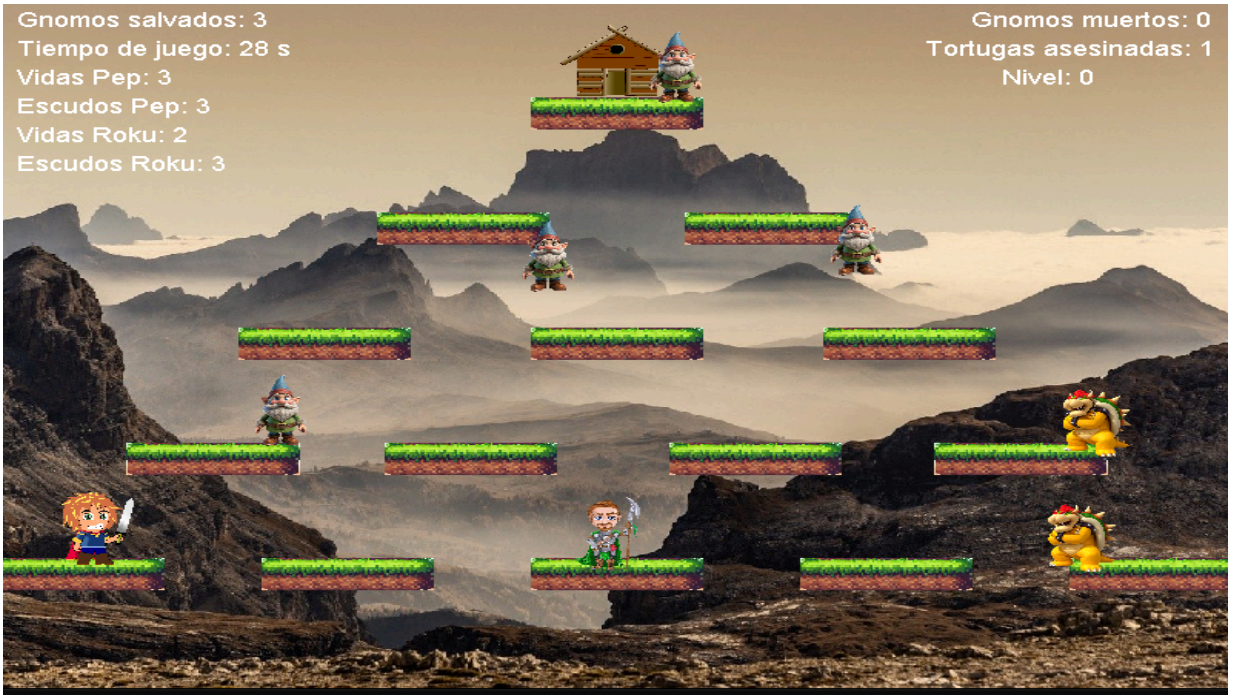
Rescate de Gnomos



Universidad: Universidad Nacional de General Sarmiento

Carrera: Tecnicatura Universitaria en Informática

Materia: Programación I

Comisión: 04

Docentes: Bidart Gauna Lucas Ezequiel y Zorzoli Giuliana

Banda Horaria: martes y jueves de 18hs a 22hs

Estudiantes: Lauzan Mateo, Toledo Nicolás y Aldeco Becette Joaquín

Correos electrónicos: mateolauzan@gmail.com, nicolastoledojym@gmail.com, joaquinaldeco@gmail.com

Introducción

En este informe se hará una introducción acerca del código implementado en la realización del trabajo práctico. Se explicarán las funcionalidades de lo que ha sido agregado y las dificultades que tuvimos durante la realización del trabajo mencionado.

El trabajo consiste en hacer un videojuego de un guerrero medieval que va rescatando gnomos y matando tortugas, desplazándose sobre unas islas que están suspendidas en el aire. En un principio ya fueron hechas algunas funcionalidades básicas como la pantalla donde se juega, así como también se dejaron hechas muchas funciones para poder reutilizar (como la captación de teclas o el procesamiento de imágenes, sonidos y otros) una clase **Entorno,** que permite crear un objeto capaz de encargarse de la interfaz gráfica y de la interacción con el usuario, una clase **Juego** que cuenta con el objeto entorno, que es creado en el constructor del juego y recibe el juego en cuestión iniciando el simulador. A partir de ahí, en cada instante de tiempo que pasa, el entorno ejecuta el método **tick()** del juego. Este debe actualizar el estado interno del juego para simular el paso del tiempo. Nuestro trabajo consistió en encargarnos de la implementación de la inteligencia del juego, saber qué código hay que reutilizar, cuando y de qué manera hacerlo sabiendo que lo pedido es: actualizar el estado interno de todos los objetos involucrados en la simulación, dibujar los mismos en la pantalla, verificar si algún objeto aparece o desaparece del juego, verificar si hay objetos interactuando entre sí y verificar si los usuarios están presionando alguna tecla y actuar en consecuencia. Nosotros pudimos realizar lo solicitado y adicionalmente le agregamos algunos objetos más a los obligatorios, la modalidad de jugabilidad doble, el incremento de dificultad (por niveles), mensajes temporales por pantallas, un menú de fin de juego, implementación de imágenes y efectos de sonido.

Explicación de código

Primero se crean las clases de los personajes que se van a utilizar en el juego, en nuestro caso contamos con las clases Casa, Isla, Islas, Gnomo, Tortuga, Pep, Roku, BolaFuego y Escudo.

Cada clase tiene sus constructores, getters, setters y función dibujar (que les agrega la imagen). También tienen variables en común como x, y, alto, ancho (ya que maneja las coordenadas) y también comparten una variable de tipo Imagen. Luego, cada una tiene variables adicionales que se agregan a sus propios getters, setters y constructores, y además tienen sus propios métodos.

La clase Casa y la clase Isla no presentan nada adicional.

La clase Islas contiene a todas las islas del juego en una ArrayList y su método principal es **getIslas()** que retorna el ArrayList de islas.

La Clase Gnomo y Tortuga tienen funciones y variables adicionales compartidas ya que movimientoIzquierda(en Tortuga) o movimientoDerecha(en gnomo) tienen la misma utilidad, que es para fijar direcciones de sus objetos. Además, ambos comparten la variable aterrizado que se usa para corroborar si el objeto se encuentra en el aire o no. Las funciones adicionales compartidas son las de elegir dirección (que por medio de un numero random elige para que lado moverse), la de si están sobre una isla, la de resetear aterrizado (corrobora si están en el aire), la función para saber si están dentro del entorno (dentro de la pantalla en la que se juega) , una función booleana para saber para qué lado se está moviendo y las funciones de moverse a la izquierda y derecha respectivamente. Hay que agregar que la clase Tortuga tiene 2 funciones que son solo de ella, una es para saber si está sobre la isla de la casa (así sabe que no se debe parar ahí) y la otra es para saber si una isla ya está ocupada por una tortuga (así tampoco se para ahí), pero esta última no la pudimos hacer funcionar. Por

último, hay unas variables que en este momento no se usan, como las de daño, vida, velocidad vertical y velocidad horizontal.

Las clases Pep y Roku comparten las mismas variables y funciones ya que son los 2 jugadores con los que se puede jugar. Variables adicionales: velocidadSalto, gravedad e impulso (todas se usan para generar un buen salto), vida(que cuenta las vidas que le quedan) y la de daño que en realidad tampoco se usa. Funciones adicionales: al igual que antes, las que son para saber si encuentran sobre una isla o si están dentro del entorno, las de moverse para la izquierda o derecha, la de que si chocan la cabeza con alguna isla no puedan seguir el salto, y las de salto que son las de iniciar salto(la cual le va restando impulso para que en algún momento no suba más) y la de actualizar salto (que corrobora todo el tiempo en que fase del salto está).

