

---

## 1. programa.wpgm

```
1 import wollok.game.* //Biblioteca de Wollok Game
2 import juego.*       //Biblioteca de Juego
3
4 program iniciar      //Inicia el programa Bomberman utilizando Wollok Game
5 {
6     juego.jugar()    //Llama al metodo jugar del objeto juego
7 }
```

## 2. juego.wlk

```
1 import wollok.game.* //Biblioteca de Wollok Game
2 import bomberman.*
3 import cajas.*
4 import rivales.*
5 import menues.*
6
7 object juego
8 {
9     //Atributos
10    const musicamenu      = game.sound("./assets/sounds/music-menu.mp3")
11    var estaMenu          = true
12    var spawnRivales      = false
13    var property rivales  = [ new Rival(position = game.at(5,5)),
14                              new Rival(position = game.at(7,5)),
15                              new Rival(position = game.at(5,7)),
16                              new Rival(position = game.at(7,7)),
17                              boss
18    const velocidadDeRivales = 1000
19    var property jugadores = []
20
21    //Metodos
22    method inicializar()
23    {
24        game.addVisual(background) //Muestra el fondo
25        zonaDeJuego.generarMapa()
26        zonaDeJuego.generarCajas()
27        game.addVisual(menu)
28        game.addVisual(selector)
29        game.onTick(500, "animacion menu" ,{menu.animar()})
30        keyboard.up().onPressDo({menu.arriba()})
31        keyboard.down().onPressDo({menu.abajo()})
32        keyboard.enter().onPressDo{self.inicializarJugadores()}
33    }
34
35    method reiniciar()
36    {
37        estaMenu = true
38        spawnRivales = false
39        rivales = [
40            new Rival(position = game.at(5,5)),
```

---

```

41         new Rival(position = game.at(7,5)),
42         new Rival(position = game.at(5,7)),
43         new Rival(position = game.at(7,7)),
44         boss
45     ]
46
47     zonaDeJuego.reiniciarPosiciones()
48     self.inicializar()
49 }
50
51 method jugar()
52 {
53     game.title("Bomberman")
54     game.width(15)
55     game.height(15)
56     self.inicializar()
57     game.schedule(5000, {musicamenu.play()})
58     game.start()
59 }
60
61 method inicializarJugadores()
62 {
63     if(estaMenu){
64         if(!musicamenu.paused())
65             musicamenu.pause()
66         game.sound("./assets/sounds/stage-start.mp3").play()
67         game.removeVisual(menu)
68         game.removeVisual(selector)
69         estaMenu = false
70         if(menu.seleccion() == 0)
71             self.jugador1()
72         if(menu.seleccion() == 1)
73         {
74             self.jugador1()
75             self.jugador2()
76         }
77         if(menu.seleccion() == 2)
78             game.stop()
79     }
80
81     if (!spawnRivales)
82     {
83         self.inicializarRivales()
84         spawnRivales = true
85     }
86 }
87
88 method inicializarRivales()
89 {
90     if(!game.hasVisual(menu))
91     {
92         rivales.forEach
93         {
94             rival => game.addVisual(rival)
95             game.onTick(velocidadDeRivales, "movimiento", {rival.
                movimientoAleatorio()})

