

<<interface>>
ControladorVistasInterfaz
+ mostrarPantallaInicio();
+ mostrarPantallaJuegoModoOriginal();
+ mostrarPantallaJuegoModoSecundario();
+ mostrarPantallaRanking();

PanelPantallaJuego
panelMapa: JPanel
panelInformacion: JPanel
panelScrollMapa: JScrollPane
entidadJugador: EntidadLogicaJugador
imagenFondoPanelMapa: JLabel
imagenFondoPanelInformacion: JLabel
labelPuntaje: JLabel
- posicionPrevixa = -1
rutaFondo: String
+ incorporarEntidadJugador(EntidadLogicaJugador entidadJugador);
+ incorporarEntidad(EntidadLogicaJugador entidadLogica);
+ incorporarSilueta(EntidadLogica entidadLogica);
+ actualizarInfoJugador(EntidadLogicaJugador entidadLogicaJugador);
+ actualizarScrollHaciaJugador(EntidadLogicaJugador entidadJugador);
- agregarPanelMapaConFondoYScroll();
- agregarPanelInformacion();
- agregarImagenFondoPanelInformacion();
- agregarLabelsEditablesPanelInformacion();
- decorarlLabelsInformacion();
+ setFondoJuego(Juego juego);
+ getRutaFondo(); String

PanelPantallaInicio
botonIniciarJuegoModoOriginal: JButton
botonIniciarJuegoModoSecundario: JButton
botonVerRanking: JButton
imagenFondo: JLabel
controladorVistas: ControladorVistas
+ agregarFondoImagen();
+ agregarBotonVerRanking();
+ agregarBotonIniciarJuegoModoOriginal();
+ agregarBotonIniciarJuegoModoSecundario();
+ registrarOyenteBotonIniciarJuegoModoOriginal();
+ registrarOyenteBotonIniciarJuegoModoSecundario();
+ registrarOyenteBotonVerRanking();
+ configurarBotonIniciarJuegoModoOriginal();
+ configurarBotonIniciarJuegoModoSecundario();
+ configurarBotonVerRanking();
+ ocultarBoton();

Jugador
nombre: String
puntos: entero
vidas: entero
+ getNombre(); String
+ setNombre(String n);
+ getPuntos(); entero
+ sumarPuntos(entero p);
+ getVidas(); entero
+ setVidas(entero v);

Enemigo*
multiplicadorVel: float
movingRight: boolean
enAire: boolean
multiplicadorVelocidad: float
izquierda: Sprite
derecha: Sprite
+ serAfectadoPorBolaDeFuego(Mario mario)*;
+ serAfectadoPorJugador(Mario mario)*;
+ cambiarSpriteEnColision();
+ setAire(Boolean b);
+ moverseIzquierda();
+ moverseDerecha();
+ getMovingRight(); boolean
+ setMovingRight(); boolean
+ getAire(); boolean
+ atezrar();
+ actualizarPosicion();
+ visit(Plataforma plataforma, int lado)
+ visit(Vacio vacio, int lado)

BuzzyBeetle
PUNTOS_A_SUMAR: int
PUNTOS_A_RESTAR: int
+ acceptMario(VisitorMario visitorMario);
+ acceptBolaDeFuego(VisitorBolaDeFuego visitorBolaDeFuego)
+ serAfectadoPorBolaDeFuego(Mario mario);
+ atacar(Mario mario);

Spiny
PUNTOS_A_SUMAR: int
PUNTOS_A_RESTAR: int
+ acceptMario(VisitorMario visitorMario);
+ acceptBolaDeFuego(VisitorBolaDeFuego visitorBolaDeFuego)
+ serAfectadoPorBolaDeFuego(Mario mario);
+ atacar(Mario mario);

Lakitu
intervaloArrojarSpiny: entero
+ acceptMario(VisitorMario visitorMario);
+ acceptBolaDeFuego(VisitorBolaDeFuego visitorBolaDeFuego)
+ serAfectadoPorBolaDeFuego(Mario mario);
+ atacar(Mario mario);