La clase Escudo va a fijar siempre su posición sobre las coordenadas que cuando se invoquen van a ser las de sus guerreros respectivos. Variables adicionales: solo la de vida (para poder ser utilizado). Funciones adicionales: no tiene.

La clase BolaFuego, variables adicionales: la de direc (es un booleano que se utilizará para saber en qué dirección tendrá que moverse). Funciones adicionales: la que corrobora si se encuentra dentro del entorno, y por supuesto las de moverse a la derecha o izquierda.

La clase Juego es la clase principal ya que contiene todo el entorno y el funcionamiento de cada clase a la hora de que este es ejecutado. En la primera parte cuenta con algunas funciones importantes como dibujarFondo() y hayColision() la segunda recibe como parámetros las coordenadas, el ancho y el alto de 2 personajes para chequear si se chocan entre ellos. En la clase Juego se definen las primeras propiedades para que el juego inicie correctamente, entre ellos los personajes. Luego le sigue el método principal

que es el **tick()** que contiene todos los métodos que se ejecutan en cada tick. Métodos de la clase Juego dentro de **tick()**:

dibujarFondo(): Agarra la imagen que se va a usar de fondo para el juego y la dibuja.

terminarJuego(): Coloca la variable juegoTerminado en true.

dibujarMenuFinal(String mensajeFinal): Dibuja el menú que aparece al ganar o perder una partida y recibe un mensaje que le aparecerá al usuario. También escribe las estadísticas de la partida. Si se presiona la r llama al método **reiniciarJuego()** que limpia las propiedades y coloca la variable juegoTerminado en false.

En caso de que Pep o Roku no sean nulos, se les dibuja, se colocan los atributos de sus escudos. Se asignan las teclas de movimiento y se les pone una condición de que si no están sobre una isla se llame a sus métodos para que se caigan. Los personajes cuentan con una muerte por límite que, en caso de que se encuentre fuera del entorno, se les resta una vida. Para las bolas de fuego se chequea si se toca la tecla para que se disparen. Estos se agregan a una lista y se elige la dirección de acuerdo hacia dónde se dirigió el personaje la última vez.

Para el Gnomo se usa un tiempoDeGeneración que suma 1 pero lo divide por 60 ya que se debe a los ticks que hay aproximadamente por segundo y suma un gnomo si hay menos de 2 o cada 2 segundos. Para el Gnomo y la Tortuga se verifica que si está sobre la isla se actualice su propiedad de estaAterrizado y se elija una dirección aleatoria para que se dirija. En caso de que no esté dentro del entorno se borran. A la tortuga también se le corrobora de que si salió de la isla y se estaba moviendo para un lado, cambie para el otro. Seguidamente, se encuentran las funcionalidades de colisión. La primera recorre a los gnomos y dentro a las tortugas. En caso de que Pep o Ruku existan y los gnomos estén por debajo de 300 px en el eje “y”, estos

podrán ser salvados si es que hay colisión entre sí. Al agarrar 30 gnomos se gana, cada vez que se salva a un gnomo se aumenta 1 escudo y se elimina al gnomo. La muerte de Pep o Roku se da si hay colisión también con las Tortugas y no tengan el escudo puesto. Los gnomos mueren siempre que caen de las islas o los toca una tortuga. Para la muerte de una tortuga se chequea la colisión entre una bola de fuego y la misma. Se remueven las bolas de fuego del arreglo de disparos tanto de Roku como de Pep y a las tortugas.

Conclusión

En conclusión, sentimos que pudimos cumplir con los objetivos necesarios para una buena presentación del trabajo y pudimos volcar algunos de nuestros conocimientos vistos en clase durante estos meses dentro del mismo, para realizar un trabajo eficiente y con un buen funcionamiento. Además, nos permitió aprender y aplicar distintos conceptos de desarrollo de juegos.

Las dificultades que se nos presentaron fueron algunos asuntos de tiempos y organización que complicaron el desarrollo del juego pero encaramos el proyecto con muchas ganas y estamos satisfechos con el resultado logrado.

Finalmente, logramos implementar nuevos detalles al juego para que el jugador pueda tener una mejor experiencia y distintas funcionalidades nuevas que fuimos ampliando para ofrecer nuevas características.

Implementación del código

PEP

```
package juego;
import java.awt.Image;
import java.util.ArrayList;
import entorno.Entorno;
import entorno.Herramientas;
public class Pep {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private double velocidad;
	private int vida;
	private double daño;
	private Image imagen;
	private double velocidadSalto = 0;
	private final double gravedad = 0.5;
	private final double impulsoSalto = 16;
	public Pep(double x, double y, double alto, double ancho, double velocidad, int vida, double
daño, Image imagen) {
		this.setX(x);
		this.setY(y);
		this.setAlto(alto);
		this.setAncho(ancho);
		this.setVelocidad(velocidad);
		this.setVida(vida);
		this.setDaño(daño);
		this.setImagen(imagen);
	}
```

```
	public void dibujar(Entorno entorno) {
		Image imagenPep = Herramientas.cargarImagen("imagenes/pep.png");
		entorno.dibujarImagen(imagenPep, this.getX(), this.getY(), Math.toRadians(0), 0.05);
	}
	public void iniciarSalto() {
		this.velocidadSalto = -impulsoSalto;
	}
	public void actualizarSalto(ArrayList<Isla> islas) {
		if (pepChocaCabeza(islas)) {
			this.velocidadSalto = gravedad;
		} else if (!pepSobreIsla(islas) || this.velocidadSalto < 0) {
			this.y += velocidadSalto;
			this.velocidadSalto += gravedad;
		} else {
			this.velocidadSalto = 0;
		}
	}
	public boolean pepSobreIsla(ArrayList<Isla> islas) {
		for (Isla isla : islas) {
			if (this.x >= isla.getX() - isla.getAncho() / 2 && this.x <= isla.getX() +
isla.getAncho() / 2) {
				if (this.y + this.alto / 2 >= isla.getY() - isla.getAlto() / 2
						&& this.y + this.alto / 2 <= isla.getY() + isla.getAlto() /
2) {
					return true;
				}
			}
		}
		return false;
	}
	public boolean pepChocaCabeza(ArrayList<Isla> islas) {
		for (Isla isla : islas) {
			if (this.x >= isla.getX() - isla.getAncho() / 2 && this.x <= isla.getX() +
isla.getAncho() / 2) {
				if (this.y - this.alto / 2 <= isla.getY() + isla.getAlto() / 2
						&& this.y - this.alto / 2 >= isla.getY() - isla.getAlto() /
2) {
					return true;
				}
			}
```

```
                }
                return false;
        }
        public void moverDerecha() {
                this.x = this.x + 3;
        }
        public void moverIzquierda() {
                this.x = this.x - 3;
        }
        public void moverAbajo() {
                this.y = this.y + 6;
        }
        public boolean dentroDelEntorno(Entorno entorno) {
                return  this.y <= entorno.alto() && this.x <= entorno.ancho() && this.y >= -100 &&
this.x >= 0;
        }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x = x;
        }
        public double getY() {
                return y;
        }
        public void setY(double y) {
                this.y = y;
        }
        public double getAlto() {
                return alto;
        }
        public void setAlto(double alto) {
                this.alto = alto;
        }
        public double getAncho() {
                return ancho;
        }
        public void setAncho(double ancho) {
                this.ancho = ancho;
        }
```

```
	public double getVelocidad() {
		return velocidad;
	}
	public void setVelocidad(double velocidad) {
		this.velocidad = velocidad;
	}
	public int getVida() {
		return vida;
	}
	public void setVida(int vida) {
		this.vida = vida;
	}
	public Image getImagen() {
		return imagen;
	}
	public void setImagen(Image imagen) {
		this.imagen = imagen;
	}
	public double getDaño() {
		return daño;
	}
	public void setDaño(double daño) {
		this.daño = daño;
	}
}
```

