```
let myHPbar = 0;
  let enemyHPbar = 0;
  let currentEnemy = 0;
  let currentMine = 0;
  let enemyStart = true;
  let myStart = true;
  let moreEnemy = 0;
  let moreMy = 0;
  let imgE;
  let imgM;
  let currentMouse = 0;
  let x = 35;
  let y = 309;
  let enemyAsleep = 0;
  let meAsleep = 0;
  let enemyParalyzed = false;
  let meParalyzed = false;
  let enemyToxiced = false;
  let meToxic = false;
  let enemyStuck = 0;
  let meStuck = 0;
  let enemyStuckType = 0;
  let myStuckType = 0;
  let enemyStuckName= '';
  let myStuckName = '';
  let enemyBurned = false;
  let meBurned = false;
  let enemySwordsDances = 1;
  let mySwordsDances = 1;
  let enemySpecial = 1;
  let mySpecial = 1;
  let enemyAgilitys = 1;
  let myAgilitys = 1;
  let enemyPoison = false;
  let myPoison = false;
  let enemyConfusion = 0;
  let myConfusion = 0;
  let enemyDef = 1;
  let myDef = 1;
  let moveUsed = '';
  let enemyFast = false;
  let xy = 0;
  let yz = 0;
  let knockedOut = false;
  let myToxicCount = 0;
  let enemyToxicCount = 0;
  let myFrozen = 0;
  let enemyFrozen = 0;
  let myRecharge = 0;
  let enemyRecharge = 0;
  let crit = 0.2;

function setup() {
  createCanvas(400, 400);
  background(245);
  fill(245);
  rect(0, 300, 360, 80);
  fill(0, 150, 0);
  ellipse(100, 270, 150, 50);
  ellipse(270, 130, 150, 50)
  fill(0);
  rect(370, 200, 5, 70);
  rect(190, 270, 180, 5);
}

function gotEmAllEnemy(data) {
  let randPokemonE = data.results[enemyTeam[currentEnemy].getDex()-1];
  loadJSON(randPokemonE.url, gotOneEnemy);
}
function gotEmAllMy(data) {
  let randPokemonM = data.results[myTeam[currentMine].getDex()-1];
  loadJSON(randPokemonM.url, gotOneMy);
}

function gotOneEnemy(data) {
  if (moreEnemy > 0){
    imgE.remove();
  }
  imgE = createImg(data.sprites.front_default, data.name);
  imgE.style("height", 200 + "px");
  imgE.position(170, -20);
  moreEnemy++;
}
function gotOneMy(data) {
  if (moreMy > 0){
    imgM.remove();
  }
  imgM = createImg(data.sprites.back_default, data.name);
  imgM.style("height", 200 + "px");
  imgM.position(0, 110);
  moreMy++;
}

function draw() {
  stroke(0);
  fill(0, 200, 0);
  rect(260, 210, 100, 10);
  rect(60, 60, 100, 10);
  fill(200, 0, 0);
  rect(360-myHPbar, 210, myHPbar, 10);
  rect(160-enemyHPbar, 60, enemyHPbar, 10);
  fill(245);
  stroke(245);
  rect(0, 60, 59, 10);
  rect(200, 210, 59, 10)
  fill(0);
  textSize(11);
  text('HP:', 40, 70);
  text('HP:', 240, 220);
  if (enemyBurned){
    fill(200, 0, 0);
    rect(0, 59, 39, 12);
    fill(255);
    text('BRN', 10, 69);
  }
  if (enemyParalyzed){
    fill(200, 200, 0);
    rect(0, 59, 39, 12);
    fill(255);
    text('PAR', 10, 69);
  }
  if (enemyAsleep > 0){
    fill(150);
    rect(0, 59, 39, 12);
    fill(255);
    text('SLP', 10, 69);
  }
  if (enemyToxiced){
    fill(200, 0, 170);
    rect(0, 59, 39, 12);
    fill(255);
    text('TOX', 10, 69);
```

```
}
if (enemyPoison){
  fill(200, 0, 170);
  rect(0, 59, 39, 12);
  fill(255);
  text('PSN', 10, 69);
}
if (enemyConfusion > 0){
  fill(150);
  rect(0, 59, 39, 12);
  fill(255);
  text('CON', 10, 69);
}
if (enemyFrozen > 0){
  fill(200, 200, 255);
  rect(0, 59, 39, 12);
  fill(255);
  text('FRZ', 10, 69);
}
if (meBurned){
  fill(200, 0, 0);
  rect(200, 209, 39, 12);
  fill(255);
  text('BRN', 210, 219);
}
if (meParalyzed){
  fill(200, 200, 0);
  rect(200, 209, 39, 12);
  fill(255);
  text('PAR', 210, 219);
}
if (meAsleep > 0){
  fill(150);
  rect(200, 209, 39, 12);
  fill(255);
  text('SLP', 210, 219);
}
if (meToxic){
  fill(200, 0, 170);
  rect(200, 209, 39, 12);
  fill(255);
  text('TOX', 210, 219);
}
if (myPoison){
  fill(200, 0, 170);
  rect(200, 209, 39, 12);
  fill(255);
  text('PSN', 210, 219);
}
if (myConfusion > 0){
  fill(150);
  rect(200, 209, 39, 12);
  fill(255);
  text('CON', 210, 219);
}
if (myFrozen > 0){
  fill(200, 200, 255);
  rect(200, 209, 39, 12);
  fill(255);
  text('FRZ', 210, 219);
}
fill(0);
textSize(25);
text(enemyTeam[currentEnemy].getName(), 20, 30);
text(':L'+enemyTeam[currentEnemy].getLvl(), 70, 55);
text(myTeam[currentMine].getName(), 230, 180);
text(':L'+myTeam[currentMine].getLvl(), 290, 205);
if (currentMouse == 0){
fill(0);
text('Fight', 50, 330);
text('Item', 50, 370);
text('Pokemon', 240, 330);
text('Run', 240, 370);
knockedOut = false;
}
if (currentMouse < 2){
beginShape();
vertex(x, y);
vertex(x, y+26);
vertex(x+10, y+13);
endShape();
if (mouseX > 200 && mouseY < 335){
 fill(245);
 rect(34, 308, 11, 70);
 rect(224, 338, 11, 40);
 x = 225;
 y = 309;
}
```

```
if (mouseX < 200 && mouseY < 335){
 fill(245);
 rect(34, 338, 11, 40);
 rect(224, 308, 11, 70);
 x = 35;
 y = 309;
}
if (mouseX > 200 && mouseY > 335){
 fill(245);
 rect(34, 308, 11, 70);
 rect(224, 308, 11, 28);
 x = 225;
 y = 349;
}
if (mouseX < 200 && mouseY > 335){
 fill(245);
 rect(34, 308, 11, 28);
 rect(224, 308, 11, 70);
 x = 35;
 y = 349;
}
}
if (currentMouse == 1){
  fill(0);
  text(myTeam[currentMine].getMove1(), 50, 330);
  text(myTeam[currentMine].getMove2(), 50, 370);
  text(myTeam[currentMine].getMove3(), 240, 330);
  text(myTeam[currentMine].getMove4(), 240, 370);
}
if (currentMouse == 2){
  if (myConfusion > 0 && Math.random() > 0.5){
    enemyAttack(40, 0, false, false, 1);
    fill(0);
    text(myTeam[currentMine].getName() + ' hit itself', 50, 330);
    text('in confusion!', 50, 360);
    currentMouse = 14;
  }
  else if (enemyConfusion > 0 && Math.random() > 0.5){
    myAttack(40, 0, false, false, 1);
    fill(0);
    text(enemyTeam[currentEnemy].getName() + ' hit itself', 50, 330);
    text('in confusion!', 50, 360);
    currentMouse = 13;
  }
```

```
    else if (meStuck > 0){
      fill(0);
      text(enemyTeam[currentEnemy].getName() + ' continues the', 50, 330);
      text(myStuckName + '!', 50, 360);
      enemyAttack(15, myStuckType, false, false, 1);
      currentMouse = 9;
      meStuck--;
    }
    else if (enemyStuck > 0){
      fill(0);
      text(myTeam[currentMine].getName() + ' continues the', 50, 330);
      text(enemyStuckName + '!', 50, 360);
      myAttack(15, enemyStuckType, false, false, 1);
      currentMouse = 9;
      enemyStuck--;
    }
    else if ((meParalyzed && !(enemyParalyzed)) || meAsleep > 0 || myFrozen > 0 || myRecharge
> 0){
      xy = (int)(Math.random()*4)+1;
      getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
      moveUsed = enemyTeam[currentEnemy].getMove(xy);
      fill(0);
      text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
      currentMouse = 13;
    }
    else if((!(meParalyzed) && enemyParalyzed) || enemyAsleep > 0 || enemyFrozen > 0 ||
enemyRecharge > 0){
      xy = (int)(Math.random()*4)+1;
      getMyAttack(myTeam[currentMine].getMove(yz));
      moveUsed = myTeam[currentMine].getMove(yz);
      fill(0);
      text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
      currentMouse = 14;
    }
    else{
    xy = (int)(Math.random()*4)+1;
    if (yz == 1){
    if (enemyFast){
      getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
      moveUsed = enemyTeam[currentEnemy].getMove(xy);
      fill(0);
      text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
    }
    else{
```

```
  getMyAttack(myTeam[currentMine].getMove1());
  moveUsed = myTeam[currentMine].getMove1();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
}
}
if (yz == 2){
if (enemyFast){
  getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
  moveUsed = enemyTeam[currentEnemy].getMove(xy);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
}
else{
  getMyAttack(myTeam[currentMine].getMove2());
  moveUsed = myTeam[currentMine].getMove2();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
}
}
if (yz == 3){
if (enemyFast){
  getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
  moveUsed = enemyTeam[currentEnemy].getMove(xy);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
}
else{
  getMyAttack(myTeam[currentMine].getMove3());
  moveUsed = myTeam[currentMine].getMove3();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
}
}
if (yz == 4){
if (enemyFast){
  getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
  moveUsed = enemyTeam[currentEnemy].getMove(xy);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
}
else{
  getMyAttack(myTeam[currentMine].getMove4());
  moveUsed = myTeam[currentMine].getMove4();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
 }
 }
 currentMouse = 12;
}
}
if (currentMouse == 3){
 fill(0);
 text('Sorry, no items!', 50, 330);
}
if (currentMouse == 5){
 fill(0);
 text('Cant switch yet, my bad.', 50, 330);
}
if (currentMouse == 7){
 fill(0);
 text('Cant run from a trainer!', 50, 330);
}
if (enemyHPbar >= 100){
 currentEnemy++;
 enemyHPbar = 0;
 fill(245);
 rect(0, 0, 200, 55);
 enemyStart =true;
 knockedOut = true;
 enemyToxiced = false;
 enemyToxicCount = 0;
 enemyPoison = false;
 enemyAsleep = 0;
 enemyBurned = false;
 enemySwordsDances = 1;
 enemySpecial = 1;
 enemyAgilitys = 1;
 enemyParalyzed = false;
 enemyStuck = 0;
 enemyConfusion = 0;
 enemyDef = 1;
 enemyFrozen = 0;
 enemyRecharge = 0;
}
 if (myHPbar >= 100){
 currentMine++;
 myHPbar = 0;
```

```
  fill(245);
  rect(200, 157, 200, 55);
  myStart = true;
  knockedOut = true;
  meToxic = false;
  myToxicCount = 0;
  myPoison = false;
  meAsleep = 0;
  meBurned = false;
  mySwordsDances = 1;
  mySpecial = 1;
  myAgilitys = 1;
  meParalyzed = false;
  meStuck = 0;
  myConfusion = 0;
  myDef = 1;
  myFrozen = 0;
  myRecharge = 0;
}
if (currentEnemy >= 6){
fill(245);
rect(1, 300, 400, 80);
fill(0);
textSize(35)
text('You defeated Trainer!', 30, 325);
noLoop();
}
if (currentMine >= 6){
fill(245);
rect(1, 300, 400, 80);
fill(0);
textSize(35)
text('You lost to Trainer...', 30, 325);
noLoop();
}
if (enemyStart){
loadJSON("https://pokeapi.co/api/v2/pokemon/?offset=0&limit=151", gotEmAllEnemy);
  enemyStart = false;
}
if (myStart){
loadJSON("https://pokeapi.co/api/v2/pokemon/?offset=0&limit=151", gotEmAllMy);
  myStart = false;
}
```

```
  if (enemyTeam[currentEnemy].getSpd()*enemyAgilitys >
myTeam[currentMine].getSpd()*myAgilitys){
    enemyFast = true;
  }
  else{
    enemyFast = false;
  }
}

  function mouseClicked(){
    if (currentMouse == 3){
      fill(245);
      rect(1, 300, 400, 80);
      currentMouse = 4;
    }
    if (currentMouse == 5){
      fill(245);
      rect(1, 300, 400, 80);
      currentMouse = 6;
    }
    if (currentMouse == 7){
      fill(245);
      rect(1, 300, 400, 80);
      currentMouse = 8;
    }
    if (currentMouse == 10){
      fill(245);
      rect(1, 300, 400, 80);
      if (yz == 1){
      if (enemyFast){
      getMyAttack(myTeam[currentMine].getMove1());
      moveUsed = myTeam[currentMine].getMove1();
      fill(0);
      text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
    }
    else{
      getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
      moveUsed = enemyTeam[currentEnemy].getMove(xy);
      fill(0);
      text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
    }
    }
      if (yz == 2){
      if (enemyFast){
```

```
  getMyAttack(myTeam[currentMine].getMove2());
  moveUsed = myTeam[currentMine].getMove2();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
}
else{
  getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
  moveUsed = enemyTeam[currentEnemy].getMove(xy);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
}
}
  if (yz == 3){
  if (enemyFast){
  getMyAttack(myTeam[currentMine].getMove3());
