

```

class Sprite {
  constructor({
    position,
    image,
    frames = {max : 1, hold:10},
    sprites,
    moving = false,
    isEnemy = false,
    name
  }) {
    this.position = position
    this.image = image
    this.frames = {...frames, val: 0, elapsed: 0, hold: 10}

    this.image.onload = () =>{
      this.width = this.image.width/this.frames.max
      this.height =this.image.height
    }
    this.moving = moving
    this.sprites = sprites
    this.opacity = 1
    this.health = 100
    this.isEnemy = isEnemy
    this.name = name
  }

  draw() {
    c.save()
    c.globalAlpha = this.opacity
    c.drawImage(
      this.image,
      this.frames.val * this.width,
      0,
      this.width,
      this.height,
      this.position.x,
      this.position.y,
      this.width,
      this.height
    )
    c.restore()

    if(!this.moving) return

    if(this.frames.max > 1){
      this.frames.elapsed++
    }
    if(this.frames.elapsed % this.frames.hold === 0){
      if(this.frames.val < this.frames.max - 1) this.frames.val++
      else this.frames.val = 0
    }
  }

  faint(){
    document.querySelector('#dialogBox').innerHTML = this.name + ' has fainted.'
    gsap.to(this.position, {

```

```

        y: this.position.y + 20
    })
    gsap.to(this.position, {
        y: this.position.y
    })
}

attack({attack, target}) {
    document.querySelector('#dialogBox').style.display = 'block'
    document.querySelector('#dialogBox').innerHTML =
        this.name + ' used ' + attack.name + '.'
    const tl = gsap.timeline()

    let healthBar = '#enemyHealthBar'
    if(this.isEnemy) healthBar = '#playerHealthBar'
    let movementDistance = 20
    if(this.isEnemy) movementDistance = -20
    target.health -= attack.damage
    if(target.health <= 0){target.health = 0}
    tl.to(this.position, {
        x: this.position.x - movementDistance,
        y: this.position.y + movementDistance
    }).to(this.position, {
        x: this.position.x + movementDistance*2,
        duration: 0.1,
        y: this.position.y - movementDistance*2,
        duration: 0.1,
        onComplete:() => {
            gsap.to(healthBar, {
                width: target.health + '%'
            })
            gsap.to(target.position, {
                x: target.position.x + 10,
                duration: 0.08,
                yoyo:true,
                repeat: 5
            })

            gsap.to(target, {
                opacity:0,
                repeat: 5,
                yoyo: true,
                duration: 0.08
            })
        }
    }).to(this.position, {
        x: this.position.x,
        y: this.position.y
    })
}

}

class Boundary {
    static width = 64

```

```
static height = 64
constructor({position,}) {
  this.position = position
  this.width = 64
  this.height = 64
}

draw(){
  c.fillStyle = 'rgba(255,0,0,0.0)'
  c.fillRect(this.position.x, this.position.y, this.width, this.height)
}
}
```