Goomba
PUNTOS_A_SUMAR: int
PUNTOS_A_RESTAR: int
+ acceptMario(VisitorMario visitorMario);
+ acceptBolaDeFuego(VisitorBolaDeFuego visitorBolaDeFuego)
+ serAfectadoPorBolaDeFuego(Mario mario);
+ atacar(Mario mario);

ControladorVistas
ventana: JFrame
juego: Juego
panelPantallaJuego: PanelPantallaJuego
panelPantallaInicio: PanelPantallaInicio
panelPantallaRanking: PanelPantallaRanking
panelPantallaPerder: PanelPantallaPerder
panelPantallaGanar: PanelPantallaGanar
+ mostrarPantallaInicio();
+ mostrarPantallaJuegoModoOriginal();
+ mostrarPantallaJuegoModoSecundario();
+ mostrarPantallaRanking();
+ mostrarPantallaGanar();
+ mostrarPantallaPerder();
+ registrarEntidad(EntidadLogica entidadLogica)
+ registrarEntidadJugador(EntidadLogicaJugador entidadLogicaJugador)
+ refrescar();
+ registrarOyenteVentana(KeyHandler keyHandler);
+ configurarVentana();
+ guardarNombre();
+ reiniciarPantallaJuego();

PanelPantallaRanking
botonVolverInicio: JButton
controladorVistas: ControladorVistas
ranking: Ranking
top1: JLabel
top2: JLabel
top3: JLabel
top4: JLabel
top5: JLabel
- agregarBotonVolverInicio();
- registrarOyenteBotonVolverInicio();
- configurarBotonVolverInicio();
- cargarLabels();
- agregarImagenFondo();
+ getRanking(); Ranking
- ocultarBoton(JButton boton);

PanelPantallaGanar
botonVolverMenu: JButton
controladorVistas: ControladorVistas
imagenFondo: JLabel
+ agregarFondolomagen();
+ cargarBoton();
+ registrarOyenteBoton();
+ ocultarBoton();

PanelPantallaPerder
botonVolverMenu: JButton
controladorVistas: ControladorVistas
imagenFondo: JLabel
+ agregarFondolomagen();
+ cargarBoton();
+ registrarOyenteBoton();
+ ocultarBoton();

KeyHandler
entidadLogicaJugador: EntidadLogicaJugador
+ keyPressed(KeyEvent e);
+ keyReleased(KeyEvent e)

Posicion
x: entero
y: entero
+ getX(); entero
+ getY(); entero
+ setX(entero x);
+ setY(entero y);

BolaDeFuego
+ actualizarPosicion();
+ setOrientacion(Int orientacion)

KoopaTroopa
#vidas: int
PUNTOS_A_SUMAR: int
PUNTOS_A_RESTAR: int
+ acceptMario(VisitorMario visitorMario);
+ acceptBolaDeFuego(VisitorBolaDeFuego visitorBolaDeFuego)
+ serAfectadoPorBolaDeFuego(Mario mario);
+ atacar(Mario mario);
+ getVidas();

PiranhaPlant
estaEscondida: boolean
maximoYPossible: int
minimoYPossible: int
PUNTOS_A_SUMAR: int
PUNTOS_A_RESTAR: int
+ actualizarPosicion();
+ acceptMario(VisitorMario visitorMario);
+ acceptBolaDeFuego(VisitorBolaDeFuego visitorBolaDeFuego)
+ serAfectadoPorBolaDeFuego(Mario mario);
+ atacar(Mario mario);

MarioParpadeante
sprite: Sprite
mario: Mario
fabricaSprites: FabricaSprites
+ recibirGolpe(int puntosARestar);
+ getSprite(); Sprite
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ sumarPuntos(int puntosASumar);
+ setEstado(EstadoMario estado);
+ atacar(Enemigo enemigo);
+ atacar(Spiny spiny);
+ atacar(PiranhaPlant piranhaPlant);
+ getSpriteMovidososeDerecha(); Sprite
+ getSpriteMovidoselzquierda(); Sprite
+ getSpriteSaltandolzquierda(); Sprite
+ getSpriteSaltandoderecha(); Sprite
+ getSpriteIledeDerecha(); Sprite
+ getSpriteIlelzelzquierda(); Sprite
+ chocarLadriiloSolido(); boolean
+ crearBolaDeFuego();
+ getEstadoAnterior(); EstadoMario
+ getAlto(); int