  moveUsed = myTeam[currentMine].getMove3();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
}
else{
  getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
  moveUsed = enemyTeam[currentEnemy].getMove(xy);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
}
}
  if (yz == 4){
  if (enemyFast){
  getMyAttack(myTeam[currentMine].getMove4());
  moveUsed = myTeam[currentMine].getMove4();
  fill(0);
  text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
}
else{
  getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
  moveUsed = enemyTeam[currentEnemy].getMove(xy);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
}
}
  currentMouse = 11;
}
if (currentMouse == 12){
if (knockedOut){
      currentMouse = 9;
    }
    else{
      currentMouse = 10;
    }
  }
    if(mouseX < 200 && mouseY < 335 && currentMouse == 1){
      fill(245);
      rect(1, 300, 400, 80);
      yz = 1;
      currentMouse = 2;
    }
    if(mouseX < 200 && mouseY > 335 && currentMouse == 1){
      fill(245);
      rect(1, 300, 400, 80);
      yz = 2;
      currentMouse = 2;
    }
    if(mouseX > 200 && mouseY < 335 && currentMouse == 1){
      fill(245);
      rect(1, 300, 400, 80);
      yz = 3;
      currentMouse = 2;
    }
    if(mouseX > 200 && mouseY > 335 && currentMouse == 1){
      fill(245);
      rect(1, 300, 400, 80);
      yz = 4;
      currentMouse = 2;
    }
    if (mouseX < 200 && mouseY < 335 && currentMouse == 0){
      fill(245);
      rect(1, 300, 400, 80);
      currentMouse = 1;
    }
    if (mouseX < 200 && mouseY > 335 && currentMouse == 0){
      fill(245);
      rect(1, 300, 400, 80);
      currentMouse = 3;
    }
    if (mouseX > 200 && mouseY < 335 && currentMouse == 0){
      fill(245);
      rect(1, 300, 400, 80);
      currentMouse = 5;
```

```
}
if (mouseX > 200 && mouseY > 335 && currentMouse == 0){
  fill(245);
  rect(1, 300, 400, 80);
  currentMouse = 7;
}
if (currentMouse == 4){
  currentMouse = 0;
}
if (currentMouse == 6){
  currentMouse = 0;
}
if (currentMouse == 8){
  currentMouse = 0;
}
if (currentMouse == 9){
  fill(245);
  rect(1, 300, 400, 80);
  if (enemyToxiced){
    enemyToxicCount++;
    enemyHPbar += 6*enemyToxicCount;
  }
  if (meToxic){
    myToxicCount++;
    myHPbar += 6*myToxicCount;
  }
  if (enemyPoison){
    enemyHPbar += 12;
  }
  if (myPoison){
    myHPbar += 12;
  }
  if (enemyBurned){
    enemyHPbar += 12;
  }
  if (meBurned){
    myHPbar += 12;
  }
  currentMouse = 0;
}
if (currentMouse == 11){
  if (enemyConfusion > 0){
    enemyConfusion--;
    if (enemyConfusion == 0){
     fill(245);
     rect(1, 300, 400, 80);
     fill(0);
     text(enemyTeam[currentEnemy].getName() + ' can focus again!', 50, 330);
    }
   }
   if (myConfusion > 0){
    myConfusion--;
    if (myConfusion == 0){
     fill(245);
     rect(1, 300, 400, 80);
     fill(0);
     text(myTeam[currentMine].getName() + ' can focus again!', 50, 360);
    }
   }
   currentMouse = 9;
  }
  if (currentMouse == 13){
   fill(245);
   rect(1, 300, 400, 80);
   if (meAsleep > 0){
    fill(0);
    text(myTeam[currentMine].getName() + ' is asleep!', 50, 330);
    meAsleep--;
   }
   else if (myFrozen > 0){
    fill(0);
    text(myTeam[currentMine].getName() + ' is frozen!', 50, 330);
    myFrozen--;
   }
   else if (myRecharge > 0){
    fill(0);
    text(myTeam[currentMine].getName() + ' must recharge!', 50, 330);
    myRecharge--;
   }
   else if (meParalyzed && Math.random() >= 0.75){
    fill(0);
    text(myTeam[currentMine].getName() + ' is paralyzed!', 50, 330);
    text('It cant move!', 50, 360);
   }
   else if (myConfusion > 0 && Math.random() > 0.5){
   enemyAttack(40, 0, false, false, 1);
   fill(0);
   text(myTeam[currentMine].getName() + ' hit itself', 50, 330);
  text('in confusion!', 50, 360);
}
  else{
    getMyAttack(myTeam[currentMine].getMove(yz));
    moveUsed = myTeam[currentMine].getMove(yz);
    fill(0);
    text(myTeam[currentMine].getName() + ' used ' + moveUsed + '!', 50, 330);
  }
  currentMouse = 11;
}
if (currentMouse == 14){
  xy = (int)(Math.random()*4)+1;
  fill(245);
  rect(1, 300, 400, 80);
  if (enemyAsleep > 0){
    fill(0);
    text(enemyTeam[currentEnemy].getName() + ' is asleep!', 50, 330);
    enemyAsleep--;
  }
  else if (enemyFrozen > 0){
    fill(0);
    text(enemyTeam[currentEnemy].getName() + ' is frozen!', 50, 330);
    enemyFrozen--;
  }
  else if (enemyRecharge > 0){
    fill(0);
    text(enemyTeam[currentEnemy].getName() + ' must recharge!', 50, 330);
    enemyRecharge--;
  }
  else if (enemyParalyzed && Math.random() >= 0.75){
    fill(0);
    text(enemyTeam[currentEnemy].getName() + ' is paralyzed!', 50, 330);
    text('It cant move!', 50, 360);
  }
  else if (enemyConfusion > 0 && Math.random() > 0.5){
  myAttack(40, 0, false, false, 1);
  fill(0);
  text(enemyTeam[currentEnemy].getName() + ' hit itself', 50, 330);
  text('in confusion!', 50, 360);
}
  else{
    getEnemyAttack(enemyTeam[currentEnemy].getMove(xy));
    moveUsed = enemyTeam[currentEnemy].getMove(xy);
    fill(0);
```

```
        text(enemyTeam[currentEnemy].getName() + ' used ' + moveUsed + '!', 50, 330);
      }
      currentMouse = 11;
    }
  }

function myAttack(power, type, range, critUp, acc){
  let damage = 0;
  if (Math.random() > acc){
    fill(0);
    text(myTeam[currentMine].getName() + ' missed!', 50, 330);
  }
  else{
  if (range){
    damage +=
(int)(((2*myTeam[currentMine].getLvl())/5+2)*power*((myTeam[currentMine].getSpatk()*mySpeci
al)/(enemyTeam[currentEnemy].getSpdef() * enemySpecial)/50))+2;
  }
  else{
  damage +=
(((2*myTeam[currentMine].getLvl())/5+2)*power*((myTeam[currentMine].getAtk()*mySwordsDan
ces)/(enemyTeam[currentEnemy].getDef()* enemyDef)/50))+2;
  }
  if (supereffectiveType1(type)){
    damage *= 2;
  }
  if (supereffectiveType2(type)){
    damage *= 2;
  }
  if (resistsType1(type)){
    damage /= 2;
  }
  if (resistsType2(type)){
    damage /= 2;
  }
  if (immuneType1(type)){
    damage = 0;
  }
  if (immuneType2(type)){
    damage = 0;
  }
  if (meBurned && !(range)){
    damage /= 2;
  }
  if (critUp){
    crit = 0.7;
  }
  if (Math.random() < crit){
    damage *= 1.5;
  }
    if (type == myTeam[currentMine].getType1() || type == myTeam[currentMine].getType2()){
      damage *= 1.5;
    }
  enemyHPbar += damage * (100/enemyTeam[currentEnemy].getHP());
    crit = 0.2;
}
}
function enemyAttack(power, type, range, critUp, acc){
  let damage = 0;
  if (Math.random() > acc){
    fill(0);
    text(enemyTeam[currentEnemy].getName() + ' missed!', 50, 330);
  }
  else{
  if (range){
    damage +=
(int)(((2*enemyTeam[currentEnemy].getLvl())/5+2)*power*((enemyTeam[currentEnemy].getSpat
k()*enemySpecial)/(myTeam[currentMine].getSpdef()*mySpecial))/50)+2;
  }
  else{
  damage +=
(((2*enemyTeam[currentEnemy].getLvl())/5+2)*power*((enemyTeam[currentEnemy].getAtk()*en
emySwordsDances)/(myTeam[currentMine].getDef()*myDef))/50)+2;
  }
  if (enemySupereffectiveType1(type)){
    damage *= 2;
  }
  if (enemySupereffectiveType2(type)){
    damage *= 2;
  }
  if (enemyResistsType1(type)){
    damage /= 2;
  }
  if (enemyResistsType2(type)){
    damage /= 2;
  }
  if (enemyImmuneType1(type)){
    damage = 0;
  }
  if (enemyImmuneType2(type)){
    damage = 0;
  }
  if (enemyBurned && !(range)){
    damage /= 2;
  }
  if (critUp){
    crit = 0.7;
  }
  if (Math.random() < crit){
    damage *= 1.5;
  }
  if (type == enemyTeam[currentEnemy].getType1() || type ==
enemyTeam[currentEnemy].getType2()){
    damage *= 1.5;
  }
  myHPbar += damage * (100/myTeam[currentMine].getHP());
  crit = 0.2;
}
}

function getMyAttack(attackName){
  if (attackName == 'Sleep Powder'){
      mySleepPowder();
      }
  if (attackName == 'Razor Leaf'){
      myRazorLeaf();
      }
  if (attackName == 'Body Slam'){
      myBodySlam();
      }
  if (attackName == 'Toxic'){
      myToxic();
      }
  if (attackName == 'Fire Spin'){
      myFireSpin();
      }
  if (attackName == 'Fire Blast'){
      myFireBlast();
      }
  if (attackName == 'Earthquake'){
      myEarthquake();
      }
```

```
if (attackName == 'Swords Dance'){
    mySwordsDance();
    }
if (attackName == 'Surf'){
    mySurf();
    }
if (attackName == 'Blizzard'){
    myBlizzard();
    }
if (attackName == 'Rest'){
    myRest();
    }
if (attackName == 'Stun Spore'){
    myStunSpore();
    }
if (attackName == 'Psychic'){
    myPsychic();
    }
if (attackName == 'Mega Drain'){
    myMegaDrain();
    }
if (attackName == 'Agility'){
    myAgility();
    }
if (attackName == 'Twineedle'){
    myTwineedle();
    }
if (attackName == 'Hyper Beam'){
    myHyperBeam();
    }
if (attackName == 'Double-Edge'){
    myDoubleEdge();
    }
if (attackName == 'Mirror Move'){
    myMirrorMove();
    }
if (attackName == 'Super Fang'){
    mySuperFang();
    }
if (attackName == 'Thunderbolt'){
    myThunderbolt();
    }
if (attackName == 'Drill Peck'){
    myDrillPeck();
    }
if (attackName == 'Glare'){
    myGlare();
    }
if (attackName == 'Wrap'){
    myWrap();
    }
if (attackName == 'Thunder Wave'){
    myThunderWave();
    }
if (attackName == 'Rock Slide'){
    myRockSlide();
    }
if (attackName == 'Confuse Ray'){
    myConfuseRay();
    }
if (attackName == 'Counter'){
    myCounter();
    }
if (attackName == 'Screech'){
    myScreech();
    }
if (attackName == 'Spore'){
    mySpore();
    }
if (attackName == 'Slash'){
    mySlash();
    }
if (attackName == 'Amnesia'){
    myAmnesia();
    }
if (attackName == 'Hydro Pump'){
    myHydroPump();
    }
if (attackName == 'Submission'){
    mySubmission();
    }
if (attackName == 'Reflect'){
    myReflect();
    }
if (attackName == 'Recover'){
    myRecover();
    }
if (attackName == 'Explosion'){
    myExplosion();
    }
if (attackName == 'Sludge'){
    mySludge();
    }
if (attackName == 'Clamp'){
    myClamp();
    }
if (attackName == 'Hypnosis'){
    myHypnosis();
    }
if (attackName == 'Night Shade'){
    myNightShade();
    }
if (attackName == 'Crabhammer'){
    myCrabhammer();
    }
if (attackName == 'Thunder'){
    myThunder();
    }
if (attackName == 'Meditate'){
    myMeditate();
    }
if (attackName == 'High Jump Kick'){
    myHighJumpKick();
    }
if (attackName == 'Seismic Toss'){
    mySeismicToss();
    }
if (attackName == 'Soft-Boiled'){
    mySoftBoiled();
    }
if (attackName == 'Growth'){
    myGrowth();
    }
if (attackName == 'Lovely Kiss'){
    myLovelyKiss();
    }
if (attackName == 'Transform'){
    myTransform();
    }
if (attackName == 'Double Kick'){
    myDoubleKick();
    }
```

```
}

function getEnemyAttack(attackName){
  if (attackName == 'Sleep Powder'){
      enemySleepPowder();
      }
  if (attackName == 'Razor Leaf'){
      enemyRazorLeaf();
      }
  if (attackName == 'Body Slam'){
      enemyBodySlam();
      }
  if (attackName == 'Toxic'){
      enemyToxic();
      }
  if (attackName == 'Fire Spin'){
      enemyFireSpin();
      }
  if (attackName == 'Fire Blast'){
      enemyFireBlast();
      }
  if (attackName == 'Earthquake'){
      enemyEarthquake();
      }
  if (attackName == 'Swords Dance'){
      enemySwordsDance();
      }
  if (attackName == 'Surf'){
      enemySurf();
      }
  if (attackName == 'Blizzard'){
      enemyBlizzard();
      }
  if (attackName == 'Rest'){
      enemyRest();
      }
  if (attackName == 'Stun Spore'){
      enemyStunSpore();
      }
  if (attackName == 'Psychic'){
      enemyPsychic();
      }
  if (attackName == 'Mega Drain'){
      enemyMegaDrain();
```

```