ROKU

```
package juego;
import java.awt.Image;
import java.util.ArrayList;
import entorno.Entorno;
import entorno.Herramientas;
public class Roku {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private double velocidad;
```

```
	private int vida;
	private double daño;
	private Image imagen;
	private double velocidadSalto = 0;
	private final double gravedad = 0.5;
	private final double impulsoSalto = 16;
	public Roku(double x, double y, double alto, double ancho, double velocidad, int vida, double
daño, Image imagen) {
		this.setX(x);
		this.setY(y);
		this.setAlto(alto);
		this.setAncho(ancho);
		this.setVelocidad(velocidad);
		this.setVida(vida);
		this.setDaño(daño);
		this.setImagen(imagen);
	}
	public void dibujar(Entorno entorno) {
		Image imagenPep = Herramientas.cargarImagen("imagenes/roku.png");
		entorno.dibujarImagen(imagenPep, this.getX(), this.getY(), Math.toRadians(0), 0.05);
	}
	public void iniciarSalto() {
		this.velocidadSalto = -impulsoSalto;
	}
	public void actualizarSalto(ArrayList<Isla> islas) {
		if (rokuChocaCabeza(islas)) {
			this.velocidadSalto = gravedad;
		} else if (!rokuSobreIsla(islas) || this.velocidadSalto < 0) {
			this.y += velocidadSalto;
			this.velocidadSalto += gravedad;
		} else {
			this.velocidadSalto = 0;
		}
	}
	public boolean rokuSobreIsla(ArrayList<Isla> islas) {
		for (Isla isla : islas) {
			if (this.x >= isla.getX() - isla.getAncho() / 2 && this.x <= isla.getX() +
isla.getAncho() / 2) {
				if (this.y + this.alto / 2 >= isla.getY() - isla.getAlto() / 2
```

```
                                        && this.y + this.alto / 2 <= isla.getY() + isla.getAlto() /
2) {
                                                return true;
                                        }
                                }
                        }
                        return false;
                }
                public boolean rokuChocaCabeza(ArrayList<Isla> islas) {
                        for (Isla isla : islas) {
                                if (this.x >= isla.getX() - isla.getAncho() / 2 && this.x <= isla.getX() +
isla.getAncho() / 2) {
                                        if (this.y - this.alto / 2 <= isla.getY() + isla.getAlto() / 2
                                                        && this.y - this.alto / 2 >= isla.getY() - isla.getAlto() /
2) {
                                                return true;
                                        }
                                }
                        }
                        return false;
                }
                public void moverDerecha() {
                        this.x = this.x + 3;
                }
                public void moverIzquierda() {
                        this.x = this.x - 3;
                }
                public void moverAbajo() {
                        this.y = this.y + 6;
                }
                public boolean dentroDelEntorno(Entorno entorno) {
                        return this.y <= entorno.alto() && this.x <= entorno.ancho() && this.y >= -100 &&
this.x >= 0;
                }
                public double getX() {
                        return x;
                }
                public void setX(double x) {
                        this.x = x;
                }
```

```
	public double getY() {
		return y;
	}
	public void setY(double y) {
		this.y = y;
	}
	public double getAlto() {
		return alto;
	}
	public void setAlto(double alto) {
		this.alto = alto;
	}
	public double getAncho() {
		return ancho;
	}
	public void setAncho(double ancho) {
		this.ancho = ancho;
	}
	public double getVelocidad() {
		return velocidad;
	}
	public void setVelocidad(double velocidad) {
		this.velocidad = velocidad;
	}
	public int getVida() {
		return vida;
	}
	public void setVida(int vida) {
		this.vida = vida;
	}
	public Image getImagen() {
		return imagen;
	}
	public void setImagen(Image imagen) {
		this.imagen = imagen;
	}
	public double getDaño() {
		return daño;
	}
	public void setDaño(double daño) {
```

```
                this.daño = daño;
        }
}
```

Bola De Fuego

```
package juego;
import java.awt.Image;
import entorno.Entorno;
import entorno.Herramientas;
public class BolaFuego {
        private double x;
        private double y;
        private double alto;
        private double ancho;
        private Image imagen;
        private boolean direc;


        public BolaFuego(double x, double y, double alto, double ancho, Image imagen, boolean
direc) {
                super();
                this.x = x;
                this.y = y;
                this.alto = alto;
                this.ancho = ancho;
                this.imagen = imagen;
                this.direc= direc;
        }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x = x;
        }
        public double getY() {
```

```
                return y;
        }
        public void setY(double y) {
                this.y = y;
        }
        public double getAlto() {
                return alto;
        }
        public void setAlto(double alto) {
                this.alto = alto;
        }
        public double getAncho() {
                return ancho;
        }
        public void setAncho(double ancho) {
                this.ancho = ancho;
        }
        public Image getImagen() {
                return imagen;
        }
        public void setImagen(Image imagen) {
                this.imagen = imagen;
        }


        public boolean isDirec() {
                return direc;
        }
        public void setDirec(boolean direc) {
                this.direc = direc;
        }
        public void dibujar(Entorno entorno) {
                Image imagenBolaFuego = Herramientas.cargarImagen("imagenes/bolaFuego.jpg");
                entorno.dibujarImagen(imagenBolaFuego, this.getX(), this.getY(), Math.toRadians(0),
0.02);
        }

        public void moverDerecha() {
                this.x = this.x + 5;
        }
```

```
	public void moverIzquierda() {
		this.x = this.x - 5;
	}
	public boolean dentroDelEntorno(Entorno entorno) {
		return this.y <= entorno.alto() && this.x <= entorno.ancho();
	}
}
```

Casa

```
package juego;
import java.awt.Image;
import entorno.Entorno;
import entorno.Herramientas;
public class Casa {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private Image imagen;
	public Casa(double x, double y, double alto, double ancho, Image Imagen) {
		this.setX(x);
		this.setY(y);
		this.setAlto(alto);
		this.setAncho(ancho);
		this.setImagen(imagen);
	}
	public double getX() {
		return x;
	}
	public void setX(double x) {
		this.x = x;
	}
	public double getY() {
		return y;
	}
	public void setY(double y) {
		this.y = y;
	}
	public double getAlto() {
		return alto;
```

```
	}
	public void setAlto(double alto) {
		this.alto = alto;
	}
	public double getAncho() {
		return ancho;
	}
	public void setAncho(double ancho) {
		this.ancho = ancho;
	}
	public Image getImagen() {
		return imagen;
	}
	public void setImagen(Image imagen) {
		this.imagen = imagen;
	}
	public void dibujar(Entorno entorno) {
		Image imagenCasa = Herramientas.cargarImagen("imagenes/casa.png");
		entorno.dibujarImagen(imagenCasa, this.getX(), this.getY(), Math.toRadians(0),
0.06);
	}
}
```

Escudo

```
package juego;
import java.awt.Image;
import entorno.Entorno;
import entorno.Herramientas;
public class Escudo {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private int vida;
	private Image imagen;


	public Escudo(double x, double y, double alto, double ancho, int vida, Image imagen) {
```

```
                super();
                this.x = x;
                this.y = y;
                this.alto = alto;
                this.ancho = ancho;
                this.vida = vida;
                this.imagen = imagen;
        }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x = x;
        }
        public double getY() {
                return y;
        }
        public void setY(double y) {
                this.y = y;
        }
        public double getAlto() {
                return alto;
        }
        public void setAlto(double alto) {
                this.alto = alto;
        }
        public double getAncho() {
                return ancho;
        }
        public void setAncho(double ancho) {
                this.ancho = ancho;
        }
        public int getVida() {
                return vida;
        }
        public void setVida(int vida) {
                this.vida = vida;
        }
        public Image getImagen() {
                return imagen;
```

```
	}
	public void setImagen(Image imagen) {
		this.imagen = imagen;
	}

	public void dibujar(Entorno entorno) {
		Image imagenEscudo = Herramientas.cargarImagen("imagenes/escudo.png");
		entorno.dibujarImagen(imagenEscudo,  this.getX(),  this.getY(),  Math.toRadians(0),
0.07);
	}



}
```