```

```

96         }
97         boss.muestraVida()
98     }
99 }
100
101 method jugador1()
102 {
103     keyboard.w().onPressDo({jugador1.irArriba()})
104     keyboard.s().onPressDo({jugador1.irAbajo()})
105     keyboard.a().onPressDo({jugador1.irIzquierda()})
106     keyboard.d().onPressDo({jugador1.irDerecha()})
107     game.addVisual(jugador1)
108     self.jugadores().add(jugador1)
109     jugador1.iniciarPropiedades()
110     jugador1.muestraVida()
111     keyboard.space().onPressDo({jugador1.ponerBomba()})
112     game.onCollideDo(jugador1,{enemigo => enemigo.matar(jugador1)})
113     game.onCollideDo(jugador1,{powerUp => powerUp.efecto(jugador1)})
114 }
115
116 method jugador2()
117 {
118     keyboard.up().onPressDo({jugador2.irArriba()})
119     keyboard.down().onPressDo({jugador2.irAbajo()})
120     keyboard.left().onPressDo({jugador2.irIzquierda()})
121     keyboard.right().onPressDo({jugador2.irDerecha()})
122     game.addVisual(jugador2)
123     self.jugadores().add(jugador2)
124     jugador2.iniciarPropiedades()
125     jugador2.muestraVida()
126     keyboard.shift().onPressDo({jugador2.ponerBomba()})
127     game.onCollideDo(jugador2,{enemigo => enemigo.matar(jugador2)})
128     game.onCollideDo(jugador2,{powerUp => powerUp.efecto(jugador2)})
129 }
130
131 }
132
133 object jugador1 inherits Bombardero ( position = game.at(1,1),
134                                     image = "./assets/bomberman1/
135                                             bomberman1frente.png",
136                                     num = "1",
137                                     posVidas = game.at(0,14) )
138 {}
139
140 object jugador2 inherits Bombardero ( position = game.at(13,13),
141                                     image = "./assets/bomberman2/
142                                             bomberman2frente.png",
143                                     num = "2",
144                                     posVidas = game.at(11,14) )
145 {}
146
147 object zonaDeJuego
148 {
149     var property posicionesOcupadas = []
150     var property posicionesOcupadasCajas = []
151     var property posicionesNoRemovibles = []

```

```

150
151 method agregarPos(posicion)
152 {
153     posicionesOcupadas.add(posicion)
154 }
155
156 method sacarPos(posicion)
157 {
158     posicionesOcupadas.remove(posicion)
159 }
160
161 method agregarCajaEn(posicion){
162     posicionesOcupadasCajas.add(posicion)
163 }
164
165 method sacarCajaEn(posicion){
166     posicionesOcupadasCajas.remove(posicion)
167 }
168
169 method sePuedeRomper(posicion)
170 {
171     return !posicionesNoRemovibles.contains(posicion)
172 }
173
174 method estaOcupada(posicion){
175     return posicionesOcupadas.contains(posicion) or
176     posicionesNoRemovibles.contains(posicion) or
177     posicionesOcupadasCajas.contains(posicion)
178 }
179
180 method reiniciarPosiciones()
181 {
182     posicionesOcupadas = []
183     posicionesOcupadasCajas = []
184 }
185
186 method generarMapa()
187 {
188     6.times{ //Agrego bloques fijos en el mapa: una posicion sin bloque,
189         una con
190         i => 6.times
191         {
192             j => posicionesNoRemovibles.add(game.at(i*2,j*2))
193         }
194     }
195     14.times
196     { //Agrego paredes del borde de mapa
197         i =>
198             posicionesNoRemovibles.add(game.at(i,0))
199             posicionesNoRemovibles.add(game.at(i,14))
200             posicionesNoRemovibles.add(game.at(0,i))
201             posicionesNoRemovibles.add(game.at(14,i))
202     }
203 }
204

```

```

205 method generarCajas()
206 {
207     const entornoJugador1 = [game.at(1,1), game.at(1,2),game.at(2,1)]
208     const entornoJugador2 = [game.at(13,13), game.at(13,12), game.at
        (12,13)]
209     const entornoEnemigos = [game.at(5,5), game.at(5,7), game.at(7,5),
        game.at(7,7)]
210     const posDondeNoPuedeAparecer = entornoJugador1 + entornoJugador2 +
        entornoEnemigos
211     14.times
212     {
213         i => 14.times
214         {
215             j => const probDeAparecer = 1.randomUpTo(10)
216                 if(probDeAparecer < 5 && !posicionesNoRemovibles.contains(
                    game.at(i,j)) && !posDondeNoPuedeAparecer.contains(game
                        .at(i,j)))
217                     new Caja(position = game.at(i,j)).ponerCaja()
218         }
219     }
220 }
221 }

