	ReboundThread	
		PacsThread	
2	application	Main	launch()
		PlaygroundController	initialize()
		Playground.fxml	
	model	Level	
		PacMan	
		Score	
	Thread	ReboundThread	
		PacsThread	
3	application	Main	launch()
		PlaygroundController	initialize(),menuLoad()
		Playground.fxml	
	model	Level	
		PacMan	
		Score	
	Thread	ReboundThread	
		PacsThread	hitHorizontalBounds, hitVerticalBounds, run, move
4	application	Main	launch()
		PlaygroundController	initialize, menuLoad
		Playground.fxml	
	model	Level	
		PacMan	
		Score	
	Thread	ReboundThread	
		PacsThread	
5	application	Main	launch()
		PlaygroundController	initialize(), menuLoad()
		Playground.fxml	
	model	Level	collisionPacMan
		PacMan	
		Score	
	Thread	ReboundThread	

		PacsThread	run,move
6	application	Main	launch()
		PlaygroundController	initialize
		Playground.fxml	
	model	Level	submitScore,saveScore,loadScore
		PacMan	
		Score	Class
	Thread	ReboundThread	
		PacsThread	
7	application	Main	launch()
		PlaygroundController	initialize,saveData
		Playground.fxml	
	model	Level	save,getPacs
		PacMan	getPosX,getPosY
		Score	
	Thread	ReboundThread	
		PacsThread	
8	application	Main	launch()
		PlaygroundController	initialize
		Playground.fxml	
	model	Level	load,add
		PacMan	
		Score	
	Thread	ReboundThread	
		PacsThread	
9	application	Main	launch()
		PlaygroundController	initialize, load
		Playground.fxml	
	model	Level	load,add, getRebounds,setRebounds
		PacMan	
		Score	
	Thread	ReboundThread	run
		PacsThread	
10	application	Main	launch()
		PlaygroundController	initialize,menuDisplayScore
		Playground.fxml	
	model	Level	displayScore
		PacMan	
		Score	
	Thread	ReboundThread	
		PacsThread	