Gnomo

```
package juego;
import java.awt.Image;
import java.util.ArrayList;
import entorno.Entorno;
import entorno.Herramientas;
public class Gnomo {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private double velocidad;
	private double vida;
	private double daño;
	private Image imagen;
	private boolean aterrizado;
	private boolean movimientoDerecha;
	public Gnomo(double x, double y, double alto, double ancho, double velocidad, double vida,
double daño,
			Image imagen, boolean aterrizado) {
		this.setX(x);
		this.setY(y);
		this.setAlto(alto);
```

```
                this.setAncho(ancho);
                this.setVelocidad(velocidad);
                this.setVida(vida);
                this.setDaño(daño);
                this.setImagen(imagen);
                this.aterrizado = false;
        }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x = x;
        }
        public double getY() {
                return y;
        }
        public void setY(double y) {
                this.y = y;
        }
        public double getAlto() {
                return alto;
        }
        public void setAlto(double alto) {
                this.alto = alto;
        }
        public double getAncho() {
                return ancho;
        }
        public void setAncho(double ancho) {
                this.ancho = ancho;
        }
        public double getVelocidad() {
                return velocidad;
        }
        public void setVelocidad(double velocidad) {
                this.velocidad = velocidad;
        }
        public double getVida() {
                return vida;
        }
```

```
	public void setVida(double vida) {
		this.vida = vida;
	}
	public Image getImagen() {
		return imagen;
	}
	public void setImagen(Image imagen) {
		this.imagen = imagen;
	}
	public double getDaño() {
		return daño;
	}
	public void setDaño(double daño) {
		this.daño = daño;
	}
	public void dibujar(Entorno entorno) {
		Image imagenGnomo = Herramientas.cargarImagen("imagenes/gnomo.png");
		entorno.dibujarImagen(imagenGnomo, this.getX(), this.getY(), Math.toRadians(0),
0.05);
	}
	public void moverDerecha() {
		this.x = this.x + 0.7;
	}
	public void moverIzquierda() {
		this.x = this.x - 0.7;
	}
	public void moverAbajo() {
		this.y = this.y + 2;
	}
	public void setAterrizado(boolean aterrizado) {
		this.aterrizado = aterrizado;
	}
	public boolean estaAterrizado() {
		return aterrizado;
	}
	public void resetearAterrizado() {
		this.aterrizado = false;
	}
	public boolean isMovimientoDerecha() {
		return movimientoDerecha;
```

```
	}
	public void elegirDireccion() {
		double numeroRandom = Math.random();
		this.movimientoDerecha = numeroRandom >= 0.5;
	}
	public boolean dentroDelEntorno(Entorno entorno) {
		return this.y <= entorno.alto() && this.x <= entorno.ancho();
	}
	public boolean gnomoSobreIsla(ArrayList<Isla> islas) {
		for (Isla isla : islas) {
			if (this.x >= isla.getX() - isla.getAncho() / 2 && this.x <= isla.getX() +
isla.getAncho() / 2) {
				if (this.y + this.alto / 2 >= isla.getY() - isla.getAlto() / 2
					&& this.y + this.alto / 2 <= isla.getY() + isla.getAlto() /
2) {
					return true;
				}
			}
		}
		return false;
	}
}
```

ISLA

```
package juego;
import java.awt.Image;
import entorno.Entorno;
import entorno.Herramientas;
public class Isla {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private Image imagen;
	public Isla(double x, double y, double alto, double ancho, Image Imagen) {
		this.setX(x);
		this.setY(y);
```

```
                this.setAlto(alto);
                this.setAncho(ancho);
                this.setImagen(imagen);
        }
        public void dibujar(Entorno entorno) {
                Image imagenIsla = Herramientas.cargarImagen("imagenes/isla.png");
                entorno.dibujarImagen(imagenIsla, this.getX(), this.getY(), Math.toRadians(0), 0.2);
        }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x = x;
        }
        public double getY() {
                return y;
        }
        public void setY(double y) {
                this.y = y;
        }
        public double getAlto() {
                return alto;
        }
        public void setAlto(double alto) {
                this.alto = alto;
        }
        public double getAncho() {
                return ancho;
        }
        public void setAncho(double ancho) {
                this.ancho = ancho;
        }
        public Image getImagen() {
                return imagen;
        }
        public void setImagen(Image imagen) {
                this.imagen = imagen;
        }
}
```

ISLAS

```
package juego;
import java.util.ArrayList;
public class Islas {
	private ArrayList<Isla> islas;
	public Islas() {
		this.islas = new ArrayList<>();
		// Inicialización de las islas en el constructor
		islas.add(new Isla(400, 100, 30, 110, null));
		islas.add(new Isla(300, 200, 30, 110, null));
		islas.add(new Isla(500, 200, 30, 110, null));
		islas.add(new Isla(210, 300, 30, 110, null));
		islas.add(new Isla(400, 300, 30, 110, null));
		islas.add(new Isla(590, 300, 30, 110, null));
		islas.add(new Isla(137.5, 400, 30, 110, null));
		islas.add(new Isla(305, 400, 30, 110, null));
		islas.add(new Isla(490, 400, 30, 110, null));
		islas.add(new Isla(662.5, 400, 30, 110, null));
		islas.add(new Isla(50, 500, 30, 110, null));
		islas.add(new Isla(225, 500, 30, 110, null));
		islas.add(new Isla(400, 500, 30, 110, null));
		islas.add(new Isla(575, 500, 30, 110, null));
		islas.add(new Isla(750, 500, 30, 110, null));
	}
	// Método para obtener todas las islas
	public ArrayList<Isla> getIslas() {
		return islas;
	}
}
```

