```
let suncounter;
let skysun = [];
let ylim = 0;
let skyfalls = [];
let spawnrate = 150;
let startspawn = false;
let scale = 8;
let zombies = [];
let browncoat;
let pokemon = [];
let h = 30;
let projectiles = [];
let froakie;
let watershuriken;
let cyndaquil;
let fire;
let menu = 0
let namebutton
let inputs = []
let namec = false
let nameinput
let daytheme
let choosetheme
let readybutton
let starterchosen = 0
let juniper
let speech
let cyndaquilbutton
let froakiebutton
let snivybutton
let selected = []
let world = 1
let level = 3
let sunkern
let larvesta
let helioptile
let selectedbuttons = []
let readyplace = false
let placing = ''
let voltorb
let pineco
let explosionsound
let powiecount = 0
let explosion
let place1
let place2
let collectsun
let spacetaken = []
let projectilesound
let splat
let zombiespawn = false
let zombiespawncount = 750
let zombiewave = 0
let bite
let gulp
let bonsly
let wynaut
let colors = []
let rechargecount = []
let conehead
let flag
let gameend
let victory
let coming
let stop = false

function setup() {
  createCanvas(scale * 100, scale * 45);
  frameRate(30);
  maintheme.loop()
  namebutton = createButton('Click Here to Set Your Profile');
namebutton.style('background-color', color(100, 70, 50));
namebutton.position(4*scale, 12*scale);
    namebutton.mousePressed(changename)
  for (let i = 0; i < 9; i++){
    spacetaken[i] = [];
  }
  for(let i = 0; i < 6; i++){
    for(let j = 0; j < 9; j++){
      spacetaken[j][i] = false;
    }
  }
  for(let i = 0; i < 8; i++){
    rechargecount[i] = 200
  }
}

function preload() {
  sun = loadImage("images/sun.png");
  frontyard = loadImage("images/Frontyard.png");
  browncoat = loadImage("images/browncoat.png");
  snivy = loadImage("images/snivy.png");
  leaf = loadImage("images/leaf.png");
  froakie = loadImage("images/froakie.png");
  watershuriken = loadImage("images/water-shuriken.png");
  cyndaquil = loadImage("images/cyndaquil.png");
  fire = loadImage("images/fire.png");
  main = loadImage("images/pvz-main.jpg")
  maintheme = loadSound("music/main.mp3")
  daytheme = loadSound("music/day.mp3")
  choosetheme = loadSound("music/choose.mp3")
  juniper = loadImage("images/Juniper.png")
  speech = loadImage("images/speech.png")
  helioptile = loadImage("images/helioptile.png")
  larvesta = loadImage("images/larvesta.png")
  sunkern = loadImage("images/sunkern.png")
  voltorb = loadImage("images/voltorb.png")
  pineco = loadImage("images/pineco.png")
  explosionsound = loadSound("sounds/explosion.mp3")
  explosion = loadImage("images/powie.png")
  place1 = loadSound("sounds/place.mp3")
  place2 = loadSound("sounds/place2.mp3")
  collectsun = loadSound('sounds/collectsun.mp3')
  projectilesound = loadSound('sounds/projectile.mp3')
  splat = loadSound('sounds/splat.mp3')
  bite = loadSound('sounds/bite.mp3')
  gulp = loadSound('sounds/gulp.mp3')
  bonsly = loadImage('images/bonsly.png')
  wynaut = loadImage('images/wynaut.png')
  conehead = loadImage('images/conehead.png')
  flag = loadImage('images/flag.png')
  gameend = loadSound('sounds/gameover.mp3')
  victory = loadSound('sounds/victory.mp3')
  coming = loadSound('sounds/coming.mp3')
}

function draw() {
  if(menu == 2){
    background(100, 70, 50)
    fill(180)
    rect(70*scale, 0, 30*scale, 45*scale)
    fill(0)
    textSize(30)
    text('choose your pokemon', scale*9, scale*4)
    if(starterchosen < 5){
      image(juniper, 0, 15*scale, 20*scale, 30*scale)
      image(speech, 10*scale, 5*scale, 50*scale, 15*scale)
      textSize(10)