```

### 3. bombas.wlk

```

1 import wollok.game.*
2 import juego.*
3 import cajas.*
4
5 class Bomba
6 {
7     var property position
8     var bombardero
9     var tamanio
10    var atraviesanParedes
11    var property image = "./assets/bombas/bombachica.png"
12
13    method colocarBomba(){
14        game.addVisual(self)
15        game.schedule(800, {image = "./assets/bombas/bombamediana.png"})
16        game.schedule(1700, {image = "./assets/bombas/bombagrande.png"})
17        game.schedule(2500,{self.eliminar() game.sound("./assets/sounds/bomb
            -explodes.mp3").play()})
18        zonaDeJuego.agregarPos(position)
19    }
20
21    method explota(){
22        bombardero.sacarBomba(self)
23        self.aparecerFuego()
24    }
25
26    method eliminar(){
27        zonaDeJuego.sacarPos(position)

```

```

28     game.removeVisual(self)
29     self.explota()
30 }
31
32 method aparecerFuego(){
33     const fuego = new Fuego(position = position,
34                             tamano = tamano,
35                             direcciones = [position.down(1), position.up(1),
36                                           position.left(1), position.right(1)],
37                             atraviesanParedes = atraviesanParedes
38     )
39     fuego.generarFuego()
40 }
41
42 method matar(alguien){}
43 method muere(){}
44 method efecto(jugador){}
45 }
46
47 class Fuego{
48     var property position
49     var tamano
50     var property image = "../assets/fuegos/fuego.png"
51     var property direcciones
52     var atraviesanParedes
53
54     method generarFuego(){
55         game.addVisual(self)
56         game.schedule(600, {game.removeVisual(self)})
57         if(tamano > 0)
58             self.expandirse()
59         game.onCollideDo(self, {algo => algo.muere()})
60     }
61
62     method matar(algo){
63         algo.muere()
64     }
65
66     method puedeExpandirse(posicion){
67         if(zonaDeJuego.sePuedeRomper(posicion))
68             position = posicion
69     }
70
71     method expandirse(){
72         direcciones.forEach{
73             direccion =>
74                 const siguienteDireccionX = (2 * direccion.x()) - position.x()
75                 const siguienteDireccionY = (2 * direccion.y()) - position.y()
76                 const siguienteDireccion = game.at(siguienteDireccionX,
77                                                     siguienteDireccionY)
78
79                 // Inicio calculo de imagen para animacion
80                 var imagen
81                 var auxiliar = ""

```

```

82
83     if(tamano == 1){auxiliar = "fin"}
84
85     if (siguienteDireccionX == position.x()){
86         if (siguienteDireccionY > position.y()) direc = "arriba"
87         else direc = "abajo"
88     }
89     if (siguienteDireccionY == position.y()){
90         if(siguienteDireccionX > position.x()) direc = "derecha"
91         else direc = "izquierda"
92     }
93
94     imagen = "./assets/fuegos/fuego" + direc + auxiliar + ".png"
95     // Fin de calculo de imagen para animacion
96
97     var nuevotamano = tamano -1
98
99     if(zonaDeJuego.sePuedeRomper(direccion)){
100         if(!atraviesanParedes && zonaDeJuego.posicionesOcupadasCajas()
101             .contains(direccion)){nuevotamano = 0}
102         const otroFuego = new Fuego(
103             position = direccion,
104             tamano = nuevotamano,
105             direcciones = [siguienteDireccion],
106             image = imagen, // Indico la imagen que calcule
107                             antes
108             atraviesanParedes = atraviesanParedes
109         )
110         otroFuego.generarFuego()
111     }
112 }
113
114 method muere(){}
115 method efecto(jugador){}

```

#### 4. bomberman.wlk

```

1  import wollok.game.*
2  import bombas.*
3  import juego.*
4  import menues.*
5
6  class Bombardero
7  {
8      var property vidas = 3
9      var property position
10     var property image
11     var property capacidadDeBombas = 2
12     var property bombas = []
13     var property tamanoDeBomba = 1
14     const posicionInicial = position