TORTUGA

```
package juego;
import java.awt.Image;
import java.util.ArrayList;
import entorno.Entorno;
import entorno.Herramientas;
public class Tortuga {
	private double x;
	private double y;
	private double alto;
	private double ancho;
	private double velocidad;
	private double vida;
	private double daño;
	private Image imagen;
	private boolean aterrizado;
	private boolean movimientoIzquierda;
	private double velocidadVertical;
	private double velocidadHorizontal;



	public Tortuga(double x, double y, double alto, double ancho, double velocidad, double vida,
double daño,
				Image imagen, boolean aterrizado, boolean movimientoIzquierda, double
velocidadvertical,double velocidadHorizontal) {
		super();
		this.x = x;
		this.y = y;
		this.alto = alto;
		this.ancho = ancho;
```

```
                this.velocidad = velocidad;
                this.vida = vida;
                this.daño = daño;
                this.imagen = imagen;
                this.aterrizado = aterrizado;
                this.movimientoIzquierda = movimientoIzquierda;
                this.velocidadVertical =  velocidadvertical;
                this.velocidadHorizontal = velocidadHorizontal;

        }
        // Dibuja la tortuga
public void dibujar(Entorno entorno) {
    if (imagen == null) {
        imagen = Herramientas.cargarImagen("imagenes/tortuga.png");
    }
    entorno.dibujarImagen(imagen, this.x, this.y, 0, 0.05);
}



        public boolean isMovimientoIzquierda() {
                return movimientoIzquierda;
        }
        public void setMovimientoIzquierda(boolean movimientoIzquierda) {
                this.movimientoIzquierda = movimientoIzquierda;
        }
        public boolean isAterrizado() {
                return aterrizado;
        }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x = x;
        }
        public double getY() {
                return y;
        }
        public void setY(double y) {
                this.y = y;
```

```
    }
    public double getAlto() {
        return alto;
    }
    public void setAlto(double alto) {
        this.alto = alto;
    }
    public double getAncho() {
        return ancho;
    }
    public void setAncho(double ancho) {
        this.ancho = ancho;
    }
    public double getVelocidad() {
        return velocidad;
    }
    public void setVelocidad(double velocidad) {
        this.velocidad = velocidad;
    }
    public double getVida() {
        return vida;
    }
    public void setVida(double vida) {
        this.vida = vida;
    }
    public double getDaño() {
        return daño;
    }
    public void setDaño(double daño) {
        this.daño = daño;
    }
    public Image getImagen() {
        return imagen;
    }
    public void setImagen(Image imagen) {
        this.imagen = imagen;
    }
```

```
public void setAterrizado(boolean aterrizado) {
	this.aterrizado = aterrizado;
}


public boolean estaAterrizado() {
	return aterrizado;
}


public void resetearAterrizado() {
	this.aterrizado = false;
}

public double getVelocidadVertical() {
	return velocidadVertical;
}

public void setVelocidadVertical(double velociadVertical) {
	this.velocidadVertical = velociadVertical;
}

public double getVelocidadHorizontal() {
	return velocidadHorizontal;
}
public void setVelocidadHorizontal(double velocidadHorizontal) {
	this.velocidadHorizontal = velocidadHorizontal;
}

private Tortuga getIsla() {
	// TODO Auto-generated method stub
	return null;
}
public void moverAbajo() {
	this.y = this.y + 2;
}
```

```
// DIRECCION
public void moverDerecha() {
        this.x = this.x + 0.5;
}
public void moverIzquierda() {
        this.x = this.x - 0.5;
}

public boolean isMovimientoIzquierda() {

        return movimientoIzquierda;
}

public void direccionTortuga() {
        double numeroRandom = Math.random();
        this.movimientoIzquierda = numeroRandom >= 0.5;

}



//resetea el estado de aterrizaje

public void resetaerAterrizado() {
        aterrizado = false;
}


//Verifica si la tortuga esta sobre una isla

public boolean tortugaSobreCasa(Casa casa) {
        if(this.x >= casa.getX() - casa.getAncho() /2-30 && this.x <= casa.getX() +
                        casa.getAncho()/2+30) {
                if (this.y + this.alto /2 >= casa.getY() - casa.getAlto() /2-40 &&
                                this.y + this.alto / 2 <= casa.getY() + casa.getAlto() / 2+40) {
  return true;
                }

        }
        return false;
```

```
        }

        public boolean tortugaSobreIsla(ArrayList<Isla> islas) {
                for (Isla isla : islas) {
                        if (this.x >= isla.getX() - isla.getAncho() / 2 && this.x <= isla.getX() +
isla.getAncho() / 2) {
                                if (this.y + this.alto / 2 >= isla.getY() - isla.getAlto() / 2
                                                && this.y + this.alto / 2 <= isla.getY() + isla.getAlto() /
2)
                        return true;
                    }
                }
                return false;
        }
        public boolean dentroDelEntorno(Entorno entorno) {
                return this.y <= entorno.alto() && this.x <= entorno.ancho()+10;
        }

        public boolean islaOcupada(ArrayList<Isla> islas, ArrayList<Tortuga> tortugas) {   // sin
terminar
                for (Isla isla : islas) {
                        for(Tortuga tortu : tortugas) {
                                if(tortu.tortugaSobreIsla(islas)) {
                                        return true;
                                }
                        }
                }
                return false;
        }

}
```