    }
if (attackName == 'Agility'){
    enemyAgility();
    }
if (attackName == 'Twineedle'){
    enemyTwineedle();
    }
if (attackName == 'Hyper Beam'){
    enemyHyperBeam();
    }
if (attackName == 'Double-Edge'){
    enemyDoubleEdge();
    }
if (attackName == 'Mirror Move'){
    enemyMirrorMove();
    }
if (attackName == 'Super Fang'){
    enemySuperFang();
    }
if (attackName == 'Thunderbolt'){
    enemyThunderbolt();
    }
if (attackName == 'Drill Peck'){
    enemyDrillPeck();
    }
if (attackName == 'Glare'){
    enemyGlare();
    }
if (attackName == 'Wrap'){
    enemyWrap();
    }
if (attackName == 'Thunder Wave'){
    enemyThunderWave();
    }
if (attackName == 'Rock Slide'){
    enemyRockSlide();
    }
if (attackName == 'Confuse Ray'){
    enemyConfuseRay();
    }
if (attackName == 'Counter'){
    enemyCounter();
    }
if (attackName == 'Screech'){
    enemyScreech();
    }
if (attackName == 'Spore'){
    enemySpore();
    }
if (attackName == 'Slash'){
    enemySlash();
    }
if (attackName == 'Amnesia'){
    enemyAmnesia();
    }
if (attackName == 'Hydro Pump'){
    enemyHydroPump();
    }
if (attackName == 'Submission'){
    enemySubmission();
    }
if (attackName == 'Reflect'){
    enemyReflect();
    }
if (attackName == 'Recover'){
    enemyRecover();
    }
if (attackName == 'Explosion'){
    enemyExplosion();
    }
if (attackName == 'Sludge'){
    enemySludge();
    }
if (attackName == 'Clamp'){
    enemyClamp();
    }
if (attackName == 'Hypnosis'){
    enemyHypnosis();
    }
if (attackName == 'Night Shade'){
    enemyNightShade();
    }
if (attackName == 'Crabhammer'){
    enemyCrabhammer();
    }
if (attackName == 'Thunder'){
    enemyThunder();
    }
```

```
  if (attackName == 'Meditate'){
      enemyMeditate();
      }
  if (attackName == 'High Jump Kick'){
      enemyHighJumpKick();
      }
  if (attackName == 'Seismic Toss'){
      enemySeismicToss();
      }
  if (attackName == 'Soft-Boiled'){
      enemySoftBoiled();
      }
  if (attackName == 'Growth'){
      enemyGrowth();
      }
  if (attackName == 'Lovely Kiss'){
      enemyLovelyKiss();
      }
  if (attackName == 'Transform'){
      enemyTransform();
      }
  if (attackName == 'Double Kick'){
      enemyDoubleKick();
      }
}

class Pokemon{
   constructor(name, type1, type2, HP, atk, def, spatk, spdef, spd, lvl, dex, move1, move2,
move3, move4){
   this.name = name;
   this.type1 = type1;
   this.type2 = type2;
   this.HP = HP;
   this.atk = atk;
   this.def = def;
   this.spatk = spatk;
   this.spdef = spdef;
   this.spd = spd;
   this.lvl = lvl;
   this.dex = dex;
   this.move1 = move1;
   this.move2 = move2;
   this.move3 = move3;
   this.move4 = move4;
```

```
}

getName(){
  return this.name;
}

getType1(){
  return this.type1;
}

getType2(){
  return this.type2;
}

getHP(){
  return this.HP;
}

getAtk(){
  return this.atk;
}

getDef(){
  return this.def;
}

getSpatk(){
  return this.spatk;
}

getSpdef(){
  return this.spdef;
}

getSpd(){
  return this.spd;
}

getLvl(){
  return this.lvl;
}

getDex(){
  return this.dex;
  }

  getMove1(){
    return this.move1;
  }

  getMove2(){
    return this.move2;
  }

  getMove3(){
    return this.move3;
  }

  getMove4(){
    return this.move4;
  }

  getMove(num){
    if (num == 1){
      return this.move1;
    }
    if (num == 2){
      return this.move2;
    }
    if (num == 3){
      return this.move3;
    }
    if (num == 4){
      return this.move4;
    }
  }

  setHP(newHP){
    this.HP = newHP;
  }
}

 var allPokemon = [
 Bulbasaur = new Pokemon('Bulbasaur', 6, 7, 222, 137, 137, 165, 165, 129, 88, 1, 'Sleep
Powder', 'Razor Leaf', 'Body Slam', 'Toxic'),
 Ivysaur = new Pokemon('Ivysaur', 6, 7, 238, 152, 154, 183, 183, 149, 84, 2, 'Sleep Powder',
'Razor Leaf', 'Body Slam', 'Toxic'),
  Venusaur = new Pokemon('Venusaur', 6, 7, 250, 171, 172, 199, 199, 168, 77, 3, 'Sleep
Powder', 'Razor Leaf', 'Body Slam', 'Toxic'),
  Charmander = new Pokemon('Charmander', 2, 0, 212, 142, 126, 156, 138, 165, 88, 4, 'Fire
Spin', 'Fire Blast', 'Earthquake', 'Swords Dance'),
  Charmeleon = new Pokemon('Charmeleon', 2, 0, 235, 156, 146, 183, 157, 183, 84, 5, 'Fire
Spin', 'Fire Blast', 'Earthquake', 'Swords Dance'),
  Charizard = new Pokemon('Charizard', 2, 5, 259, 183, 173, 223, 184, 209, 81, 6, 'Fire Spin',
'Fire Blast', 'Earthquake', 'Swords Dance'),
  Squirtle = new Pokemon('Squirtle', 4, 0, 221, 135, 165, 138, 163, 126, 88, 7, 'Surf', 'Blizzard',
'Body Slam', 'Rest'),
  Wartortle = new Pokemon('Wartortle', 4, 0, 236, 154, 183, 157, 183, 146, 84, 8, 'Surf',
'Blizzard', 'Body Slam', 'Rest'),
  Blastoise = new Pokemon('Blastoise', 4, 0, 261, 181, 209, 184, 217, 173, 81, 9, 'Surf',
'Blizzard', 'Body Slam', 'Rest'),
  Caterpie = new Pokemon('Caterpie', 13, 0, 222, 103, 112, 85, 85, 129, 88, 10, 'Sleep Powder',
'Stun Spore', 'Psychic', 'Mega Drain'),
  Metapod = new Pokemon('Metapod', 13, 0, 221, 82, 141, 90, 90, 99, 84, 11, 'Sleep Powder',
'Stun Spore', 'Psychic', 'Mega Drain'),
  Butterfree = new Pokemon('Butterfree', 13, 5, 230, 120, 128, 192, 176, 160, 81, 12, 'Sleep
Powder', 'Stun Spore', 'Psychic', 'Mega Drain'),
  Weedle = new Pokemon('Weedle', 13, 7, 214, 112, 103, 85, 85, 138, 88, 13, 'Agility', 'Swords
Dance', 'Twineedle', 'Hyper Beam'),
  Kakuna = new Pokemon('Kakuna', 13, 7, 213, 90, 132, 90, 90, 107, 84, 14, 'Agility', 'Swords
Dance', 'Twineedle', 'Hyper Beam'),
  Beedrill = new Pokemon('Beedrill', 13, 7, 238, 192, 111, 120, 176, 168, 81, 15, 'Agility', 'Swords
Dance', 'Twineedle', 'Hyper Beam'),
  Pidgey = new Pokemon('Pidgey', 1, 5, 214, 129, 121, 112, 112, 149, 88, 16, 'Double-Edge',
'Hyper Beam', 'Mirror Move', 'Agility'),
  Pidgeotto = new Pokemon('Pidgeotto', 1, 5, 243, 149, 141, 132, 132, 167, 84, 17,
'Double-Edge', 'Hyper Beam', 'Mirror Move', 'Agility'),
  Pidgeot = new Pokemon('Pidgeot', 1, 5, 267, 176, 168, 160, 160, 210, 81, 18, 'Double-Edge',
'Hyper Beam', 'Mirror Move', 'Agility'),
  Rattata = new Pokemon('Rattata', 1, 0, 196, 149, 112, 94, 112, 177, 88, 19, 'Super Fang',
'Hyper Beam', 'Body Slam', 'Thunderbolt'),
  Raticate = new Pokemon('Raticate', 1, 0, 222, 178, 144, 128, 160, 204, 81, 20, 'Super Fang',
'Hyper Beam', 'Body Slam', 'Thunderbolt'),
  Spearow = new Pokemon('Spearow', 1, 5, 214, 156, 103, 105, 105, 173, 88, 21, 'Drill Peck',
'Double-Edge', 'Hyper Beam', 'Agility'),
  Fearow = new Pokemon('Fearow', 1, 5, 238, 192, 152, 145, 145, 209, 81, 22, 'Drill Peck',
'Double-Edge', 'Hyper Beam', 'Agility'),
  Ekans = new Pokemon('Ekans', 7, 0, 205, 156, 128, 121, 145, 147, 88, 23, 'Glare', 'Wrap',
'Earthquake', 'Hyper Beam'),
  Arbok = new Pokemon('Arbok', 7, 0, 230, 201, 158, 152, 175, 176, 81, 24, 'Glare', 'Wrap',
'Earthquake', 'Hyper Beam'),
  Pikachu = new Pokemon('Pikachu', 8, 0, 205, 147, 121, 138, 138, 209, 88, 25, 'Thunderbolt',
'Surf', 'Thunder Wave', 'Agility'),
  Raichu = new Pokemon('Raichu', 8, 0, 227, 190, 134, 190, 174, 222, 80, 26, 'Thunderbolt',
'Surf', 'Thunder Wave', 'Agility'),
  Sandshrew = new Pokemon('Sandshrew', 9, 0, 231, 182, 200, 85, 103, 121, 88, 27, 'Swords
Dance', 'Earthquake', 'Hyper Beam', 'Rock Slide'),
  Sandslash = new Pokemon('Sandslash', 9, 0, 254, 209, 225, 120, 136, 152, 81, 28, 'Swords
Dance', 'Earthquake', 'Hyper Beam', 'Rock Slide'),
  NidoranF = new Pokemon('NidoranF', 7, 0, 240, 133, 142, 121, 121, 122, 88, 29, 'Earthquake',
'Thunderbolt', 'Fire Blast', 'Body Slam'),
  Nidorina = new Pokemon('Nidorina', 7, 0, 255, 152, 161, 141, 141, 142, 84, 30, 'Earthquake',
'Thunderbolt', 'Fire Blast', 'Body Slam'),
  Nidoqueen = new Pokemon('Nidoqueen', 7, 9, 278, 196, 188, 168, 184, 170, 81, 31,
'Earthquake', 'Thunderbolt', 'Fire Blast', 'Body Slam'),
  NidoranM = new Pokemon('NidoranM', 7, 0, 224, 151, 121, 121, 121, 138, 88, 32,
'Earthquake', 'Thunderbolt', 'Fire Blast', 'Body Slam'),
  Nidorino = new Pokemon('Nidorino', 7, 0, 240, 169, 144, 141, 141, 157, 84, 33, 'Earthquake',
'Thunderbolt', 'Fire Blast', 'Body Slam'),
  Nidoking = new Pokemon('Nidoking', 7, 9, 264, 212, 171, 184, 168, 184, 81, 34, 'Earthquake',
'Thunderbolt', 'Fire Blast', 'Body Slam'),
  Clefairy = new Pokemon('Clefairy', 1, 0, 255, 124, 129, 149, 157, 107, 84, 35, 'Thunder Wave',
'Body Slam', 'Thunderbolt', 'Hyper Beam'),
  Clefable = new Pokemon('Clefable', 1, 0, 287, 160, 165, 201, 192, 144, 81, 36, 'Thunder
Wave', 'Body Slam', 'Thunderbolt', 'Hyper Beam'),
  Vulpix = new Pokemon('Vulpix', 2, 0, 210, 122, 121, 138, 165, 165, 88, 37, 'Fire Spin', 'Fire
Blast', 'Body Slam', 'Confuse Ray'),
  Ninetales = new Pokemon('Ninetales', 2, 0, 251, 170, 168, 178, 209, 209, 81, 38, 'Fire Spin',
'Fire Blast', 'Body Slam', 'Confuse Ray'),
  Jigglypuff = new Pokemon('Jigglypuff', 1, 0, 346, 129, 85, 129, 94, 85, 88, 39, 'Thunder Wave',
'Hyper Beam', 'Blizzard', 'Counter'),
  Wigglytuff = new Pokemon('Wigglytuff', 1, 0, 359, 160, 120, 184, 128, 120, 81, 40, 'Thunder
Wave', 'Hyper Beam', 'Blizzard', 'Counter'),
  Zubat = new Pokemon('Zubat', 7, 5, 214, 129, 112, 103, 121, 147, 88, 41, 'Confuse Ray',
'Double-Edge', 'Screech', 'Hyper Beam'),
  Golbat = new Pokemon('Golbat', 7, 5, 254, 176, 160, 152, 168, 192, 81, 42, 'Confuse Ray',
'Double-Edge', 'Screech', 'Hyper Beam'),
  Oddish = new Pokemon('Oddish', 6, 7, 222, 138, 147, 182, 165, 103, 88, 43, 'Sleep Powder',
'Swords Dance', 'Mega Drain', 'Hyper Beam'),
  Gloom = new Pokemon('Gloom', 6, 7, 238, 157, 166, 191, 174, 115, 84, 44, 'Sleep Powder',
'Swords Dance', 'Mega Drain', 'Hyper Beam'),
  Vileplume = new Pokemon('Vileplume', 6, 7, 254, 176, 184, 225, 192, 128, 81, 45, 'Sleep
Powder', 'Swords Dance', 'Mega Drain', 'Hyper Beam'),
  Paras = new Pokemon('Paras', 13, 6, 205, 173, 147, 129, 147, 94, 88, 46, 'Spore', 'Stun
Spore', 'Swords Dance', 'Hyper Beam'),
  Parasect = new Pokemon('Parasect', 13, 6, 230, 201, 176, 144, 176, 95, 81, 47, 'Spore', 'Stun
Spore', 'Swords Dance', 'Hyper Beam'),
  Venonat = new Pokemon('Venonat', 13, 7, 249, 147, 138, 121, 147, 129, 88, 48, 'Sleep
Powder', 'Stun Spore', 'Psychic', 'Mega Drain'),
  Venomoth = new Pokemon('Venomoth', 13, 7, 246, 152, 144, 192, 168, 192, 81, 49, 'Sleep
Powder', 'Stun Spore', 'Psychic', 'Mega Drain'),
  Diglett = new Pokemon('Diglett', 9, 0, 161, 147, 94, 112, 129, 217, 88, 50, 'Earthquake', 'Rock
Slide', 'Slash', 'Toxic'),
  Dugtrio = new Pokemon('Dugtrio', 9, 0, 180, 199, 122, 122, 152, 229, 77, 51, 'Earthquake',
'Rock Slide', 'Slash', 'Toxic'),
  Meowth = new Pokemon('Meowth', 1, 0, 214, 129, 112, 121, 121, 209, 88, 52, 'Slash', 'Hyper
Beam', 'Thunderbolt', 'Toxic'),
  Persian = new Pokemon('Persian', 1, 0, 227, 152, 137, 145, 145, 222, 77, 53, 'Slash', 'Hyper
Beam', 'Thunderbolt', 'Toxic'),
  Psyduck = new Pokemon('Psyduck', 4, 0, 231, 142, 135, 165, 138, 147, 88, 54, 'Amnesia',
'Hydro Pump', 'Blizzard', 'Rest'),
  Golduck = new Pokemon('Golduck', 4, 0, 259, 177, 171, 198, 174, 182, 80, 55, 'Amnesia',
'Hydro Pump', 'Blizzard', 'Rest'),
  Mankey = new Pokemon('Mankey', 3, 0, 214, 191, 112, 112, 129, 173, 88, 56, 'Submission',
'Body Slam', 'Rock Slide', 'Hyper Beam'),
  Primeape = new Pokemon('Primeape', 3, 0, 238, 217, 144, 144, 160, 201, 81, 57,
'Submission', 'Body Slam', 'Rock Slide', 'Hyper Beam'),
  Growlithe = new Pokemon('Growlithe', 2, 0, 240, 173, 129, 173, 138, 156, 88, 58, 'Agility', 'Fire
Blast', 'Body Slam', 'Hyper Beam'),
  Arcanine = new Pokemon('Arcanine', 2, 0, 278, 225, 176, 209, 176, 201, 81, 59, 'Agility', 'Fire
Blast', 'Body Slam', 'Hyper Beam'),
