



«interface» ControladorGrafico
+ registrarEntidad(p:PlataformaMovil): ObserverPlataformaMovil + registrarEntidad(e: EntidadEstatica): void + registrarEntidad(e: EntidadDinamica): void + registrarEntidad(j: Jugador): void + setModoJuego(modos:String): void

«interface» ControladorDeVista
+ cambiarInicioJuego(): void + cambiarPantallaRanking(): void + cambiarPantallaModo(): void + cambiarPantallaGameOver(): void + cambiarPantallaTransicion(): void + cambiarPantallaPrincipal(): void + cambiarPantallaFinDelJuego(): void

ConstantesDificultad
+ POTENCIA_DISPARO_AUMENTADA: int=2 + POTENCIA_DISPARO_NORMAL: int=1 + VELOCIDAD_DISPARO: int=15 + ALTURA_MAX_SALTO: int=72 + SALUD_DEMONIO_ROJO: int=2 + SALUD_TROLL_AMARILLO: int=1 + SALUD_RANA_DE_FUEGO: int=1 + SALUD_MOGHERA: int=3 + SALUD_KAMAKICHI: int=3 + VELOCIDAD_DEMONIO_ROJO: int=1 + VELOCIDAD_TROLL_AMARILLO: int=2 + VELOCIDAD_RANA_DE_FUEGO: int=1 + VELOCIDAD_MOGHERA: int=8 + VELOCIDAD_KAMAKICHI: int=8 + VELOCIDAD_CALABAZA: int=1 + VELOCIDAD_FANTASMA: int=1 + VELOCIDAD_BOLADENIEVE: int=2 + VELOCIDAD_MAXIMA_BOLADENIEVE: int=15 + VELOCIDAD_BOLADEFUEGO: int=6 + VELOCIDAD_ENTIDAD_DINAMICA_X: int=8 + VELOCIDAD_ENTIDAD_DIANMICA_Y: int=4 + VELOCIDAD_PLATAFORMA_MOVIL: int=1 + VELOCIDAD_JUGADOR: int=5 + VELOCIDAD_SALTO: int=4 + TOPE_TIEMPO_CONTRARRELOJ: int=120 + CONDICION_SUPERVIVENCIA: int=5000; + VIDAS_INICIO: int=3 + EFECTO_BOTELLA_AZUL: int=10

ConstantesVistas
+ VENTANA_ALTO: int=660 + VENTANA_ANCHO: int=832 + PANEL_ALTO: int=608 + PANEL_ANCHO: int=832 + PANEL_JUEGO_ANCHO: int=768 + PANEL_JUEGO_ALTO: int=600

ConstantesNiveles
+ NIVEL1: String="SnowBros/src/Niveles/nivel1.txt" + NIVEL2: String="SnowBros/src/Niveles/nivel2.txt" + NIVEL3: String="SnowBros/src/Niveles/nivel3.txt" + NIVEL4: String="SnowBros/src/Niveles/nivel4.txt" + OLEADA1: String="SnowBros/src/Niveles/oleada1.txt" + OLEADA2: String= "SnowBros/src/Niveles/oelada2.txt"

ConstantesTiempos
+ TIEMPO_CALABAZA: int=60 + TIEMPO_BOLA_DE_FUEGO: int=10 + TIEMPO_BOLA_DE_NIEVE: int=10 + TIEMPO_DISPARO_JUGADOR: int=12 + TIEMPO_VIDA_POWER_UP: int=10 + TIEMPO_EFECTO_POWER_UP: int=7 + COOLDOWN_DISPARO: long=300 + TIEMPO_LAPSO_INVENCIBILIDAD: long=100 + TIEMPO_CREACION_FANTASMA: int=15 + TIEMPO_CONGELADO_ENEMIGOS: int=10 + TIEMPO_CAIDA_DISPARO: int=800

ConstantesEntidades
+ AIRE: int=20000 + SUELO: int=20001 + ARRIBA: int=15000 + ARRIBA_IZQUIERDA: int=15001 + ARRIBA_DERECHA: int=15002 + IZQUIERDA: int=15003 + DERECHA: int=15004 + ABAJO: int=15005 + QUIETO: int=0 + MARGEN_HIT_BOX: int=1 + X_CALABAZA: int=320 + Y_CALABAZA: int=120 + X_MOGHERA: int=730

	<b>EstadosDePatron</b>
	+ SALTAR: int=16002 + MOVERIZQUIERDA: int=16003 + MOVERDERECHA: int=16004 + ESPERAR: int=16005 + MOVERARRIBA: int=16006 + MOVERABAJO: int= 16007