JUEGO

```
package juego;
import java.awt.Color;
import java.awt.Image;
import java.util.ArrayList;
import java.util.Random;
import entorno.Entorno;
import entorno.Herramientas;
import entorno.InterfaceJuego;
public class Juego extends InterfaceJuego {
	private Entorno entorno;
	private ArrayList<Gnomo> gnomos;
	private double tiempoGeneracion;
	private int intervaloGeneracion;
	private ArrayList<Tortuga> tortugas;
	private Random rand = new Random();
	private Pep pep;
	private Casa casa;
	private Islas islas;
	private boolean direcBola;
	private ArrayList<BolaFuego> disparos;
	private int gnomosSalvados;
	private int gnomosMuertos;
	private int tiempoJuego;
	private int tiempoTranscurrido;
	private int tiempoUltimaMuerte;
	private int tiempoUltimoSalvado;
	private boolean juegoTerminado = false;
	private boolean juegoGanado;
	private Escudo escudo;
	private int tortugasAsesinadas;
```

```
	private int nivel;
	private Roku roku;
	private Escudo escudo2;
	private boolean direcBola2;
	private ArrayList<BolaFuego> disparos2;
	private int finalizar;
	private int pepSalvados;
	private int rokuSalvados;
	private int pepAsesinatos;
	private int rokuAsesinatos;
	public void dibujarFondo() {
		Image imagenFondo = Herramientas.cargarImagen("imagenes/fondo.jpg");
		entorno.dibujarImagen(imagenFondo, 400, 300, Math.toRadians(0), 0.7);
	}
	private boolean hayColision(double x1, double y1, double ancho1, double alto1, double x2,
double y2, double ancho2,
			double alto2) {
		return x1 < x2 + ancho2 / 2 && x1 + ancho1 / 2 > x2 && y1 < y2 + alto2 / 2 && y1 +
alto1 / 2 > y2;
	}
	public void terminarJuego() {
		juegoTerminado = true;
	}
	Juego() {
		this.entorno = new Entorno(this, "Aldeco-Lauzan-Toledo-tp-p1", 800, 600);
		this.gnomos = new ArrayList<>();
		this.tortugas = new ArrayList<Tortuga>();
		this.casa = new Casa(400, 60, 50, 50, null);
		this.islas = new Islas();
		this.pep = new Pep(50, 460, 50, 30, 0, 3, 0, null);
		this.escudo = new Escudo(this.pep.getX(), this.pep.getY(), this.pep.getAlto(),
this.pep.getAncho(), 3, null);
		this.roku = new Roku(750, 460, 50, 30, 0, 3, 0, null);
		this.escudo2 = new Escudo(this.roku.getX(), this.roku.getY(), this.roku.getAlto(),
this.roku.getAncho(), 3,
				null);
		this.disparos = new ArrayList<>();
		this.disparos2 = new ArrayList<>();
		finalizar = 3;
		this.entorno.iniciar();
```

```
                this.tiempoGeneracion = 0;
                this.intervaloGeneracion = 2;
        }
        public void tick() {
                dibujarFondo();
                if (pep == null && roku == null) {
                        juegoGanado = false;
                        terminarJuego();
                }
                if (nivel == finalizar) {
                        juegoGanado = true;
                        terminarJuego();
                }
                if (juegoTerminado) {
                        if (juegoGanado) {
                                dibujarMenuFinal("Ganaste");
                        } else {
                                dibujarMenuFinal("Perdiste");
                        }
                        return;
                }
                entorno.cambiarFont("DialogInput", 26, Color.RED, 1);
                if (tortugasAsesinadas == 1 && (tiempoJuego - tiempoUltimaMuerte) < 3) {
                        entorno.escribirTexto("¡TORTUGA A LA PARRILLA!", 240, 140);
                } else if (tortugasAsesinadas == 5 && (tiempoJuego - tiempoUltimaMuerte) < 3) {
                        entorno.escribirTexto("¡SICARIO!", 320, 140);
                } else if (tortugasAsesinadas == 10 && (tiempoJuego - tiempoUltimaMuerte) < 3) {
                        entorno.escribirTexto("¡EL TERROR DE LAS TORTUGAS!", 205, 140);
                } else if (tortugasAsesinadas == 20 && (tiempoJuego - tiempoUltimaMuerte) < 3) {
                        entorno.escribirTexto("¡FURIA DESATADA!", 265, 140);
                }
                entorno.cambiarFont("DialogInput", 26, Color.CYAN, 1);
                if (gnomosSalvados == 1 && (tiempoJuego - tiempoUltimoSalvado) < 3) {
                        entorno.escribirTexto("¡GNOMO A SALVO!", 520, 170);
                } else if (gnomosSalvados == 5 && (tiempoJuego - tiempoUltimoSalvado) < 3) {
                        entorno.escribirTexto("¡HÉROE DE GNOMOS!", 500, 170);
                } else if (gnomosSalvados == 15 && (tiempoJuego - tiempoUltimoSalvado) < 3) {
                        entorno.escribirTexto("¡NIÑERA PROFESIONAL!", 460, 170);
                } else if (gnomosSalvados == 25 && (tiempoJuego - tiempoUltimoSalvado) < 3) {
                        entorno.escribirTexto("¡EL SALVADOR!", 540, 170);
```

```
                } else if (gnomosSalvados == nivel * 10 && (tiempoJuego - tiempoUltimoSalvado) < 3
&& nivel != 0) {
                        entorno.escribirTexto("¡SUBES DE NIVEL!", 540, 170);
                }
                entorno.cambiarFont("Arial", 18, Color.white);
                entorno.escribirTexto("Gnomos salvados: " + gnomosSalvados, 10, 25);
                entorno.escribirTexto("Gnomos muertos: " + gnomosMuertos, 630, 25);
                entorno.escribirTexto("Tortugas asesinadas: " + tortugasAsesinadas, 600, 50);
                entorno.escribirTexto("Nivel: " + nivel, 650, 75);
                entorno.escribirTexto("Tiempo de juego: " + tiempoJuego + " s", 10, 50);
                if (this.pep != null) {
                        entorno.escribirTexto("Vidas Pep: " + this.pep.getVida(), 10, 75);
                        entorno.escribirTexto("Escudos Pep: " + this.escudo.getVida(), 10, 100);
                }
                if (this.roku != null) {
                        entorno.escribirTexto("Vidas Roku: " + this.roku.getVida(), 10, 125);
                        entorno.escribirTexto("Escudos Roku: " + this.escudo2.getVida(), 10, 150);
                }
                this.casa.dibujar(entorno);
                for (Isla isla : this.islas.getIslas()) {
                        isla.dibujar(entorno);
                }
                tiempoTranscurrido++;
                if (tiempoTranscurrido >= 60) {
                        tiempoJuego++;
                        tiempoTranscurrido = 0;
                }
                // NIVELES
                if (gnomosSalvados % 10 == 0 && gnomosSalvados != nivel * 10) {
                        nivel++;
                        if (nivel != finalizar) {
                                try {
                                        Herramientas.play("sonidos/nivel2.wav");
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                }
                // PEP
                if (this.pep != null) {
```

```
                this.pep.dibujar(entorno);
                // ESCUDO
                this.escudo.setX(this.pep.getX());
                this.escudo.setY(this.pep.getY());
                this.escudo.setAlto(this.pep.getAlto());
                this.escudo.setAncho(this.pep.getAncho());
                if (entorno.estaPresionada('s') && this.escudo.getVida() > 0) {
                        this.escudo.dibujar(entorno);
                }
                if (entorno.seLevanto('s') && this.escudo.getVida() > 0) {
                        this.escudo.setVida(this.escudo.getVida() - 1);
                }
                // MOVILIDAD PEP
                if (entorno.estaPresionada('d') && this.pep.getX() + 10 <
this.entorno.ancho()) {
                        this.pep.moverDerecha();
                        direcBola = true;
                }
                if (entorno.estaPresionada('a') && this.pep.getX() - 10 > 0) {
                        this.pep.moverIzquierda();
                        direcBola = false;
                }
                if (this.pep.pepSobreIsla(this.islas.getIslas()) == false &&
this.pep.dentroDelEntorno(entorno)) {
                        this.pep.moverAbajo();
                }
                if (entorno.sePresiono('w') && this.pep.pepSobreIsla(this.islas.getIslas())) {
                        this.pep.iniciarSalto();
                }
                this.pep.actualizarSalto(this.islas.getIslas());
                // muerte por limite
                if (!this.pep.dentroDelEntorno(entorno)) {
                        try {
                                Herramientas.play("sonidos/muerte.wav");
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                        if (this.pep.getVida() > 1) {
                                this.pep.setVida(this.pep.getVida() - 1);
                                this.pep.setX(400);
```

```
                                        this.pep.setY(1);
                                } else {
                                        this.pep = null;
                                        try {
                                                Herramientas.play("sonidos/perder2.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                        }
                        // DISPARO PEP
                        if (entorno.sePresiono('v') ||
entorno.sePresionoBoton(entorno.BOTON_IZQUIERDO)) {
                                disparos.add(new BolaFuego(this.pep.getX(), this.pep.getY() +
this.pep.getAlto() / 4, 50, 30, null,
                                                direcBola));
                                try {
                                        Herramientas.play("sonidos/disparo.wav");
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                        // BOLA DE FUEGO
                        for (int i = 0; i < disparos.size(); i++) {
                                BolaFuego bola = disparos.get(i);
                                if (bola != null) {
                                        // MOVILIDAD BOLA
                                        if (bola.isDirec()) {
                                                bola.moverDerecha();
                                        }
                                        if (!bola.isDirec()) {
                                                bola.moverIzquierda();
                                        }
                                        bola.dibujar(entorno);
                                        if (!bola.dentroDelEntorno(entorno)) {
                                                bola = null;
                                                disparos.remove(i);
                                                i--;
                                        }
                                }
```

```
                        }
                }
                // ROKU
                if (this.roku != null) {
                        this.roku.dibujar(entorno);
                        // ESCUDO2
                        this.escudo2.setX(this.roku.getX());
                        this.escudo2.setY(this.roku.getY());
                        this.escudo2.setAlto(this.roku.getAlto());
                        this.escudo2.setAncho(this.roku.getAncho());
                        if (entorno.estaPresionada(entorno.TECLA_ABAJO) &&
this.escudo2.getVida() > 0) {
                                this.escudo2.dibujar(entorno);
                        }
                        if (entorno.seLevanto(entorno.TECLA_ABAJO) && this.escudo2.getVida() >