  Poliwag = new Pokemon('Poliwag', 4, 0, 214, 138, 121, 121, 121, 209, 88, 60, 'Blizzard',
'Amnesia', 'Surf', 'Submission'),
  Poliwhirl = new Pokemon('Poliwhirl', 4, 0, 246, 157, 157, 132, 132, 199, 84, 61, 'Blizzard',
'Amnesia', 'Surf', 'Submission'),
  Poliwrath = new Pokemon('Poliwrath', 4, 3, 275, 198, 198, 158, 190, 158, 80, 62, 'Blizzard',
'Amnesia', 'Surf', 'Submission'),
  Abra = new Pokemon('Abra', 10, 0, 187, 85, 77, 235, 147, 209, 88, 63, 'Psychic', 'Reflect',
'Thunder Wave', 'Recover'),
  Kadabra = new Pokemon('Kadabra', 10, 0, 188, 98, 91, 229, 152, 206, 77, 64, 'Psychic',
'Reflect', 'Thunder Wave', 'Recover'),
  Alazkazam = new Pokemon('Alakazam', 10, 0, 206, 119, 111, 246, 186, 224, 75, 65, 'Psychic',
'Reflect', 'Thunder Wave', 'Recover'),
  Machop = new Pokemon('Machop', 3, 0, 266, 191, 138, 112, 112, 112, 88, 66, 'Submission',
'Earthquake', 'Body Slam', 'Hyper Beam'),
  Machoke = new Pokemon('Machoke', 3, 0, 272, 216, 166, 132, 149, 124, 84, 67, 'Submission',
'Earthquake', 'Body Slam', 'Hyper Beam'),
  Machamp = new Pokemon('Machamp', 3, 0, 278, 257, 176, 152, 184, 136, 81, 68,
'Submission', 'Earthquake', 'Body Slam', 'Hyper Beam'),
  Bellsprout = new Pokemon('Bellsprout', 6, 7, 231, 182, 112, 173, 103, 121, 88, 69, 'Sleep
Powder', 'Wrap', 'Razor Leaf', 'Hyper Beam'),
  Weepinbell = new Pokemon('Weepinbell', 6, 7, 246, 199, 132, 191, 124, 141, 84, 70, 'Sleep
Powder', 'Wrap', 'Razor Leaf', 'Hyper Beam'),
  Victreebel = new Pokemon('Victreebel', 6, 7, 244, 201, 141, 194, 149, 149, 75, 71, 'Sleep
Powder', 'Wrap', 'Razor Leaf', 'Hyper Beam'),
  Tentacool = new Pokemon('Tentacool', 4, 7, 214, 121, 112, 138, 226, 173, 88, 72, 'Surf',
'Blizzard', 'Wrap', 'Rest'),
  Tentacruel = new Pokemon('Tentacruel', 4, 7, 250, 152, 145, 168, 229, 199, 77, 73, 'Surf',
'Blizzard', 'Wrap', 'Rest'),
  Geodude = new Pokemon('Geodude', 11, 9, 214, 191, 226, 103, 103, 85, 88, 74, 'Rock Slide',
'Earthquake', 'Explosion', 'Body Slam'),
  Graveler = new Pokemon('Gravever', 11, 9, 230, 208, 241, 124, 124, 107, 84, 75, 'Rock Slide',
'Earthquake', 'Explosion', 'Body Slam'),
  Golem = new Pokemon('Golem', 11, 9, 262, 241, 257, 136, 152, 120, 81, 76, 'Rock Slide',
'Earthquake', 'Explosion', 'Body Slam'),
  Ponyta = new Pokemon('Ponyta', 2, 0, 231, 200, 147, 165, 165, 209, 88, 77, 'Fire Spin', 'Fire
Blast', 'Agility', 'Toxic'),
  Rapidash = new Pokemon('Rapidash', 2, 0, 238, 209, 160, 176, 176, 217, 81, 78, 'Fire Spin',
'Fire Blast', 'Agility', 'Toxic'),
  Slowpoke = new Pokemon('Slowpoke', 4, 10, 302, 165, 165, 121, 121, 77, 88, 79, 'Amnesia',
'Psychic', 'Thunder Wave', 'Rest'),
  Slowbro = new Pokemon('Slowbro', 4, 10, 266, 156, 209, 194, 164, 89, 75, 80, 'Amnesia',
'Psychic', 'Thunder Wave', 'Rest'),
  Magnemite = new Pokemon('Magnemite', 8, 0, 187, 112, 173, 217, 147, 129, 88, 81, 'Thunder
Wave', 'Thunderbolt', 'Double-Edge', 'Rest'),
  Magneton = new Pokemon('Magneton', 8, 0, 214, 144, 201, 241, 160, 160, 81, 82, 'Thunder
Wave', 'Thunderbolt', 'Double-Edge', 'Rest'),
  Farfetchd = new Pokemon('Farfetchd', 1, 5, 217, 192, 136, 141, 147, 144, 81, 83, 'Agility',
'Swords Dance', 'Slash', 'Body Slam'),
  Doduo = new Pokemon('Doduo', 1, 5, 205, 200, 129, 112, 112, 182, 88, 84, 'Body Slam', 'Drill
Peck', 'Hyper Beam', 'Agility'),
  Dodrio = new Pokemon('Dodrio', 1, 5, 219, 214, 152, 137, 137, 214, 77, 85, 'Body Slam', 'Drill
Peck', 'Hyper Beam', 'Agility'),
  Seel = new Pokemon('Seel', 4, 0, 258, 129, 147, 129, 173, 129, 88, 86, 'Blizzard', 'Surf', 'Body
Slam', 'Rest'),
  Dewgong = new Pokemon('Dewgong', 4, 12, 278, 160, 176, 160, 201, 160, 81, 87, 'Blizzard',
'Surf', 'Body Slam', 'Rest'),
  Grimer = new Pokemon('Grimer', 7, 0, 284, 191, 138, 121, 138, 94, 88, 88, 'Sludge', 'Body
Slam', 'Thunderbolt', 'Explosion'),
  Muk = new Pokemon('Muk', 7, 0, 303, 217, 168, 152, 209, 128, 81, 89, 'Sludge', 'Body Slam',
'Thunderbolt', 'Explosion'),
  Shellder = new Pokemon('Shellder', 4, 0, 196, 165, 226, 129, 94, 121, 88, 90, 'Clamp',
'Blizzard', 'Hyper Beam', 'Explosion'),
  Cloyster = new Pokemon('Cloyster', 4, 12, 199, 186, 314, 171, 111, 149, 75, 91, 'Clamp',
'Blizzard', 'Hyper Beam', 'Explosion'),
  Gastly = new Pokemon('Gastly', 15, 7, 196, 112, 103, 226, 112, 191, 88, 92, 'Hypnosis', 'Night
shade', 'Thunderbolt', 'Explosion'),
  Haunter = new Pokemon('Haunter', 15, 7, 196, 122, 114, 222, 129, 191, 77, 93, 'Hypnosis',
'Night shade', 'Thunderbolt', 'Explosion'),
  Gengar = new Pokemon('Gengar', 15, 7, 214, 141, 134, 239, 156, 209, 75, 94, 'Hypnosis',
'Night shade', 'Thunderbolt', 'Explosion'),
  Onix = new Pokemon('Onix', 11, 9, 189, 120, 306, 95, 120, 160, 81, 95, 'Wrap', 'Toxic',
'Earthquake', 'Explosion'),
  Drowzee = new Pokemon('Drowzee', 10, 0, 249, 135, 129, 126, 209, 124, 88, 96, 'Psychic',
'Rest', 'Thunder Wave', 'Hypnosis'),
  Hypno = new Pokemon('Hypno', 10, 0, 257, 157, 152, 157, 222, 148, 77, 97, 'Psychic', 'Rest',
'Thunder Wave', 'Hypnosis'),
  Krabby = new Pokemon('Krabby', 4, 0, 196, 235, 209, 94, 94, 138, 88, 98, 'Swords Dance',
'Hyper Beam', 'Crabhammer', 'Body Slam'),
  Kingler = new Pokemon('Kingler', 4, 0, 222, 257, 233, 128, 128, 168, 81, 99, 'Swords Dance',
'Hyper Beam', 'Crabhammer', 'Body Slam'),
  Voltorb = new Pokemon('Voltorb', 8, 0, 214, 103, 138, 147, 147, 226, 88, 100, 'Thunder Wave',
'Thunderbolt', 'Explosion', 'Thunder'),
  Electrode = new Pokemon('Electrode', 8, 0, 230, 128, 160, 176, 176, 290, 81, 101, 'Thunder
Wave', 'Thunderbolt', 'Explosion', 'Thunder'),
  Exeggcute = new Pokemon('Exeggcute', 6, 10, 249, 121, 191, 156, 129, 121, 88, 102, 'Sleep
Powder', 'Psychic', 'Explosion', 'Mega Drain'),
  Exeggutor = new Pokemon('Exeggutor', 6, 10, 266, 186, 171, 231, 156, 126, 75, 103, 'Sleep
Powder', 'Psychic', 'Explosion', 'Mega Drain'),
  Cubone = new Pokemon('Cubone', 9, 0, 231, 138, 217, 121, 138, 112, 88, 104, 'Earthquake',
'Body Slam', 'Blizzard', 'Counter'),
  Marowak = new Pokemon('Marowak', 9, 0, 230, 176, 225, 128, 176, 120, 81, 105,
'Earthquake', 'Body Slam', 'Blizzard', 'Counter'),
  Hitmonlee = new Pokemon('Hitmonlee', 3, 0, 214, 241, 132, 103, 225, 188, 81, 106, 'Meditate',
'High Jump Kick', 'Body Slam', 'Mega Kick'),
  Hitmonchan = new Pokemon('Hitmonchan', 3, 0, 214, 217, 175, 103, 225, 170, 81, 107,
'Submission', 'Body Slam', 'Agility', 'Counter'),
  Lickitung = new Pokemon('Lickitung', 1, 0, 278, 136, 168, 144, 168, 95, 81, 108, 'Swords
Dance', 'Body Slam', 'Earthquake', 'Wrap'),
  Koffing = new Pokemon('Koffing', 7, 0, 214, 165, 217, 156, 129, 112, 88, 109, 'Explosion', 'Fire
Blast', 'Thunderbolt', 'Sludge'),
  Weezing = new Pokemon('Weezing', 7, 0, 238, 192, 241, 184, 160, 144, 81, 110, 'Explosion',
'Fire Blast', 'Thunderbolt', 'Sludge'),
  Rhyhorn = new Pokemon('Rhyhorn', 9, 11, 284, 200, 217, 103, 103, 94, 88, 111, 'Earthquake',
'Rock Slide', 'Body Slam', 'Double-Edge'),
  Rhydon = new Pokemon('Rhydon', 9, 11, 281, 239, 224, 111, 111, 104, 75, 112, 'Earthquake',
'Rock Slide', 'Body Slam', 'Double-Edge'),
  Chansey = new Pokemon('Chansey', 1, 0, 499, 51, 51, 96, 201, 119, 75, 113, 'Reflect',
'Seismic Toss', 'Soft-Boiled', 'Thunder Wave'),
  Tangela = new Pokemon('Tangela', 6, 0, 227, 129, 222, 199, 106, 137, 77, 114, 'Sleep
Powder', 'Stun Spore', 'Growth', 'Mega Drain'),
  Kangaskhan = new Pokemon('Kangaskhan', 1, 0, 288, 191, 168, 106, 168, 183, 77, 115, 'Body
Slam', 'Hyper Beam', 'Earthquake', 'Toxic'),
  Horsea = new Pokemon('Horsea', 4, 0, 196, 121, 173, 173, 94, 156, 88, 116, 'Agility', 'Surf',
'Blizzard', 'Hyper Beam'),
  Seadra = new Pokemon('Seadra', 4, 0, 222, 152, 201, 201, 120, 184, 81, 117, 'Agility', 'Surf',
'Blizzard', 'Hyper Beam'),
  Goldeen = new Pokemon('Goldeen', 4, 0, 222, 168, 156, 112, 138, 161, 88, 118, 'Agility',
'Double-Edge', 'Hyper Beam', 'Surf'),
  Seaking = new Pokemon('Seaking', 4, 0, 262, 196, 152, 152, 176, 157, 81, 119, 'Agility',
'Double-Edge', 'Hyper Beam', 'Surf'),
  Staryu = new Pokemon('Staryu', 4, 0, 196, 129, 147, 173, 147, 200, 88, 120, 'Recover',
'Thunder Wave', 'Thunderbolt', 'Psychic'),
  Starmie = new Pokemon('Starmie', 4, 10, 214, 156, 171, 194, 171, 216, 75, 121, 'Recover',
'Thunder Wave', 'Thunderbolt', 'Psychic'),
  MrMime = new Pokemon('Mr. Mime', 10, 0, 197, 120, 152, 209, 241, 192, 81, 122, 'Thunder
Wave', 'Psychic', 'Thunderbolt', 'Seismic Toss'),
  Scyther = new Pokemon('Scyther', 13, 5, 246, 225, 176, 136, 176, 217, 81, 123, 'Swords
Dance', 'Agility', 'Slash', 'Hyper Beam'),
  Jynx = new Pokemon('Jynx', 12, 10, 221, 119, 96, 216, 186, 186, 75, 124, 'Lovely Kiss',
'Blizzard', 'Psychic', 'Rest'),
  Electabuzz = new Pokemon('Electabuzz', 8, 0, 227, 172, 132, 191, 175, 206, 77, 125,
'Thunderbolt', 'Thunder Wave', 'Psychic', 'Seismic Toss'),
  Magmar = new Pokemon('Magmar', 2, 0, 238, 201, 139, 209, 184, 197, 81, 126, 'Fire Blast',
'Psychic', 'Confuse Ray', 'Body Slam'),
  Pinsir = new Pokemon('Pinsir', 13, 0, 238, 249, 209, 136, 160, 184, 81, 127, 'Swords Dance',
'Hyper Beam', 'Slash', 'Wrap'),
  Tauros = new Pokemon('Tauros', 1, 0, 236, 194, 186, 104, 149, 209, 75, 128, 'Body Slam',
'Hyper Beam', 'Blizzard', 'Earthquake'),
  Magikarp = new Pokemon('Magikarp', 4, 0, 178, 68, 147, 77, 85, 191, 88, 129, 'Body Slam',
'Hyper Beam', 'Blizzard', 'Thunderbolt'),
  Gyarados = new Pokemon('Gyarados', 4, 5, 273, 237, 166, 137, 199, 169, 77, 130, 'Body
Slam', 'Hyper Beam', 'Blizzard', 'Thunderbolt'),
  Lapras = new Pokemon('Lapras', 4, 12, 319, 171, 164, 171, 186, 134, 75, 131, 'Blizzard',
'Thunderbolt', 'Hyper Beam', 'Confuse Ray'),
  Ditto = new Pokemon('Ditto', 1, 0, 210, 124, 124, 124, 124, 124, 81, 132, 'Transform', '', '', ''),
  Eevee = new Pokemon('Eevee', 1, 0, 240, 147, 138, 129, 165, 147, 88, 133, 'Surf',
'Thunderbolt', 'Fire Blast', 'Hyper Beam'),
  Vaporeon = new Pokemon('Vaporeon', 4, 0, 327, 145, 137, 214, 191, 145, 77, 134, 'Surf',
'Blizzard', 'Body Slam', 'Rest'),
  Jolteon = new Pokemon('Jolteon', 8, 0, 221, 141, 134, 209, 186, 239, 75, 135, 'Thunder Wave', 'ThunderBolt', 'Double Kick', 'Rest'),
  Flareon = new Pokemon('Flareon', 2, 0, 238, 257, 144, 201, 225, 152, 81, 136, 'Fire Blast', 'Body Slam' ,'Hyper Beam', 'Fire Spin'),
  Porygon = new Pokemon('Porygon', 1, 0, 238, 144, 160, 184, 168, 111, 81, 137, 'Thunderbolt', 'Psychic' ,'Thunder Wave', 'Recover'),
  Omanyte = new Pokemon('Omanyte', 11, 4, 205, 121, 226, 209, 147, 112, 88, 138, 'Surf', 'Blizzard', 'Body Slam', 'Rest'),
  Omastar = new Pokemon('Omastar', 11, 4, 234, 137, 237, 222, 152, 129, 77, 139, 'Surf', 'Blizzard', 'Body Slam', 'Rest'),
  Kabuto = new Pokemon('Kabuto', 11, 4, 196, 191, 209, 147, 129, 147, 88, 140, 'Slash', 'Swords Dance', 'Hyper Beam', 'Surf'),
  Kabutops = new Pokemon('Kabutops', 11, 4, 230, 233, 217, 152, 160, 176, 81, 141, 'Slash', 'Swords Dance', 'Hyper Beam', 'Surf'),
  Aerodactyl = new Pokemon('Aerodactyl', 11, 5, 262, 217, 152, 144, 168, 257, 81, 142, 'Double-Edge', 'Hyper Beam', 'Fire Blast', 'Agility'),
  Snorlax = new Pokemon('Snorlax', 1, 0, 364, 209, 141, 141, 209, 89, 75, 143, 'Body Slam', 'Hyper Beam', 'Explosion', 'Earthquake'),
  Articuno = new Pokemon('Articuno', 12, 5, 265, 175, 199, 191, 237, 175, 77, 144, 'Blizzard', 'Ice Beam', 'Hyper Beam', 'Agility'),
  Zapdos = new Pokemon('Zapdos', 8, 5, 259, 179, 171, 231, 179, 194, 75, 145, 'Thunderbolt', 'Drill Peck', 'Thunder Wave', 'Agility'),
  Moltres = new Pokemon('Moltres', 2, 5, 278, 209, 192, 249, 184, 192, 81, 146, 'Fire Spin', 'Fire Blast', 'Toxic', 'Agility'),
