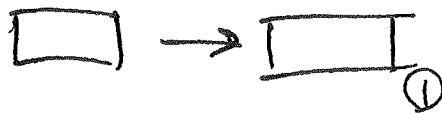


v2



hC = \emptyset ;

for (q = 1; q <= maxBricks; q++) {

xB = _root["brick" + q].x;

yB = _root["brick" + q].y;

wB = _root["brick" + q].width / 2;

hB = _root["brick" + q].height / 2;

quad = \emptyset ; side = \emptyset ;

if (xB - wB >= xN + wN &&

yB + hB <= yN - hN) {

quad = 1; aL = 1; bL = 1; aH = -1; bH = -1; }

else {

if (xB + wB <= xN - wN &&

yB + hB <= yN - yB) {

quad = 2; aL = -1; bL = 1; aH = 1; bH = -1; }

else {

if (xB + wB <= xN - wN &&

yB - hB >= yN + hN) {

quad = 3; aL = -1; bL = -1; aH = 1; bH = +1; }

else {

if (xB - wB >= xN + wN &&

yB - hB >= yN + hN) {

quad = 4; aL = 1; bL = -1; aH = -1; bH = 1; }

}

}

```

}
if(quad == 0){
    if(xB - wB < xN + wN)
        xB + wB > xN - wN){
        if(yB + hB <= yN - hN){
            side = "t"; aL = 1; bL = 1; aH = 1; bH = -1;
        }
        else{
            side = "b"; aL = -1; bL = -1; aH = -1; bH = 1;
        }
    }
}
else{
    if(xB + wB <= xN - wN){
        side = "l"; aL = -1; bL = 1; aH = 1; bH = +1;
    }
    else{
        side = "r"; aL = 1; bL = -1; aH = 1; bH = -1;
    }
}
}
}

```

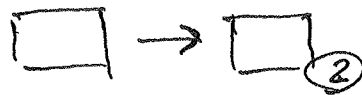
// get low

```

a = yN - (yB + ((hB + hN) * aL));
b = (xN - (xB + ((wB + wN) * bL))) * -1;
low = Math.atan2(a, b) * rd;
if(low < 0){
    low += 360;
}

```

√2



```
// get High
```

```
a = yN - (yB + ((hB + hN) * aH));
```

```
b = (xN - (xB + ((wB + wN) * bH))) * -1;
```

```
high = Math.atan2(a, b) * rd;
```

```
if (high < 0) {
```

```
    high += 360; }
```

```
if ((tD > low && tD < high) ||
```

```
    (tD - 360 > low && tD - 360 < high)) {
```

```
    // in sight
```

v2



```
if(quad==1){
  a = yN - (yB + hB + hN);
  b = (xN - (xB - wB - wN)) * -1;
  C1 = Math.atan2(a, b) * rd;
  if(tD < C1){
    a2 = Math.abs(a);
    if(a2 == 0){
      a2 = .01;
    }
    g = (1 / (Math.sin(tD * dg))) * a2;
    if(tV > g){
      hC++;
      eval("x" + hC) = Math.cos(tD * dg) * g;
      eval("y" + hC) = Math.sin(tD * dg) * g;
      eval("p" + hC) = g;
      eval("rebound" + hC) = tD - (180 - (2 * tD));
    }
  }
}
else {
  b2 = Math.abs(b);
  if(b2 == 0){
    b2 = .01;
  }
  g = (1 / (Math.cos(tD * dg))) * b2;
  if(tV > g){
    eval("x" + hC) = Math.cos(tD * dg) * g;
    hC++;
  }
}
```

```

eval("y"+hC) = Math.sin(tD*dg)*g;
eval("p"+hC) = g;
eval("rebound"+hC) = tD + (180 - (2*tD));
    }
}
}
if (quad == 2) {
    a = yN - (yB + hB + hN);
    b = (xN - (xB + wB + wN)) * -1;
    C1 = Math.atan2(a, b) * rd;
    if (tD < C1) {
        b2 = Math.abs(b);
        if (b2 == 0) {
            b2 = .01;
        }
        g = (1 / (Math.cos(tD*dg))) * b2;
        if (tV > g) {
            hC++;
            eval("x"+hC) = Math.cos(tD*dg)*g;
            eval("y"+hC) = Math.sin(tD*dg)*g;
            eval("p"+hC) = g;
            eval("rebound"+hC) = tD - (180 - (2*(180-tD)));
        }
    }
} else {
    a2 = Math.abs(a);