0) {
                                this.escudo2.setVida(this.escudo2.getVida() - 1);
                        }
                        // MOVILIDAD ROKU
                        if (entorno.estaPresionada(entorno.TECLA_DERECHA) && this.roku.getX() +
10 < this.entorno.ancho()) {
                                this.roku.moverDerecha();
                                direcBola2 = true;
                        }
                        if (entorno.estaPresionada(entorno.TECLA_IZQUIERDA) && this.roku.getX()
- 10 > 0) {
                                this.roku.moverIzquierda();
                                direcBola2 = false;
                        }
                        if (this.roku.rokuSobreIsla(this.islas.getIslas()) == false &&
this.roku.dentroDelEntorno(entorno)) {
                                this.roku.moverAbajo();
                        }
                        if (entorno.sePresiono(entorno.TECLA_ARRIBA) &&
this.roku.rokuSobreIsla(this.islas.getIslas())) {
                                this.roku.iniciarSalto();
                        }
                        this.roku.actualizarSalto(this.islas.getIslas());
                        // muerte por limite
                        if (!this.roku.dentroDelEntorno(entorno)) {
```

```
                                try {
                                        Herramientas.play("sonidos/muerte.wav");
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                                if (this.roku.getVida() > 1) {
                                        this.roku.setVida(this.roku.getVida() - 1);
                                        this.roku.setX(400);
                                        this.roku.setY(1);
                                } else {
                                        this.roku = null;
                                        try {
                                                Herramientas.play("sonidos/perder.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                        }
                        // DISPARO ROKU
                        if (entorno.sePresiono('l') ||
entorno.sePresionoBoton(entorno.BOTON_DERECHO)) {
                                disparos2.add(new BolaFuego(this.roku.getX(), this.roku.getY() +
this.roku.getAlto() / 4, 50, 30, null,
                                                direcBola2));
                                try {
                                        Herramientas.play("sonidos/disparo.wav");
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                        // BOLA DE FUEGO 2
                        for (int i = 0; i < disparos2.size(); i++) {
                                BolaFuego bola2 = disparos2.get(i);
                                if (bola2 != null) {
                                        // MOVILIDAD BOLA
                                        if (bola2.isDirec()) {
                                                bola2.moverDerecha();
                                        }
                                        if (!bola2.isDirec()) {
                                                bola2.moverIzquierda();
```

```
                                        }
                                        bola2.dibujar(entorno);
                                        if (!bola2.dentroDelEntorno(entorno)) {
                                                bola2 = null;
                                                disparos2.remove(i);
                                                i--;
                                        }
                                }
                        }
                }
                // GNOMO
                tiempoGeneracion += 1.0 / 60;
                if (tiempoGeneracion >= intervaloGeneracion) {
                        if (gnomos.size() < 4) {
                                gnomos.add(new Gnomo(400, 60, 55, 95, 0, 0, 0, null, false));
                        }
                        tiempoGeneracion = 0;
                }
                if (gnomos.size() < 2) {
                        gnomos.add(new Gnomo(400, 60, 55, 95, 0, 0, 0, null, false));
                }
                for (int i = 0; i < gnomos.size(); i++) {
                        Gnomo gnomo = gnomos.get(i);
                        if (gnomo != null) {
                                gnomo.dibujar(entorno);
                                // MOVILIDAD GNOMO
                                if (gnomo.gnomoSobreIsla(this.islas.getIslas())) {
                                        if (!gnomo.estaAterrizado()) {
                                                gnomo.elegirDireccion();
                                                gnomo.setAterrizado(true);
                                        }
                                        if (gnomo.isMovimientoDerecha()) {
                                                gnomo.moverDerecha();
                                        } else {
                                                gnomo.moverIzquierda();
                                        }
                                } else {
                                        gnomo.moverAbajo();
                                        gnomo.resetearAterrizado();
                                        if (!gnomo.dentroDelEntorno(entorno)) {
```

```
                                        gnomo = null;
                                        gnomos.remove(i);
                                        i--;
                                }
                        }
                }
        }
        // TORTUGA
        tiempoGeneracion += 1 / 60.0;
        if (tiempoGeneracion >= intervaloGeneracion) {
                if (tortugas.size() < nivel * 2 + 2) {
                        int x = rand.nextInt(entorno.getWidth());
                        tortugas.add(new Tortuga(x, 0, 35, 50, 0, 0, 0, null, false, true, 2, 2));
                }
                tiempoGeneracion = 0;
        }
        for (int i = 0; i < tortugas.size(); i++) {
                Tortuga tortuga = tortugas.get(i);
                if (tortuga != null) {
                        tortuga.dibujar(entorno);
                        // MOVILIDAD TORTUGA
                        if (!tortuga.estaAterrizado()) {
                                tortuga.moverAbajo();
                                tortuga.resetearAterrizado();
                        }
                        if (tortuga.tortugaSobreIsla(this.islas.getIslas())
                                        && !tortuga.tortugaSobreCasa(casa) /* &&
!tortuga.islaOcupada(islas, tortugas) */) {
                                tortuga.setAterrizado(true);
                                if (tortuga.isMovimientoIzquierda()) {
                                        tortuga.moverIzquierda();
                                } else {
                                        tortuga.moverDerecha();
                                }
                                if (!tortuga.tortugaSobreIsla(this.islas.getIslas())) { // si se va
de limites de isla vuelve a

                                                // entrar con direccion
                                        // opuesta
                                        tortuga.moverDerecha();
```

```
                                        tortuga.setMovimientoIzquierda(false);
                                        if (!tortuga.tortugaSobreIsla(this.islas.getIslas())) {
                                                tortuga.moverIzquierda();
                                                tortuga.moverIzquierda();
                                                tortuga.setMovimientoIzquierda(true);
                                        }
                                }
                                if (!tortuga.dentroDelEntorno(entorno)) {
                                        tortuga = null;
                                        tortugas.remove(i);
                                        i--;
                                }
                        }
                }
        }
//              COLISION 1 PEP
        for (int j = 0; j < gnomos.size(); j++) {
                Gnomo gnomo = gnomos.get(j);
                for (int i = 0; i < tortugas.size(); i++) {
                        Tortuga tortuga = tortugas.get(i);
                        // rescate pep
                        if (this.pep != null && this.pep.getY() > 300 && gnomo != null
                                        && hayColision(this.pep.getX(), this.pep.getY(),
this.pep.getAncho(), this.pep.getAlto(),
                                                        gnomo.getX(), gnomo.getY(),
gnomo.getAncho(), gnomo.getAlto())) {
                                tiempoUltimoSalvado = tiempoJuego;
                                pepSalvados++;
                                gnomosSalvados++;
                                if (gnomosSalvados == finalizar * 10) {
                                        try {
                                                Herramientas.play("sonidos/ganar.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                                if (gnomosSalvados == 1 || gnomosSalvados == 5 ||
gnomosSalvados == 15 || gnomosSalvados == 25) {
                                        try {
```

```
Herramientas.play("sonidos/logrosalva.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                } else {
                                        try {
                                                Herramientas.play("sonidos/salvado.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                                gnomo = null;
                                gnomos.remove(j);
                                j--;
                                if (this.escudo.getVida() < 3) {
                                        this.escudo.setVida(this.escudo.getVida() + 1);
                                }
                        }
                        // rescate roku
                        if (this.roku != null && this.roku.getY() > 300 && gnomo != null
                                        && hayColision(this.roku.getX(), this.roku.getY(),
this.roku.getAncho(), this.roku.getAlto(),
                                                        gnomo.getX(), gnomo.getY(),
gnomo.getAncho(), gnomo.getAlto())) {
                                tiempoUltimoSalvado = tiempoJuego;
                                rokuSalvados++;
                                gnomosSalvados++;
                                if (gnomosSalvados == finalizar * 10) {
                                        try {
                                                Herramientas.play("sonidos/ganar.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                                if (gnomosSalvados == 1 || gnomosSalvados == 5 ||
gnomosSalvados == 15 || gnomosSalvados == 25) {
                                        try {

Herramientas.play("sonidos/logrosalva.wav");
```

```
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                } else {
                                        try {