  Dratini = new Pokemon('Dratini', 14, 0, 215, 163, 129, 138, 138, 138, 88, 147, 'Blizzard', 'Hyper Beam', 'Wrap', 'Agility'),
  Dragonair = new Pokemon('Dragonair', 14, 0, 240, 189, 157, 166, 166, 166, 84, 148, 'Blizzard', 'Hyper Beam', 'Wrap', 'Agility'),
  Dragonite = new Pokemon('Dragonite', 14, 5, 267, 251, 191, 199, 199, 168, 77, 149, 'Blizzard', 'Hyper Beam', 'Wrap', 'Agility'),
  Mewtwo = new Pokemon('Mewtwo', 10, 0, 275, 203, 174, 267, 174, 232, 73, 150, 'Amnesia', 'Psychic', 'Ice Beam', 'Recover'),
  Mew = new Pokemon('Mew', 10, 0, 266, 188, 188, 188, 188, 188, 73, 151, 'Swords Dance', 'Earthquake', 'Hyper Beam', 'Soft-Boiled')
  ];

const ea = Math.floor(Math.random() * allPokemon.length);
const eb = Math.floor(Math.random() * allPokemon.length);
const ec = Math.floor(Math.random() * allPokemon.length);
const ed = Math.floor(Math.random() * allPokemon.length);
const ee = Math.floor(Math.random() * allPokemon.length);
const ef = Math.floor(Math.random() * allPokemon.length);
const ma = Math.floor(Math.random() * allPokemon.length);
const mb = Math.floor(Math.random() * allPokemon.length);
const mc = Math.floor(Math.random() * allPokemon.length);
const md = Math.floor(Math.random() * allPokemon.length);
const me = Math.floor(Math.random() * allPokemon.length);
const mf = Math.floor(Math.random() * allPokemon.length);

var enemyTeam = [6];
enemyTeam[0] = allPokemon[ea];
enemyTeam[1] = allPokemon[eb];
enemyTeam[2] = allPokemon[ec];
enemyTeam[3] = allPokemon[ed];
enemyTeam[4] = allPokemon[ee];
enemyTeam[5] = allPokemon[ef];
var myTeam = [6];
myTeam[0] = allPokemon[ma];
myTeam[1] = allPokemon[mb];
myTeam[2] = allPokemon[mc];
myTeam[3] = allPokemon[md];
myTeam[4] = allPokemon[me];
myTeam[5] = allPokemon[mf];


  function supereffectiveType1(t)
{
  if ((t == 2 && enemyTeam[currentEnemy].getType1() === 6) || (t == 2 &&
enemyTeam[currentEnemy].getType1() === 12) || (t == 2 &&
enemyTeam[currentEnemy].getType1() === 13) || (t==2 &&
enemyTeam[currentEnemy].getType1() === 17)){
    return true;
  }
  else if((t == 3 && enemyTeam[currentEnemy].getType1() === 1) || (t == 3 &&
enemyTeam[currentEnemy].getType1() === 11) || (t == 3 &&
enemyTeam[currentEnemy].getType1() === 12) || (t == 3 &&
enemyTeam[currentEnemy].getType1() === 16) || (t == 3 &&
enemyTeam[currentEnemy].getType1() === 17)){
    return true;
  }
  else if((t == 4 && enemyTeam[currentEnemy].getType1() === 2) || (t == 4 &&
enemyTeam[currentEnemy].getType1() === 9) || (t == 4 &&
enemyTeam[currentEnemy].getType1() === 11)){
    return true;
  }
  else if((t == 5 && enemyTeam[currentEnemy].getType1() === 6) || (t == 5 &&
enemyTeam[currentEnemy].getType1() === 3) || (t == 5 &&
enemyTeam[currentEnemy].getType1() === 13)){
    return true;
  }
  else if((t == 6 && enemyTeam[currentEnemy].getType1() === 4) || (t == 6 &&
enemyTeam[currentEnemy].getType1() === 9) || (t == 4 &&
enemyTeam[currentEnemy].getType1() === 11)){
    return true;
  }
  else if((t == 7 && enemyTeam[currentEnemy].getType1() === 6) || (t == 7 &&
enemyTeam[currentEnemy].getType1() === 18)){
    return true;
  }
  else if((t == 8 && enemyTeam[currentEnemy].getType1() === 4) || (t == 8 &&
enemyTeam[currentEnemy].getType1() === 5)){
    return true;
  }
  else if((t == 9 && enemyTeam[currentEnemy].getType1() === 2) || (t == 9 &&
enemyTeam[currentEnemy].getType1() === 8) || (t == 9 &&
enemyTeam[currentEnemy].getType1() === 7) || (t == 9 &&
enemyTeam[currentEnemy].getType1() === 11) || (t == 9 &&
enemyTeam[currentEnemy].getType1() === 17)){
    return true;
  }
  else if((t == 10 && enemyTeam[currentEnemy].getType1() === 3) || (t == 10 &&
enemyTeam[currentEnemy].getType1() === 7)){
    return true;
  }
  else if((t == 11 && enemyTeam[currentEnemy].getType1() === 2) || (t == 11 &&
enemyTeam[currentEnemy].getType1() === 12) || (t == 11 &&
enemyTeam[currentEnemy].getType1() === 5) || (t == 11 &&
enemyTeam[currentEnemy].getType1() === 13)){
    return true;
  }
  else if((t == 12 && enemyTeam[currentEnemy].getType1() === 6) || (t == 12 &&
enemyTeam[currentEnemy].getType1() === 9) || (t == 12 &&
enemyTeam[currentEnemy].getType1() === 5) || (t == 12 &&
enemyTeam[currentEnemy].getType1() === 14)){
    return true;
  }
  else if((t == 13 && enemyTeam[currentEnemy].getType1() === 6) || (t == 13 &&
enemyTeam[currentEnemy].getType1() === 10) || (t == 13 &&
enemyTeam[currentEnemy].getType1() === 16)){
    return true;
  }
  else if((t == 14 && enemyTeam[currentEnemy].getType1() === 14)){
    return true;
  }
  else if((t == 15 && enemyTeam[currentEnemy].getType1() === 10) || (t == 15 &&
enemyTeam[currentEnemy].getType1() === 15)){
    return true;
  }
  else if((t == 16 && enemyTeam[currentEnemy].getType1() === 10) || (t == 16 &&
enemyTeam[currentEnemy].getType1() === 15)){
    return true;
  }
  else if((t == 17 && enemyTeam[currentEnemy].getType1() === 12) || (t == 17 &&
enemyTeam[currentEnemy].getType1() === 11) || (t == 17 &&
enemyTeam[currentEnemy].getType1() === 18)){
    return true;
  }
  else if((t == 18 && enemyTeam[currentEnemy].getType1() === 3) || (t == 18 &&
enemyTeam[currentEnemy].getType1() === 14) || (t == 18 &&
enemyTeam[currentEnemy].getType1() === 16)){
    return true;
  }
  else
    return false;
}

function supereffectiveType2(t)
{
  if ((t == 2 && enemyTeam[currentEnemy].getType2() === 6) || (t == 2 &&
enemyTeam[currentEnemy].getType2() === 12) || (t == 2 &&
enemyTeam[currentEnemy].getType2() === 13) || (t==2 &&
enemyTeam[currentEnemy].getType2() === 17)){
    return true;
  }
  else if((t == 3 && enemyTeam[currentEnemy].getType2() === 1) || (t == 3 &&
enemyTeam[currentEnemy].getType2() === 11) || (t == 3 &&
enemyTeam[currentEnemy].getType2() === 12) || (t == 3 &&
enemyTeam[currentEnemy].getType2() === 16) || (t == 3 &&
enemyTeam[currentEnemy].getType2() === 17)){
    return true;
  }
  else if((t == 4 && enemyTeam[currentEnemy].getType2() === 2) || (t == 4 &&
enemyTeam[currentEnemy].getType2() === 9) || (t == 4 &&
enemyTeam[currentEnemy].getType2() === 11)){
    return true;
  }
```

```
 else if((t == 5 && enemyTeam[currentEnemy].getType2() === 6) || (t == 5 &&
enemyTeam[currentEnemy].getType2() === 3) || (t == 5 &&
enemyTeam[currentEnemy].getType2() === 13)){
  return true;
 }
 else if((t == 6 && enemyTeam[currentEnemy].getType2() === 4) || (t == 6 &&
enemyTeam[currentEnemy].getType2() === 9) || (t == 4 &&
enemyTeam[currentEnemy].getType2() === 11)){
  return true;
 }
 else if((t == 7 && enemyTeam[currentEnemy].getType2() === 6) || (t == 7 &&
enemyTeam[currentEnemy].getType2() === 18)){
  return true;
 }
 else if((t == 8 && enemyTeam[currentEnemy].getType2() === 4) || (t == 8 &&
enemyTeam[currentEnemy].getType2() === 5)){
  return true;
 }
 else if((t == 9 && enemyTeam[currentEnemy].getType2() === 2) || (t == 9 &&
enemyTeam[currentEnemy].getType2() === 8) || (t == 9 &&
enemyTeam[currentEnemy].getType2() === 7) || (t == 9 &&
enemyTeam[currentEnemy].getType2() === 11) || (t == 9 &&
enemyTeam[currentEnemy].getType2() === 17)){
  return true;
 }
 else if((t == 10 && enemyTeam[currentEnemy].getType2() === 3) || (t == 10 &&
enemyTeam[currentEnemy].getType2() === 7)){
  return true;
 }
 else if((t == 11 && enemyTeam[currentEnemy].getType2() === 2) || (t == 11 &&
enemyTeam[currentEnemy].getType2() === 12) || (t == 11 &&
enemyTeam[currentEnemy].getType2() === 5) || (t == 11 &&
enemyTeam[currentEnemy].getType2() === 13)){
  return true;
 }
 else if((t == 12 && enemyTeam[currentEnemy].getType2() === 6) || (t == 12 &&
enemyTeam[currentEnemy].getType2() === 9) || (t == 12 &&
enemyTeam[currentEnemy].getType2() === 5) || (t == 12 &&
enemyTeam[currentEnemy].getType2() === 14)){
  return true;
 }
 else if((t == 13 && enemyTeam[currentEnemy].getType2() === 6) || (t == 13 &&
enemyTeam[currentEnemy].getType2() === 10) || (t == 13 &&
enemyTeam[currentEnemy].getType2() === 16)){
    return true;
  }
  else if((t == 14 && enemyTeam[currentEnemy].getType2() === 14)){
    return true;
  }
  else if((t == 15 && enemyTeam[currentEnemy].getType2() === 10) || (t == 15 &&
enemyTeam[currentEnemy].getType2() === 15)){
    return true;
  }
  else if((t == 16 && enemyTeam[currentEnemy].getType2() === 10) || (t == 16 &&
enemyTeam[currentEnemy].getType2() === 15)){
    return true;
  }
  else if((t == 17 && enemyTeam[currentEnemy].getType2() === 12) || (t == 17 &&
enemyTeam[currentEnemy].getType2() === 11) || (t == 17 &&
enemyTeam[currentEnemy].getType2() === 18)){
    return true;
  }
  else if((t == 18 && enemyTeam[currentEnemy].getType2() === 3) || (t == 18 &&
enemyTeam[currentEnemy].getType2() === 14) || (t == 18 &&
enemyTeam[currentEnemy].getType2() === 16)){
    return true;
  }
  else
    return false;
}

function enemySupereffectiveType1(t)
{
  if ((t == 2 && myTeam[currentMine].getType1() === 6) || (t == 2 &&
myTeam[currentMine].getType1() === 12) || (t == 2 && myTeam[currentMine].getType1() ===
13) || (t==2 && myTeam[currentMine].getType1() === 17)){
    return true;
  }
  else if((t == 3 && myTeam[currentMine].getType1() === 1) || (t == 3 &&
myTeam[currentMine].getType1() === 11) || (t == 3 && myTeam[currentMine].getType1() ===
12) || (t == 3 && myTeam[currentMine].getType1() === 16) || (t == 3 &&
myTeam[currentMine].getType1() === 17)){
    return true;
  }
  else if((t == 4 && myTeam[currentMine].getType1() === 2) || (t == 4 &&
myTeam[currentMine].getType1() === 9) || (t == 4 && myTeam[currentMine].getType1() ===
11)){
    return true;
  }
  else if((t == 5 && myTeam[currentMine].getType1() === 6) || (t == 5 &&
myTeam[currentMine].getType1() === 3) || (t == 5 && myTeam[currentMine].getType1() ===
13)){
    return true;
  }
  else if((t == 6 && myTeam[currentMine].getType1() === 4) || (t == 6 &&
myTeam[currentMine].getType1() === 9) || (t == 4 && myTeam[currentMine].getType1() ===
11)){
    return true;
  }
  else if((t == 7 && myTeam[currentMine].getType1() === 6) || (t == 7 &&
myTeam[currentMine].getType1() === 18)){
    return true;
  }
  else if((t == 8 && myTeam[currentMine].getType1() === 4) || (t == 8 &&
myTeam[currentMine].getType1() === 5)){
    return true;
  }
  else if((t == 9 && myTeam[currentMine].getType1() === 2) || (t == 9 &&
myTeam[currentMine].getType1() === 8) || (t == 9 && myTeam[currentMine].getType1() === 7) ||
(t == 9 && myTeam[currentMine].getType1() === 11) || (t == 9 &&
myTeam[currentMine].getType1() === 17)){
    return true;
  }
  else if((t == 10 && myTeam[currentMine].getType1() === 3) || (t == 10 &&
myTeam[currentMine].getType1() === 7)){
    return true;
  }
  else if((t == 11 && myTeam[currentMine].getType1() === 2) || (t == 11 &&
myTeam[currentMine].getType1() === 12) || (t == 11 && myTeam[currentMine].getType1() ===
5) || (t == 11 && myTeam[currentMine].getType1() === 13)){
    return true;
  }
  else if((t == 12 && myTeam[currentMine].getType1() === 6) || (t == 12 &&
myTeam[currentMine].getType1() === 9) || (t == 12 && myTeam[currentMine].getType1() === 5)
|| (t == 12 && myTeam[currentMine].getType1() === 14)){
    return true;
  }
  else if((t == 13 && myTeam[currentMine].getType1() === 6) || (t == 13 &&
myTeam[currentMine].getType1() === 10) || (t == 13 && myTeam[currentMine].getType1() ===
16)){
    return true;
  }
  else if((t == 14 && myTeam[currentMine].getType1() === 14)){
    return true;
  }
  else if((t == 15 && myTeam[currentMine].getType1() === 10) || (t == 15 &&
myTeam[currentMine].getType1() === 15)){
    return true;
  }
  else if((t == 16 && myTeam[currentMine].getType1() === 10) || (t == 16 &&
