Model::Main

Jugador
#Mario: Mario
+getMario(): Mario
+moverDerecha(): void
+moverIzquierda(): void
+detener(): void

Sonido
-sonido: Sonido
+getInstance()
+play()
+pause()

```
public static Singleton getIntance(){
    if(sonido == null){
        sonido = new Sonido();
    }
    return sonido;
}
```

nivelBuilder
#mapaActual: Mapa
#nivelActual: Nivel
#fabrica_elem: FabricaElementos
+buildBloques(): void
+buildEntidades(): void
-crearEntidad(id: int, coord: Coordenada): Entidad
+getNivel(): Nivel

Nivel
#listaEntidades: List<Entidades>
#listaPlataforma: List<Plataforma>
#mario: Mario
+setEntidad(ent: Entidad): void
+setPlataforma(plat: Plataforma, Parameter1: void)
+getEntidad(): List<Entidad>
+getPlataforma(): List<Plataforma>

Observer
+*actualizar(): void*

Enemigo
+esSaltable: boolean
+mover()
+*colisionar(mario: Mario): void*
+accept(visitor: ColisionVisitor): void
+isCollisionCheck(ente: Entidad): void
+mostrarDetalle(): void
+getTipo(): String

Juego
#gui
#player: Jugador
#sound: Sonido
#map: Mapa
+initialize()
+stop()
+loadMap(path: String)

Nota: durante el metodo initialize se recolecta por cada clave del mapeo cada coordenada y se va creando a cada entidad jugable del mapa

Parseo
+MapaDesdeDibujo(): void
+posicionInicialMario(): Coordenada
+procesarArchivo(string: String): void

Coordenada
-x: int
-y: int
+getX(): int
+getY(): int
+setX(x: int): void
+setY(y: int): void
+distancia(otra: Coordenada): double

ObserverGrafica
-1L: serialVersionUID
-entidad: Entidad
+actualizar(): void
+actualizar_imagen(): void
+actuilzar_posicion_tamaño(): void

Constantes
+WHIDHT: int = 600
+HEIGHT: int = 480
+OFFSET: int = 16

koopa_troopa
+colision: Hit-Box
+colisionar(mario: Mario)
+move()

Buzy_beetle
{leaf}
+colision: Hit-Box
+colisionar(mario: Mario)
+move()

Marco
-fondo_mapa: Image
+getFondoMapa(): Image

loadMap: Se recibe un path (localizacion del archivo .txt)del nivel deseado a cargar y luego se hace la fase de recoleccion

cuando muere, se elimina la hit box y su sprite baja asta fuera del mapa donde es eliminado

Plataforma
#coordena: Coordenada
#spr: Sprite
#observer: Observer
#hitbox: Hit-Box
+getCoordenada(): Coordenada
+getHitBox(): Hit-Box
+getSprite(): Sprite
+setObserver(observer: Observer): void

ObserverMario

ObserverEntidades

ControladorVista
-jugador: Jugador
-ruta_imagen: String
#ventana: JFrame
#obs_mario1: Observer
#obs_mario: ObserverMario
#obs_entidades: ObserverEntidades
#nivel_actual: nivel
+setNivel(nivel_actual: Nivel): void
+setObserverEntidades(obs_entidades): void
+setObserverMario(obs_mario: Observer): void
+iniciar(): void
+escucharTeclado(): void
#paintComponent(g: Graphics): void
+repaint(): void
+run(): void
+actionPerformed(e: ActionEvent): void
+checkCollision(e1: Entidad, e2: Entidad): boolean

Sprite
+get_Ruta_Imagen(): String

Lakitu
+colision: Hit-Box
+lanzarSpiny()
+colisionar(mario: Mario)
+move()

lakitu deberia tener un angulo que mantenga el control dela cantidad de spinies lanzados?

Piranha
+colision: Hit-Box
+colisionar(mario: Mario)
+move()

Spiny
+colision: Hit-Box
+colisionar(mario: Mario)
+move()

Bola Fuego
+colision: Hit-Box
+colisionar(enemy: Enemy)
+move()

tuberia
+mi_piraña: Piranha
+colision: Hit-Box
+colisionar(mario: Mario)

vacio
+colision: Hit-Box
+colisionar(mario: Mario)

bloque_Solid
+colision: Hit-Box
+colisionar(mario: Mario)

ladrillo_solid
+getTipo(mario: Mario): String
+colision(entidad: Entidad): void
+accept(visitor: ColisionVisitor): void
+CollisionIsCheck(entidadEntrante: Entidad): void
+mostrarDetalle(): void