```

```

eval("y"+hC)=Math.sin(tD*dg)*g;
eval("p"+hC)=g;
eval("rebound"+hC)=tD-(2*(tD-180));

```

```

}

```

```

}

```

```

else {

```

```

    b2=Math.abs(b);

```

```

    if(b2==0){

```

```

        b2=.01;
    }

```

```

    g=(1/(Math.cos(tD*dg)))*b2;

```

```

    if(tV>g){

```

```

        hC++;

```

```

        eval("x"+hC)=Math.cos(tD*dg)*g;

```

```

        eval("y"+hC)=Math.sin(tD*dg)*g;

```

```

        eval("p"+hC)=g;

```

```

        eval("rebound"+hC)=tD-(180-(2*(tD-180)));
    }

```

```

}

```

```

}

```

```

}

```

```

else { if(quad==4){

```

```

#quad 4

```

```

a=yN-(yB-hB-hN);

```

```

b=(xN-(xB-wB-wN))*-1;

```

```

C1=Math.atan2(a,b)*rd;

```

```

C1+=360;

```

```

if(tD<C1){

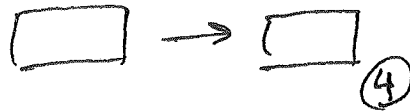
```

```

    a2=Math.abs(a);

```

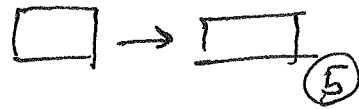
v2



```
if(a2 == 0){
    a2 = .01;
}
g = (1 / (Math.sin(tD * dg))) * a2;
if(tV > g){
    hC++;
    eval("x" + hC) = Math.cos(tD * dg) * g;
    eval("y" + hC) = Math.sin(tD * dg) * g;
    eval("p" + hC) = g;
    eval("rebound" + hC) = tD + (2 * (180 - tD));
}
}
```

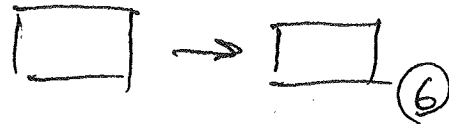
```
if(quad == 3){
    a = yN - (yB - hB - hN);
    b = (xN - (xB + wB + wN)) * -1;
    C1 = Math.atan2(a, b) * rd;
    C1 += 360;
    if(tD < C1){
        a2 = Math.abs(a);
        if(a2 == 0){
            a2 = .01;
        }
        g = (1 / (Math.sin(tD * dg))) * a2;
        if(tV > g){
            hC++;
            eval("x" + hC) = Math.cos(tD * dg) * g;
        }
    }
}
```

v2



```
if(a2==0){
    a2=.01;}
g=(1/(Math.sin(tD*dg)))*a2;
if(tV>g){
    hC++;
    eval("x"+hC)=Math.cos(tD*dg)*g;
    eval("y"+hC)=Math.sin(tD*dg)*g;
    eval("p"+hC)=g;
    eval("rebound"+hC)=tD-(180-(2*(360-tD)));
}
}
else{
    b2=Math.abs(b);
    if(b2==0){
        b2=.01;}
    g=(1/(Math.cos(tD*dg)))*b2;
    if(tV>g){
        hC++;
        eval("x"+hC)=Math.cos(tD*dg)*g;
        eval("y"+hC)=Math.sin(tD*dg)*g;
        eval("p"+hC)=g;
        eval("rebound"+hC)=tD+(180-(2*(tD-270)));
    }
}
}
```


