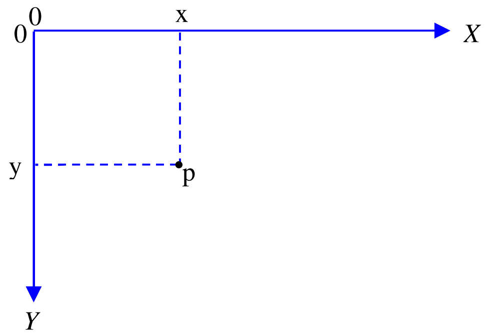
5

5 Col·lisions.

En aquesta lliçó veurem diferents algoritmes de detecció de col·lisió entre diferents objectes: cercle-cercle, rectangle-rectangle, rectangle-cercle i segment-segment.

5.1 Punt.

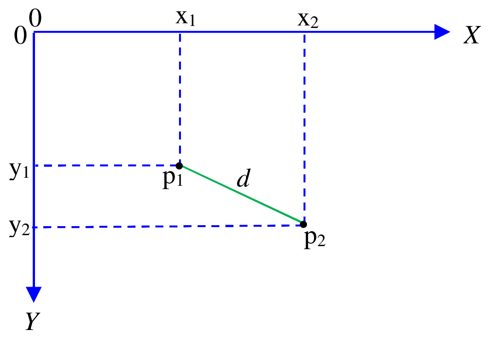
Un punt p en el pla 2D, ve determinat per la parella de valors x y, atenent a un sistema de coordenades cartesià donat.




```
function Punt(x,y){
  this.x=x;
  this.y=y;
}
```

5.1.1 Distància entre dos punts.

Es defineix la distància entre dos punts com la longitud del segment mínim que uneix els dos punts i es calcula utilitzant el Teorema de Pitàgores:

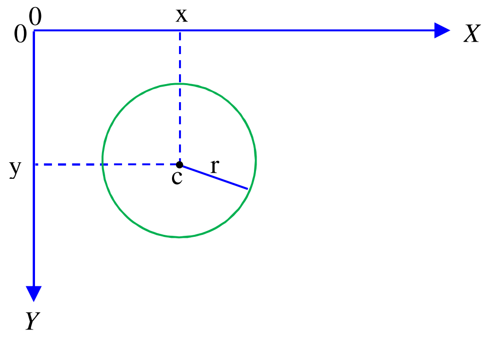


$$distància, d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
function distancia(p1,p2){
  return Math.sqrt(Math.power(p2.x-p1.x,2)+ Math.power(p2.y-p1.y,2));
}
```

5.2 Cercle.

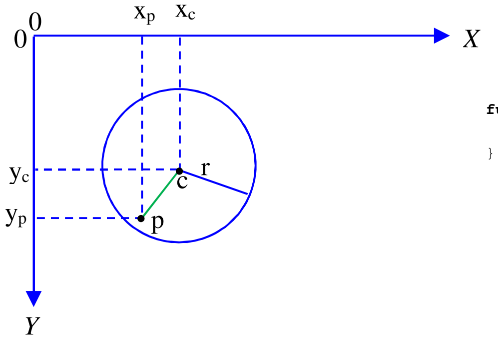
Un cercle en el pla 2D, ve determinat pel seu centre (un punt) i el seu radi.




```
function Cercle(centre,radi){
  this.c=centre;  // és un Punt
  this.r=radi;
}
```

5.2.1 Punt interior a un cercle.

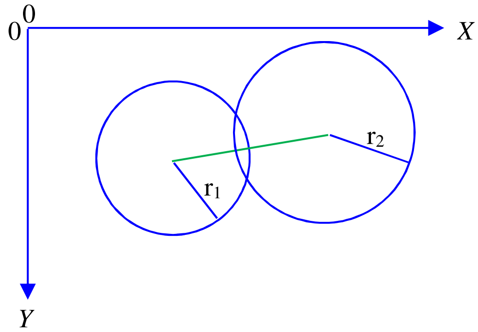
Un punt p és interior a un cercle si la distància entre el punt p i el centre del cercle és menor al radi r del cercle,




```
function puntInteriorCercle(cercle,p){
  return distancia(cercle.c,p) < cercle.r ;
}
```

5.2.2 Col·lisió entre dos cercles.

Dos cercles col·lisionen si la distancia entre els seus centres és menor a la suma del radis dels dos cercles,

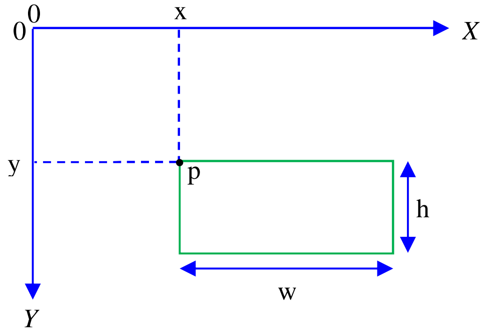