Goomba
+colisionar(mario: Mario)

```
public void colisionar(Mario mario) {
    string estado = mario.getState();

    //si mario es fuego pasaria a
    super
    mario.getState().colision(mario)


}
Bueno nos quedaria ifs si la clase del estado es diferrete
 eso en el caso de que las coordenas den para que mario cambie de estado
```

Ranking
-lista_ranking: lista
+mostrarLista()
+añadirElemento(nombre: string, puntaje: integer)

Mapa
+mapa_coordenadas: Hashmap<<Integer, List<Coordenadas>>
+cargarMapa(): void
+getMapaCoordenadas(): Hashmap<Integer, List<Coordenada>>
+obtenerCoordenada(id: int): List<Coordenada>

Bloque_pregunta
+colision: Hit-Box
+colisionar(mario: Mario)

Entidad
#hitbox: Hit-Box
#coordenada: Coordenada
#observer: Observer
#spr: Sprite
+*colisionar(ente: Entidad): void*
+*accept(ColisionVisitor: ColisionVisitor): void*
+*isColisionCheck(entidad: Entidad): void*
+getCoordenada(): Coodenada
+getSrite(): Sprite
+gerHitBox(): Hit-Box
+setObserver(observer: Observer): void
+*mostrarDetalle()*
+*mover(): void*

power-up
+colision: Hit-Box
+*colisionar(mario: Mario)*

bloque_Piso
+colision: Hit-Box
+Operation1()

Hit-Box
(from Untitled)
+altura: entero
+ancho: entero
+getAlto(): int
+getAncho(): int

Mario
#state: State
#puntaje: Puntaje
#instancia_mario: Mario
+setState(state: State)
+getState(): MarioState
+setPuntaje(puntaje:Puntaje)
+getPuntaje(): Puntaje
+accept(visitor: ColisionVisitor): void
+colisionar(entidad: Entidad): void
+mostrarDetalle(): void
+accept(visitor: ColisionVisitor): void
+mover(dx: int): void
+detener(dx: int): void
+isCollisionCheck(ente: Entidad): void
+getTipo(): String

```
public Puntaje getPuntaje(){

    return puntaje.getPuntaje();
}
```

```
public int getPuntaje(){
    return puntaje;
}
```

Puntaje
#puntaje: integer
+setPuntaje(valor: int): void
+getPuntaje(): int

Super-champ
+colision: Hit-Box
+colisionar(mario: Mario)

Moneda
+colision: Hit-Box
+colisionar(mario: Mario)

florFuego
+colision: Hit-Box
+colisionar(mario: Mario)

Estrella
+colision: Hit-Box
+colisionar(mario: Mario)

champVerde
+colision: Hit-Box
+colisionar(mario: Mario)

ColisionVisitor
+*visitMario(mario: Mario): void*
+*visitEnemigo(enemigo: Enemigo): void*
+*visitPowerUp(power_up: power-up): void*
+*visitPlataforma(plataforma: Plataforma): void*

Delegamos el comportamiento del get y set puntaje de la clase puntaje

```
public void crash(Entity entity) {

    entity.colisionar(this);
}
```

```
public void accept(ColisionVisitor visitor){
    visitor.visitMario(this)
}
```

```
public void setPuntaje(int puntaje){
    this.puntaje = puntaje;
}
```

MarioColisionVisitor
-mario: Mario
+*visitMario(mario: Mario): void*
+*visitEnemigo(enemigo: Enemigo): void*
+*visitPlataforma(plataforma: Plataforma): void*
+*visitPowerUp(power_up: power-up): void*

ColisionVisitorEnemigo
-enemigo: Enemigo
+visitMario(mario: Mario)
+visit(enemigo: Enemigo)
+visit(plataforma: Plataforma)
+visit(power_up: power-up)

```
public void colisionar(Entidad entidad) {
        // Mario delega la colisión al visitor
        MarioColisionVisitor visitor = new MarioColisionVisitor(this);
        entidad.aceptar(visitor);  // Pasar el visitor a la entidad
    }
```

```
public void setPuntaje(Puntaje puntaje){
    puntaje.setPuntaje(puntaje.getPuntaje());
}
```

MarioState
+mover(): void
+*getChampa(mario: Mario): void*
+*getEstrella(mario: Mario): void*
+*dañado(mario: Mario): void*
+*getFlor(mario: Mario): void*
+*Muerto(mario: Mario): void*