      text('Hello! Welcome to the wonderful world of pokemon! ... Is what', 12*scale, 8*scale)
      text("I would say if it wasn't overrun with zombies at the moment.", 12*scale, 10*scale)
      text('Say, you look like the hero type. Maybe you can save us!', 12*scale, 12*scale)
      text('Here are the last pokemon we have; choose wisely!', 12*scale, 14*scale)
    }
    if(starterchosen == 1){
      image(cyndaquil, 20*scale, 25*scale, 15*scale, 15*scale)
      image(froakie, 38*scale, 25*scale, 15*scale, 15*scale)
      image(snivy, 55*scale, 25*scale, 15*scale, 15*scale)
    }
    if(starterchosen == 2){
      image(sunkern, 20*scale, 25*scale, 15*scale, 15*scale)
      image(helioptile, 38*scale, 25*scale, 15*scale, 15*scale)
      image(larvesta, 55*scale, 25*scale, 15*scale, 15*scale)
    }
    if(starterchosen == 3){
      image(voltorb, 20*scale, 25*scale, 15*scale, 15*scale)
      image(pineco, 38*scale, 25*scale, 15*scale, 15*scale)
    }
    if(starterchosen == 4){
      image(bonsly, 20*scale, 25*scale, 15*scale, 15*scale)
      image(wynaut, 38*scale, 25*scale, 15*scale, 15*scale)
    }
  }
  if(menu == 0){
    background(main);
    //image(wynaut, 0, 0, 50, 50)
    fill(0)
    rect(38*scale, 5*scale, scale*25, 3*scale)
    fill(100, 70, 50)
        if(keyCode === RETURN){
          fill(240, 240, 49)
          text(nameinput.value(), scale*7, scale*11)
          nameinput.remove()
        }
  }
  if(menu == 1){
  background(frontyard);
```

```
fill(100, 70, 50);
rect(scale, scale, 15 * scale, 5 * scale);
rect(scale, 7*scale, 15*scale, 37*scale)
image(sun, scale, scale, 5 * scale, 5 * scale);
spawnrate--;
if (spawnrate == 0) {
  append(
    skyfalls,
    new Skysun(
      random(3 * scale, 97 * scale),
      random(30 * scale, 40 * scale),
      0,
      225
    )
  );
  startspawn = true;
  spawnrate = 300;
}
  if (pokemon.length > 0) {
  for (let i = 0; i < pokemon.length; i++) {
    if(!pokemon[i].getexist()){
      spacetaken[pokemon[i].getxsquare() - 1][pokemon[i].getysquare() - 1] = false
      pokemon.splice(i, 1)
    }
    else{
      if(pokemon[i].getpaid() == false){
      pokemon[i].pay()
    }
      if(pokemon[i].getexist()){
    pokemon[i].display();
    pokemon[i].attack();
      }
      if(pokemon[i].getHP() <= 0){
        pokemon[i].remove()
      }
    }
  }
}
if (startspawn) {
  for (let i = 0; i < skyfalls.length; i++) {
    skyfalls[i].fall(i);
    if (skyfalls.length > 0)
      skyfalls[i].display();
  }
```

```
}
suncount.display();
if (zombies.length > 0) {
  for (let i = 0; i < zombies.length; i++) {
    zombies[i].display();
    zombies[i].move();
    zombies[i].attack();
    if (zombies[i].getHP() <= 0) {
      zombies.splice(i, 1);
    }
  }
}
if (projectiles.length > 0) {
  for (let i = 0; i < projectiles.length; i++) {
    projectiles[i].display();
    projectiles[i].move(i);
    projectiles[i].hit(i);
  }
}
  for(let i = 0; i < selectedbuttons.length; i++){
    selectedbuttons[i].style('background-color', colors[i])
  }
  if(zombiespawn){
    zombiespawncount--
    if(zombiespawncount == 0){
      if(zombiewave == 0){
        coming.play()
        spawnzombies(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      }
    }
    if(zombies.length == 0 || zombiespawncount == 0){
      if(zombiewave == 1 || zombiewave == 2){
        spawnzombies(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      }
      else if(zombiewave == 3 || zombiewave == 5){
        spawnzombies(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      }