<b>Ranking</b>
# topClasico: Map<String, Integer> # topSupervivencia: Map<String, Integer> # topContraTiempo: Map<String, Integer>
+ setJugador(nombre: String, puntaje: int, modo: String): void - actualizarRanking(mapa: LinkedHashMap<String, Integer>, nombre: String, puntaje: int): void

# nivel: Nivel # puntajeJugador # tiempo: Tiempo # reiniciarJugador # jugador: Jugador # ranking: Ranking # controladorGame # fabricaSprite # fabricaEntidad # parserActual - tiempoAntes - tiempoActual # hiloColision: # hiloGlobal: Thread # juegoEnEjecucion # nombreJugador # modoDeJuego - ultimoDisparo # dominio: String # gestor: Gestor # jefeCreado: IJefe - oleadas: int - habilitarCondicion
+ setControlador + iniciar(): void - iniciarHilos(): void + detenerJuego + reiniciarNivel + iniciarNivel() + superarNivel + cumplirCondicion + actualizarPuntaje + setReinicio(tiempo: Tiempo) + getControlador + getNivelActual + getPuntaje() + getJugador() + getNivelActual + getTiempoActual + getTimer(): Timer + getRanking() + getJugador() + setNombreJugador + setDominio(dominio: String) + setNivel(nivel: Nivel) + getFabricaEntidad + registrarObservador + registrarObservador + registrarObservador void + registrarObservador extends Entidad + registrarObservador extends Entidad + crearPowerUp + actualizarVida + efectoVidaExpirado + efectoVidaExpirado











<<Interface>> Observer
+ actualizar(): void + actualizarBolaDeNieve(b: boolean):void + borrarObserver(): void

ObserverEntidad
# entidadObservada: EntidadDinamica # plataformaObservada: PlataformaMovil - imagen: JLabel
+ actualizar(): void # actualizarImagen(): void # actualizarPosicionTamano(): void # actualizarPosicionTamanoPlataforma(): void + actualizarBolaDeNieve(activo: boolean): void + borrarObserver(): void + getImagen(): JLabel

ObserverPlataformaMovil
+ actualizar(): void # actualizarImagen(): void

PanelPantallaTransicion
# imagenFondo: JLabel # textoNivel: JLabel # miControladorGrafico: GUI
+ mostrarNivel(): void - agregarImagenFondo(): void - agregarTextoNivel(): void - iniciarTemporizadorTransicion(): void - refrescar(): void

PanelPantallaGameOver
# imagenFondo: JLabel # botonSalir: JButton # botonRanking: JButton # miControladorGrafico: GUI
+ agregarImagenFondo(): void - agregarBotonSalir() - agregarBotonRanking(): void - transparentarBoton(boton: JButton): void - registrarOyenteBotonSalir(): void - registrarOyenteBotonRanking(): void

```

- setModoJuego(modos: String): void
+ setDominio( dominio: String) : void

```

```

+ cambiarPantallaFinDelJuego(): void
+ cambiarPantallaDominio(): void

```

GUI
<pre> # ventana: JFrame # panelPantallaPrincipal: PanelPantallaPrincipal # panelPantallaJuego: PanelPantallaJuego # panelPantallaModo: PanelPantallaModo # panelPantallaDominio: PanelPantallaDominio # panelPantallaTransicion: PanelPantallaTransicion # panelPantallaFinDelJuego: PanelPantallaFinDelJuego # panelPantallaRanking: PanelPantallaRanking # panelPantallaGameOver: PanelPantallaGameOver # miJuego: Juego - movDerecha: boolean - movIzquierda: boolean - disparando : boolean - empujando: boolean - juegoIniciado: boolean # dominio: String # modoDeJuego: String </pre>
<pre> - configurarVentana():void # registrarOyenteVentana():Void - addWindowKeyListener():void + moverJugadorSegunInput(): void - cambiarPantalla(panel: JPanel): void + cambiarPantallaTransicion(): void + cambiarPantallaGameOver():void + cambiarPantallaFinDelJuego(): void + cambiarrPantallaPrincipal(): void + cambiarPantallaModo():void + cambiarPantallaDominio():void + cambiarPantallaRanking(): void + cambiarInicioJuego(): void + setDominio(dominio: String): void + setModoJuego(modos: String): void + setNombre(modos: String): void + registrarEntidad(entidad: EntidadDinamica): void + registrarEntidadEstatica(e: EntidadEstatica): void + registrarEntidad(jugador: Jugador): void + registrarEntidad(entidad: PlataformaMovil): ObserverPlataformaMovil + setJugadorVisibleGraficamente(): void + setJugadorInvisibleGraficamente(): void + setMoverIzq(b: boolean): void + setMoverDer(b: boolean): void + setEmpujando(b: boolean): void + refrescar(): void + getJuego(): Juego + getPantalla(): PanelPantallaJuego + getVentana(): JFrame </pre>

