## Clases del juego

* BolaDeFuego
* Cofre
* Moneda
* Caballero
* Esqueleto
* Isla
* Juego

### Características de las clases

A excepción de la clase Juego, todas las clases poseen los 4 mismos atributos. X, Y, Alto, Ancho. Necesarias para determinar la posición de los distintos objetos, así como para detectar las colisiones entre ellos.

Las clases Esqueleto y Caballero poseen muchas similitudes entre ellas. Además de los ya mencionados atributos, propios de cada clase. Ambas poseen los métodos **tocaAbajo()**. Los cuales reciben un objeto de tipo isla por parámetro, y determina si se esta colisionando con él, a través de un valor booleano.

public boolean tocaAbajo(Isla isla){

        boolean tocaX = this.x >= isla.getX() - isla.getAncho()/2 && this.x <= isla.getX() + isla.getAncho()/2;

        boolean tocaY = this.y + this.alto/2 == isla.getY() - isla.getAlto()/2;

        return tocaY && tocaX;

    }

Este método tiene además, una variante reservada para el array que contiene todas las islas del escenario. El mismo es el método **estaTocandoAlgunaIsla()**. Teniendo una versión de este método para el caballero **caballeroEstaTocandoAlgunaIsla()**, y otra para el esqueleto **esqueletoEstaTocandoAlgunaIsla().** Ambos métodos reciben por parámetro un array con las islas del juego, y a través de un for y del propio método tocaAbajo(); determina si se esta tocando alguna isla presente en el array. De ser así, el método devuelve la isla que se tocó, y en caso de que no se toque ninguna, devuelve null.

public Isla esqueletoEstaTocandoAlgunaIsla(Isla[] islas){

        for (int i = 0; i < islas.length; i++) {

            Isla isla = islas[i];

            if(esqueleto.tocaAbajo(isla)){

                return isla;

            }

        }

        return null;

    }

public Isla caballeroEstaTocandoAlgunaIsla(Isla[] islas){

        for (int i = 0; i < islas.length; i++) {

            Isla isla = islas[i];

            if(caballero.tocaAbajo(isla)){

                return isla;

            }

        }

        return null;

    }

La razón de dividir este método en dos, es simplemente para evitar posibles bugs al estar comparando el estado de dos objetos diferentes en un mismo método.

### Algunos otros métodos de la clase Esqueleto

El método **mover()** que no hace mas que incrementar el valor de x del esqueleto, lo que hace que avanze, por defecto, hacia la derecha.

public void mover(){

        this.x = this.x + this.velocidad;

    }

El método **cambiarDireccion(),** que es utilizado cuando se detecta el borde de una isla, para cambiar la dirección en la que se mueve el esqueleto y así evitar que caiga al vacío (Mas información sobre esto en el apartado de la clase Juego).

public void cambiarDireccion(){

        this.imagenEsqueleto = Herramientas.cargarImagen("images/esqueleto1.jpg");

        this.velocidad = this.velocidad \*(-1);

    }

### Algunos otros métodos de la clase Caballero

El caballero, a diferencia de la clase Esqueleto, posee un método para moverse en cada dirección (izquierda o derecha). Incrementando o decrementando el valor de X para hacerlo. A esto se le suma el método **saltar(),** que siguiendo la misma lógica, decrementa el valor de Y para dar la ilusión de salto en el personaje.

 public void moverDerecha(){

        this.x = this.x + velocidad;

    }

    public void moverIzquierda(){

        this.x = this.x - velocidad;

    }

    public void saltar(){

        this.y = this.y - velocidad\*2;

    }

## Clase juego

### Algunos métodos utilizados

Entre los métodos utilizado en la clase Juego, se encuentran los anteriormente mencionados **esqueletoTocaAlgunaIsla()** y **caballeroTocaAlgunaIsla**. También se encuentra el método **generarIslas()** que no hace mas que instanciar los 15 objeto de tipo islas que conforman el escenario. Adicionalmente, está el método **dibujarIslas()**, el cual a través de un ciclo for, recorre el array con las islas y las dibuja una por una.