```
function col_lisioCercles(c1,c2){
  return distancia(c1.c,c2.c) < c1.r + c2.r ;
}
```

http://ssh.eupmt.tecnocampus.cat/~jou/FP/capitol_3/exemples/exXocExterior2.html

5.3 Rectangle.

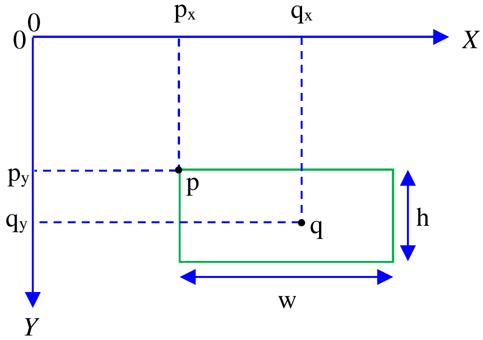
Un rectangle en el pla 2D, ve determinat per la seva posició (un punt, normalment la cantonada esquerra superior del rectangle) i la seva amplada i alçada.




```
function Rectangle(posicio, amplada, alçada){
  this.p=posicio;  // és un Punt
  this.w=amplada;
  this.h=alçada;
}
```

5.3.1 Punt interior a un rectangle.

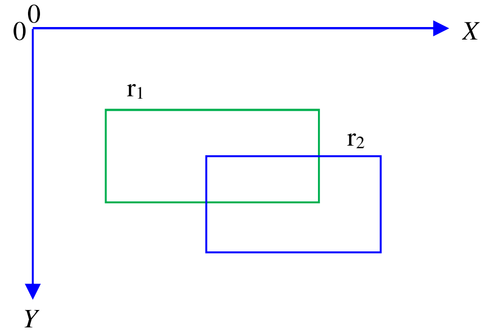
Un punt q és interior a un rectangle si es compleix la següent condició,




```
function puntInteriorRectangle(r,q){
  return q.x >= r.p.x && q.x <= r.p.x+r.w &&
         q.y >= r.p.y && q.y <= r.p.y+r.h ;
}
```

5.3.2 Col·lisió entre dos rectangles.

Dos rectangles col·lisionen si compleixen la següent condició,



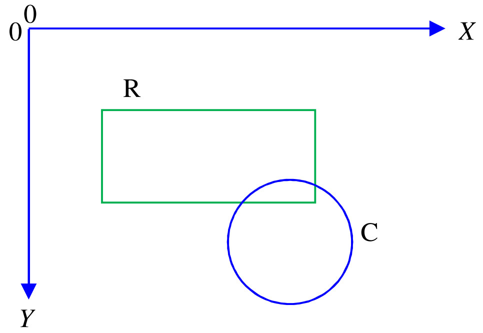

```
function col_lisioRectangles(r1,r2){
  return !(r1.p.x+r1.w < r2.p.x || r1.p.x > r2.p.x+r2.w ||
           r1.p.y+r1.h < r2.p.y || r1.p.y > r2.p.y+r2.h ) ;
}
```

Observeu que és més fàcil calcular la (no col·lisió) i retornar la no (no col·lisió)

http://ssh.eupmt.tecnocampus.cat/~jou/FP/capitol_3/exemples/exXocExterior1.html

5.3.3 Col·lisió entre un cercle i un rectangle.

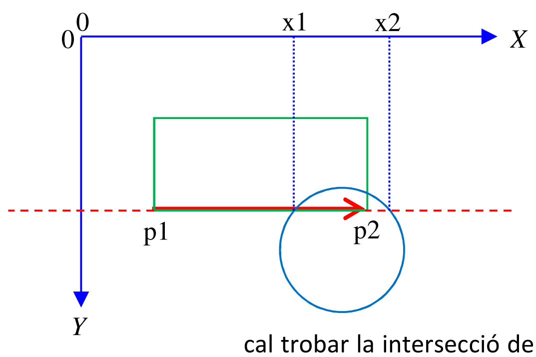
Un cercle col·lisiona amb un rectangle si compleixen la següent condició,




```
function interseccioRectangleCercle(R,C){
  return iHC({p1:R.posicio,                         p2:{x:R.posicio.x+R.amplada, y:R.posicio.y}          },C) ||
         iHC({p1:{x:R.posicio.x, y:R.posicio.y+R.alçada},  p2:{x:R.posicio.x+R.amplada, y:R.posicio.y+R.alçada}},C) ||
         iVC({p1:R.posicio,                         p2:{x:R.posicio.x,           y:R.posicio.y+R.alçada}},C) ||
         iVC({p1:{x:R.posicio.x+R.amplada, y:R.posicio.y}, p2:{x:R.posicio.x+R.amplada, y:R.posicio.y+R.alçada}},C) ;