PanelPantallaPrincipal
<pre> + serialVersionUID: long=1L # imagenFondo: JLabel # botonIniciar: JButton # botonRanking: JButton # miControladorGrafico: GUI </pre>
<pre> # agregarImagenFondo(): void # agregarBotonIniciar(): void - agregarBotonRanking(): void - registrarOyenteBotonIniciar(): void - registrarOyenteBotonRanking(): void - transparentarBoton(boton: JButton): void </pre>

PanelPantallaDominio
<pre> # imagenFondo: JLabel # botonOriginal: JButton # botonAlternativo: JButton # miControladorGrafico: GUI </pre>
<pre> + agregarImagenFondo(): void + agregarBotonOriginal(): void + agregarBotonAlternativo(): void + registrarOyenteBotonOriginal(): void + registrarOyenteBotonAlternativo(): void + transparentarBoton(boton: JButton): void </pre>

PanelPantallaModo
<pre> # imagenFondo: JLabel # botonClasico: JButton # botonContrarreloj: JButton # botonSupervivencia: JButton # miControladorGrafico: GUI </pre>
<pre> + agregarImagenFondo(): void + agregarBotonClasico(): void + agregarBotonSupervivencia(): void + agregarBotonContrarreloj(): void + registrarOyenteBotonClasico(): void + registrarOyenteBotonContrarreloj(): void + registrarOyenteBotonSupervivencia(): void + transparentarBoton(boton: JButton): void </pre>

+ Y\_MOGHERA: int=70  
+ X\_KAMAKICHI: int=  
+ Y\_KAMAKICHI: int=

<<Interface>> InterfazEliminar
+ eliminarPlataforma(p: Plataforma): void + eliminarPared(p: Pared): void + eliminarJugador(j: Jugador): void + eliminarEnemigo(e: Enemigo): void + eliminarPowerUp(p: PowerUp): void + eliminarObstaculo(o: Obstaculo): void + eliminarBolaDeFuego(b: BolaDeFuego): v + eliminarBolaDeNieve(b: BolaDeNieve): vo + eliminarDisparo(d: DisparoJugador): void + eliminarSilueta(s: Silueta): void + eliminarFantasma(f: Fantasma): void + eliminarCalabaza(): void + eliminarPMovil(p: PlataformaMovil): void + eliminarResbaladizo(s: SueloResbaladizo): void

<<Interface>> LocalizadorJugador
+ getJugadorX(): void

<<Interface>> Visitante
+ aceptar(): void

PanelPantallaJuego
# panelPanelPantallaJuego: JPanel # panelInformacion: JPanel # imagenFondoJuego: JLabel # labelPuntaje: JLabel # labelNivel: JLabel # labelTiempo: JLabel # panelVidas: JLabel # miControladorGrafico: ControladorGrafico
# agregarPanelInformacion(): void # agregarPanelJuego(): void + mostrarPlataformas(): void + mostrarObstaculos(): void + mostrarEntidades(entidades: List<T>, ajustarZOrder: boolean): void

PanelPantallaFinDelJuego
# imagenFondo: JLabel # botonSalir: JButton # botonNuevaPartida: JButton # botonRanking: JButton # miControladorGrafico: GUI
+ agregarImagenFondo(): void + agregarBotonSalir(): void + agregarBotonRanking(): void + agregarBotonNuevaPartida(): void + registrarOyenteBotonSalir(): void + registrarOyenteBotonNuevaPartida(): void + registrarOyenteBotonRanking(): void + transparentarBoton(boton: JButton): void

PanelPantallaRank
# imagenFondo: JLabel # botonSalir: JButton # miControladorGrafico: GUI
+ agregarImagenFondo(): void + agregarBotonSalir(): void + registrarOyenteBotonSalir(): void + transparentarBoton(boton: JButton): void + agregarRankings(): void + agregarColumnaRanking(titulo: String, LinkedHashMap<String, Integer>, x: int, filaInicialY: int, separacion: int): void + actualizarRanking(ranking: Ranking): void