public void generarIslas(){

        islas[0] = new Isla(100, 500, 100, 30);

        islas[1] = new Isla(250, 500, 100, 30);

        islas[2] = new Isla(400, 500, 100, 30);

        islas[3] = new Isla(550, 500, 100, 30);

        islas[4] = new Isla(700, 500, 100, 30);

        islas[5] = new Isla(175, 400, 100, 30);

        islas[6] = new Isla(325, 400, 100, 30);

        islas[7] = new Isla(475, 400, 100, 30);

        islas[8] = new Isla(625, 400, 100, 30);

        islas[9] = new Isla(250, 300, 100, 30);

        islas[10] = new Isla(400, 300, 100, 30);

        islas[11] = new Isla(550, 300, 100, 30);

        islas[12] = new Isla(325, 200, 100, 30);

        islas[13] = new Isla(475, 200, 100, 30);

        islas[14] = new Isla(400, 100, 100, 30);

    }

    public void dibujarIslas(Isla[] isla){

        for (int i = 0; i < isla.length; i++) {

            islas[i].dibujarIsla(entorno);

        }

    }

### Método tick()

El método **tick()** es el encargado de controlar el tiempo transcurrido en el juego. A grandes rasgos, funciona como un bucle while constante mente abierto. Dentro de este método esta toda a la lógica referente al comportamiento de los distintos objetos dentro del escenario.

### Movimiento del esqueleto en la isla:

El razonamiento inicial para programar esta parte fue el siguiente. El esqueleto tendrá como movimiento por defecto, caer, salvo que toque una isla. Si toca una isla el esqueleto empezara a moverse hacia la derecha, hasta que toque el borde de la isla, lo que hará que se empiece a mover hacia la izquierda, hasta que toque el borde de nuevo. Y así sucesivamente. Para ello se crean dos booleanos que comprueban si se tocan los bordes de la isla

boolean tocaDerecha = esqueleto.getX() + esqueleto.getAncho()/2 == isla.getX() + isla.getAncho()/2;

boolean tocaIzquierda = esqueleto.getX() - esqueleto.getAncho()/2 == isla.getX() - isla.getAncho()/2;

A su vez la parte encargada del movimiento del esqueleto en la isla es el siguiente.

if(esqueleto.tocaAbajo(isla) != true){

            esqueleto.caer();

        }else{

            if (tocaDerecha == false) {

                esqueleto.moverDerecha();

            }else {

                if(tocaIzquierda == false) {

                    esqueleto.moverIzquierda();

            }

        }

Sin embargo esto provoca que al llegar al borde de la isla, el esqueleto empiece a solapar los métodos **moverDerecha()** y **moverIzquierda()**, haciendo que el esqueleto se bugue en el borde de la isla moviéndose de izquierda a derecha pero en la misma posición.

Lo primero que se hace es cambiar los métodos **moverDerecha()** y **moverIzquierda(),** por los métodos **mover()** y **cambiarDireccion().** Sin embargo el error persiste, incluso cambiando la estructura de los condicionales. Esto debido a la manera que tiene de procesar el tiempo el método **tick(),** que actúa como si se estuviesen ejecutando todas las acciones en bucle.

Luego de varios intentos, se da con la estructura condicional que funciona

if(esqueleto.tocaAbajo(isla)){

            if(tocaDerecha || tocaIzquierda){

                esqueleto.cambiarDireccion();

                esqueleto.mover(); //Si no se añade esta línea, al tocar un borde, el esqueleto se queda completamente quieto en su lugar.

            }else{

                esqueleto.mover();

            }

        }else{

            esqueleto.caer();

        }

En esta parte del código resulta fundamental el buen uso de los if y los elses para determinar el final de cada acción y el inicio de la siguiente.

Para empezar el método **caer()**, ya no se mantiene ejecutándose siempre, si no que únicamente lo hace cuando no se toca una isla. Por otro lado, y siguiendo la misma lógica, el método **mover(),** únicamente se ejecutara si no se toca ningún borde de la isla.

### Movimiento del caballero

El movimiento del caballero, sigue la misma lógica que la del esqueleto. Con la diferencia de que los movimientos los realiza el jugador a través del teclado.

El movimiento por defecto del caballero es la de caer, a excepción de que colisione con alguna isla, en cuyo caso se detendrá su caída.

if(caballeroEstaTocandoAlgunaIsla(islas)==null){

            caballero.caer();

            if(entorno.estaPresionada(entorno.TECLA\_ESPACIO)){

                caballero.saltar();

            }

        }

        if(estaPresionadaDerecha){

            caballero.moverDerecha();

        }

        if (estaPresionadaIzquierda) {

            caballero.moverIzquierda();

        }

        if (entorno.estaPresionada(entorno.TECLA\_ESPACIO)) {

            caballero.saltar();

        }

### Disparo de la bola de fuego

Primero a través de una condicional se corrobora que se haya presionado la tecla ctrl. De ser asi, se instanciará el objeto boladefuego, el cual es creado con anterioridad dentro de la clase Juego. Este objeto aparece en las mismas coordenadas que el caballero.