      else if(zombiewave == 4){
        spawnzombies(2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      }
      else if(zombiewave == 6 || zombiewave == 8){
        spawnzombies(1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      }
      else if(zombiewave == 7){
        spawnzombies(3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    }
      else if(zombiewave == 9){
        spawnzombies(5, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  }
  }
  for(let i = 0; i < 8; i++){
    if(rechargecount[i] < 200){
    colors[i] = color(rechargecount[i], rechargecount[i], rechargecount[i], 255)
    }
    else
      colors[i] = color(0, 200, 0)
  }
  if(rechargecount[0] < 200)
  rechargecount[0] += 0.9;
  if(rechargecount[1] < 200)
  rechargecount[1] += 0.9;
  if(rechargecount[2] < 200)
  rechargecount[2] += 4/30;
  if(rechargecount[3] < 200)
  rechargecount[3] += 6.6/30;
  textSize(15)
  text('100', selectedbuttons[0].width+2*scale, 10*scale)
  text('50', selectedbuttons[1].width+2*scale, 14*scale)
  text('150', selectedbuttons[2].width+2*scale, 18*scale)
  text('50', selectedbuttons[3].width+2*scale, 22*scale)
}
if(zombies.length == 0 && zombiewave == 10){
  win()
}
    for(let k = 0; k < pokemon.length; k++){
      for(let i = 0; i < 9; i++){
        for(let j = 0; j < 6; j++){

        }
      }
    }
//console.log(projectiles.length)
if(zombies.length > 0){
  for(let i = 0; i < zombies.length; i++){
          if(zombies[i].getx() <= scale*15){
      gameover()
    }
  }
  }
  if(stop){
    fill(255)
    rect(scale*43, scale*21, scale*12, scale*6)
    fill(0)
    text('PAUSED', scale*45, scale*25)
  }
}
}

function mousePressed() {
  if(menu == 2){
    if(starterchosen == 0){
      choosestarter()
    }
    else if(starterchosen == 2){
      cyndaquilbutton.remove()
      froakiebutton.remove()
      snivybutton.remove()
    }
    else if(starterchosen == 3){
      larvestabutton.remove()
      sunkernbutton.remove()
      helioptilebutton.remove()
    }
    else if(starterchosen == 4){
      voltorbbutton.remove()
      pinecobutton.remove()
    }
    else if(starterchosen == 5){
      bonslybutton.remove()
      wynautbutton.remove()
    }
  }
  if(menu == 0){
    if(mouseX > scale*51 && mouseY < scale*14 && mouseX < scale*93 && mouseY >
scale*12){
       menu = 2
       namebutton.remove()
       maintheme.stop()
       choosetheme.loop()
       readybutton = createButton('READY')
       readybutton.position(scale*80, scale*40)
       readybutton.mousePressed(ready)
    }
 }
 if(menu == 1){
 for (let i = 0; i < skyfalls.length; i++) {
  if (
   abs(skyfalls[i].getX() - mouseX) < 4 * scale &&
   abs(skyfalls[i].getY() - mouseY < 4 * scale)
  ) {
   skyfalls[i].collect(i);
   collectsun.play()
  }
 }
  if(readyplace && mouseX > scale*17 && !spacetaken[floor((mouseX - scale*17)/(7.8*scale)) - 1][floor((mouseY - 2*scale)/(7.3*scale))]){
   if((placing == "snivy" || placing == "cyndaquil" || placing == "froakie") && rechargecount[0] >= 200){
    append(pokemon, new Pokemon(floor((mouseX - scale*17)/(7.8*scale)), floor((mouseY - 2*scale)/(7.3*scale)) + 1, placing, 60, 60, 1, 4*scale, 100, false, false, false, false, true, 15, false))
    rechargecount[0] = 0
   }
   if((placing == "sunkern" || placing == "larvesta" || placing == "helioptile") && rechargecount[1] >= 200){
    append(pokemon, new Pokemon(floor((mouseX - scale*17)/(7.8*scale)), floor((mouseY - 2*scale)/(7.3*scale)) + 1, placing, 720, floor(random(120, 240)), 0, 0, 50, false, false, true, false, true, 15, false))