```

---

```

15  var property estaVivo = true
16  const num
17  const posVidas
18  var bombasAtraviesanParedes = false
19
20  method iniciarPropiedades(){
21      vidas = 3
22      capacidadDeBombas = 1
23      tamanoDeBomba = 1
24      estaVivo = true
25      position = posicionInicial
26  }
27
28  method puedeMoverse(posicion) = !zonaDeJuego.estaOcupada(posicion)
29
30  method irArriba(){
31      self.moverse(position.up(1))
32      image = "./assets/bomberman"+ num + "/bomberman"+ num + "dorso.png"
33  }
34
35  method irAbajo(){
36      self.moverse(position.down(1))
37      image = "./assets/bomberman"+ num + "/bomberman"+ num + "frente.png"
38  }
39  method irDerecha(){
40      self.moverse(position.right(1))
41      image = "./assets/bomberman"+ num + "/bomberman"+ num + "derecha.png"
42  }
43  method irIzquierda(){
44      self.moverse(position.left(1))
45      image = "./assets/bomberman"+ num + "/bomberman"+ num + "izquierda.
46      png"
47  }
48  method moverse(posicion){
49      if(self.puedeMoverse(posicion))
50          position = posicion
51  }
52
53  method ponerBomba(){
54      if(!zonaDeJuego.estaOcupada(position) && bombas.size() <
55          capacidadDeBombas && estaVivo){
56          game.sound("./assets/sounds/place-bomb.mp3").play()
57          const bomba = new Bomba(position = position, bombardero = self,
58              tamano = tamanoDeBomba, atraviesanParedes =
59              bombasAtraviesanParedes)
60          bombas.add(bomba)
61          bomba.colocarBomba()
62      }
63  }
64
65  method sacarBomba(bomb){
66      bombas.remove(bomb)
67  }

```



```

66     method muere(){
67         if(self.vidas() > 1){
68             vidas = vidas -1
69             game.removeVisual(self)
70             estaVivo = false
71             game.schedule(1000, {
72                 game.addVisual(self)
73                 position = posicionInicial
74                 estaVivo = true
75             })
76         }
77         else{
78             vidas = 0
79             juego.jugadores().remove(self)
80             game.removeVisual(self)
81             estaVivo = false
82             game.schedule(500, {menuperdiste.mostrar()})
83         }
84         self.muestraVida()
85     }
86
87     method vidaExtra(){
88         if(vidas < 5)
89             vidas++
90         self.muestraVida()
91     }
92
93     method muestraVida(){
94         const vidasAMostrar = new MostrarVidas(
95             position = posVidas,
96             image = "./assets/corazones/" + self.vidas().toString() + "
               corazones.png"
97         )
98         game.addVisual(vidasAMostrar)
99     }
100
101     method bombaAtraviesaParedes(){bombasAtraviesanParedes = true}
102     method bombaExtra(){capacidadDeBombas = 2}
103     method bombaMasGrande(){tamanioDeBomba = 2}
104     method efecto(a){}
105     method matar(a){
106         const frases = ["Hola!", "Tengo " + self.vidas() + " vidas!", "Movete!"
            , "Quedan " + juego.rivales().size() + " rivales."]
107         const frase = frases.anyOne()
108         game.say(self,frase)
109     }
110 }
111
112
113 class MostrarVidas{
114     var property position
115     var property image
116 }

```

---

## 5. cajas.wlk

```
1  import wollok.game.*
2  import juego.*
3  import powerups.*
4
5  class Caja
6  {
7      var property position
8      var property image = "../assets/muros/murodestructible.png"
9
10     method ponerCaja()
11     {
12         game.addVisual(self)
13         zonaDeJuego.agregarCajaEn(position)
14     }
15
16     method muere()
17     {
18         game.removeVisual(self)
19         self.posibilidadDePowerUp()
20         zonaDeJuego.sacarCajaEn(position)
21     }
22
23     method posibilidadDePowerUp()
24     {
25         const probabilidad = 1.randomUpTo(10)
26         const powerups = [
27             new UnaVidaExtra(position = position),
28             new UnaBombaExtra(position = position),
29             new BombaMasGrande(position = position),
30             new BombaAtraviesaParedes(position = position)
31         ]
32         if(probabilidad < 3)
33         {
34             const powerupAleatorio = powerups.anyOne()
35             powerupAleatorio.aparecer()
36         }
37     }
38
39     method efecto(jugador){}
40
41 }
```