Además se cambia a true el booleano confirmar. El cual servirá mas adelante para accionar el evento encargado de disparar como tal la bola de fuego

if(entorno.sePresiono(entorno.TECLA\_CTRL)){

            confirmar = true;

            boladefuego = new BolaDeFuego(caballero.getX(), caballero.getY(),10, 10, 10);

        }

Otra condicional dentro del método tick(), se encarga de verificar que el booleano confirmar sea igual a true, para así dibujar la bola de fuego, y mover la bola de fuego hasta que colisione con el borde de la pantalla, o hasta que se dispare otra bola de fuego.

Si en la trayectoria, la bola de fuego toca las coordenadas x e y de algún esqueleto, cambiara su estado a null. Es decir, lo “eliminara”.

La razón de utilizar un booleano para disparar este evento, es que al método tick() funcionar como un ciclo while abierto, eso significa que todas las líneas de código se mantendrán ejecutándose, por lo que será necesario estar presionando todo el tiempo la tecla ctrl para que la bola de fuego se dibuje. Al controlar el evento por el booleano, permitimos que con accionar una única vez la tecla ctrl, la bola d efuego se cree, se dibuje y se mueva hasta que colisione con algún objeto o salga de la pantalla.

if(confirmar == true){

            if(boladefuego.x <= entorno.ancho()){

                boladefuego.dibujarBolaDeFuego(entorno);

                boladefuego.mover();

            }

            if(boladefuego.x == esqueleto.getX() && boladefuego.y == esqueleto.getY()){

                esqueleto = null;

            }

        }

# Codigo completo

## Clase caballero

package juego;

import java.awt.Color;

import java.awt.Image;

import entorno.Entorno;

import entorno.Herramientas;

public class Caballero {

    public BolaDeFuego boladefuego;

    private Image imagenCaballero;

    public int alto,ancho;

    private int x,y;

    private int velocidad;

    public Caballero(int x, int y, int velocidad,int ancho, int alto){

        this.x = x;

        this.y = y;

        this.velocidad = velocidad;

        this.ancho = ancho;

        this.alto = alto;

        this.imagenCaballero = Herramientas.cargarImagen("images/caballero.png");

    }

    public void moverDerecha(){

        this.x = this.x + velocidad;

    }

    public void moverIzquierda(){

        this.x = this.x - velocidad;

    }

    public void saltar(){

        this.y = this.y - velocidad\*2;

    }

    public void caer(){

        this.y = this.y + velocidad\*2;

    }

    public boolean tocaAbajo(Isla isla){

        boolean tocaX = this.x >= isla.getX() - isla.getAncho()/2 && this.x <= isla.getX() + isla.getAncho()/2;

        boolean tocaY = this.y + this.alto/2 == isla.getY() - isla.getAlto()/2;

        return tocaY && tocaX;

    }

    public void dibujarCaballero(Entorno entorno){

        entorno.dibujarRectangulo(x, y, ancho, alto, 0, Color.RED);

        entorno.dibujarImagen(this.imagenCaballero, this.x, this.y, 0, 0.1);

    }

    ////////Setter y Getter///////

    //Getters

    public int getX(){

        return this.x;

    }

    public int getY(){

        return this.y;

    }

    public int getVelocidad(){

        return this.velocidad;

    }

    public Image getImagenCaballero(){

        return this.imagenCaballero;

    }

    //Setters

    public void setX(int x){

        this.x = x;

    }

    public void setY(int y){

        this.y = y;

    }

    public void setVelocidad(int velocidad){

        this.velocidad = velocidad;

    }

}

## Clase esqueleto

package juego;

import java.awt.Color;

import java.awt.Image;

import entorno.Entorno;

import entorno.Herramientas;

public class Esqueleto {

    private int x,y;

    private int ancho,alto;

    private int velocidad;

    Image imagenEsqueleto;

    public Esqueleto(int x, int y, int ancho, int alto, int velocidad){

        this.x = x;

        this.y = y;

        this.ancho = ancho;

        this.alto = alto;

        this.velocidad = velocidad;

        this.imagenEsqueleto = Herramientas.cargarImagen("images/esqueleto.png");

    }

    public void saltar(){

        this.y = this.y - velocidad;

    }

    public void caer(){

        this.y = this.y + velocidad;

    }

    public void mover(){

        this.x = this.x + this.velocidad;