    rechargecount[1] = 0
   }
   if((placing == "voltorb" || placing == "pineco") && rechargecount[2] >= 200){
    append(pokemon, new Pokemon(floor((mouseX - scale*17)/(7.8*scale)), floor((mouseY - 2*scale)/(7.3*scale)) + 1, placing, 45, 45, 75, 0, 150, false, false, false, true, true, 9999999, false))
    rechargecount[2] = 0
   }
   if((placing == 'bonsly' || placing == 'wynaut') && rechargecount[3] >= 200){
    append(pokemon, new Pokemon(floor((mouseX - scale*17)/(7.8*scale)), floor((mouseY - 2*scale)/(7.3*scale)) + 1, placing, 0, 0, 0, 0, 50, false, false, false, false, true, 180, true))
    rechargecount[3] = 0
   }
   spacetaken[floor((mouseX - scale*17)/(7.8*scale)) - 1][floor((mouseY - 2*scale)/(7.3*scale))] = true;
   readyplace = false
   placing = ""
   if(random(0, 2) < 1)
```

```
        place1.play();
      else
        place2.play();
    }
  }
}

function spawnzombies(browncoats, flags, coneheads, polevaulters, bucketheads, newspapers,
screendoors, allstars, discos, backups, floaties, snorkels, zombonis, sleds, dolphins, cookoo,
balloons, miners, yetis, bungees, ladders, catapults, gargantuars, imps){
  for(let i = 0; i < browncoats; i++){
    append(zombies, new Zombies(floor(random(1,6)), floor(random(scale * 98, scale * 100)),
'browncoat', 5, 10, 2.5, 1, 15, 15))
  }
  for(let i = 0; i < flags; i++){
    append(zombies, new Zombies(floor(random(1,6)), floor(random(scale * 98, scale * 100)),
"flag", 6, 10, 2.5, 1, 15, 15))
  }
  for(let i = 0; i < coneheads; i++){
    append(zombies, new Zombies(floor(random(1,6)), floor(random(scale * 98, scale * 100)),
"conehead", 5, 28, 2.5, 1, 15, 15))
  }
  zombiespawncount = 750
          zombiewave++
}

function pause(){
  stop = !stop
  if(stop){
    noLoop();
  }
  else
    loop();
}

function gameover(){
  noLoop()
  textSize(50)
  fill(50, 250, 0)
  text('The zombies ate your', scale*17, scale*17)
  text('brains!', scale*40, scale*30)
  gameend.play()
}
```

```
function win(){
  noLoop()
  textSize(50)
  fill(0)
  text('You win!', scale*30, scale*20)
  daytheme.stop()
  victory.play()
}

function setstarterC(){
  append(selected, 'cyndaquil')
  starterchosen++
  choosesun()
}

function setstarterF(){
  append(selected, 'froakie')
  starterchosen++
  choosesun()
}

function setstarterS(){
  append(selected, 'snivy')
  starterchosen++
  choosesun()
}

function setsunS(){
  append(selected, 'sunkern')
  starterchosen++
  choosecherrybomb()
}

function setsunH(){
  append(selected, 'helioptile')
  starterchosen++
  choosecherrybomb()
}

function setsunL(){
  append(selected, 'larvesta')
  starterchosen++
  choosecherrybomb()
}
```

```
function setcherrybombV(){
  append(selected, "voltorb")
  starterchosen++
  choosewallnut()
}

function setcherrybombP(){
  append(selected, "pineco")
  starterchosen++
  choosewallnut()
}

function setwallnutB(){
  append(selected, 'bonsly')
  starterchosen++
}

function setwallnutW(){
  append(selected, 'wynaut')
  starterchosen++
}

function choosesun(){
  sunkernbutton = createButton('sunkern')
  sunkernbutton.mousePressed(setsunS)
    sunkernbutton.position(scale*20, scale*20)
    helioptilebutton = createButton('helioptile')
  helioptilebutton.mousePressed(setsunH)
    helioptilebutton.position(scale*40, scale*20)
    larvestabutton = createButton('larvesta')
  larvestabutton.mousePressed(setsunL)
    larvestabutton.position(scale*60, scale*20)
}

function choosestarter(){