## 6. menus.wlk

```
1  import wollok.game.*
2  import juego.*
3
4  const dirmenu = "../assets/menus/"
5
6  object selector{
```

```

7      var property position = game.origin()
8      var property image = dirmenu + "menu-seleccion-normal.png"
9
10     method normal(){image = dirmenu + "menu-seleccion-normal.png"}
11     method coop(){image = dirmenu + "menu-seleccion-coop.png"}
12     method salir(){image = dirmenu + "menu-seleccion-salir.png"}
13 }
14
15 object menu{
16     var property image = dirmenu + "menu-base.png"
17     var property position = game.origin()
18     var property seleccion = 0
19
20     method abajo(){
21         seleccion++
22         self.actualizarmenu()
23     }
24
25     method arriba(){
26         seleccion--
27         self.actualizarmenu()
28     }
29
30     method actualizarmenu(){
31         if(seleccion > 2)
32             seleccion = 0
33         if(seleccion < 0)
34             seleccion = 2
35
36         if(seleccion == 0)
37             self.normal()
38         if(seleccion == 1)
39             self.coop()
40         if(seleccion == 2)
41             self.salir()
42     }
43
44     method normal(){selector.normal()}
45     method coop(){selector.coop()}
46     method salir(){selector.salir()}
47
48     method animar(){
49         if(image == dirmenu + "menu-base.png")
50             image = dirmenu + "menu-base2.png"
51         else
52             image = dirmenu + "menu-base.png"
53     }
54 }
55
56 class Menuesfin {
57     var property image = dirmenu + "menu" + estado + "1.png"
58     var property position = game.origin()
59     var estado
60
61     method animar(){
62         if(image == dirmenu + "menu" + estado + "1.png")

```

```

63         image = dirmenu + "menu" + estado + "2.png"
64     else
65         image = dirmenu + "menu" + estado + "1.png"
66     }
67
68     method mostrar(){
69         if(juego.rivales().isEmpty()){
70             game.sound("./assets/sounds/stage-clear.mp3").play()
71             game.clear()
72             game.addVisual(self)
73             game.onTick(500, "animacion menu" + estado , {self.animar()})
74             keyboard.r().onPressDo({
75                 game.removeVisual(self)
76                 juego.reiniciar()
77             })
78         }
79     }
80 }
81
82 object menuganaste inherits Menuesfin(estado = "ganaste"){
83
84 object menuperdiste inherits Menuesfin(estado = "perdiste") {
85     override method mostrar(){
86         if(juego.jugadores().isEmpty()){
87             game.sound("./assets/sounds/enemy-dies.mp3").play()
88             game.clear()
89             game.addVisual(self)
90             game.onTick(500, "animacion menuperdiste" , {self.animar()})
91             keyboard.r().onPressDo({
92                 game.removeVisual(self)
93                 juego.reiniciar()
94             })
95         }
96     }
97 }
98
99 object background{
100     var property image = dirmenu + "background.png"
101     var property position = game.origin()
102 }

```

## 7. rivales.wlk

```

1  import juego.*
2  import wollok.game.*
3  import menues.*
4  import bomberman.*
5
6  class Rival {
7      var property position
8      var property image = "./assets/rivales/rivalfrente.png"
9
10     method moverse(posicion){