SuperMarioFuego
sprite: Sprite
mario: Mario
fabricaSprites: FabricaSprites
+ recibirGolpe(int puntosARestar);
+ getSprite(); Sprite
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ sumarPuntos(int puntosASumar);
+ setEstado(EstadoMario estado);
+ atacar(Enemigo enemigo);
+ atacar(Spiny spiny);
+ atacar(PiranhaPlant piranhaPlant);
+ getSpriteMovidososeDerecha(); Sprite
+ getSpriteMovidoselzquierda(); Sprite
+ getSpriteSaltandolzquierda(); Sprite
+ getSpriteSaltandoderecha(); Sprite
+ getSpriteIledeDerecha(); Sprite
+ getSpriteIlelzelzquierda(); Sprite
+ chocarLadriiloSolido(); boolean
+ crearBolaDeFuego();
+ getEstadoAnterior(); EstadoMario
+ getAlto(); int

MarioInvencible
sprite: Sprite
mario: Mario
fabricaSprites: FabricaSprites
+ recibirGolpe(int puntosARestar);
+ getSprite(); Sprite
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ sumarPuntos(int puntosASumar);
+ setEstado(EstadoMario estado);
+ atacar(Enemigo enemigo);
+ atacar(Spiny spiny);
+ atacar(PiranhaPlant piranhaPlant);
+ getSpriteMovidososeDerecha(); Sprite
+ getSpriteMovidoselzquierda(); Sprite
+ getSpriteSaltandolzquierda(); Sprite
+ getSpriteSaltandoderecha(); Sprite
+ getSpriteIledeDerecha(); Sprite
+ getSpriteIlelzelzquierda(); Sprite
+ chocarLadriiloSolido(); boolean
+ crearBolaDeFuego();
+ getEstadoAnterior(); EstadoMario
+ getAlto(); int

Mario
juego: Juego
estadoActual: EstadoMario
izquiera: boolean
derecha: boolean
saltar: boolean
gravedad: float
velocidadSalto: float
fuerzaSalto: float
enAire: boolean
punaje: int
vidas: int
ultimoPos: int
fabricaSprites: FabricaSprites
orientacion: int
MARIO_IDLE_RIGHT
MARIO_IDLE_LEFT
MARIO_MOVING_RIGHT
MARIO_MOVING_LEFT
MARIO_JUMPING_RIGHT
MARIO_JUMPING_LEFT
+ Mario(Sprite sprite, int x, int y, int ancho, int alto, Juego juego);
+ moverseDerecha();
+ moverseIzquierda();
+ moverseAmbal();
+ recibirGolpe(int puntosARestar);
+ notficarObserver();
+ setMovidososeDerecha(boolean derecha);
+ setMovidoselzquierda(boolean izquierda);
+ setSaltar(boolean saltar);
+ getAire(); boolean
+ setAire(boolean enAire);
+ actualizarTamaño();
+ moviendoseDerecha(); boolean
+ moviendoseIzquierda(); boolean
+ sumarVidas();
+ restarVidas();
+ moverse();
+ setVelocidadSalto(int velocidadSalto);
+ actualizarPosicionGravedad();
+ saltar();
+ setEstado(EstadoMario estadoNuevo);
+ getEstado(); EstadoMario
+ setSprite(Sprite sprite);
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ getPuntaje(); int
+ sumarPuntaje(int p);
+ getJuego(); Juego
+ atacar(Enemigo enemigo);
+ crearBolaDeFuego();
- chequearSiTieneQueCaer();
+ actualizarSprite();
+ getFabrica(); FabricaSprites
+ getOrientacion(); int
+ setOrientacion(int orientacion);
+ caerAVacio();
+ getVidas(); int
+ visit(BloqueSolido bloqueSolido, int lado);
+ visit(BloqueDePreguntas bloqueDePreguntas, int lado);
+ visit(LadriiloSolido ladriiloSolido, int lado);
+ visit(Tuberia tuberia, int lado);
+ visit(Vacio vacio, int lado);
+ visit(BuzzyBeetle buzzyBeetle, int lado);
+ visit(Goomba goomba, int lado);
+ visit(KoopaTroopa koopaTroopa, int lado);
+ visit(Lakitu lakitu, int lado);
+ visit(PiranhaPlant piranhaPlant, int lado);
+ visit(Spiny spiny, int lado);
+ visit(ChampionVerde championVerde);
+ visit(Estrella estrella);
+ visit(FlorDeFuego florDeFuego);
+ visit(Moneda moneda);
+ visit(SuperChampion superChampion);
+ visit(Legada llegada);
- corregirPosicion(EntidadLogica entidadLogica, int lado);
- corregirPosicionEnColisionPorEncima(EntidadLogica entidadColisionada);
- corregirPosicionEnColisionPorIzquierda(EntidadLogica entidadColisionada);
- corregirPosicionEnColisionPorDerecha(EntidadLogica entidadColisionada);
- corregirPosicionEnColisionPorAbajo(EntidadLogica entidadColisionada);