    cyndaquilbutton = createButton('cyndaquil')
  cyndaquilbutton.mousePressed(setstarterC)
    cyndaquilbutton.position(scale*20, scale*20)
    froakiebutton = createButton('froakie')
  froakiebutton.mousePressed(setstarterF)
    froakiebutton.position(scale*40, scale*20)
    snivybutton = createButton('snivy')
  snivybutton.mousePressed(setstarterS)
```

```
    snivybutton.position(scale*60, scale*20)
  starterchosen++
}

function choosecherrybomb(){
  voltorbbutton = createButton('voltorb')
 voltorbbutton.mousePressed(setcherrybombV)
  voltorbbutton.position(scale*20, scale*20)
  pinecobutton = createButton('pineco')
  pinecobutton.mousePressed(setcherrybombP)
  pinecobutton.position(scale*40, scale*20)
}

function choosewallnut(){
  bonslybutton = createButton('bonsly')
  bonslybutton.mousePressed(setwallnutB)
  bonslybutton.position(scale*20, scale*20)
  wynautbutton = createButton('wynaut')
  wynautbutton.mousePressed(setwallnutW)
  wynautbutton.position(scale*40, scale*20)
}

function changename(){
    if(!namec){
  nameinput = createInput('')
  nameinput.position(scale*4, scale*8)
      namec = true
  }
}

function ready(){
  if(starterchosen == 5){
  menu = 1
  readybutton.remove()
  choosetheme.stop()
  daytheme.loop()
  for(let i = 0; i < selected.length; i++){
    button = createButton(selected[i], selected[i])
    append(selectedbuttons, button)
    selectedbuttons[i].position(2*scale, (8 + 4*i)*scale)
    selectedbuttons[i].mousePressed(place)
  }
  zombiespawn = true
  pausebutton = createButton('pause')
  pausebutton.mousePressed(pause)
  pausebutton.position(scale*90, scale*2)
}
}

function place(){
  if(!readyplace){
    placing = this.value()
    readyplace = true;
  }
  else{
    placing = ''
    readyplace = false;
  }
}

function powie(x, y){
  image()
}

class Suncounter {
  constructor(count) {
    this.count = count;
  }
  update(cost) {
    this.count -= cost;
  }
  display() {
    fill(255);
    textSize(4 * scale);
    text(str(this.count), 6 * scale, 5 * scale);
    fill(0);
    textSize(scale);
  }
  getcount(){
    return this.count
  }
}

let suncount = new Suncounter(50);

class Pokemon {
  constructor(
    xsquare,
    ysquare,
    name,
    atkspeed,
    atkcountdown,
    atkpower,
    atkhitbox,
    price,
    paid,
    atkframe,
    sunmaker,
     detonator,
    exist,
     HP,
     wall
  ) {
    this.xsquare = xsquare;
    this.ysquare = ysquare;
    this.name = name;
    this.atkspeed = atkspeed;
    this.atkcountdown = atkcountdown;
    this.atkpower = atkpower;
    this.atkhitbox = atkhitbox;
    this.price = price;
    this.paid = paid;
    this.atkframe = atkframe
    this.sunmaker = sunmaker
      this.detonator = detonator
      this.exist = exist
      this.HP = HP
      this.wall = wall
  }
  display() {
    if (this.name == "snivy") {
      image(
        snivy,
        this.xsquare * 7.8 * scale + scale * 17,
        (this.ysquare - 1) * scale * 7.3 + 2 * scale,
        8 * scale,
        8 * scale
      );
    }
    if (this.name == "froakie") {
      image(
        froakie,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
      if (this.name == "cyndaquil") {
    image(
      cyndaquil,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if (this.name == "sunkern") {
    image(
      sunkern,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if (this.name == "larvesta") {
    image(
      larvesta,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if (this.name == "helioptile") {
    image(