    }

    public void cambiarDireccion(){

        this.imagenEsqueleto = Herramientas.cargarImagen("images/esqueleto1.jpg");

        this.velocidad = this.velocidad \*(-1);

    }

    public boolean tocaAbajo(Isla isla){

        boolean tocaX = this.x >= isla.getX() - isla.getAncho()/2 && this.x <= isla.getX() + isla.getAncho()/2;

        boolean tocaY = this.y + this.alto/2 == isla.getY() - isla.getAlto()/2;

        return tocaY && tocaX;

    }

    public void dibujarEsqueleto(Entorno entorno){

        entorno.dibujarRectangulo(x, y, ancho, alto, 0, Color.WHITE);

    }

    //Setter y Getters

    //Getters

    public int getX(){

        return this.x;

    }

    public int getY(){

        return this.y;

    }

    public int getVelocidad(){

        return this.velocidad;

    }

    public int getAncho(){

        return this.ancho;

    }

    public int getAlto(){

        return this.alto;

    }

    //Setters

    public void setX(int x){

        this.x = x;

    }

    public void setY(int y){

        this.y = y;

    }

    public void setVelocidad(int velocidad){

        this.velocidad = velocidad;

    }

}

## Clase Isla

package juego;

import java.awt.Color;

import java.awt.Image;

import entorno.Entorno;

import entorno.Herramientas;

public class Isla {

    private Image imagenIsla;

    private int x,y;

    private int ancho, alto;

    Image isla;

    public Isla(int x, int y, int ancho, int alto){

        this.x = x;

        this.y = y;

        this.ancho = ancho;

        this.alto = alto;

        this.imagenIsla = Herramientas.cargarImagen("images/isla.png");

    }

    public void dibujarIsla(Entorno entorno){

        entorno.dibujarRectangulo(x, y, ancho, alto, 0, Color.CYAN);

        //entorno.dibujarImagen(this.imagenIsla, this.x, this.y, 0, 0.6);

    }

    //Metodos getter

    public int getX(){

        int x = this.x;

        return x;

    }

    public int getY(){

        int y = this.y;

        return y;

    }

    public Image getImagen(){

        return this.imagenIsla;

    }

    public int getAncho(){

        return this.ancho;

    }

    public int getAlto(){

        return this.alto;

    }

}

## Clase BolaDeFuego

package juego;

import java.awt.Color;

import java.awt.Image;

import entorno.Entorno;

import entorno.Herramientas;

public class BolaDeFuego {

    Image imagenboladefuego = Herramientas.cargarImagen("images/boladefuego1.gif");

    int x,y;

    int ancho,alto;

    int velocidad;

    public BolaDeFuego(int x, int y, int ancho, int alto, int velocidad){

        this.x = x;

        this.y = y;

        this.ancho = ancho;

        this.alto = alto;

        this.velocidad = velocidad;

        //this.imagenboladefuego = entorno.Herramientas.cargarImagen("src/images/boladefuego.gif");

    }

    public void dibujarBolaDeFuego(Entorno entorno){

        entorno.dibujarRectangulo(this.x, this.y, this.ancho, this.alto, 0, Color.ORANGE);

        entorno.dibujarImagen(imagenboladefuego, x, y, 0, 0.2);

    }

    public void mover(){

        this.x = this.x + velocidad;

    }

}

## Clase Juego

package juego;

import entorno.Entorno;

import entorno.InterfaceJuego;

public class Juego extends InterfaceJuego

{

    // El objeto Entorno que controla el tiempo y otros

    private Entorno entorno;

    // Variables y métodos propios de cada grupo

    Cofre cofre;

    public BolaDeFuego boladefuego;

    public Isla[] islas = new Isla[15];

    public Caballero caballero;

    public Esqueleto esqueleto;

    boolean confirmar = false;

    int yCaballero = 0;

    Juego()

    {

        // Inicializa el objeto entorno

        this.entorno = new Entorno(this, "Al Rescate de los Gnomos", 800, 600);

        // Inicializar lo que haga falta para el juego

        generarIslas();

        cofre = new Cofre(400,50,30,30);

        caballero = new Caballero(100, 50, 5,20,30);

        esqueleto = new Esqueleto(150, 100, 20, 30, 5);

        //hitboxIsla = new Hitbox(200, 550,200,50);

        //hitboxCaballero = new Hitbox(100, 50, 30, 30);

        // Inicia el juego!

        this.entorno.iniciar();

    }

    /\*\*

     \* Durante el juego, el método tick() será ejecutado en cada instante y

     \* por lo tanto es el método más importante de esta clase. Aquí se debe

     \* actualizar el estado interno del juego para simular el paso del tiempo

     \* (ver el enunciado del TP para mayor detalle).