	<b>&lt;&lt;Interface&gt;&gt; Visitador</b>
void id	+ visitar(e:Enemigo): void + visitar(p:Plataforma): void + visitar(e: PlataformaMovil): void + visitar(e: PlataformaQuebradiza): void + visitar(b: BotellaAzul): void + visitar(b: BotellaRoja): void + visitar(b: BotellaVerde): void + visitar(f: Fruta): void + visitar(v: VidaExtra): void + visitar(e: Pared): void + visitar(e:ParedDestructible): void + visitar(e: Escalera): void + visitar(e:SueloResbaladizo): void + visitar(e:TrampaPinchos): void + visitar(e: Jugador): void

ce>> ole
void

ing
void
ig, mapeo: it, tituloY: int,
void

+ getTopClasico(): LinkedHashMap<String, Integer> + getTopSupervivencia(): LinkedHashMap<String, Integer> + getTopContrarreloj(): LinkedHashMap<String, Integer> - ordenar(mapeo: LinkedHashMap<String, Integer>): void
--

<b>Nivel</b>
# sprite: Sprite # jugador: Jugador # numeroNivel: int # silueta: Silueta # listaEnemigos: LinkedList<Enemigo> # listaPowerUps: LinkedList<PowerUp> # listaObstaculos: LinkedList<Obstaculos> # listaPlataformas: LinkedList<Plataformas> # listaDisparoJugador: LinkedList<DisparoJugador> # listaBolaDeNieve: LinkedList<BolaDeNieve> # listaBolaDeFuego: LinkedList<BolaDeFuego> # listaParedes: LinkedList<Pared> # listaPmovil: LinkedList<PlataformaMovil> # calabaza: Calabaza # listaFantasmas: LinkedList<Fantasma> # listaResbaladizos: LinkedList<SueloResbaladizo>
+ setNivel(nivel: int): void + getEnemigos(): LinkedList<Enemigo> + getPowerUps(): LinkedList<PowerUp> + getObstaculos(): LinkedList<Obstaculo> + getParedes(): LinkedList<Pared> + getPlataformas(): LinkedList<Plataformas> + getPlataformasMovil(): LinkedList<PlataformaMovil> + getDisparos(): LinkedList<DisparoJugador> + getBolasDeNieve(): LinkedList<BolaDeNieve> + getBolasDeFuego(): LinkedList<BolaDeFuego> + getFantasmas(): LinkedList<Fantasma> + getCalabaza(): Calabaza + getResbaladizos(): LikedList<SueloResbaladizo> + getNivelActual(): int + eliminarPlataforma(p: Plataforma): void + eliminarJugador(j: Jugador): void + eliminarEnemigo(e: Enemigo): void + eliminarPowerUp(p: PowerUp): void + eliminarObstaculo(o: Obstaculo): void + eliminarEntidadesEstaticas(): void + eliminarPared(p: Pared): void + eliminarSilueta(s: Silueta): void + eliminarBolaDeFuego(b: BolaDeFuego): void + eliminarDisparo(b: DisparoJugador): void + eliminarBolaDeNieve(b: BolaDeNieve): void + eliminarFantasma(f: Fantasma): void + eliminarCalabaza(): void + eliminarPMovil(p: PlataformaMovil): void + eliminarResbaladizo(s: SueloResbaladizo): void + agregarJugador(j: Jugador): void + agregarPlataforma(p: Plataforma): void + agregarPlataformaMovil(p: PlataformaMovil): void + agregarObstaculo(o: Obstaculo): void + agregarPared(p: Pared): void + agregarPowerUp(p: PowerUp): void + agregarEnemigo(e: Enemigo): void + agregarDisparo(b: DisparoJugador): void + agregarBolaDeNieve(b: BolaDeNieve): void + agregarBolaDeFuego(b: BolaDeFuego): void + agregarFantasma(f: Fantasma): void + agregarCalabaza(c: Calabaza): void + agregarResbaladizo(s: SueloResbaladizo): void + efectoBotellaVerde(): void + getJuego(): Juego + getJugadorX(): int + getJugador(): Jugador + getSilueta(): Silueta + notificarObservers(): void