myTeam[currentMine].getType1() === 15)){
    return true;
  }
  else if((t == 17 && myTeam[currentMine].getType1() === 12) || (t == 17 &&
myTeam[currentMine].getType1() === 11) || (t == 17 && myTeam[currentMine].getType1() ===
18)){
    return true;
  }
  else if((t == 18 && myTeam[currentMine].getType1() === 3) || (t == 18 &&
myTeam[currentMine].getType1() === 14) || (t == 18 && myTeam[currentMine].getType1() ===
16)){
    return true;
  }
  else
    return false;
}

function enemySupereffectiveType2(t)
{
  if ((t == 2 && myTeam[currentMine].getType2() === 6) || (t == 2 &&
myTeam[currentMine].getType2() === 12) || (t == 2 && myTeam[currentMine].getType2() ===
13) || (t==2 && myTeam[currentMine].getType2() === 17)){
    return true;
  }
  else if((t == 3 && myTeam[currentMine].getType2() === 1) || (t == 3 &&
myTeam[currentMine].getType2() === 11) || (t == 3 && myTeam[currentMine].getType2() ===
12) || (t == 3 && myTeam[currentMine].getType2() === 16) || (t == 3 &&
myTeam[currentMine].getType2() === 17)){
    return true;
  }
  else if((t == 4 && myTeam[currentMine].getType2() === 2) || (t == 4 &&
myTeam[currentMine].getType2() === 9) || (t == 4 && myTeam[currentMine].getType2() ===
11)){
    return true;
  }
  else if((t == 5 && myTeam[currentMine].getType2() === 6) || (t == 5 &&
myTeam[currentMine].getType2() === 3) || (t == 5 && myTeam[currentMine].getType2() ===
13)){
    return true;
  }
  else if((t == 6 && myTeam[currentMine].getType2() === 4) || (t == 6 &&
myTeam[currentMine].getType2() === 9) || (t == 4 && myTeam[currentMine].getType2() ===
11)){
    return true;
  }
  else if((t == 7 && myTeam[currentMine].getType2() === 6) || (t == 7 &&
myTeam[currentMine].getType2() === 18)){
    return true;
  }
  else if((t == 8 && myTeam[currentMine].getType2() === 4) || (t == 8 &&
myTeam[currentMine].getType2() === 5)){
    return true;
  }
  else if((t == 9 && myTeam[currentMine].getType2() === 2) || (t == 9 &&
myTeam[currentMine].getType2() === 8) || (t == 9 && myTeam[currentMine].getType2() === 7) ||
(t == 9 && myTeam[currentMine].getType2() === 11) || (t == 9 &&
myTeam[currentMine].getType2() === 17)){
    return true;
  }
  else if((t == 10 && myTeam[currentMine].getType2() === 3) || (t == 10 &&
myTeam[currentMine].getType2() === 7)){
    return true;
  }
  else if((t == 11 && myTeam[currentMine].getType2() === 2) || (t == 11 &&
myTeam[currentMine].getType2() === 12) || (t == 11 && myTeam[currentMine].getType2() ===
5) || (t == 11 && myTeam[currentMine].getType2() === 13)){
    return true;
  }
  else if((t == 12 && myTeam[currentMine].getType2() === 6) || (t == 12 &&
myTeam[currentMine].getType2() === 9) || (t == 12 && myTeam[currentMine].getType2() === 5)
|| (t == 12 && myTeam[currentMine].getType2() === 14)){
    return true;
  }
  else if((t == 13 && myTeam[currentMine].getType2() === 6) || (t == 13 &&
myTeam[currentMine].getType2() === 10) || (t == 13 && myTeam[currentMine].getType2() ===
16)){
    return true;
  }
  else if((t == 14 && myTeam[currentMine].getType2() === 14)){
```

```
  return true;
 }
 else if((t == 15 && myTeam[currentMine].getType2() === 10) || (t == 15 &&
myTeam[currentMine].getType2() === 15)){
  return true;
 }
 else if((t == 16 && myTeam[currentMine].getType2() === 10) || (t == 16 &&
myTeam[currentMine].getType2() === 15)){
  return true;
 }
 else if((t == 17 && myTeam[currentMine].getType2() === 12) || (t == 17 &&
myTeam[currentMine].getType2() === 11) || (t == 17 && myTeam[currentMine].getType2() ===
18)){
  return true;
 }
 else if((t == 18 && myTeam[currentMine].getType2() === 3) || (t == 18 &&
myTeam[currentMine].getType2() === 14) || (t == 18 && myTeam[currentMine].getType2() ===
16)){
  return true;
 }
 else
  return false;
}
function resistsType1(t){
 if (t == 1){
  if (enemyTeam[currentEnemy].getType1() == 11 || enemyTeam[currentEnemy].getType1() ==
17)
   return true;
 }
 if (t == 2){
  if (enemyTeam[currentEnemy].getType1() == 2 || enemyTeam[currentEnemy].getType1() == 4
|| enemyTeam[currentEnemy].getType1() == 11 || enemyTeam[currentEnemy].getType1() == 14)
    return true;
 }
 if (t == 3){
  if (enemyTeam[currentEnemy].getType1() == 7 || enemyTeam[currentEnemy].getType1() == 5
|| enemyTeam[currentEnemy].getType1() == 10 || enemyTeam[currentEnemy].getType1() == 13
|| enemyTeam[currentEnemy].getType1() == 18)
   return true;
 }
 if (t == 4){
  if (enemyTeam[currentEnemy].getType1() == 4 || enemyTeam[currentEnemy].getType1() == 6
|| enemyTeam[currentEnemy].getType1() == 14)
   return true;
  }
  if (t == 5){
    if (enemyTeam[currentEnemy].getType1() == 8 || enemyTeam[currentEnemy].getType1() ==
11 || enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 6){
    if (enemyTeam[currentEnemy].getType1() == 2 || enemyTeam[currentEnemy].getType1() == 6
|| enemyTeam[currentEnemy].getType1() == 7 || enemyTeam[currentEnemy].getType1() == 5 ||
enemyTeam[currentEnemy].getType1() == 13 || enemyTeam[currentEnemy].getType1() == 14 ||
enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 7){
    if (enemyTeam[currentEnemy].getType1() == 7 || enemyTeam[currentEnemy].getType1() == 9
|| enemyTeam[currentEnemy].getType1() == 11 || enemyTeam[currentEnemy].getType1() == 15)
      return true;
  }
  if (t == 8){
    if (enemyTeam[currentEnemy].getType1() == 8 || enemyTeam[currentEnemy].getType1() == 6
|| enemyTeam[currentEnemy].getType1() == 14)
      return true;
  }
  if (t == 9){
    if (enemyTeam[currentEnemy].getType1() == 6 || enemyTeam[currentEnemy].getType1() ==
13)
      return true;
  }
  if (t == 10){
    if (enemyTeam[currentEnemy].getType1() == 10 || enemyTeam[currentEnemy].getType1() ==
17)
      return true;
  }
  if (t == 11){
    if (enemyTeam[currentEnemy].getType1() == 3 || enemyTeam[currentEnemy].getType1() == 9
|| enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 12){
    if (enemyTeam[currentEnemy].getType1() == 2 || enemyTeam[currentEnemy].getType1() == 4
|| enemyTeam[currentEnemy].getType1() == 12 || enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 13){
    if (enemyTeam[currentEnemy].getType1() == 2 || enemyTeam[currentEnemy].getType1() == 3
|| enemyTeam[currentEnemy].getType1() == 7 || enemyTeam[currentEnemy].getType1() == 5 ||
enemyTeam[currentEnemy].getType1() == 15 || enemyTeam[currentEnemy].getType1() == 17 ||
enemyTeam[currentEnemy].getType1() == 18)
      return true;
  }
  if (t == 14){
    if (enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 15){
    if (enemyTeam[currentEnemy].getType1() == 16)
      return true;
  }
  if (t == 16){
    if (enemyTeam[currentEnemy].getType1() == 3 || enemyTeam[currentEnemy].getType1() ==
16 || enemyTeam[currentEnemy].getType1() == 18)
      return true;
  }
  if (t == 17){
    if (enemyTeam[currentEnemy].getType1() == 2 || enemyTeam[currentEnemy].getType1() == 4
|| enemyTeam[currentEnemy].getType1() == 8 || enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 18){
    if (enemyTeam[currentEnemy].getType1() == 2 || enemyTeam[currentEnemy].getType1() == 7
|| enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  else
    return false;
}
function resistsType2(t){
  if (t == 1){
    if (enemyTeam[currentEnemy].getType2() == 11 || enemyTeam[currentEnemy].getType2() ==
17)
      return true;
  }
  if (t == 2){
    if (enemyTeam[currentEnemy].getType2() == 2 || enemyTeam[currentEnemy].getType2() == 4
|| enemyTeam[currentEnemy].getType2() == 11 || enemyTeam[currentEnemy].getType2() == 14)
        return true;
  }
  if (t == 3){
    if (enemyTeam[currentEnemy].getType2() == 7 || enemyTeam[currentEnemy].getType2() == 5
|| enemyTeam[currentEnemy].getType2() == 10 || enemyTeam[currentEnemy].getType2() == 13
|| enemyTeam[currentEnemy].getType2() == 18)
      return true;
  }
  if (t == 4){
    if (enemyTeam[currentEnemy].getType2() == 4 || enemyTeam[currentEnemy].getType2() == 6
|| enemyTeam[currentEnemy].getType2() == 14)
      return true;
  }
  if (t == 5){
    if (enemyTeam[currentEnemy].getType2() == 8 || enemyTeam[currentEnemy].getType2() ==
11 || enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 6){
    if (enemyTeam[currentEnemy].getType2() == 2 || enemyTeam[currentEnemy].getType2() == 6
|| enemyTeam[currentEnemy].getType2() == 7 || enemyTeam[currentEnemy].getType2() == 5 ||
enemyTeam[currentEnemy].getType2() == 13 || enemyTeam[currentEnemy].getType2() == 14 ||
enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 7){
    if (enemyTeam[currentEnemy].getType2() == 7 || enemyTeam[currentEnemy].getType2() == 9
|| enemyTeam[currentEnemy].getType2() == 11 || enemyTeam[currentEnemy].getType2() == 15)
      return true;
  }
  if (t == 8){
    if (enemyTeam[currentEnemy].getType2() == 8 || enemyTeam[currentEnemy].getType2() == 6
|| enemyTeam[currentEnemy].getType2() == 14)
      return true;
  }
  if (t == 9){
    if (enemyTeam[currentEnemy].getType2() == 6 || enemyTeam[currentEnemy].getType2() ==
13)
      return true;
  }
  if (t == 10){
    if (enemyTeam[currentEnemy].getType2() == 10 || enemyTeam[currentEnemy].getType2() ==
17)
      return true;
  }
  if (t == 11){
    if (enemyTeam[currentEnemy].getType2() == 3 || enemyTeam[currentEnemy].getType2() == 9
|| enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 12){
    if (enemyTeam[currentEnemy].getType2() == 2 || enemyTeam[currentEnemy].getType2() == 4
|| enemyTeam[currentEnemy].getType2() == 12 || enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 13){
    if (enemyTeam[currentEnemy].getType2() == 2 || enemyTeam[currentEnemy].getType2() == 3
|| enemyTeam[currentEnemy].getType2() == 7 || enemyTeam[currentEnemy].getType2() == 5 ||
enemyTeam[currentEnemy].getType2() == 15 || enemyTeam[currentEnemy].getType2() == 17 ||
enemyTeam[currentEnemy].getType2() == 18)
      return true;
  }
  if (t == 14){
    if (enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 15){
    if (enemyTeam[currentEnemy].getType2() == 16)
      return true;
  }
  if (t == 16){
    if (enemyTeam[currentEnemy].getType2() == 3 || enemyTeam[currentEnemy].getType2() ==
16 || enemyTeam[currentEnemy].getType2() == 18)
      return true;
  }
  if (t == 17){
    if (enemyTeam[currentEnemy].getType2() == 2 || enemyTeam[currentEnemy].getType2() == 4
|| enemyTeam[currentEnemy].getType2() == 8 || enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 18){
    if (enemyTeam[currentEnemy].getType2() == 2 || enemyTeam[currentEnemy].getType2() == 7
|| enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  else
    return false;
}
   function enemyResistsType1(t){
  if (t == 1){
```

```
  if (myTeam[currentMine].getType1() == 11 || myTeam[currentMine].getType1() == 17)
   return true;
 }
 if (t == 2){
  if (myTeam[currentMine].getType1() == 2 || myTeam[currentMine].getType1() == 4 ||
myTeam[currentMine].getType1() == 11 || myTeam[currentMine].getType1() == 14)
    return true;
 }
 if (t == 3){
  if (myTeam[currentMine].getType1() == 7 || myTeam[currentMine].getType1() == 5 ||
myTeam[currentMine].getType1() == 10 || myTeam[currentMine].getType1() == 13 ||
myTeam[currentMine].getType1() == 18)
   return true;
 }
 if (t == 4){
  if (myTeam[currentMine].getType1() == 4 || myTeam[currentMine].getType1() == 6 ||
myTeam[currentMine].getType1() == 14)
   return true;
 }
 if (t == 5){
  if (myTeam[currentMine].getType1() == 8 || myTeam[currentMine].getType1() == 11 ||
myTeam[currentMine].getType1() == 17)
   return true;
 }
 if (t == 6){
