**DOCUMENTACION**

**LABORATORIO 3**

**Requerimientos funcionales:**

|  |  |
| --- | --- |
| Nombre | R0: Crear y dibujar los Pac-man |
| Resumen | Crea e inserta los Pac-man en el escenario |
| Entrada | ActionEvent |
| Salida | Se inicia el juego y aparecen los Pac-man |

|  |  |
| --- | --- |
| Nombre | R1: generar movimiento horizontal y vertical para cada Pac-man |
| Resumen | Genera el movimiento horizontal y vertical único de cada Pac-man |
| Entrada | N/A |
| Salida | Los objetos Pac-man se mueven de forma horizontal y vertical en el escenario |

|  |  |
| --- | --- |
| Nombre | R2: Hacer al Pac-man rebotar al tocar los bordes u otro Pac-man |
| Resumen | Cuando un Pac-man toca el borde del escenario, este rebota y cambia su dirección. Lo mismo pasa si toca otro Pac-man. |
| Entrada | N/A |
| Salida | Los Pac-man rebotan en contacto con los bordes y entre ellos. |

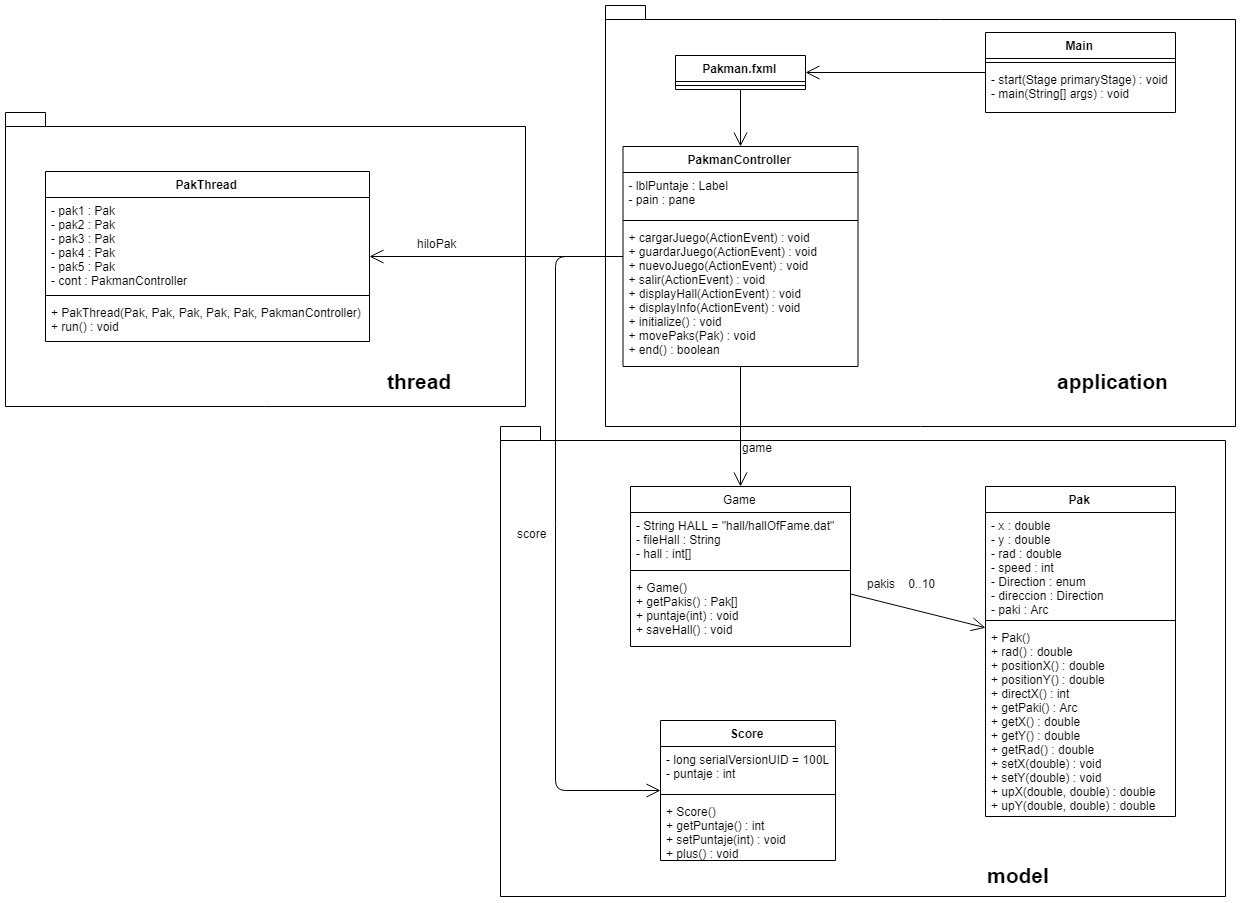
|  |  |
| --- | --- |
| Nombre | R3: Aumentar el puntaje y eliminar un Pac-man al realizar clic sobre este |
| Resumen | Cada vez que se da clic sobre un Pac-man, este desaparece y aumenta el puntaje del jugador. Al acabar con los 5 Pac-man el juego termina. |
| Entrada | MouseEvent |
| Salida | Se aumenta el puntaje al acabar con un Pac-man. |

|  |  |
| --- | --- |
| Nombre | R4: Almacenar los 10 mejores puntajes utilizando serializacion |
| Resumen | Al acabar un juego, si el puntaje es lo suficientemente alto, este se guarda en el hall de la fama que persiste utilizando serializacion |
| Entrada | Puntaje |
| Salida | Entra o no en el hall de la fama |

|  |  |
| --- | --- |
| Nombre | R5: Cargar un archivo de juego existente |
| Resumen | En vez de realizar un juego nuevo, se puede cargar un juego preexistente. |
| Entrada | Archivo plano de juego |
| Salida | Inicia el juego desde el punto de guardado |

|  |  |
| --- | --- |
| Nombre | R6: Guardar un archivo de juego |
| Resumen | Se guarda el juego actual como un archivo de texto plano |
| Entrada | Contexto del juego actual |
| Salida | Se guarda la información del juego en el archivo |

**Diagrama de clases:**



**Tabla de trazabilidad de requerimientos vs métodos:**

|  |  |  |
| --- | --- | --- |
| Requerimiento | Clase | Método |
| Crear y dibujar los Pac-man | PakmanController | nuevoJuego(ActionEvent) : void |
| Game | Game() |
| Pak | Pak() |
| generar movimiento horizontal y vertical para cada Pac-man | PakmanController | MovePaks(Pak) : void |
| Game | getPakis() : Pak[] |
| Pak | upX(double, double) : double  upY(double, double) : double |
| Hacer al Pac-man rebotar al tocar los bordes u otro Pac-man | PakmanController | MovePaks(Pak) : void |
| Game | getPakis() : Pak[] |
| Pak | upX(double, double) : double  upY(double, double) : double |
| Aumentar el puntaje y eliminar un Pac-man al realizar clic sobre este | PakmanController | nuevoJuego(ActionEvent) : void |
| Score | plus() : int |
| Almacenar los 10 mejores puntajes utilizando serializacion | PakmanController | End() : boolean |
| Game | Puntaje(int) : void |
| Score | getPuntaje() : int |
| Cargar un archivo de juego existente | PakmanController | CargarJuego(ActionEvent) : void |
| Guardar un archivo de juego | PakmanController | GuardarJuego(ActionEvent) : void |