```

```

11         if(!zonaDeJuego.estaOcupada(posicion))
12             position = posicion
13     }
14
15     method movimientoAleatorio() {
16         const direccionesPosibles = [position.up(1), position.down(1),
17             position.left(1), position.right(1)]
18         const direccionAleatoria = direccionesPosibles.anyOne()
19         if(direccionAleatoria == position.up(1))
20             image = "./assets/rivales/rivaldorso.png"
21         if(direccionAleatoria == position.down(1))
22             image = "./assets/rivales/rivalfrente.png"
23         if(direccionAleatoria == position.right(1))
24             image = "./assets/rivales/rivalderecha.png"
25         if(direccionAleatoria == position.left(1))
26             image = "./assets/rivales/rivalizquierda.png"
27         self.move(se, direccionAleatoria)
28     }
29
30     method matar(jugador){
31         jugador.muere()
32     }
33
34     method muere(){
35         game.sound("./assets/sounds/enemy-dies.mp3").play()
36         game.removeVisual(self)
37         juego.rivales().remove(self)
38         game.schedule(500, {menuganaste.mostrar()})
39     }
40
41     method efecto(jugador){}
42 }
43
44 object boss inherits Rival(position = game.at(5,5), image = "./assets/
45     rivales/bossfrente.png")
46 {
47     var property vidas = 5
48
49     override method move(se, posicion){
50         if(zonaDeJuego.sePuedeRomper(posicion))
51             position = posicion
52     }
53
54     override method movimientoAleatorio() {
55         const direccionesPosibles = [position.up(1), position.down(1),
56             position.left(1), position.right(1)]
57         const direccionAleatoria = direccionesPosibles.anyOne()
58         if(direccionAleatoria == position.up(1))
59             image = "./assets/rivales/bossdorso.png"
60         if(direccionAleatoria == position.down(1))
61             image = "./assets/rivales/bossfrente.png"
62         if(direccionAleatoria == position.right(1))
63             image = "./assets/rivales/bossderecha.png"
64         if(direccionAleatoria == position.left(1))
65             image = "./assets/rivales/bossizquierda.png"

```

---

```

64         self.moverse(direccionAleatoria)
65     }
66     override method muere(){
67         game.sound("./assets/sounds/enemy-dies.mp3").play()
68         if(vidas > 1){
69             vidas--
70             position = zonaDeJuego.posicionesOcupadasCajas().anyOne()
71         }
72         else{
73             game.removeVisual(self)
74             juego.rivales().remove(self)
75             game.schedule(500, {menuganaste.mostrar()})
76         }
77         self.muestraVida()
78     }
79     method muestraVida(){
80         const vidasAMostrar = new MostrarVidas(
81             position = game.at(5,14),
82             image = "./assets/corazones/bossvida" + self.vidas().toString() +
83                 ".png"
84         )
85         game.addVisual(vidasAMostrar)
86     }

```

## 8. powerups.wlk

```

1  import wollok.game.*
2
3  class PowerUp{
4      var property position
5
6      method efecto(jugador){game.removeVisual(self) game.sound("./assets/
7          sounds/item-get.mp3").play()}
8
9      method aparecer(){
10         game.addVisual(self)
11     }
12
13     method matar(noSeUsa){}
14     method muere(){ }
15 }
16
17 class UnaVidaExtra inherits PowerUp{
18     var property image = "./assets/powerups/vidaextra.png"
19     override method efecto(jugador){
20         jugador.vidaExtra()
21         super(jugador)
22     }
23 }
24
25 class UnaBombaExtra inherits PowerUp{
26     var property image = "./assets/powerups/bombaextra.png"

```

---

```
26     override method efecto(jugador){
27         jugador.bombaExtra()
28         super(jugador)
29     }
30 }
31
32 class BombaMasGrande inherits PowerUp{
33     var property image = "../assets/powerups/bombamasgrande.png"
34     override method efecto(jugador){
35         jugador.bombaMasGrande()
36         super(jugador)
37     }
38 }
39
40 class BombaAtraviesaParedes inherits PowerUp{
41     var property image = "../assets/powerups/bombaatraviesaparedes.png"
42     override method efecto(jugador){
43         jugador.bombaAtraviesaParedes()
44         super(jugador)
45     }
46 }
```