v2



```
if(side == "t"){
  a = yN - (yB + hB + hN);
  a2 = Math.abs(a);
  if(a2 == 0){
    a2 = .01;
  }
  g = (1 / (Math.sin(tD * dg))) * a2;
  if(tV > g){
    hc++;
    eval("x" + hc) = Math.cos(tD * dg) * g;
    eval("y" + hc) = Math.sin(tD * dg) * g;
    eval("p" + hc) = g;
    if(tD < 90){
      eval("rebound" + hc) = tD - (2 * (tD));
    }
    else {
      eval("rebound" + hc) = tD + (2 * (180 - tD));
    }
  }
}

if(side == "l"){
  b = (xN - (xB + wB + wN)) * -1;
  b2 = Math.abs(b);
  if(b2 == 0){
    b2 = .01;
  }
  g = (1 / (Math.cos(tD * dg))) * b2;
```

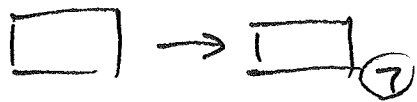
```

if(tV > g) {
    hC++;
    eval("x"+hC) = Math.cos(tD*dg)*g;
    eval("y"+hC) = Math.sin(tD*dg)*g;
    eval("p"+hC) = g;
    if(tD < 180) {
        eval("rebound"+hC) = tD - (2*(tD-90));
    }
    else {
        eval("rebound"+hC) = tD + (2*(270-tD));
    }
}
}

if(side == "b") {
    a = yN - (yB + hB + hN);
    a2 = Math.abs(a);
    if(a2 == 0) {
        a2 = .01;
    }
    g = (1/(Math.sin(tD*dg))) * a2;
    if(tV > g) {
        hC++;
        eval("x"+hC) = Math.cos(tD*dg)*g;
        eval("y"+hC) = Math.sin(tD*dg)*g;
        eval("p"+hC) = g;
        if(tD < 270) {
            eval("rebound"+hC) = tD - (2*(tD-180));
        }
    }
}

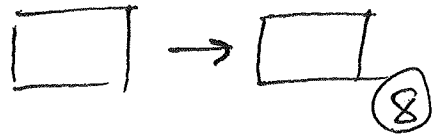
```

v2



```
    else {
        eval("rebound"+hC) = tD + (2*(360 - tD));
    }
}
}
}
else { if (side == "r") {
    // side == "r"
    b = (xN - (xB - wB - wN)) * -1;
    b2 = Math.abs(b);
    if (b2 == 0) {
        b2 = .01;
    }
    g = (1 / (Math.cos(tD * dg))) * b2;
    if (tV > g) {
        hC++;
        eval("x"+hC) = Math.cos(tD * dg) * g;
        eval("y"+hC) = Math.sin(tD * dg) * g;
        eval("p"+hC) = g;
        if (tD > 270) {
            eval("rebound"+hC) = tD - (2*(tD - 270));
        }
        else {
            eval("rebound"+hC) = tD + (2*(90 - tD));
        }
    }
}
}
}
```

√2



```
winner = 10,000;
if (hc > 0) {
  for (q = 1; q <= hc; q++) {
    if (eval("p" + q) < winner) {
      winner = eval("p" + q);
      winNum = q;
    }
  }
  xProp1 = eval("x" + winNum);
  yProp1 = eval("y" + winNum);
  proxy1 = eval("p" + winNum);
  reboundProp1 = eval("rebound" + winNum);
}
```

}}}

insert □ → ○

get back xProp2 / yProp2
proxy 2 / reboundProp2

}}}

```
if (proxy1 < proxy2) {
  xProp = xProp1; reboundProp = reboundProp1;
  yProp = yProp1; }
else {
  xProp = xProp2; reboundProp = reboundProp2;
  yProp = yProp2; }
```