EntidadDinamica*
vel: entero
vida: entero
+ puedeMoverse(entero x, entero y): boolean
+ getVelocidad(entero y);
+ setVida(entero v);
+ setVelocidad(); entero
+ getVida(); entero

SuperMario
sprite: Sprite
mario: Mario
fabricaSprites: FabricaSprites
+ recibirGolpe(int puntosARestar);
+ getSprite(); Sprite
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ sumarPuntos(int puntosASumar);
+ setEstado(EstadoMario estado);
+ atacar(Enemigo enemigo);
+ atacar(Spiny spiny);
+ atacar(PiranhaPlant piranhaPlant);
+ getSpriteMovidososeDerecha(); Sprite
+ getSpriteMovidoselzquierda(); Sprite
+ getSpriteSaltandolzquierda(); Sprite
+ getSpriteSaltandoderecha(); Sprite
+ getSpriteIledeDerecha(); Sprite
+ getSpriteIlelzelzquierda(); Sprite
+ chocarLadriiloSolido(); boolean
+ crearBolaDeFuego();
+ getEstadoAnterior(); EstadoMario
+ getAlto(); int

MarioNormal
sprite: Sprite
mario: Mario
fabricaSprites: FabricaSprites
+ recibirGolpe(int puntosARestar);
+ getSprite(); Sprite
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ sumarPuntos(int puntosASumar);
+ setEstado(EstadoMario estado);
+ atacar(Enemigo enemigo);
+ atacar(Spiny spiny);
+ atacar(PiranhaPlant piranhaPlant);
+ getSpriteMovidososeDerecha(); Sprite
+ getSpriteMovidoselzquierda(); Sprite
+ getSpriteSaltandolzquierda(); Sprite
+ getSpriteSaltandoderecha(); Sprite
+ getSpriteIledeDerecha(); Sprite
+ getSpriteIlelzelzquierda(); Sprite
+ chocarLadriiloSolido(); boolean
+ crearBolaDeFuego();
+ getEstadoAnterior(); EstadoMario
+ getAlto(); int

Juego
jugadorActual: Jugador
managerMovimientoMario: ManagerMovimientoMario
managerMovimientoBolaDeFuego: MangerMovimientoBolasDeFuego
managerMovimientoEnemigos: ManagerMovimientoEnemigos
nivelActual: Nivel
controladorVistas: ControladorVistas
generadorNiveles: GeneradorNiveles
fabricaEntidades: FabricaEntidades
fabricaSprites: fabricaSprites
jugador: Jugador
nombre: String
ranking: Ranking
+ Juego(); Juego
+ iniciarMusicaMenu(); void
+ detenerMusicaMenu(); void
+ sonidoBolaDeFuego(); void
+ agarrarMoneda(); void
+ reproducirSonidoSalto(); void
+ reproducirSonidoPowerUp(); void
+ reproducirSonidoMuerteMario(); void
+ reproducirSonidoBump(); void
+ reproducirSonidoLadriiloRoto(); void
+ reproducirSonidoNivelTerminado(); void
+ cargarFabricaYGenerador(); void
+ setControladorVistas(ControladorVistas controladorVistas); void
+ setFabricaSprites(FabricaSprites fabricaSprites); void
+ getFabricaSprites(); FabricaSprites
+ getControladorVistas(); ControladorVistas
+ iniciar(); void
+ cambiarNivel(int nivel); void
+ registrarBolaDeFuego(BolaDeFuego bolaDeFuego); void
+ registrarObserverEntidad(Entidad entidad); void
+ getNivelActual(); Nivel
+ terminarJuego(); void
+ ganarJuego(); void
+ crearJugador(); void
+ recibirNombre(String nombre); void
+ marioEstaEnEstrella(); void
+ iniciarHilos(); void
+ registrarObservers(); void
+ registrarObserverJugador(Mario mario); void
+ registrarObserverEntidades(List< extends Entidad> entidades); void
- actualizarRanking(); void