  if (myTeam[currentMine].getType1() == 2 || myTeam[currentMine].getType1() == 6 ||
myTeam[currentMine].getType1() == 7 || myTeam[currentMine].getType1() == 5 ||
myTeam[currentMine].getType1() == 13 || myTeam[currentMine].getType1() == 14 ||
myTeam[currentMine].getType1() == 17)
   return true;
 }
 if (t == 7){
  if (myTeam[currentMine].getType1() == 7 || myTeam[currentMine].getType1() == 9 ||
myTeam[currentMine].getType1() == 11 || myTeam[currentMine].getType1() == 15)
   return true;
 }
 if (t == 8){
  if (myTeam[currentMine].getType1() == 8 || myTeam[currentMine].getType1() == 6 ||
myTeam[currentMine].getType1() == 14)
   return true;
 }
 if (t == 9){
  if (myTeam[currentMine].getType1() == 6 || myTeam[currentMine].getType1() == 13)
   return true;
  }
  if (t == 10){
    if (myTeam[currentMine].getType1() == 10 || myTeam[currentMine].getType1() == 17)
      return true;
  }
  if (t == 11){
    if (myTeam[currentMine].getType1() == 3 || myTeam[currentMine].getType1() == 9 ||
myTeam[currentMine].getType1() == 17)
      return true;
  }
  if (t == 12){
    if (myTeam[currentMine].getType1() == 2 || myTeam[currentMine].getType1() == 4 ||
myTeam[currentMine].getType1() == 12 || myTeam[currentMine].getType1() == 17)
      return true;
  }
  if (t == 13){
    if (myTeam[currentMine].getType1() == 2 || myTeam[currentMine].getType1() == 3 ||
myTeam[currentMine].getType1() == 7 || myTeam[currentMine].getType1() == 5 ||
myTeam[currentMine].getType1() == 15 || myTeam[currentMine].getType1() == 17 ||
myTeam[currentMine].getType1() == 18)
      return true;
  }
  if (t == 14){
    if (myTeam[currentMine].getType1() == 17)
      return true;
  }
  if (t == 15){
    if (myTeam[currentMine].getType1() == 16)
      return true;
  }
  if (t == 16){
    if (myTeam[currentMine].getType1() == 3 || myTeam[currentMine].getType1() == 16 ||
myTeam[currentMine].getType1() == 18)
      return true;
  }
  if (t == 17){
    if (myTeam[currentMine].getType1() == 2 || myTeam[currentMine].getType1() == 4 ||
myTeam[currentMine].getType1() == 8 || myTeam[currentMine].getType1() == 17)
      return true;
  }
  if (t == 18){
    if (myTeam[currentMine].getType1() == 2 || myTeam[currentMine].getType1() == 7 ||
myTeam[currentMine].getType1() == 17)
      return true;
  }
     else
       return false;
}
   function enemyResistsType2(t){
  if (t == 1){
    if (myTeam[currentMine].getType2() == 11 || myTeam[currentMine].getType2() == 17)
      return true;
  }
  if (t == 2){
    if (myTeam[currentMine].getType2() == 2 || myTeam[currentMine].getType2() == 4 ||
myTeam[currentMine].getType2() == 11 || myTeam[currentMine].getType2() == 14)
        return true;
  }
  if (t == 3){
    if (myTeam[currentMine].getType2() == 7 || myTeam[currentMine].getType2() == 5 ||
myTeam[currentMine].getType2() == 10 || myTeam[currentMine].getType2() == 13 ||
myTeam[currentMine].getType2() == 18)
      return true;
  }
  if (t == 4){
    if (myTeam[currentMine].getType2() == 4 || myTeam[currentMine].getType2() == 6 ||
myTeam[currentMine].getType2() == 14)
      return true;
  }
  if (t == 5){
    if (myTeam[currentMine].getType2() == 8 || myTeam[currentMine].getType2() == 11 ||
myTeam[currentMine].getType2() == 17)
      return true;
  }
  if (t == 6){
    if (myTeam[currentMine].getType2() == 2 || myTeam[currentMine].getType2() == 6 ||
myTeam[currentMine].getType2() == 7 || myTeam[currentMine].getType2() == 5 ||
myTeam[currentMine].getType2() == 13 || myTeam[currentMine].getType2() == 14 ||
myTeam[currentMine].getType2() == 17)
      return true;
  }
  if (t == 7){
    if (myTeam[currentMine].getType2() == 7 || myTeam[currentMine].getType2() == 9 ||
myTeam[currentMine].getType2() == 11 || myTeam[currentMine].getType2() == 15)
      return true;
  }
  if (t == 8){
```

```
  if (myTeam[currentMine].getType2() == 8 || myTeam[currentMine].getType2() == 6 ||
myTeam[currentMine].getType2() == 14)
   return true;
 }
 if (t == 9){
  if (myTeam[currentMine].getType2() == 6 || myTeam[currentMine].getType2() == 13)
   return true;
 }
 if (t == 10){
  if (myTeam[currentMine].getType2() == 10 || myTeam[currentMine].getType2() == 17)
   return true;
 }
 if (t == 11){
  if (myTeam[currentMine].getType2() == 3 || myTeam[currentMine].getType2() == 9 ||
myTeam[currentMine].getType2() == 17)
   return true;
 }
 if (t == 12){
  if (myTeam[currentMine].getType2() == 2 || myTeam[currentMine].getType2() == 4 ||
myTeam[currentMine].getType2() == 12 || myTeam[currentMine].getType2() == 17)
   return true;
 }
 if (t == 13){
  if (myTeam[currentMine].getType2() == 2 || myTeam[currentMine].getType2() == 3 ||
myTeam[currentMine].getType2() == 7 || myTeam[currentMine].getType2() == 5 ||
myTeam[currentMine].getType2() == 15 || myTeam[currentMine].getType2() == 17 ||
myTeam[currentMine].getType2() == 18)
   return true;
 }
 if (t == 14){
  if (myTeam[currentMine].getType2() == 17)
   return true;
 }
 if (t == 15){
  if (myTeam[currentMine].getType2() == 16)
   return true;
 }
 if (t == 16){
  if (myTeam[currentMine].getType2() == 3 || myTeam[currentMine].getType2() == 16 ||
myTeam[currentMine].getType2() == 18)
   return true;
 }
 if (t == 17){
    if (myTeam[currentMine].getType2() == 2 || myTeam[currentMine].getType2() == 4 ||
myTeam[currentMine].getType2() == 8 || myTeam[currentMine].getType2() == 17)
      return true;
  }
  if (t == 18){
    if (myTeam[currentMine].getType2() == 2 || myTeam[currentMine].getType2() == 7 ||
myTeam[currentMine].getType2() == 17)
      return true;
  }
     else
       return false;
}
function immuneType1(t){
  if (t == 1 || t == 3){
    if (enemyTeam[currentEnemy].getType1() == 15)
      return true;
  }
  if (t == 8){
    if (enemyTeam[currentEnemy].getType1() == 9)
      return true;
  }
  if (t == 7){
    if (enemyTeam[currentEnemy].getType1() == 17)
      return true;
  }
  if (t == 9){
    if (enemyTeam[currentEnemy].getType1() == 5)
      return true;
  }
  if (t == 10){
    if (enemyTeam[currentEnemy].getType1() == 16)
      return true;
  }
  if (t == 15){
    if (enemyTeam[currentEnemy].getType1() == 1)
      return true;
  }
  if (t == 14){
    if (enemyTeam[currentEnemy].getType1() == 18)
      return true;
  }
  else
    return false;
}
```

```
function immuneType2(t){
  if (t == 1 || t == 3){
    if (enemyTeam[currentEnemy].getType2() == 15)
      return true;
  }
  if (t == 8){
    if (enemyTeam[currentEnemy].getType2() == 9)
      return true;
  }
  if (t == 7){
    if (enemyTeam[currentEnemy].getType2() == 17)
      return true;
  }
  if (t == 9){
    if (enemyTeam[currentEnemy].getType2() == 5)
      return true;
  }
  if (t == 10){
    if (enemyTeam[currentEnemy].getType2() == 16)
      return true;
  }
  if (t == 15){
    if (enemyTeam[currentEnemy].getType2() == 1)
      return true;
  }
  if (t == 14){
    if (enemyTeam[currentEnemy].getType2() == 18)
      return true;
  }
  else
    return false;
}
function enemyImmuneType1(t){
  if (t == 1 || t == 3){
    if (myTeam[currentMine].getType1() == 15)
      return true;
  }
  if (t == 8){
    if (myTeam[currentMine].getType1() == 9)
      return true;
  }
  if (t == 7){
    if (myTeam[currentMine].getType1() == 17)
      return true;
  }
  if (t == 9){
    if (myTeam[currentMine].getType1() == 5)
      return true;
  }
  if (t == 10){
    if (myTeam[currentMine].getType1() == 16)
      return true;
  }
  if (t == 15){
    if (myTeam[currentMine].getType1() == 1)
      return true;
  }
  if (t == 14){
    if (myTeam[currentMine].getType1() == 18)
      return true;
  }
  else
    return false;
}
function enemyImmuneType2(t){
  if (t == 1 || t == 3){
    if (myTeam[currentMine].getType2() == 15)
      return true;
  }
  if (t == 8){
    if (myTeam[currentMine].getType2() == 9)
      return true;
  }
  if (t == 7){
    if (myTeam[currentMine].getType2() == 17)
      return true;
  }
  if (t == 9){
    if (myTeam[currentMine].getType2() == 5)
      return true;
  }
  if (t == 10){
    if (myTeam[currentMine].getType2() == 16)
      return true;
  }
  if (t == 15){
    if (myTeam[currentMine].getType2() == 1)
      return true;
  }
  if (t == 14){
    if (myTeam[currentMine].getType2() == 18)
      return true;
  }
  else
    return false;
}
function mySleepPowder(){
  if ((Math.random() > 0.25) && enemyAsleep == 0){
    enemyStatusReset();
    enemyAsleep += (int)(Math.random()*4)+2;
  }
}
function myRazorLeaf(){
  myAttack(55, 6, true, true, 0.95);
}
function myBodySlam(){
  if (Math.random() > 0.7 && (enemyTeam[currentEnemy].getType1() != 8 &&
enemyTeam[currentEnemy].getType2() != 8)){
    enemyStatusReset();
    enemyParalyzed = true;
  }
  myAttack(85, 1, false, false, 1);
}
function myToxic(){
  if (Math.random() > 0.15){
    enemyStatusReset();
    enemyToxiced = true;
  }
}
function myFireSpin(){
  if (Math.random() > 0.3 && enemyStuck == 0){
    FS = (int)(Math.random()*4)+2;
    enemyStatusReset();
    enemyStuck += FS;
    enemyStuckName = 'Fire Spin';
    enemyStuckType = 2;
    myAttack(15, 2, true, false, 1);
  }
  else if (enemyStuck > 0){
    myAttack(15, 2, true, false, 1);
  }
}
```

```
function myFireBlast(){
  if (Math.random() > 0.7 && (enemyTeam[currentEnemy].getType1() != 2 &&
enemyTeam[currentEnemy].getType2() != 2)){
    enemyStatusReset();
    enemyBurned = true;
  }
    myAttack(120, 2, true, false, 0.85);
}
function myEarthquake(){
  myAttack(100, 9, false, false, 1);
}
function mySwordsDance(){
  if (mySwordsDances < 3.5){
  mySwordsDances++;
  }
  else{
    mySwordsDances = 4;
  }
}
function mySurf(){
  myAttack(95, 4, true, false, 1);
}
function myBlizzard(){
  if (Math.random() > 0.9 && (enemyTeam[currentEnemy].getType1() != 12 &&
enemyTeam[currentEnemy].getType2() != 12) && enemyFrozen == 0){
    enemyStatusReset();
    enemyFrozen += 5;
  }
  myAttack(120, 12, true, false, 0.9);
}
function myRest(){
  if (myAsleep == 0){
  myStatusReset();
meAsleep += 2;
  myHPbar = 0;
  }
}
function myStunSpore(){
  if (Math.random() > 0.25 && (enemyTeam[currentEnemy].getType1() != 8 &&
enemyTeam[currentEnemy].getType2() != 8)){
    enemyStatusReset();
    enemyParalyzed = true;
  }
}
```

```
function myPsychic(){
  if (Math.random() > 0.66 ){
    if (enemySpecial <= 1){
    if (enemySpecial == 1){
      enemySpecial = 0.66;
    }
    if (enemySpecial == 0.66){
      enemySpecial = 0.5;
    }
    if (enemySpecial == 0.5){
      enemySpecial = 0.33;
    }
    if (enemySpecial == 0.33){
      enemySpecial = 0.25;
    }
    }
    else{
      enemySpecial -= 0.5;
    }
  }
  myAttack(90, 10, true, false, 1);
}
function myMegaDrain(){
  myAttack(40, 6, true, false, 1);
  if (myHPbar >= 15){
  myHPbar -= 15;
}
  else{
    myHPbar = 0;
  }
}
function myAgility(){
  myAgilitys += 1;
}
function myTwineedle(){
  if (Math.random() > 0.8){
    enemyStatusReset();
    enemyPoison = true;
  }
  myAttack(25, 13, false, false, 1);
  myAttack(25, 13, false, false, 1);
}
function myHyperBeam(){
  myAttack(150, 1, false, false, 0.9);
  if (enemyHPbar < 100){
    myRecharge += 1;
  }
}
function myDoubleEdge(){
  myAttack(100, 1, false, false, 1);
  myHPbar += 10;
}
function myMirrorMove(){
  myAttack(myHPbar, 1, false, false, 1);
}
function mySuperFang(){
  enemyHPbar += (100-enemyHPbar)/2;
}
function myThunderbolt(){
  if (Math.random() > 0.9 && (enemyTeam[currentEnemy].getType1() != 8 &&
enemyTeam[currentEnemy].getType2() != 8)){
    enemyStatusReset();
    enemyParalyzed = true;
  }
  myAttack(95, 8, true, false, 1);
