



Trabajo Práctico

Sheldon Eats

UNIVERSIDAD TECNOLÓGICA NACIONAL

Paradigmas de Programación

Curso K2002

Docentes: Lucas Spigariol, Ignacio Coluccio

Presentación grupo PdePatrona



Martina Abril
Dominguez

martdominguez@frba.utn.edu.ar



Lucas
Pangaro

lpangaro@frba.utn.edu.ar



Irina Pérez
Gribnicow

iprezgribnicow@frba.utn.edu.ar

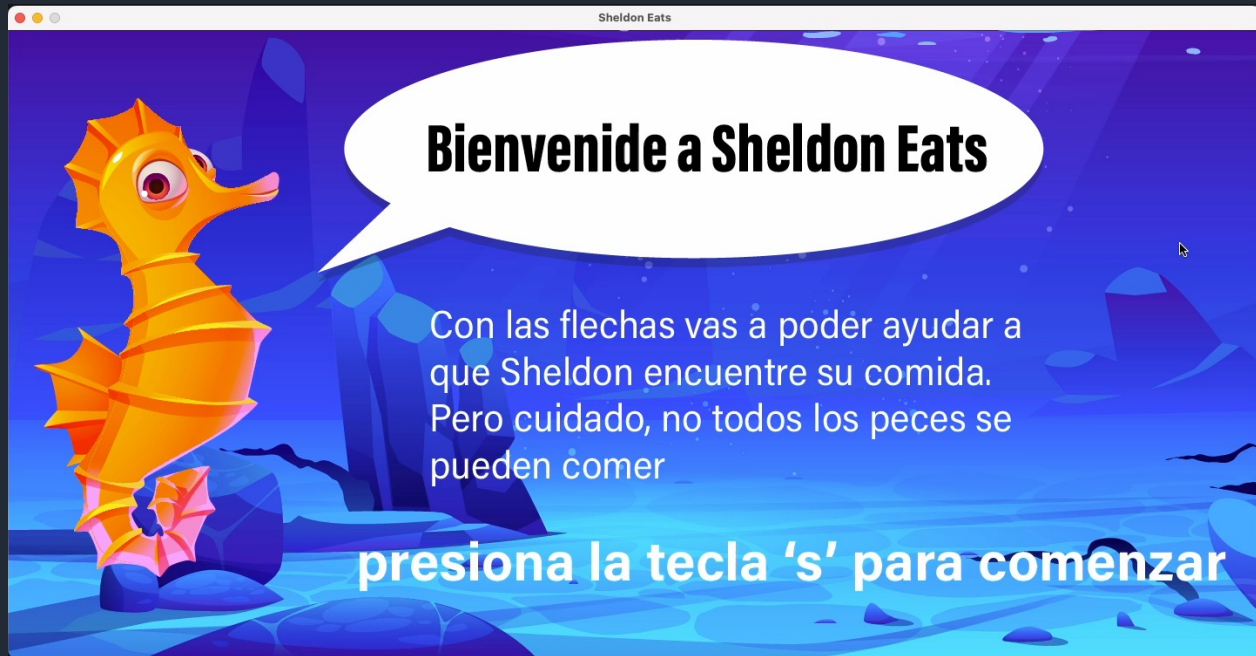


Índice de archivos

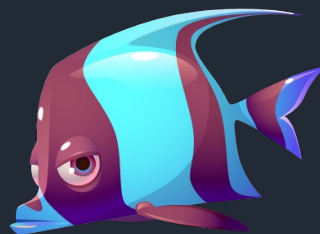
- > **nivel.wlk**: contiene un objeto llamado nivel que es el encargado de toda la configuracion basica, tanto inicial como del nivel en si. Contiene los metodos para cuando se pierde y gana, ademas de incertar las primeras visuales de personajes
- > **visuales.wlk**: Contiene la clase Visual y las definiciones de los fondos
- > **teclado.wlk**: Contiene la configuracion necesaria del teclado
- > **borde.wlk**: Se encarga de dibujar un borde solido para que el personaje principal no puede irse de pantalla
- > **elementos.wlk**: Dentro de este archivo se definen todos los peces, sus metodos en comun y los particulares



bit.ly/sheldonEats

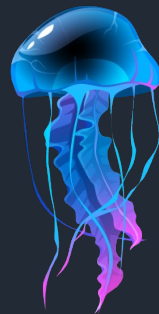


Personajes que Suman Puntos



bit.ly/sheldonEats

Personajes que Restan Puntos





elementos.wlk

```
object sheldon inherits Visual(  
  image = "sheldon.png",  
  position = new Position(x=1,y=1)) {  
  
  var property puntos = 0  
  
  method sumaPuntos(cantidad) { puntos += cantidad }  
  
  method comer(pez){  
    self.sumaPuntos(pez.puntos())  
    game.say(self, "Tengo " + self.puntos() + " puntos!!")  
  }  
}
```



visuales.wlk

```
class Visual {  
  var property image  
  var property position = game.origin()  
}
```



nivel.wlk

```
method configurate() {
  game.clear()
  game.addVisual(fondo)

  //Sheldon
  game.addVisual(sheldon)
  movimiento.configurarFlechas(sheldon)
  game.whenCollideDo(sheldon, {elemento =>
    elemento.colisionadoPor(sheldon) self.puntosParaGanar()
  })
}
```

elementos.wlk

```
class Peces inherits Visual{
  const property puntos

  method colisionadoPor(personaje) {
    personaje.comer(self)
    nivel.ubicarAleatoriamente(self)
  }
}

object tiburon inherits Visual(
  image= "tiburon.png", position = new Position(x=9,y=1)
) {
  method colisionadoPor(personaje) {
    game.sound("comido.mp3").play()
    nivel.gameOver()
  }
}
```