Entidad*
pos: Posicion
ancho: entero
alto: entero
observer: Observer
sprite: Sprite
spriteActual: int
+ getX(); entero
+ getY(); entero
+ setX(entero x);
+ setY(entero y);
+ setAncho(entero a);
+ setAlto(entero a);
+ getAncho(); entero
+ getAlto(); entero
+ getSprite(); Sprite
+ setSprite(Sprite s);
+ notficarObserver();
+ registrarObserver();
+ getSpriteActual();

Nivel
listaEnemigos CopyOnWriteArrayList<Enemigo>
listaPlataformas CopyOnWriteArrayList<Plataforma>
listaPowerUps CopyOnWriteArrayList<PowerUp>
listaBolaDeFuego: CopyOnWriteArrayList<BolaDeFuego>
listaLakitus: CopyOnWriteArrayList<Lakitu>
mario: Mario
numero: int
+ getNumero(); entero
+ getMario(); Mario
+ getEnemigos(); List<Enemigos>
+ getLakitu(); List<Lakitu>
+ getBolasDeFuego(); List<BolaDeFuego>
+ agregarBolaDeFuego(BolaDeFuego bolaDeFuego)
+ agregarLakitu(Lakitu lakitu)
+ getPowerUps(); List<PowerUps>
+ setPlataformas(); List<Plataformas>
+ agregarEnemigo(Enemigo enemigo);
+ agregarPlataforma(Plataforma plataforma);
+ agregarPowerUp(PowerUP powerUp);
+ agregarJugador(Mario mario)
+ eliminarEnemigo(Enemigo enemigo)

ConstantesVentanas
+ VENTANA_ALTO = 640
+ VENTANA_ANCHO = 1120
+ PANEL_PRINCIPAL_ALTO = 640
+ PANEL_PRINCIPAL_ANCHO = 1120
+ PANEL_JUEGO_ALTO = 480
+ PANEL_JUEGO_ANCHO = 512
+ PANEL_INFORMACION_ALTO = 128
+ PANEL_INFORMACION_ANCHO = 1120
+ PANEL_MAPA_ALTO = 512
+ PANEL_MAPA_ANCHO = 10753
ESCALA = 1

<<Interface>>
EstadoMario
+ recibirGolpe(int puntosARestar);
+ getSprite(); Sprite
+ serAfectadoPorPowerUp(PowerUP powerUp);
+ sumarPuntos(int puntosASumar);
+ setEstado(EstadoMario estado);
+ atacar(Enemigo enemigo);
+ atacar(PiranhaPlant piranhaPlant);
+ getSpriteMovidososeDerecha(); Sprite
+ getSpriteMovidoselzquierda(); Sprite
+ getSpriteSaltandolzquierda(); Sprite
+ getSpriteSaltandoderecha(); Sprite
+ getSpriteIledeDerecha(); Sprite
+ getSpriteIlelzelzquierda(); Sprite
+ chocarLadriiloSolido(); boolean
+ crearBolaDeFuego();
+ getEstadoAnterior(); EstadoMario
+ getAlto(); int