+ eliminarPow + eliminarEntic void + eliminarPlata PlataformaQue + getModoDeJ + setModoDeJ + crearCalaba + crearKamaki + crearMogher + crearBolaDe puntajeEmpuja + crearBolaDe BolaDeFuego + crearFantas + shootJugadc + empujarBola + invencibilida + estaCerca(ei entidadEvalua void
--

erUp(p: PowerUp): void  
ladGraficamente(d: EntidadDinamica):  
  
aformaQuebradiza(p:  
:bradiza): void  
luego(): String  
luego(modo: String): void  
za(): void  
ichi(): void  
ra(): void  
:Nieve(posX: int, posY: int,  
do: int): void  
:Fuego(posX: int, posY: int, direccion: int):  
  
ma(posX: int, posY:int): void  
or(): void  
iDeNieve(): void  
d(): void  
ntidadPrincipal: Entidad,  
da: Entidad, umbralDeRango: double):

- instancia:Sonido - mapeo: Map <String,Clip> - clipMusical: Clip
+ getInstancia(): Sonido + reproducirMusica(s: String): void + frenarMusica(): void + reproducirEfectoDeSonido(s: String): void - cargarSonido(r: String) :Clip

Temporizador
- tiempoActual: int - tope: int - hilo: Thread - enMarcha: boolean - accionAlFinalizar: Runnable
+ iniciar(): void + setAccionAlFinalizar(accion: Runnable): void + detener(): void + reiniciar(): void + resetear(): void + incrementar(): void + llegoAlTope(): boolean + getTiempoActual(): int + getStringDeTiempo(): String + getStringTiempoContrarreloj(): String

«Abstr FabricaS
# rutaACarpeta: String
+ get_silueta(): Sprite + getSpriteJugador(): Spr + getSpriteDemonioRojo() + getSpriteTrollAmarillo(): + getSpriteRanaDeFuego + getSpriteCalabaza(): Sp + getSpriteFantasma(): S + getSpriteBotellaAzul(): S + getSpriteBotellaRoja(): S + getSpriteBotellaVerde(): + getSpriteFruta(): Sprite + getSpriteVidaExtra(): Spr + getSpriteEscalera(): Spr + getSpritePared(): Sprite + getSpriteParedDestructi + getSpriteSueloResbalac + getSpriteTrampaPincho: + getSpritePlataforma(): S + getSpritePlataformaMov + getSpritePlataformaQue + getSpriteMoghera(): Spr + getSpriteKamakichi(): S + getSpriteBolaNieve(): S + getSpriteBolaFuego(): S + getSpriteDisparoJugado

Alternativo

Sprite
+ mapeoDeSprites: HashMap<String,String>
+ getRutaImagen(clave: String): String

HitBox
- margen: int= ConstantesEntidades.MARGEN_HIT_BOX - hitBox: Rectangle - ancho: int - alto: int
+ actualizarHitbox(x: int, y: int, ancho: int, alto: int): void + getRectangle(): Rectangle + getTopHitBox(): Rectangle + getBottomHitBox(): Rectangle + getLeftHitBox(): Rectangle + getRightHitBox(): Rectangle + getX(): int + getY(): int + getAncho(): int + getAlto(): int

ParserNivel
- fabrica: FabricaEntidades - superoNivelAnterior: boolean
+ generarNivel(modo:int): Nivel - cargarNivel(archivo:String, modo:int): Nivel + generarOleada(nivel: Nivel, archivo: String): void + obtenerArchivoNivel(modo:int):String - agregarDemonioRojo(nivel: Nivel, x: int, y: int) - agregarTrollAmarillo(nivel: Nivel, x: int, y: int) - agregarRanaDeFuego(nivel: Nivel, x: int, y: int) + setSuperoNivelAnterior(b: boolean): void + getSuperoNivelAnterior(): boolean

act»
Sprites
ite
): Sprite
Sprite
() : Sprite
rite
prite
Sprite
Sprite
: Sprite
rite
rite
.
ible(): Sprite
dizo(): Sprite
s(): Sprite
Sprite
il(): Sprite
bradiza(): Sprite
rite
prite
prite
Sprite
r(): Sprite