}
function myDrillPeck(){
  myAttack(80, 5, false, false, 1);
}
function myGlare(){
  if (Math.random() > 0.25 && (enemyTeam[currentEnemy].getType1() != 8 &&
enemyTeam[currentEnemy].getType2() != 8)){
    enemyStatusReset();
    enemyParalyzed = true;
  }
}
function myWrap(){
  if (Math.random() > 0.3 && enemyStuck == 0){
    W = (int)(Math.random()*4)+2;
    enemyStatusReset();
    enemyStuck += W;
    enemyStuckName = 'Wrap';
    enemyStuckType = 1;
    myAttack(15, 1, true, false, 1);
  }
  else if (enemyStuck > 0){
    myAttack(15, 1, true, false, 1);
  }
```

```
}
function myThunderWave(){
  if ((enemyTeam[currentEnemy].getType1() != 8 && enemyTeam[currentEnemy].getType2() !=
8)){
  enemyStatusReset();
  enemyParalyzed = true;
  }
}
function myRockSlide(){
  myAttack(75, 11, false, false, 0.9);
}
function myConfuseRay(){
  enemyStatusReset();
  enemyConfusion += (int)(Math.random()*4)+2;
}
function myCounter(){
  myAttack(myHPbar, 1, false, false, 1);
}
function myScreech(){
  if (enemyDef <= 1)
  enemyDef /= 2;
  else if (enemyDef <= 4){
    enemyDef--;
  }
}
function mySpore(){
  if (enemyAsleep == 0){
  enemyStatusReset();
  enemyAsleep += (int)(Math.random()*4)+2;
}
}
function mySlash(){
  myAttack(70, 1, false, true, 1);
}
function myAmnesia(){
  if (mySpecial < 0.6){
    mySpecial *= 2;
  }
  else if (mySpecial == 0.66){
    mySpecial = 1.5;
  }
  else{
  mySpecial += 1;
  }
  }
function myHydroPump(){
  myAttack(120, 4, true, false, 0.8);
}
function mySubmission(){
  myAttack(80, 3, false, false, 0.8);
  myHPbar += 10;
}
function myReflect(){
  if (myDef <= 1)
  myDef *= 2;
  else if (myDef < 4)
    myDef++;
}
function myRecover(){
  if (myHPbar > 50){
    myHPbar -= 50;
  }
  else{
    myHPbar = 0;
  }
}
function myExplosion(){
  myAttack(150, 1, false, false, 1);
  myHPbar = 101;
}
function mySludge(){
  if (Math.random() > 0.6){
    enemyStatusReset();
    enemyPoison = true;
  }
  myAttack(65, 7, false, false, 1);
}
function myClamp(){
  if (Math.random() > 0.3 && enemyStuck == 0){
    C = (int)(Math.random()*4)+2;
    enemyStatusReset();
    enemyStuck += C;
    enemyStuckName = 'Clamp';
    enemyStuckType = 4;
    myAttack(15, 4, true, false, 1);
  }
  else if (enemyStuck > 0){
    myAttack(15, 4, true, false, 1);
  }
}
function myHypnosis(){
  if ((Math.random() > 0.4) && enemyAsleep == 0){
    enemyStatusReset();
    enemyAsleep += (int)(Math.random()*4)+2;
  }
}
function myNightShade(){
  myAttack(myTeam[currentMine].getLvl(), 0, true, false, 1);
}
function myCrabhammer(){
  myAttack(90, 4, false, true, 0.85);
}
function myThunder(){
  if (Math.random() > 0.9 && (enemyTeam[currentEnemy].getType1() != 8 &&
enemyTeam[currentEnemy].getType2() != 8)){
    enemyStatusReset();
    enemyParalyzed = true;
  }
  myAttack(120, 8, true, false, 0.7);
}
function myMeditate(){
  if (mySwordsDances < 4){
  mySwordsDances += 0.5;
  }
}
function myHighJumpKick(){
  myAttack(85, 3, false, false, 0.9)
}
function mySeismicToss(){
  myAttack(myTeam[currentMine].getLvl(), 0, false, false, 1);
}
function mySoftBoiled(){
  if (myHPbar > 50){
    myHPbar -= 50;
  }
  else{
    myHPbar = 0;
  }
}
function myGrowth(){
  if (mySpecial < 1){
    if (mySpecial == 0.25){
      mySpecial = 0.33;
    }
    if (mySpecial == 0.33){
      mySpecial = 0.5;
    }
    if (mySpecial == 0.5){
      mySpecial = 0.66;
    }
    if (mySpecial == 0.66){
      mySpecial = 1;
    }
  }
  else if (mySpecial < 4){
    mySpecial += 0.5;
  }
}
function myLovelyKiss(){
  if ((Math.random() > 0.25) && enemyAsleep == 0){
    enemyStatusReset();
    enemyAsleep += (int)(Math.random()*4)+2;
  }
}
function myTransform(){
  enemyHPbar = 101;
  myHPbar = 101;
}
function myDoubleKick(){
  myAttack(30, 3, false, false, 1);
  myAttack(30, 3, false, false, 1);
}
function myMegaKick(){
  myAttack(120, 1, false, false, 0.75)
}
function enemySleepPowder(){
  if ((Math.random() > 0.25) && meAsleep == 0){
    myStatusReset();
    meAsleep += (int)(Math.random()*4)+2;
  }
}
function enemyRazorLeaf(){
  enemyAttack(55, 6, true, true, 0.95);
}
function enemyBodySlam(){
  if (Math.random() > 0.7 && (myTeam[currentMine].getType1() != 8 &&
myTeam[currentMine].getType2() != 8)){
    myStatusReset();
    meParalyzed = true;
  }
  enemyAttack(85, 1, false, false, 1);
}
function enemyToxic(){
  if (Math.random() > 0.15){
    myStatusReset();
    meToxic = true;
  }
}
function enemyFireSpin(){
  if (Math.random() > 0.3 && meStuck == 0){
    FS = (int)(Math.random()*4)+2;
    myStatusReset();
    meStuck += FS;
    myStuckType = 2;
    myStuckName = 'Fire Spin';
    enemyAttack(15, 2, true, false, 1);
  }
  else if (meStuck > 0){
    enemyAttack(15, 2, true, false, 1);
  }
}
function enemyFireBlast(){
  if (Math.random() > 0.7 && (myTeam[currentMine].getType1() != 2 &&
myTeam[currentMine].getType2() != 2)){
    myStatusReset();
    meBurned = true;
  }
    enemyAttack(120, 2, true, false, 0.85);

}
function enemyEarthquake(){
  enemyAttack(100, 9, false, false, 1);
}
function enemySwordsDance(){
  if (enemySwordsDances < 3.5)
  enemySwordsDances++;
  else
    enemySwordsDances = 4;
}
```

```
function enemySurf(){
 enemyAttack(95, 4, true, false, 1);
}
function enemyBlizzard(){
 if ((Math.random() > 0.9) && myFrozen == 0 && (myTeam[currentMine].getType1() != 12 &&
myTeam[currentMine].getType2() != 12)){
  myStatusReset();
  myFrozen += 5;
 }
 enemyAttack(120, 12, true, false, 0.9);
}
function enemyRest(){
 if (enemyAsleep == 0){
 enemyStatusReset();
 enemyAsleep += 2;
 enemyHPbar = 0;
 }
}
function enemyStunSpore(){
 if (Math.random() > 0.25 && (myTeam[currentMine].getType1() != 8 &&
myTeam[currentMine].getType2() != 8)){
  myStatusReset();
  meParalyzed = true;
 }
}
function enemyPsychic(){
 if (Math.random() > 0.66){
  if (mySpecial <= 1){
  if (mySpecial == 1){
   mySpecial = 0.66;
  }
  if (mySpecial == 0.66){
   mySpecial = 0.5;
  }
  if (mySpecial == 0.5){
   mySpecial = 0.33;
  }
  if (mySpecial == 0.33){
   mySpecial = 0.25;
  }
  }
  else{
   mySpecial -= 0.5;
  }
 }
 enemyAttack(90, 10, true, false, 1);
}
function enemyMegaDrain(){
 enemyAttack(40, 6, true, false, 1);
 if (enemyHPbar >= 15){
 enemyHPbar -= 15;
 }
 else{
  enemyHPbar = 0;
 }
}
function enemyAgility(){
 enemyAgilitys += 1;
}
function enemyTwineedle(){
 if (Math.random() > 0.8){
  myStatusReset();
  myPoison = true;
 }
 enemyAttack(25, 13, false, false, 1);
 enemyAttack(25, 13, false, false, 1);
}
function enemyHyperBeam(){
 enemyAttack(150, 1, false, false, 0.9);
 if (myHPbar < 100){
  enemyRecharge += 1;
 }
}
function enemyDoubleEdge(){
 enemyAttack(100, 1, false, false, 1);
 enemyHPbar += 10;
}
function enemyMirrorMove(){
 enemyAttack(enemyHPbar, 1, false, false, 1);
}
function enemySuperFang(){
 myHPbar += (100-myHPbar)/2;
}
function enemyThunderbolt(){
 if (Math.random() > 0.9 && (myTeam[currentMine].getType1() != 8 &&
myTeam[currentMine].getType2() != 8)){
  myStatusReset();
  meParalyzed = true;
  }
  enemyAttack(95, 8, true, false, 1);
}
function enemyDrillPeck(){
  enemyAttack(80, 5, false, false, 1);
}
function enemyGlare(){
  if (Math.random() > 0.25 && (myTeam[currentMine].getType1() != 8 &&
myTeam[currentMine].getType2() != 8)){
    myStatusReset();
    meParalyzed = true;
  }
}
function enemyWrap(){
  if (Math.random() > 0.3 && meStuck == 0){
    W = (int)(Math.random()*4)+2;
    myStatusReset();
    meStuck += W;
    myStuckType = 1;
    myStuckName = 'Wrap';
    enemyAttack(15, 1, true, false, 1);
  }
  else if (meStuck > 0){
    enemyAttack(15, 1, true, false, 1);
  }
}
function enemyThunderWave(){
  if (myTeam[currentMine].getType1() != 8 && myTeam[currentMine].getType2() != 8){
  myStatusReset();
  meParalyzed = true;
}
}
function enemyRockSlide(){
  enemyAttack(75, 11, false, false, 0.9);
}
function enemyConfuseRay(){
  myStatusReset();
  myConfusion += (int)(Math.random()*4)+2;
}
function enemyCounter(){
  enemyAttack(enemyHPbar, 1, false, false, 1);
}
function enemyScreech(){
  if (myDef <= 1)
  myDef /= 2;
  else if (myDef <= 4){
    myDef--;
  }
}
function enemySpore(){
  if (meAsleep == 0){
  myStatusReset();
  meAsleep += (int)(Math.random()*4)+2;
}
}
function enemySlash(){
  enemyAttack(70, 1, false, true, 1);
}
function enemyAmnesia(){
  if (enemySpecial < 0.6){
    enemySpecial *= 2;
  }
  else if (enemySpecial == 0.66){
    enemySpecial = 1.5;
  }
  else{
  enemySpecial += 1;
  }
}
function enemyHydroPump(){
  enemyAttack(120, 4, true, false, 0.8);
}
function enemySubmission(){
  enemyAttack(80, 3, false, false, 0.8);
  enemyHPbar += 10;
}
function enemyReflect(){
  if (enemyDef <= 1)
  enemyDef *= 2;
  else if (enemyDef < 4)
    enemyDef++;
}
function enemyRecover(){
  if (enemyHPbar > 50){
    enemyHPbar -= 50;
  }
  else{
    enemyHPbar = 0;
  }
}
function enemyExplosion(){
  enemyAttack(150, 1, false, false, 1);
  enemyHPbar = 101;
}
function enemySludge(){
  if (Math.random() > 0.6 && (enemyTeam[currentEnemy].getType1() != 8 &&
enemyTeam[currentEnemy].getType2() != 8)){
    myStatusReset()
    myPoison = true;
  }
  myAttack(65, 7, false, false, 1);
}
function enemyClamp(){
  if (Math.random() > 0.3 && meStuck  == 0){
    C = (int)(Math.random()*4)+2;
    myStatusReset();
    meStuck += C;
    myStuckType = 4;
    myStuckName = 'Clamp';
    enemyAttack(15, 4, true, false, 1);
  }
  else if (meStuck > 0){
    enemyAttack(15, 4, true, false, 1);
  }
}
function enemyHypnosis(){
  if ((Math.random() > 0.4) && meAsleep == 0){
    myStatusReset();
    meAsleep += (int)(Math.random()*4)+2;
  }
}
function enemyNightShade(){
  enemyAttack(enemyTeam[currentEnemy].getLvl(), 0, true, false, 1);
}
function enemyCrabhammer(){
  enemyAttack(90, 4, false, true, 0.85);
}
function enemyThunder(){
  if (Math.random() > 0.9 && (myTeam[currentMine].getType1() != 8 &&
myTeam[currentMine].getType2() != 8)){
    myStatusReset();
    meParalyzed = true;
  }
  enemyAttack(120, 8, true, false, 0.7);
}
function enemyMeditate(){
  if (enemySwordsDances < 4)
  enemySwordsDances += 0.5;
}
function enemyHighJumpKick(){
  enemyAttack(85, 3, false, false, 0.9)
}
function enemySeismicToss(){
  enemyAttack(enemyTeam[currentEnemy].getLvl(), 0, false, false, 1);
}
function enemySoftBoiled(){
  if (enemyHPbar > 50){
    enemyHPbar -= 50;
  }
  else{
    enemyHPbar = 0;
  }
}
function enemyGrowth(){
  if (enemySpecial < 1){
    if (enemySpecial == 0.25){
      enemySpecial = 0.33;
    }
    if (enemySpecial == 0.33){
      enemySpecial = 0.5;
    }
    if (enemySpecial == 0.5){
      enemySpecial = 0.66;
    }
    if (enemySpecial == 0.66){
      enemySpecial = 1;
    }
  }
  else if (enemySpecial < 4){
    enemySpecial += 0.5;
  }
}
function enemyLovelyKiss(){
  if ((Math.random() > 0.25) && meAsleep == 0){
    myStatusReset();
    meAsleep += (int)(Math.random()*4)+2;
  }
}
function enemyTransform(){
  enemyHPbar = 101;
  myHPbar = 101;
}
function enemyDoubleKick(){
  enemyAttack(30, 3, false, false, 1);
  enemyAttack(30, 3, false, false, 1);
}
function enemyMegaKick(){
  enemyAttack(120, 1, false, false, 0.75)
}
function enemyStatusReset(){
enemyAsleep = 0;
enemyParalyzed = false;
enemyToxiced = false;
enemyStuck = 0;
enemyBurned = false;
enemyPoison = false;
enemyConfusion = 0;
enemyFrozen = false;
}
function myStatusReset(){
meAsleep = 0;
meParalyzed = false;
meToxic = false;
meStuck = 0;
meBurned = false;
myPoison = false;
myConfusion = 0;
myFrozen = false;
}
// "It just works."//
```