GeneradorNiveles
- TAMANO_BLOQUE: int
- TAMANO_BLOQUE_Y_MEDIO: int
- fabricaEntidades: FabricaEntidades
- nivelAGenerar: String[]
+ generarNivel(int numeroNivel): Nivel
- cargarNivelDesdeArchivo(String rutaArchivo): String[]

PowerUp*
+PowerUp(sprite: Sprite, x: int, y: int, ancho: int, alto: int)
+ aplicarEfecto(marioNormal: MarioNormal): void
+ aplicarEfecto(superMario: SuperMario): void
+ aplicarEfecto(superMarioFuego: SuperMarioFuego): void
+ aplicarEfecto(marioInvencible: MarioInvencible): void
+ getX(); int
+ getY(); int
+ setX(a: int): void
+ setY(y: int): void
+ setAncho(a: int): void
+ setAlto(a: int): void
+ getAncho(); int
+ getAlto(); int

ReproductorSonido
- clips : Map<String, Clip>
- volumenGeneral : float
+ cargarSonido(nombre: String, rutaArchivo: String) : void
+ reproducir(nombre: String) : void
+ detener(nombre: String) : void
+ ajustarVolumenGeneral(nivelDecibelios: float) : void
+ ajustarVolumenClip(clip: Clip, nivelDecibelios: float) : void

SuperChampionin
- PUNTOS_MARIO_NORMAL: int
- PUNTOS_MARIO_SUPERMARIO: int
+ acceptMario(visitorMario: VisitorMario): void
+ aplicarEfecto(marioNormal: MarioNormal): void
+ aplicarEfecto(superMario: SuperMario): void
+ aplicarEfecto(superMarioFuego: SuperMarioFuego): void
+ aplicarEfecto(marioInvencible: MarioInvencible): void

Moneda
-PUNTOS_MARIO_NORMAL: int
+acceptMario(visitorMario: VisitorMario): void
+aplicarEfecto(marioNormal: MarioNormal): void
+aplicarEfecto(superMario: SuperMario): void
+aplicarEfecto(superMarioFuego: SuperMarioFuego): void
+aplicarEfecto(marioInvencible: MarioInvencible): void

Ranking
- instancia: Ranking
- topRanking: Jugador[]
- size: int
- cantJugadores: int
+ Ranking();
+ getInstancia(); Ranking
+ agregarJugador(Jugador: Jugador): void
+ ordenar(); void
+ intercambiar(pos1: int, pos2: int): void
+ mostrarRanking(); void
+ cargarRankingDesdeArchivo(); void
+ guardarRankingEnArchivo(); void
+ getPrimerPuesto(); Jugador
+ getSegundoPuesto(); Jugador
+ getTercerPuesto(); Jugador
+ getCuartoPuesto(); Jugador
+ getQuintoPuesto(); Jugador

<<Interface>>
EntidadLogicaJugador
+ getVelocidad(); int
+ getVidas(); int
+ getAire(); boolean
+ getAncho(); int
+ getAlto(); int
+ moverseDerecha(); void
+ moverseIzquierda(); void
+ moviendoseDerecha(); boolean
+ moviendoseIzquierda(); boolean
+ setMovidoselzquierda(b: boolean): void
+ setMovidoseseDerecha(b: boolean): void
+ setAire(b: boolean): void
+ setSaltar(b: boolean): void
+ getPuntaje(); int
+ atacar(enemigo: Enemigo): boolean
+ atacar(spiny: Spiny): boolean
+ atacar(piranhaPlant: PiranhaPlant): boolean
+ crearBolaDeFuego(); void
+ getOrientacion(); int
+ setOrientacion(orientacion: int): void

ManagerMovimientoMario
juego : Juego
managerColisionesMario : ManagerColisionesMario
+ ManagerMovimientoMario(Juego juego) : ManagerMovimientoMario
+ run(); void
+ interrupt(); void

ManagerMovimientoEnemigos
juego : Juego
- running : boolean (volatile)
- FPS : int (static final)
- FRAME_TIME : long (static final)
+ ManagerMovimientoEnemigos(Juego juego) : ManagerMovimientoEnemigos
+ run(); void
+ interrupt(); void