     \*/

    public void tick()

    {

        // Procesamiento de un instante de tiempo

        dibujarIslas(islas);

        cofre.dibujarCofre(entorno);

        esqueleto.dibujarEsqueleto(entorno);

        caballero.dibujarCaballero(entorno);

        //SECCION DE CABALLERO Y ESQUELETO

        boolean estaPresionadaDerecha = entorno.estaPresionada(entorno.TECLA\_DERECHA);

        boolean estaPresionadaIzquierda = entorno.estaPresionada(entorno.TECLA\_IZQUIERDA);

        if(esqueletoEstaTocandoAlgunaIsla(islas) != null){

            Isla isla = esqueletoEstaTocandoAlgunaIsla(islas); //Guarda la isla que el esqueleto toco

            boolean tocaDerecha = esqueleto.getX() + esqueleto.getAncho()/2 == isla.getX() + isla.getAncho()/2;

            boolean tocaIzquierda = esqueleto.getX() - esqueleto.getAncho()/2 == isla.getX() - isla.getAncho()/2;

            if(tocaDerecha || tocaIzquierda){

                esqueleto.cambiarDireccion();

                esqueleto.mover();

            }else{

                esqueleto.mover();

            }

        }else{

            esqueleto.caer();

        }

        if(caballeroEstaTocandoAlgunaIsla(islas)==null){

            caballero.caer();

            if(entorno.estaPresionada(entorno.TECLA\_ESPACIO)){

                caballero.saltar();

            }

        }

        if(estaPresionadaDerecha){

            caballero.moverDerecha();

        }

        if (estaPresionadaIzquierda) {

            caballero.moverIzquierda();

        }

        if (entorno.estaPresionada(entorno.TECLA\_ESPACIO)) {

            caballero.saltar();

        }

        if(entorno.sePresiono(entorno.TECLA\_CTRL)){

            confirmar = true;

            boladefuego = new BolaDeFuego(caballero.getX(), caballero.getY(),10, 10, 10);

        }

        if(confirmar == true){

            if(boladefuego.x <= entorno.ancho()){

                boladefuego.dibujarBolaDeFuego(entorno);

                boladefuego.mover();

            }

            if(boladefuego.x == esqueleto.getX() && boladefuego.y == esqueleto.getY()){

                esqueleto = null;

            }

        }

        //SECCION DE COFRE Y MONEDAS

        if(faltaAlgunaMoneda(cofre)){

            añadirMonedas(cofre);

            dibujarMonedas(cofre);

        }

    }

    public void generarIslas(){

        islas[0] = new Isla(100, 500, 100, 30);

        islas[1] = new Isla(250, 500, 100, 30);

        islas[2] = new Isla(400, 500, 100, 30);

        islas[3] = new Isla(550, 500, 100, 30);

        islas[4] = new Isla(700, 500, 100, 30);

        islas[5] = new Isla(175, 400, 100, 30);

        islas[6] = new Isla(325, 400, 100, 30);

        islas[7] = new Isla(475, 400, 100, 30);

        islas[8] = new Isla(625, 400, 100, 30);

        islas[9] = new Isla(250, 300, 100, 30);

        islas[10] = new Isla(400, 300, 100, 30);

        islas[11] = new Isla(550, 300, 100, 30);

        islas[12] = new Isla(325, 200, 100, 30);

        islas[13] = new Isla(475, 200, 100, 30);

        islas[14] = new Isla(400, 100, 100, 30);

    }

    public void dibujarIslas(Isla[] isla){

        for (int i = 0; i < isla.length; i++) {

            islas[i].dibujarIsla(entorno);

        }

    }

    public Isla esqueletoEstaTocandoAlgunaIsla(Isla[] islas){

        for (int i = 0; i < islas.length; i++) {

            Isla isla = islas[i];

            if(esqueleto.tocaAbajo(isla)){

                return isla;

            }

        }

        return null;

    }

    public Isla caballeroEstaTocandoAlgunaIsla(Isla[] islas){

        for (int i = 0; i < islas.length; i++) {

            Isla isla = islas[i];

            if(caballero.tocaAbajo(isla)){

                return isla;

            }

        }

        return null;

    }

    @SuppressWarnings("unused")

    public static void main(String[] args)

    {

        Juego juego = new Juego();

    }

}