FabricaEntidades
- fabrica: FabricaSprite
- gestor : GestorEntidad
- asignarGestor(e: Enemigo): void
+ getSilueta(): Silueta
+ getJugador(x: int, y: int): Jugador
+ getDisparoJugador(x: int, y: int): DisparoJugador
+ getDemonioRojo(x: int, y: int): DemonioRojo
+ getTrollAmarillo(x: int, y: int): TrollAmarillo
+ getRanaDeFuego(x: int, y: int): RanaDeFuego
+ getCalabaza(x: int, y: int): Calabaza
+ getFantasma(x: int, y: int): Fantasma
+ getBotellaAzul(x: int, y: int): BotellaAzul
+ getBotellaRoja(x: int, y: int): BotellaRoja
+ getBotellaVerde(x: int, y: int): BotellaVerde
+ getFruta(x: int, y: int): Fruta
+ getVidaExtra(x: int, y: int): VidaExtra
+ getEscalera(x: int, y: int): Escalera
+ getPared(x: int, y: int): Pared
+ getParedDestructible(x: int, y: int): ParedDestructible
+ getSueloResbaladizo(x: int, y: int): SueloResbaladizo
+ getTrampaPinchos(x: int, y: int): TrampaPinchos
+ getPlataforma(x: int, y: int): Plataforma
+ getPlataformaMovil(x: int, y: int): PlataformaMovil
+ getPlataformaQuebradiza(x: int, y: int): PlataformaQuebradiza
+ getBolaDeNieve(x: int, y: int): BolaDeNieve
+ getBolaDeFuego(x: int, y: int): BolaDeFuego
+ getMoghera(x: int, y: int): Moghera
+ getKamakichi(x: int,y: int): Kamakichi

Original











```
boolean); void
- creaLabelDesdeEntidad(entidad: Entidad): JLabel
+ incorporarEntidad(entidad: EntidadDinamica): void
+ incorporarEntidad(entidad: PlataformaMovil):
ObserverPlataformaMovil
+ incorporarEntidadEstatica(e: EntidadEstatica): JLabel
+ incorporarEntidadJugador(jugador: Jugador): void
+ incorporarSilueta(silueta: Entidad): void
+ actualizarLabelsVidas(j: Jugador): void
+ actualizarLabelsPuntaje(puntajeASumar: int): void
+ actualizarLabelsTiempo(): void
+ actualizarLabelTiempoContrarreloj(): void
+ eliminarLabel(l: JLabel): void
+ refrescar(): void
```

<<Abstra  
EntidadDin

```
# direccion : int
# dirAnterior: int
# velocidad: int
# velocidadY: int
# saltando: boolean
# enAire: boolean
- bajandoPlataforma: bool
- alturaSalto: int
# alturaMax: int
# tocandoSuelo: int
# velocidadSalto: int
# ultimoSalto: long
# observer: ObserverEntic
- miNivel: Nivel

+ getSaltando(): boolean
+ setDireccion(direccion: i
+ setSaltando(salto: boole
+ setBajarPlataforma(b: b
+ mover(): void
+ setTocandoSuelo(estada
+ generarClaveEstado(): $
+ getTocandoSuelo(): int
+ getObserver(): Observer
+ verificarSuelo(): void
+ actualizarCaida(): void
+ iniciarSalto(): void
+ actualizarSalto(): void
# visitarPlataforma(e: Plat
# verificarPared(p: Pared)
+ seguirCaminando(e: Ent
+ eliminar(): void
+ setNivel(n: Nivel): void
+ posicionarHitbox(y: int):
```

BolaDeFuego

```
# tiempo: Temporizador
- tiempoCaida: int
- decrementos: int
- caida: boolean
- spriteJefeKamakichi: boolean=false
- spriteJefeMoghera: boolean=false
```

```
+ mover(): void
+ getTimer(): Temporizador
- subida(): void
- decrementarY(): void
+ setDireccion(direccionDisparo: int): void
+ kamakichiBomba(): boolean
- actualizarBolaMoghera(): void
+ setSpriteKamakichi(b: boolean): void
+ setSpriteMoghera(b: boolean): void
```

BolaDeNieve

```
-estaActiva: boolean
- tiempo: Temporizador
- velocidadMaxima: int
- reboto: boolean
# puntajeEmpujado: int
# gestor: GestorEntidad
```

```
+ mover(): void
+ getEstado(): boolean
+ getTimer(): Temporizad
+ disparar(): int
- terminarRebotar(): void
- aparecerPowerUp(): voic
+ empujar(): void
+ getPuntajeEmpujado(): i
+ setPuntajeEmpujado(pu
void
```

```

+ terminoEINivel(): boolean
+ eliminarse(): void

```

Silueta
+ generarClaveEstado(): String + eliminar(): void

```
ct>>
amica

```

---

```
lean

```

```
dad

```

---

```
int): void
ean): void
oolean):

```

```
o: int): void
String

```

```
rEntidad

```

```
taforma): void
l: void
tidad): void

```

```
int

```

<b>&lt;&lt;Abstract&gt;&gt;</b> <b>Entidad</b>
# sprite: Sprite # posX: int # posY: int # alto: int # ancho: int # hitbox: HitBox # nivel: InterfazEliminar
+ setPosX(x: int): void + setPosY(y: int): void + getPosX(): int + getPosY(): int + getHitbox(): HitBox + getSprite(): Sprite + <i>eliminar</i> () :void + setNivel(l: Nivel): void

