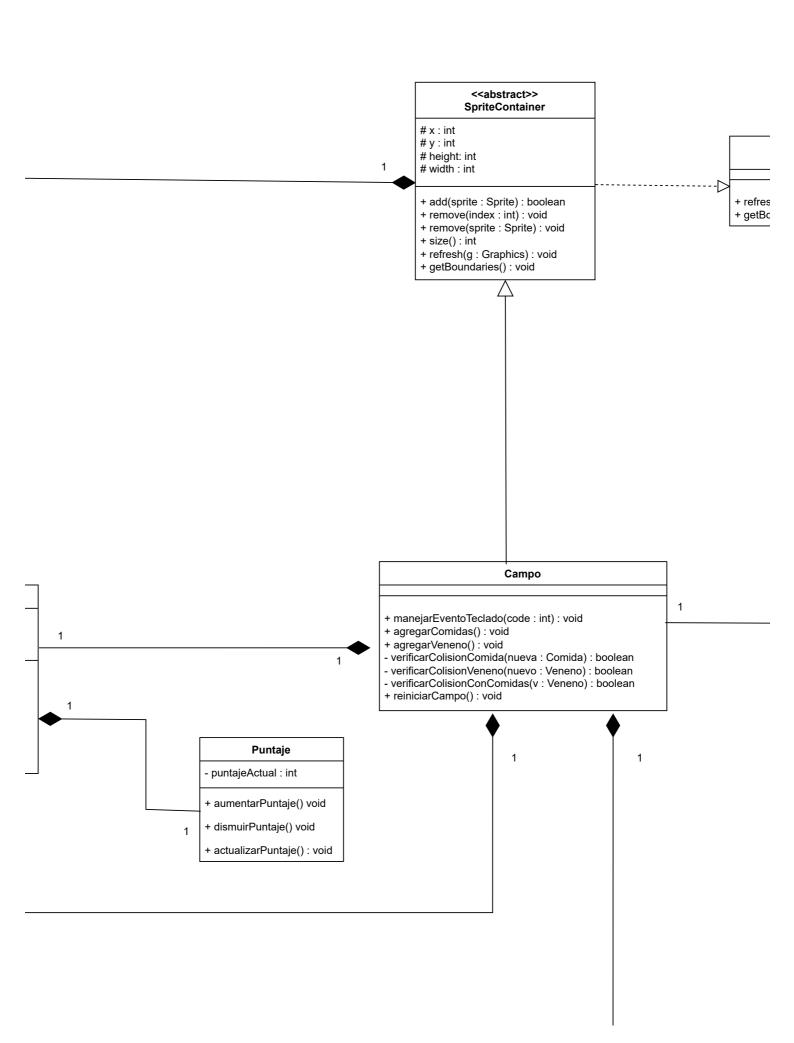
0..\* <<abstract>> Sprite # x : int # y : int # height: int # width : int # visible : boolean # color : Color # image : ImageIcon + isOutOfGraphicContainer(): boolean + isOutOfGraphicContainer(x : int, y : int, width : int, height : int) : boolean + checkCollision(other : Sprite) : boolean + paint(g : Graphics) : void <<abstract>> **SpriteMobile** # running : boolean Jugador # pausa : boolean - nombre : string # delay : long - pasos : int # hilo : Thread - jugadorlmage : Image + isRunning(): boolean + atraparComida(comidas : ArrayList<Comida>) : void + parar(): void + atraparVeneno(venenos : ArrayList<Veneno>) : void + isPausa(): boolean + move(direccion : int) : void + pausar(): void + limiteDeMapa(nx : int, ny : int) : boolean + dePausar() : void + paint(g : Graphics) : void + run(): void + reiniciarPuntaje(): void Veneno <<abstract>> Comida tipo: String - nombre: String · valor: int ‡ venenolmagen : Image - valor : int # comidalmage : Image ⊦ estaCayendo() : boolean + dibujar(g : Graphics) : void + caer(): void + estaCayendo() : boolean + paint(g : Graphics) : void + caer(): void + mover() : boolean + run(): void



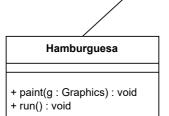
## <<interface>> GraphicContainer

sh(g : Graphics : g) : void oundaries() : Rectangle

## VideoJuego

1

- + iniciarJuego() : void + manejarEventoTeclado(code : int) : void + generarComida() : void + generarVeneno() : void + dibujarElementos(g : Graphics) : void + reiniciarJuego() : void



HotDog

+ paint(g : Graphics) : void + run() : void