      helioptile,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if(this.name == "voltorb"){
    image(
      voltorb,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if(this.name == "pineco"){
    image(
      pineco,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if(this.name == "bonsly"){
    image(
      bonsly,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
  if(this.name == "wynaut"){
    image(
      wynaut,
      this.xsquare * 7.8 * scale + scale * 17,
      (this.ysquare - 1) * scale * 7.3 + 2 * scale,
      8 * scale,
      8 * scale
    );
  }
}
attack() {
  if (zombies.length > 0 && this.getsunmaker() == false && !this.detonator && !this.wall) {
    for (let i = 0; i < zombies.length; i++) {
      if (this.ysquare == zombies[i].getLane() && this.getatkframe() == false) {
        this.atkcountdown--;
        if (this.atkcountdown == 0) {
          append(
            projectiles,
            new Projectiles(
        this.ysquare,
        this.atkpower,
        this.atkhitbox + this.xsquare * 7.8 * scale + scale * 17,
        this.name,
        false
       )
      );
      projectilesound.play()
      this.atkcountdown = this.atkspeed;
     }
     this.atkframe = true;
    }
   }
   this.atkframe = false;
  }
  else if(this.getsunmaker() == true){
   this.atkcountdown--
   if(this.atkcountdown == 0){
    append(skyfalls, new Skysun(this.xsquare*7.8*scale + scale*21 + random(-scale, scale),
(this.ysquare-1)*scale*7.3 + 7*scale, (this.ysquare-1)*scale*7.3 + 6*scale, 225))
    this.atkcountdown = this.atkspeed
   }
  }
  else if(this.detonator){
   this.atkcountdown--
   if(this.atkcountdown == 0){
    for(let i = 0; i < zombies.length; i++){
     if(abs(zombies[i].getLane() - this.ysquare) <= 1 && abs(zombies[i].getx() -
(this.xsquare*7.8*scale + scale*20.9)) <= 11.7*scale){
      zombies[i].damage(this.atkpower)
     }
    }
    explosionsound.play()
    powiecount = 15
   }
   if(powiecount > 0){
     image(explosion, this.xsquare*7.8*scale + scale*9.2, (this.ysquare - 2)*scale*7.3 +
2*scale, 24*scale, 24*scale)
     powiecount--
    if(powiecount == 0)
     this.exist = false;
    }
   }
  }
getatkcount() {
  return this.atkcountdown;
}
getxsquare() {
  return this.xsquare;
}
getpaid(){
  return this.paid
}
pay(){
  if(this.price <= suncount.getcount()){
  this.paid = true;
  suncount.update(this.price)
  }
  else
   this.exist = false
}
getprice(){
  return this.price
}
getatkframe(){
  return this.atkframe
}
getysquare(){
  return this.ysquare
}
getsunmaker(){
  return this.sunmaker
}
getexist(){
  return this.exist
}
getHP(){
  return this.HP
}
remove(){
  this.exist = false
}
damage(dam){
  this.HP -= dam
}
}

class Zombies {
```

```
constructor(lane, x, type, speed, HP, atk, slow, atkspeed, atkcount) {
  this.lane = lane;
  this.x = x;
  this.type = type;
  this.speed = speed;
  this.HP = HP;
  this.atk = atk
  this.slow = 1
  this.atkspeed = atkspeed
  this.atkcount = atkcount
}
display() {
  if(this.type == 'browncoat')
  image(
    browncoat,
    this.x,
    1.5 * scale + (this.lane - 1) * scale * 7,
    6 * scale,
    10 * scale
  );
  if(this.type == 'conehead')
    image(
    conehead,
    this.x,
    1.5 * scale + (this.lane - 1) * scale * 7,
    6 * scale,
    10 * scale
  );
  if(this.type == 'flag')
    image(
    flag,
    this.x,
    1.5 * scale + (this.lane - 1) * scale * 7,
    6 * scale,
    10 * scale
  );
}
move() {
  this.x -= (this.speed / 100) * scale * this.slow;
}
damage(dam) {
  this.HP -= dam;
}
getHP() {
  return this.HP;