<b>Disparo</b>	<b>DisparoJugador</b> # tiempoEfecto: Temporizador # potenciaDisparo: int # velocidad: int # tiempoCaida: int # aumentos: int # efectoBotella: boolean
<b>Jugador</b>	+ mover(): void + setDireccion(direccion: int, dirAnt: int): void - caida(): void + incrementarY(): void + setPotenciaAumentada(efectoActivo: boolean): void + getDaño(): int + eliminar(): void

Jugador
- vidaDelInicio: int=3 # miJuego: Juego # vidas: int # nombre: String # disparoAumentado: boolean # posXoriginal: int # posYoriginal: int # habilitadoASubirEscalera: boolean # invencible: boolean # bolaEmpujada: BolaDeNieve # chocoParedIzq: boolean=false # chocoParedDer: boolean=false
+ mover(direccionAMover: int):void + disparar(): DisparoJugador + setVelocidad(v: int): void + setVelocidadBase(): void

DemonioRojo
<pre># puntajeCongelado: int = 150 # puntajeEmpujado: int = 300 # contadorAnimacion: int=0 - tiempoEstimado: long=0 estado: int=</pre>


<<Abstract>> Enemigo
# salud: int # temporizadoCongelado: Temporizador # accionMovimientoEnemigo: Temporizador # rodando: boolean # muerto: boolean # congelado: boolean # detenido: boolean # tiempoEstado: long=0 # inicioCongelado: long=0 # estado: int=EstadoDePatron.MOVERDERECHA # activarJefe: boolean # jefeDerrotado: boolean # gestor: GestorEntidad
+ detener(): void + avanzar(): void + patronDeMovimiento(): void + getSalud(): int # verificarPared(p: Pared): void + recibirDaño(dañó: int): void + generarClaveEstado(): String + actualizarTiempoCongelado(): void + setCongelado(b: boolean): void + getCongelado(): boolean + getRodando(): boolean + getMuerto(): boolean + getDerrotado(): boolean + getInicioCongelado(): long + congelar(): void + descongelar(): void + convertirEnBola(): void + eliminar(): void

	TrollAmarillo	RanaDeFuego	Fantasma	Calabazas
	# puntajeCongelado: int = 300 # puntajeEmpujado: int = 500 - tiempoEstado: long	# puntajeCongelado: int = 150 # puntajeEmpujado: int = 300 # lanzandoBolas: boolean # direccionDisparo: int - bola1 anizada: boolean=false	- xJugador: int - yJugador: int	# tiempoCreacionFanta - tiempoEstado: long=0 - estado: int=
	+ patronDeMovimiento():		+ setPosicionJugador(x: int, y:int): void + patronDeMovimiento(): void	EstadosDePatron.MOV

<b>&lt;&lt;Abstract&gt;&gt; EntidadEstatica</b>
# miLabel: JLabel
+ generarClaveEstado(): String + setMiLabel(j: JLabel): void + getMiLabel(): void + borrarLabel(): void

<b>&lt;&lt;Abstract&gt;&gt; Obstaculo</b>
+eliminar():void

<b>TrampaPinchos</b>	<b>SueloResbaladizo</b>	<b>Escalera</b>	<b>Pared</b>	<b>ParedDestructible</b>
				+ destruirse(): void + eliminarPared(): void + getPuntaje(): int

<b>Armadura</b>	<b>Moghera</b>	<b>Kamakichi</b>
# isma: Temporizador # velocidad: int # ERIZQUIERDA	# puntajeCongelado: int = 1000 # puntajeEmpujado: int = 5000 - altoJefe: int # congelado: boolean - bolaanzada: boolean	# puntajeCongelado: int = 1000 # puntajeEmpujado: int = 5000 # activarDisparo: boolean # patronDisparo: boolean

Plataforma
+eliminar():void

PlataformaMovil
- puntaje: int = 200 # horizontal: boolean # velocidad: int # observer: ObserverPlataformaMovil # distanciaRecorrida: int # anteriorX: int - estado: int - distanciaMaxRecorrida: int # usada: boolean
+ mover(): void + getPuntaje(): int + getDeltaX(): int + getObserver(): ObserverPlataformaMovil + setDireccion(d: int): void + getDireccion(): int + getDirAnterior(): int + registrarObserver(obs: ObserverPlataformaMovil): void + setUsada():void + getUsada(): boolean

