

```

//
// BATTLE LOGIC =====
//

// Sprites needed for the battle scene-----
const battleImage = new Image()
battleImage.src = './img/battleBackground.png'

const playerMonImage = new Image()
playerMonImage.src = './img/snakeSprite.png'

const enemyMonImage = new Image()
enemyMonImage.src = './img/mogusSprite.png'

let battleBack = new Sprite({
  position: {
    x: -56,
    y: 0
  },
  image: battleImage
})
//-----

//function that animates everything in the battle scene
let battleAnimationId
// a queue for all the function called during a battle
let queue

const snake = new Sprite({
  position: {
    x: 170,
    y: 340
  },
  image: playerMonImage,
  frames: {
    max: 4,
    hold: 20
  },
  moving: true,
  name: "Snake"
})
const mogus = new Sprite({
  position: {
    x: 754,
    y: 265
  },
  image: enemyMonImage,
  frames: {
    max: 4,
    hold: 30
  },
  moving: true,
  isEnemy: true,
  name: "Mogus"
})

function animateBattle() {
  battleAnimationId = window.requestAnimationFrame(animateBattle)

```

```

battleBack.draw()
snake.draw()
mogus.draw()
}

function initBattle() {
  document.querySelector('#userInterface').style.display = 'block'
  document.querySelector('#dialogBox').style.display = 'none'
  document.querySelector('#playerHealthBar').style.width = '100%'
  document.querySelector('#enemyHealthBar').style.width = '100%'

  snake.opacity

  snake.health = 100
  mogus.health = 100
  queue = []
  //event listeners for buttons with the attack id
  document.querySelectorAll('#attack').forEach(button => {
    button.addEventListener('click', (e) => {
      const selectedAttack = attacks[e.currentTarget.innerHTML]
      snake.attack({
        attack: selectedAttack,
        target: mogus
      })
      if (mogus.health <= 0) {
        queue.push(() => {
          mogus.faint()
        })
        queue.push(() => {
          //fade back to black to go back to the world
          gsap.to('#overlappingDiv', {
            opacity: 1,
            onComplete: () => {
              battle.initiated = false
              cancelAnimationFrame(battleAnimationId)
              animate()
              document.querySelector('#userInterface').style.display = 'none'

              gsap.to('#overlappingDiv', {
                opacity: 0
              })
            }
          })
        })
      }
    })
  })

  //make the enemy attack after the player
  queue.push(() => {
    mogus.attack({
      attack: attacks.Vent,
      target: snake
    })
    if (snake.health <= 0) {
      queue.push(() => {
        snake.faint()
      })
    }
  })
}

```

```
button.addEventListener('mouseenter', (e) => {  
  const selectedAttack = attacks[e.currentTarget.innerHTML]  
  document.querySelector('#attackType').innerHTML = selectedAttack.type  
  document.querySelector('#attackType').style.color = selectedAttack.color  
})  
}
```