Model::Main

Personaje
#velocidad: int
#poder: boolean

+Personaje(n: Nivel, c: Celda, v: int, p: boolean)
+getVelocidad(): int
+getPoder(): boolean
+getCelda(): Celda
morir(): int
+setCelda(c: Celda)
+getNivel(): Nivel

Nivel
#puntaje: int
#tiempo: int
#cantDestruibles: int

+Nivel(j: Juego)
+getPuntaje(): int
+getTiempo(): int
+getCantDestruibles(): int
+getCelda(fila: int, columna int): Celda
+incrementarTiempo()
+incrementarPuntaje(n: int)
+disminuirCantDestruibles()
+finalizarJuego()
+ganarJuego()
+getAlto(): int
+getAncho(): int
+getJuego(): Juego
+getEnemigos(): coleccion de Enemigo
+setEnemigo(e: enemigo)
+eliminarEnemigo(e: Enemigo)

El constructor inicia el juego

disminuirCantDestruibles(): diminuye en 1 cantDestruibles

ganarJuego(): si cantDestrubiles es cero

incrementarTiempo(): aumenta en 1 tiempo

Juego
+cantRugulos: int
+cantAltair: int
+cantSiruis: int

+Juego(GUI gui)
+mover()
+mover(dir: int)
+getGui(): Gui

Gui

Bomberman
#cantBombas: int
#alcanceBomba: int

+Bomberman(n: Nivel, c: Celda)
+mover(direccion: int)
+duplicarVelocidad()
+setPoder(pd: boolean)
+ponerBomba()
+morir+(): int
+getNivel(): Nivel
+agregarBomba()
+duplicarAlcance()
+getCantBombas(): int
+getAlcanceBomba(): int
+setCantidadBombas(n: int)

morir(): int retornar cero

Enemigo
#valor: int

+Enemigo(n: Nivel, c: Celda, v: int, p: boolean, v: int)
mover()
+morir(): int
+getValor(): int

Celda
#fila: int
#columna: int

+Celda(f: int, c: int, nivel: Nivel, power: PowerUp, estado: Estado)
+getFila(): int
+getColumna(): int
+getNivel(): Nivel
+getPowerUp(): PowerUp
+getBomberman(): Bomberman
+getEnemigos(): colección de Enemigos
+explotar(): int
+getEstado(): Estado
+setEstado(e: Estado)
+avanzar(e: Enemigo, dir: int)
+avanzar(b: Bomberman, dir: int)
+setBomberman(b: Bomberman)
+setEnemigo(e: Enemigo)
+eliminarEnemigo(e: Enemigo)
+setPowerUp(p: PowerUp)

explotar() modifica el estado de la celda

Estado
#Estado(c: Celda)
avanzar(e: Enemigo, dir: int, c: Celda)
avanzar(b: Bobmerman)
destruir()
+getCelda(): Celda

Libre
+Libre(c: Celda)
+avanzar+(e: Enemigo, dir: int, c: Celda)
+avanzar+(b: Bomberman, dir: int, c: Celda)
+destruir+(): int

Destruible
+Destruible(c: Celda)
+avanzar+(e: Enemigo, dir: int, c: Celda)
+avanzar+(b: Bomberman, dir: int, c: Celda)
+destruir+(): int

Indestructible
+Indestruible(c: Celda)
+avanzar+(e: Enemigo, dir: int, c: Celda)
+avanzar+(b: Bomberman, dir: int, c: Celda)
+destruir+(): int

Rugulo
+Rugulo(n: Nivel, c: Celda)
+mover+()

Altair
+Altair(n: Nivel, c: Celda)
+mover+()

Siruis
+Siruis(n: Nivel, c: Celda)
+mover+()

Valor, velocidad y poder no se reciben por parámetro. Son fijos para cada Enemigo y se establecen en su

PowerUp
#valor: int

#PowerUp(c: Celda, valor: int)
+getValor(): int
comportamiento(b: Bomberman)
+getCelda(): Celda

comportamiento(Bomberman) aumenta puntaje en Nivel a traves de Bomberman

Bomba
#alcance: int

+Bomba(c: Celda, alcance: int)
+getAlcance(): int
+explotar(celda: Celda): int

explotar(Celda): verifica en la Celda y sus vecinos que tenga un estado destruible, enemigo y bomberman. Llama a los metodos explotar() del Celda y morir() de los personajes y lleva suma de lo que devuelve cada uno. Si el Bomberman no muere, aumenta el puntaje en nivel

explotar(Celda) debe tardar unos segundos en realizar la accion de destruir las celdas segun su alcence, y su efecto debe durar también unos segundos

SpeedUp
+SpeedUp(c: Celda)
+comportamiento+(b: Bomberman)

Fatality
+Fatality(c: Celda)
+comportamiento+(b: Bomberman)

Bombality
+bombality(c: Celda)
+comportamiento+(b: Bomberman)

Masacrallity
+Masacrallity(c: Celda)
+comportamiento+(b: Bomberman)