                                                Herramientas.play("sonidos/salvado.wav");
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                                gnomo = null;
                                gnomos.remove(j);
                                j--;
                                if (this.escudo2.getVida() < 3) {
                                        this.escudo2.setVida(this.escudo2.getVida() + 1);
                                }
                        }
                        // muerte pep
                        if (this.pep != null && tortuga != null
                                        && !((entorno.estaPresionada('s') ||
entorno.estaPresionada(entorno.TECLA_ABAJO))
                                                        && this.escudo.getVida() > 0)
                                        && hayColision(this.pep.getX(), this.pep.getY(),
this.pep.getAncho(), this.pep.getAlto(),
                                                        tortuga.getX(), tortuga.getY(),
tortuga.getAncho(), tortuga.getAlto())) {
                                try {
                                        Herramientas.play("sonidos/muerte.wav");
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                                if (this.pep.getVida() > 1) {
                                        this.pep.setVida(this.pep.getVida() - 1);
                                        this.pep.setX(400);
                                        this.pep.setY(1);
                                } else {
                                        this.pep = null;
                                        try {
                                                Herramientas.play("sonidos/perder2.wav");
                                        } catch (Exception e) {
```

```
                                        e.printStackTrace();
                                    }
                                }
                            }
                            // muerte roku
                            if (this.roku != null && tortuga != null
                                    && !((entorno.estaPresionada('s') ||
entorno.estaPresionada(entorno.TECLA_ABAJO))
                                            && this.escudo2.getVida() > 0)
                                    && hayColision(this.roku.getX(), this.roku.getY(),
this.roku.getAncho(), this.roku.getAlto(),
                                            tortuga.getX(), tortuga.getY(),
tortuga.getAncho(), tortuga.getAlto())) {
                                try {
                                    Herramientas.play("sonidos/muerte.wav");
                                } catch (Exception e) {
                                    e.printStackTrace();
                                }
                                if (this.roku.getVida() > 1) {
                                    this.roku.setVida(this.roku.getVida() - 1);
                                    this.roku.setX(400);
                                    this.roku.setY(1);
                                } else {
                                    this.roku = null;
                                    try {
                                        Herramientas.play("sonidos/perder.wav");
                                    } catch (Exception e) {
                                        e.printStackTrace();
                                    }
                                }
                            }
                            // muerte gnomo
                            if (gnomo != null && tortuga != null && hayColision(gnomo.getX(),
gnomo.getY(), gnomo.getAncho(),
                                    gnomo.getAlto(), tortuga.getX(), tortuga.getY(),
tortuga.getAncho(), tortuga.getAlto())) {
                                gnomosMuertos++;
                                gnomo = null;
                                gnomos.remove(j);
                                j--;
```

```
                                        }
                                }
                        }
                        // COLISION 2
                        for (int i = 0; i < tortugas.size(); i++) {
                                Tortuga tortuga = tortugas.get(i);
                                for (int w = 0; w < disparos.size(); w++) {
                                        BolaFuego bola = disparos.get(w);
                                        // muerte tortuga
                                        if (bola != null && tortuga != null && hayColision(bola.getX(),
bola.getY(), bola.getAncho(),
                                                        bola.getAlto(), tortuga.getX(), tortuga.getY(),
tortuga.getAncho(), tortuga.getAlto())) {
                                                pepAsesinatos++;
                                                tortugasAsesinadas++;
                                                tiempoUltimaMuerte = tiempoJuego;
                                                if (tortugasAsesinadas == 1 || tortugasAsesinadas == 5 ||
tortugasAsesinadas == 10
                                                                || tortugasAsesinadas == 20) {
                                                        try {
                                                                Herramientas.play("sonidos/logro2.wav");
                                                        } catch (Exception e) {
                                                                e.printStackTrace();
                                                        }
                                                } else {
                                                        try {

Herramientas.play("sonidos/asesinato2.wav");
                                                        } catch (Exception e) {
                                                                e.printStackTrace();
                                                        }
                                                }
                                                tortuga = null;
                                                tortugas.remove(i);
                                                i--;
                                                bola = null;
                                                disparos.remove(w);
                                                w--;
                                        }
                                }
```

```
                        for (int j = 0; j < disparos2.size(); j++) {
                                BolaFuego bola2 = disparos2.get(j);
                                // muerte tortuga
                                if (bola2 != null && tortuga != null && hayColision(bola2.getX(),
bola2.getY(), bola2.getAncho(),
                                                bola2.getAlto(), tortuga.getX(), tortuga.getY(),
tortuga.getAncho(), tortuga.getAlto())) {
                                        rokuAsesinatos++;
                                        tortugasAsesinadas++;
                                        tiempoUltimaMuerte = tiempoJuego;
                                        if (tortugasAsesinadas == 1 || tortugasAsesinadas == 5 ||
tortugasAsesinadas == 10
                                                        || tortugasAsesinadas == 20) {
                                                try {
                                                        Herramientas.play("sonidos/logro2.wav");
                                                } catch (Exception e) {
                                                        e.printStackTrace();
                                                }
                                        } else {
                                                try {

Herramientas.play("sonidos/asesinato2.wav");
                                                } catch (Exception e) {
                                                        e.printStackTrace();
                                                }
                                        }
                                        tortuga = null;
                                        tortugas.remove(i);
                                        i--;
                                        bola2 = null;
                                        disparos2.remove(j);
                                        j--;
                                }
                        }
                }
        }
        public void dibujarMenuFinal(String mensajeFinal) {
                Color fondoTransparente = new Color(0, 0, 0, 150);
                entorno.dibujarRectangulo(400, 300, 800, 600, 0, fondoTransparente);
                if (juegoGanado) {
```

```
                        entorno.cambiarFont("Arial", 36, Color.GREEN);
                } else {
                        entorno.cambiarFont("Arial", 36, Color.RED);
                }
                entorno.escribirTexto(mensajeFinal, 310, 200);
                entorno.cambiarFont("Arial", 24, Color.WHITE);
                entorno.escribirTexto("Tiempo jugado: " + tiempoJuego + " segundos", 250, 250);
                entorno.escribirTexto("Gnomos salvados: " + gnomosSalvados, 250, 290);
                entorno.escribirTexto("Gnomos muertos: " + gnomosMuertos, 250, 330);
                entorno.escribirTexto("Tortugas asesinadas: " + tortugasAsesinadas, 250, 370);
                entorno.escribirTexto("Nivel: " + nivel, 250, 410);
                entorno.escribirTexto("PEP: ", 50, 250);
                entorno.escribirTexto(pepSalvados + " SALVADAS ", 50, 300);
                entorno.escribirTexto(pepAsesinatos + " ASESINATOS ", 50, 350);
                entorno.escribirTexto("ROKU : ", 600, 250);
                entorno.escribirTexto(rokuSalvados + " SALVADAS ", 600, 300);
                entorno.escribirTexto(rokuAsesinatos + " ASESINATOS ", 600, 350);
                entorno.escribirTexto("Presioná 'R' para reiniciar o 'Q' para salir", 180, 470);
                if (entorno.sePresiono('r')) {
                        reiniciarJuego();
                } else if (entorno.sePresiono('q')) {
                        System.exit(0);
                }
        }
        public void reiniciarJuego() {
                juegoTerminado = false;
                tiempoJuego = 0;
                gnomos.clear();
                tortugas.clear();
                disparos.clear();
                disparos2.clear();
                gnomosSalvados = 0;
                gnomosMuertos = 0;
                tortugasAsesinadas = 0;
                nivel = 0;
                pep = new Pep(50, 460, 50, 30, 0, 3, 0, null);
                roku = new Roku(750, 460, 50, 30, 0, 3, 0, null);
                escudo = new Escudo(this.pep.getX(), this.pep.getY(), this.pep.getAlto(),
this.pep.getAncho(), 3, null);
```

```
                escudo2 = new Escudo(this.roku.getX(), this.roku.getY(), this.roku.getAlto(),
this.roku.getAncho(), 3, null);
                rokuAsesinatos = 0;
                pepAsesinatos = 0;
                pepSalvados = 0;
                rokuSalvados = 0;
        }
        public static void main(String[] args) {
                Juego juego = new Juego();
        }
}
```