 }
 getLane() {
  return this.lane;
 }
 getx() {
  return this.x;
 }
 attack(){
  if(pokemon.length > 0){
  for(let i = 0; i < pokemon.length; i++){
   if(floor(pokemon[i].getxsquare() * 7.8 * scale + scale * 24) == floor(this.x) && this.lane == pokemon[i].getysquare()){
    this.slow = 0
    if(pokemon[i].getHP() == 0){
      this.slow = 1
      gulp.play()
     }
    this.atkcount--
    if(this.atkcount == 0){
     pokemon[i].damage(this.atk)
     this.atkcount = this.atkspeed
     bite.play()
    }
   }
  }
 }
 }
}

class Skysun {
 constructor(x, ylimit, y, despawn) {
  this.x = x;
  this.ylimit = ylimit;
  this.y = y;
  this.despawn = despawn;
 }
 fall(x) {
  if (this.y >= this.ylimit) {
   this.despawn--;
   if (this.despawn == 0) {
    skyfalls.splice(x, 1);
   }
  } else this.y += scale / 10;
 }
 display() {
  image(sun, this.x - 4 * scale, this.y - 4 * scale, 8 * scale, 8 * scale);
 }
 collect(i) {
  skyfalls.splice(i, 1);
  suncount.update(-25);
 }
 getX() {
  return this.x;
 }
 getY() {
  return this.y;
 }
}

class Projectiles {
 constructor(lane, dmg, xstart, user, poke) {
  this.lane = lane;
  this.dmg = dmg;
  this.xstart = xstart;
  this.user = user;
  this.poke = poke;
 }
 display() {
  if (
   this.user == "snivy" ||
   this.user == "servine" ||
   this.user == "serperior"
  ) {
   image(
    leaf,
    this.xstart,
    (this.lane - 1) * scale * 7.3 + 5 * scale,
    3 * scale,
    3 * scale
   );
  }
  if (
   this.user == "froakie" ||
   this.user == "frogadier" ||
   this.user == "greninja"
  ) {
   image(
      watershuriken,
      this.xstart,
      (this.lane - 1) * scale * 7.3 + 5 * scale,
      3 * scale,
      3 * scale
    );
  }
      if (
    this.user == "cyndaquil" ||
    this.user == "quilava" ||
    this.user == "typhlosion"
  ) {
    image(
      fire,
      this.xstart,
      (this.lane - 1) * scale * 7.3 + 5 * scale,
      3 * scale,
      3 * scale
    );
  }
}
move(q) {
  this.xstart += scale;
}
hit(q) {
  if(this.xstart > scale*100){
    projectiles.splice(q, 1)
  }
  if (zombies.length > 0) {
    for (let i = 0; i < zombies.length; i++) {
      if (
        this.xstart >= zombies[i].getx() &&
        this.lane == zombies[i].getLane()
      ) {
        zombies[i].damage(this.dmg);
        projectiles.splice(q, 1);
        splat.play()
        i += 9999999
      }
    }
  }
}
getpoke() {
  return this.poke;
  }
}

/* browncoat: append(zombies, new Zombies(1, scale*100, "browncoat", 5, 10, 2.5, 1, 15, 15))
append(zombies, new Zombies(1, scale*100, "conehead", 5, 10, 2.5, 1, 15, 15))
append(zombies, new Zombies(1, scale*100, "flag", 6, 10, 2.5, 1, 15, 15))
suncount.update(-10000)
append(pokemon, new Pokemon(1,1,"snivy", 60, 60, 1, 4*scale, 100, false, false, false))
append(pokemon, new Pokemon(2,1,"froakie", 60, 60, 1, 4*scale, 100, false, false, false))
append(pokemon, new Pokemon(3,1,"cyndaquil", 60, 60, 1, 4*scale, 100, false, false, false))
append(pokemon, new Pokemon(1,2,"sunkern", 720, floor(random(120, 240)), 0, 0, 50, false,
false, true))
append(pokemon, new Pokemon(2,2,"helioptile", 720, floor(random(120, 240)), 0, 0, 50, false,
false, true))
append(pokemon, new Pokemon(3,2,"larvesta", 720, floor(random(120, 240)), 0, 0, 50, false,
false, true))
Sunflower will produce the first sunlight in 4 to 8 seconds, and then every 24 seconds
normally.*/

/*useful links
zombie stats:
https://pvzstrategy.fandom.com/wiki/Zombie_Stats
sounds:
https://www.findsounds.com/ISAPI/search.dll?start=31&keywords=plants%20vs&seed=5
*/
```