#miNivel 1
#miNivel
#miJuego 1
1
#miGui
#miNivel 1
#miBomberman 1
0..6 #misEnemigos
#miCelda 1
#miGrilla[][] 403
#miNivel
0..6 #misEnemigos
0..6
#miCelda
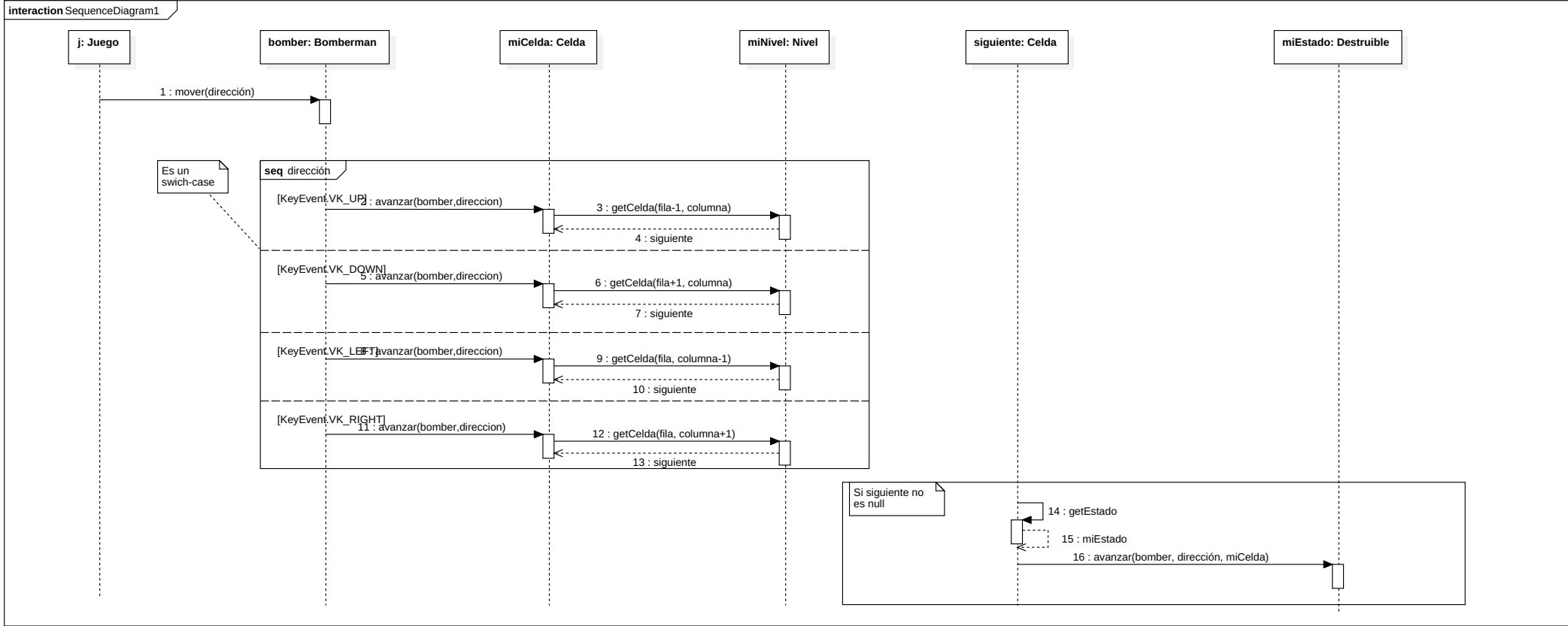1 #miEstado
#miCelda 1
#miCelda
0..1 #miPowerUp
0..1 #miBomberman

Collaboration1::Interaction1::SequenceDiagram1

interaction SequenceDiagram1

j: Juego	bomber: Bomberman	miCelda: Celda	miNivel: Nivel	siguiente: Celda	miEstado: Destruible

1 : mover(dirección)

Es un swich-case

seq dirección

[KeyEvent.VK_UP] 2 : avanzar(bomber,direccion)

3 : getCelda(fila-1, columna)

4 : siguiente

[KeyEvent.VK_DOWN] 5 : avanzar(bomber,direccion)

6 : getCelda(fila+1, columna)

7 : siguiente

[KeyEvent.VK_LEFT] 8 : avanzar(bomber,direccion)

9 : getCelda(fila, columna-1)

10 : siguiente

[KeyEvent.VK_RIGHT] 11 : avanzar(bomber,direccion)

12 : getCelda(fila, columna+1)

13 : siguiente

Si siguiente no es null

14 : getEstado

15 : miEstado

16 : avanzar(bomber, dirección, miCelda)

Collaboration2::Interaction1::SequenceDiagram1

interaction SequenceDiagram1

Avanzar en miEstado
(continuación)

siguiente: Celda   **miEstado: Destruible**   **miCelda: Celda**   **bomber: Bomberman**

1 : avanzar(bomber, dirección, miCelda)

2 : getPoder

3 : tienePoder

Si tienePoder es
verdadero

4 : setBomberman(null)

5 : setBomberman(bomber)

6 : setCelda(siguiente)