POLIMORFISMO

nivel.wlk


```
method puntosParaGanar(){
  if (sheldon.puntos() >= 150) {
    self.hasGanado()
  }
}
```


●●● nivel.wlk

```
method configurate() {  
  game.clear()  
  game.addVisual(fondo)  
  
  //Sheldon  
  game.addVisual(sheldon)  
  movimiento.configurarFlechas(sheldon)  
  game.whenCollideDo(sheldon, {elemento =>  
    elemento.colisionadoPor(sheldon) self.puntosParaGanar()  
  })  
}
```

●●● teclado.wlk

```
object normal{  
  method mover(direccion, personaje){  
    personaje.position(direccion.siguiente(personaje.position()))  
  }  
}  
object electrocutado{  
  method mover(direccion, personaje){  
  }  
}  
  
object movimiento {  
  var property estado = normal  
  
  method configurarFlechas(visual){  
    keyboard.up().onPressDo{ estado.mover(arriba, visual)}  
    keyboard.down().onPressDo{ estado.mover(abajo, visual)}  
    keyboard.left().onPressDo{ estado.mover(izquierda, visual)}  
    keyboard.right().onPressDo{ estado.mover(derecha, visual)}  
  }  
}  
  
object abajo {  
  method siguiente(position) = position.down(1)  
}  
  
object arriba {  
  method siguiente(position) = position.up(1)  
}
```



elementos.wlk

```
class Peces inherits Visual{
  const property puntos

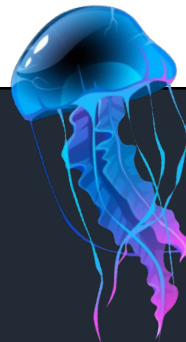
  method colisionadoPor(personaje) {
    personaje.comer(self)
    nivel.ubicarAleatoriamente(self)
  }
}
```

```
object medusa inherits Peces(image= "medusa.png", puntos = -15) {
  override method colisionadoPor(personaje) {
    super(personaje)
    sheldon.image("sheldonSacaLaManoDeAhiCarajo.png")
    game.sound("sacaLaManoDeAhi.mp3").play()

    movimiento.estado(electrocutado)
    game.schedule(4000, { sheldon.image("sheldon.png") movimiento.estado(normal)})
  }
}
```

teclado.wlk

```
object normal{
  method mover(direccion, personaje){
    personaje.position(direccion.siguiente(personaje.position()))
  }
}
object electrocutado{
  method mover(direccion, personaje){
  }
}
```



● ● ● elementos.wlk

```
class Peces inherits Visual{
  const property puntos

  method colisionadoPor(personaje) {
    personaje.comer(self)
    nivel.ubicarAleatoriamente(self)
  }
}

object pulpo inherits Peces(image= "pulpo.png", puntos=35) {
  override method colisionadoPor(personaje){
    super(personaje)
    game.removeVisual(self)
    game.addVisual(tinta)
  }
}

object abisal inherits Peces(image= "abisal.png", puntos = -35) {
  override method colisionadoPor(personaje){
    super(personaje)
    game.addVisual(pantallaOscura)
    game.sound("cortasteTodaLaLoz.mp3").play()

    game.schedule(4000, { game.removeVisual(pantallaOscura) })
  }
}
```

● ● ● nivel.wlk

```
//Peces
[globo,pez10,pez15,pez20,pez30, pulpo, abisal, medusa].forEach {
  pez => game.addVisual(pez) self.ubicarAleatoriamente(pez)}
```



elementos.wlk

```
object tiburon inherits Visual(  
  image= "tiburon.png",  
  position = new Position(x=9,y=1)) {  
  
  method nadar(posSheldon) {  
    position = game.at(  
      position.x()+(posSheldon.x()-position.x())/4,  
      position.y()+(posSheldon.y()-position.y())/4  
    )  
  }  
  method colisionadoPor(personaje) {  
    game.sound("comido.mp3").play()  
    nivel.gameOver()  
  }  
}
```



nivel.wlk

```
//Tiburon  
game.addVisual(tiburon)  
game.onTick(300, "nadar", {tiburon.nadar(sheldon.position())})
```



nivel.wlk

```
method gameOver(){
  game.clear()
  game.width(12)
  game.height(6)
  game.addVisual(perdiste)

  sheldon.puntos(0)
  movimiento.estado(normal)
  self.ubicarAleatoriamente(tiburon)

  keyboard.r().onPressDo{self.configurate()}
}

method hasGanado(){
  game.clear()
  game.width(12)
  game.height(6)
  game.addVisual(ganaste)

  sheldon.puntos(0)
  movimiento.estado(normal)
  keyboard.r().onPressDo{self.configurate()}
}
```

```
const anchoTablero = 11
const altoTablero = 5

method ubicarAleatoriamente(visual){
  var posicion = new Position (x=1.randomUpTo(anchoTablero),y=1.randomUpTo(altoTablero))
  if(game.getObjectsIn(posicion).isEmpty())
    visual.position(posicion)
  else
    self.ubicarAleatoriamente(visual)
}
```

Sheldon Eats



bit.ly/sheldonEats

bit.ly/codigoSheldon