PlataformaQuebradiza
- puntaje: int = 300 # tiempoAntesDeRomperse: Temporizador # eliminado: boolean # usada: boolean
+ getPuntaje(): int + setUsada(): void + getUsada(): boolean + eliminar(): void

VidaExtra

Fruta
+ puntaje: int = 500
+ getPuntaje(): int

BotellaAzul

<div>&lt;&lt;Abstract</div> <div>PowerUp</div>
- puntaje: int = 300 # activo: boolean # tiempoDeVida: Temporizador # tiempoDeEfecto: Temporizador
+ getPuntaje(): int + getLabel(): JLabel + setLabel(l: JLabel): void + getTiempoDeVida(): Temporizador + getTiempoDeEfecto(): Temporizador

<i>tract&gt;&gt;</i> <b>erUp</b>
idor zador
orizador mporizador


<b>BotellaRoja</b>

<b>BotellaVerde</b>







+ eliminar(): void

+ setGestor(g: GestorEnti  
+ eliminar(): void

dad): void		+ setVelocidadBase(): void + setEmpujando(b: BolaDeNieve): void + actualizarPuntaje(puntaje: int): void + getVidas(): int + getJuego(): Juego + getTiempo(): int + getNombre(): String + setNombre(nombre: String): void + getVelocidad(): int + getBolaEmpujada(): BolaDeNieve + setDireccion(direccionActual: int): void + setJuego(j: Juego): void + getChocoParedIzq: boolean + getChocoParedDer: boolean + setChocoPared(b: boolean): void + chocaLadoPlataforma(ladoIzq: boolean): void + perderVida(): void + iniciarInvencibilidad(): void + getInvencibilidad(): boolean + setVidas(): void - efectoBotellaAzul(b: BotellaAzul): void - efectoBotellaRoja(b: BotellaRoja): void + getDisparoAumentado(): boolean - efectoBotellaVerde(b: BotellaVerde): void - efectoPowerUp(b: PowerUp): void + estaHabilitadoASubirEscalera(): boolean + eliminar(): void	+ patronDeMovimiento(): void + getPuntajeCongelado(): int + getPuntajeEmpujado(): int + generarClaveEstado(): String + getSalud(): int
------------	--	--	--

ERDA	<div>- obtenerPatron(): + generarClaveEstado(): String + getPuntajeEmpujado(): int + getPuntajeCongelado(): int + getSalud(): int</div>	<div>BolaDeFuego: BolaDeFuego - jugador: LocalizadorJugador  + patronDeMovimiento(): void + lanzarFuego(): BolaDeFuego + generarClaveEstado(): String + getPuntajeCongelado(): int + getPuntajeEmpujado(): int + getSalud(): int + setNivel(n: Nivel): void</div>	<div>+ patronDeMovimiento(): void + getSalud(): int + getPuntajeCongelado(): int + getPuntajeEmpujado(): int</div>	<div>+ patronDeMovimiento() + invocarFantasma(): void + detenerTemporizador() + getSalud(): int + getPuntajeCongelado() + getPuntajeEmpujado()</div>
------	---	---	--	--

	<div><div>publicidad: boolean;</div><div>- recorrido: int</div></div>	
<div><div>(): void</div><div>oid</div><div>DeFantasma(): void</div></div>	<div><div>+ patronDeMovimiento(): void</div><div>+ lanzarFuego(): BolaDeFuego</div><div>+ setActivo(t: boolean): void</div><div>+ convertirEnBola(): void</div><div>+ getPuntajeCongelado(): int</div><div>+ getPuntajeEmpujado(): int</div><div>+ getSalud(): int</div><div>+ setNivel(n: Nivel): void</div><div>+ posicionarHitbox(y: int): int</div><div>+ seguirCaminando(e: Entidad): void</div><div>+ eliminar(): void</div></div>	<div><div>+ verificarSuelo(): void</div><div>+ patronDeMovimiento(): void</div><div>+ dispararBomba(): void</div><div>+ setActivo(t: boolean): void</div><div>+ agregarBomba(posXInicial: int, posYInicial: int, direccionDisparo: int): void</div><div>+ convertirEnBola(): void</div><div>+ getPuntajeCongelado(): int</div><div>+ getPuntajeEmpujado(): int</div><div>+ getSalud(): int</div><div>+ generarClaveEstado(): String</div><div>+ eliminar(): void</div></div>