MarioNormalColisionVisitor
+MarioNormalColisionVisitor(mario: Mario)
+visitEnemigo(enemigo: Enemigo): void
+visitPlataforma(plataforma: Plataforma): void
+visitPowerUp(power_up: power-up): void
+()

```
public MarioColisionVisitor(Mario mario) {
    this.mario = mario;
    }
    @Override
    public void visitarEnemigo(Enemigo enemigo) {
        mario.getState().colisionarConEnemigo(mario, enemigo);
    }
    @Override
    public void visitarPowerUp(PowerUp powerUp) {
        mario.getState().colisionarConPowerUp(mario, powerUp);
    }
    @Override
    public void visitarPlataforma(Plataforma plataforma) {

mario.getState().colisionarConPlataforma(mario, plataforma);
    }
```

FabricaElementos
+factory_spr: FabricaSprites
+getGoomba(origen: Coordenada): Goomba
+getLadrillo(origen: Coordenada): ladrillo_solid
+getBuzquePiso(origen: Coordenada): bloque_Piso
+getBloquePregunta(origen: Coordenada): Bloque_pregunta
+getBuzzy(origen: Coordenada): Buzy_beetle
+getKoopa(origen: Coordenada): koopa_troopa
+getLakitu(origen: Coordenada): Lakitu
+getPiranha(origen: Coordenada): Piranha
+getSpiny(origen: Coordenada): Spiny
+getTuboInferior(origen: Coordenada): TuboInferior
+getTuboSuperior(origen: Coordenada): TuboSuperior
+getVacio(origen: Coordenada): vacio

Normal
+colisionarConPowerUp(mario: Mario, power_up: power-up): void
+colisionarConEnemigo(mario: Mario, enemigo: Enemigo)
+colisionarConPlataforma(mario: Mario, plataforma: Plataforma

Super
+colisionarConPowerUp(mario: Mario, power_up: power-up): void
+colisionarConEnemigo(mario: Mario, enemigo: Enemigo)
+colisionarConPlataforma(mario: Mario, plataforma: Plataforma)

Muerto
#superState: MarioState
+Muerto(superState: MarioState)
+colisionarConPowerUp(mario: Mario, power_up: power-up): void
+colisionarConEnemigo(mario: Mario, enemigo: Enemigo)
+colisionarConPlataforma(mario: Mario, plataforma: Plataforma)

Fuego
#superState: MarioState
+Fuego(superState: MarioState): null
+colisionarConPowerUp(mario: Mario, power_up: power-up): void
+colisionarConEnemigo(mario: Mario, enemigo: Enemigo)
+colisionarConPlataforma(mario: Mario, plataforma: Plataforma)

```
public Muerto(State superState){
    this.superState = superState;
}
```

```
public Fuego(State superState){
    this.superState = superState;
}
```

FabricaSpritesBase
#ruta_al_sprite: String
+get_Bloque_Misterioso(): Sprite
+get_Buzzy(): Sprite
+gte_Champi_Rojo(): Sprite
+get_Champi_Verde(): Sprite
+get_Estrella(): Sprite
+get_Flor(): Sprite
+get_Fondo(): Sprite
+get_Goomba(): Sprite
+get_Koopa(): Sprite
+get_Ladrillo(): Sprite
+get_Lakitu(): Sprite
+get_Mario(): Sprite
+get_Moneda(): Sprite
+get_Piranha(): Sprite
+get_Spiny(): Sprite
+get_TuboInferior(): Sprite
+get_TuboSuperior(): Sprite
+get_Vacio(): Sprite
+get_Bloque_Solido(): Sprite
+get_Bloque_Piso(): Sprite

```
powerUp.colisionarConMario(mario);
delega al tipo espefico de poweUp
```

```
@Override
    public void colisionarConEnemigo(Mario mario, Enemigo enemigo) {
        System.out.println("Mario (Normal) muere al colisionar con un enemigo.");
        mario.morir();
    }
    @Override
    public void colisionarConPowerUp(Mario mario, PowerUp powerUp) {
        System.out.println("Mario (Normal) recoge un PowerUp y se convierte en Super Mario.");
        mario.setState(new SuperState()); // Cambiar a SuperState
    }
    @Override
    public void colisionarConPlataforma(Mario mario, Plataforma plataforma) {
        System.out.println("Mario (Normal) aterriza sobre una plataforma.");
        // Comportamiento específico
    }
```