Plataforma*
+ Plataforma(sprite: Sprite, x: int, y: int, ancho: int, alto: int) : Plataforma
+ serAfectadoPorJugador(mario: Mario) : void
+ serAfectadoPorBolaDeFuego(b: BolaDeFuego) : void

FlorDeFuego
- PUNTOS_MARIO_NORMAL: int
- PUNTOS_MARIO_SUPERMARIO: int
- PUNTOS_MARIO_SUPERMARIOFUEGO: int
+acceptMario(visitorMario: VisitorMario): void
+aplicarEfecto(marioNormal: MarioNormal): void
+aplicarEfecto(superMario: SuperMario): void
+aplicarEfecto(superMarioFuego: SuperMarioFuego): void
+aplicarEfecto(marioInvencible: MarioInvencible): void

Estrella
- PUNTOS_MARIO_NORMAL: int
- PUNTOS_MARIO_SUPERMARIO: int
- PUNTOS_MARIO_ESTRELLA: int
+acceptMario(visitorMario: VisitorMario): void
+aplicarEfecto(marioNormal: MarioNormal): void
+aplicarEfecto(superMario: SuperMario): void
+aplicarEfecto(superMarioFuego: SuperMarioFuego): void
+aplicarEfecto(marioInvencible: MarioInvencible): void

ChampioninVerde
- PUNTOS_MARIO_NORMAL: int
+acceptMario(visitorMario: VisitorMario): void
+aplicarEfecto(marioNormal: MarioNormal): void
+aplicarEfecto(superMario: SuperMario): void
+aplicarEfecto(superMarioFuego: SuperMarioFuego): void
+aplicarEfecto(marioInvencible: MarioInvencible): void

<<interface>>
Observer
+ actualizar(); void
+ actualizarSoloimagen(); void
ObserverJugador
- entidadLogicaJugador: EntidadLogicaJugador
- panelPantallaJuego: PanelPantallaJuego
- lastSprite: int
+ actualizar(); void
+ actualizarSoloimagen(); void

<<interface>>
EntidadLogica
+ getSprite(); Sprite
+ getX(); int
+ getY(); int
+ getAncho(); int

DetectorColisiones
- mejora: int
- colisionaCon(entidadLogica: EntidadLogica, entidadColisionada: EntidadLogica): int - colisionaConEntidadDerecha(entidadLogica: EntidadLogica, entidadColisionada: EntidadLogica): int - colisionaConEntidadIzquierda(entidadLogica: EntidadLogica, entidadColisionada: EntidadLogica): int - colisionaConEntidadAbajo(entidadLogica: EntidadLogica, entidadColisionada: EntidadLogica): int - colisionaConEntidadArriba(entidadLogica: EntidadLogica, entidadColisionada: EntidadLogica): int + colisionaCon(entidadLogica: EntidadLogica, entidadColisionada: EntidadLogica): int

ConstantesAuxiliares
+ MARIOGRANDE_ALTO: int + MARIONORMAL_ALTO: int + MARIO_ANCHO: int + TAMANOBLOQUE_ALTO: int + TAMANOBLOQUE_ANCHO: int + TAMANOBLOQUEFUEGO: int

FabricaSprites*
- rutaACarpeta: String
+ FabricaSprites(rutaACarpeta: String) + getLegada(): Sprite + getTuberiaIzquierda(): Sprite + getTuberiaDerecha(): Sprite + getTechoTuberiaIzquierda(): Sprite + getTechoTuberiaDerecha(): Sprite + getMario(): Sprite + getMarioldieRight(): Sprite + getMarioldieLeft(): Sprite + getMarioMovingRight(): Sprite + getMarioMovingLeft(): Sprite + getMarioJumpingRight(): Sprite + getMarioJumpingLeft(): Sprite + getSuperMario(): Sprite + getSuperMarioldieRight(): Sprite + getSuperMarioldieLeft(): Sprite + getSuperMarioMovingRight(): Sprite + getSuperMarioMovingLeft(): Sprite + getSuperMarioJumpingRight(): Sprite + getSuperMarioJumpingLeft(): Sprite + getSuperMarioFuego(): Sprite + getSuperMarioFuegoldieRight(): Sprite + getSuperMarioFuegoldieLeft(): Sprite + getSuperMarioFuegoMovingRight(): Sprite + getSuperMarioFuegoMovingLeft(): Sprite + getSuperMarioFuegoJumpingRight(): Sprite + getSuperMarioFuegoJumpingLeft(): Sprite + getMarioEstrella(): Sprite + getMarioEstrellaldieRight(): Sprite + getMarioEstrellaldieLeft(): Sprite + getMarioEstrellaMovingRight(): Sprite + getMarioEstrellaMovingLeft(): Sprite + getMarioEstrellaJumpingRight(): Sprite + getMarioEstrellaJumpingLeft(): Sprite + get(): Sprite + getGoomba(): Sprite + getGoombaDerecha(): Sprite + getGoombaIzquierda(): Sprite + getLakitu(): Sprite + getLakituDerecha(): Sprite + getLakituIzquierda(): Sprite + getSpiny(): Sprite + getSpinyDerecha(): Sprite + getSpinyIzquierda(): Sprite + getKoopaTroopa(): Sprite + getKoopaTroopaDerecha(): Sprite + getKoopaTroopaIzquierda(): Sprite + getBuzzyBeetle(): Sprite + getBuzzyBeetleDerecha(): Sprite + getBuzzyBeetleIzquierda(): Sprite + getFlorDeFuego(): Sprite + getEstrella(): Sprite + getChampionVerde(): Sprite + getMoneda(): Sprite + getPiranhaPlant(): Sprite + getBloqueSolido(): Sprite + getLadrilloSolido(): Sprite + getBloqueDePreguntas(): Sprite + getTuberia(): Sprite + getSuperChampionin(): Sprite + getVacio(): Sprite + getBolaDeFuego(): Sprite

FabricaSpritesModoOriginal

FabricaSpritesModoSecundario

FabricaEntidades
fabricaSprites: FabricaSprites
+ FabricaEntidades(fabricaSprites: FabricaSprites) + getMario(x: int, y: int, ancho: int, alto: int, juego: Juego): Mario + getBuzzyBeetle(x: int, y: int, ancho: int, alto: int): BuzzyBeetle + getSpiny(x: int, y: int, ancho: int, alto: int): Spiny + getGoomba(x: int, y: int, ancho: int, alto: int): Goomba + getKoopaTroopa(x: int, y: int, ancho: int, alto: int): KoopaTroopa + getLakitu(x: int, y: int, ancho: int, alto: int, fabricaSprites: FabricaSprites): Lakitu + getPiranhaPlant(x: int, y: int, ancho: int, alto: int): PiranhaPlant + getChampionVerde(x: int, y: int, ancho: int, alto: int): ChampionVerde + getEstrella(x: int, y: int, ancho: int, alto: int): Estrella + getFlorDeFuego(x: int, y: int, ancho: int, alto: int): FlorDeFuego + getMoneda(x: int, y: int, ancho: int, alto: int): Moneda + getSuperChampionin(x: int, y: int, ancho: int, alto: int): SuperChampionin + getBloqueSolido(x: int, y: int, ancho: int, alto: int): BloqueSolido + getBloqueDePreguntas(x: int, y: int, ancho: int, alto: int, fabricaSprites: FabricaSprites): BloqueDePreguntas + getLadrilloSolido(x: int, y: int, ancho: int, alto: int): LadrilloSolido + getTuberiaIzquierda(x: int, y: int, ancho: int, alto: int): Tuberia + getTuberiaDerecha(x: int, y: int, ancho: int, alto: int): Tuberia + getTechoTuberiaIzquierda(x: int, y: int, ancho: int, alto: int): Tuberia + getTechoTuberiaDerecha(x: int, y: int, ancho: int, alto: int): Tuberia + getVacio(x: int, y: int, ancho: int, alto: int): Vacio + getLegada(x: int, y: int, ancho: int, alto: int): Llegada

FabricaSonidos
- reproductorSonido: ReproductorSonido
+ FabricaSonidos() + cargarSonidos(): void + reproducir(nombre: String): void + detener(nombre: String): void