}
function iHC (S,C){
  var d = Math.pow(C.radi,2)-Math.pow(S.p1.y-C.centre.y,2);
  var x1, x2;
  if(d >= 0){
    x1 = C.centre.x - Math.sqrt(d);
    x2 = C.centre.x + Math.sqrt(d);
    return S.p1.x<=x2 && S.p2.x >=x1;
  }
  else
    return false;
}

function iVC (S,C){
  var d = Math.pow(C.radi,2)-Math.pow(S.p1.x-C.centre.x,2);
  var y1, y2;
  if(d >= 0){
    y1 = C.centre.y - Math.sqrt(d);
    y2 = C.centre.y + Math.sqrt(d);
    return S.p1.y<=y2 && S.p2.y >=y1;
  }
  else
    return false;
}
```

(no col·lisió) si `p1.x > x2 || p2.x < x1`, per tant, no (no col·lisió) serà, aplicant De Morgan,

`p1.x <= x2 && p2.x  >= x1`



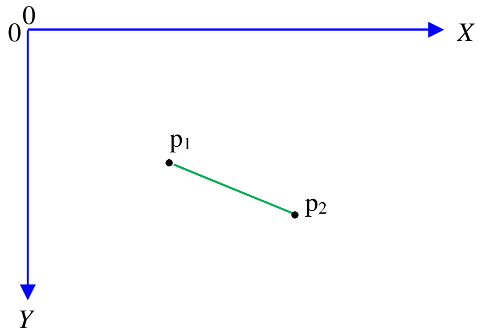
cal trobar la intersecció de la circumferència amb la recta horitzontal,

$$\begin{cases} (x - x_c)^2 + (y - y_c)^2 = r^2 \\ y = k \end{cases}$$

http://ssh.eupmt.tecnocampus.cat/~jou/FP/capitol_3/exemples/exXocExterior4.html

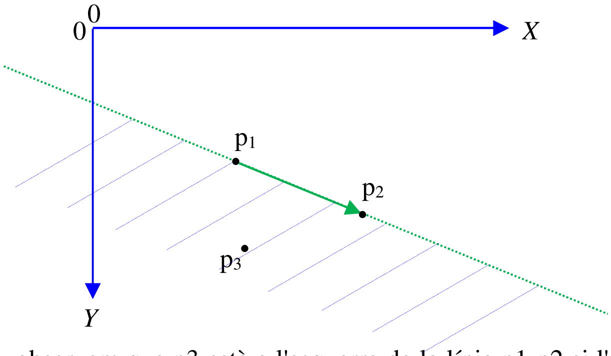
5.4 Segment.

Un segment és un tram de recta delimitat per dos punts,




```
function Segment(p1,p2){
  this.p1=p1; // és un Punt
  this.p2=p2; // un altre Punt
}
```

5.4.1 Punt en el semiplà dret d'un segment.

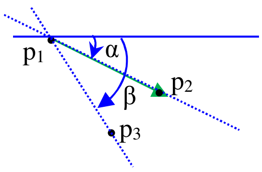


observem que p3 està a l'esquerra de la línia p1-p2 si l'angle β és més gran que l'angle α. Però, aquests angles estan directament relacionats amb el pendent de les rectes que uneixen els punts p1-p2 i p1-p3 essent aquests pendents:

$$\alpha = \frac{p2.y - p1.y}{p2.x - p1.x} \qquad \beta = \frac{p3.y - p1.y}{p3.x - p1.x}$$

per tant,

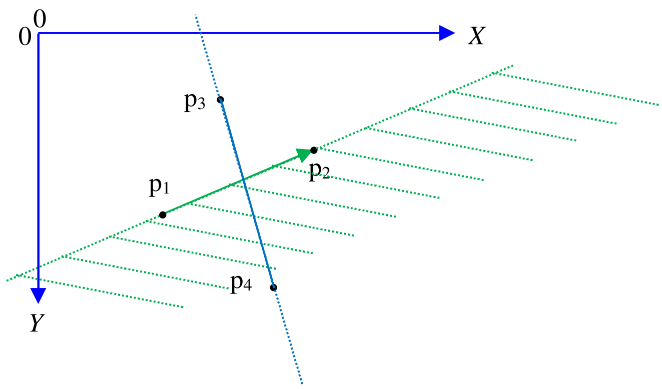
```
function check(p1,p2,p3){
  return (p2.y-p1.y)*(p3.x-p1.x) < (p3.y-p1.y)*(p2.x-p1.x);

}
```

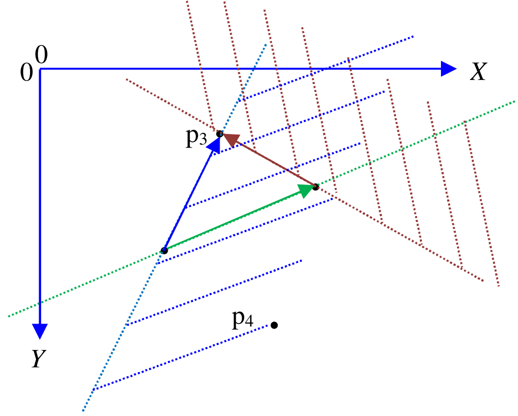


http://ssh.eupmt.tecnocampus.cat/~jou/LM/exercicis/exercici_4-2a.html

5.4.2 Intersecció de dos segments.

Dos segments, p1-p2 i p3-p4 intersequen si p3 i p4 estan en els semiplans contraris formats per la línia p1==>p2 i p4 està en semiplans contraris respecte les línies p1==>p3 i p2==>p3,





per tant,

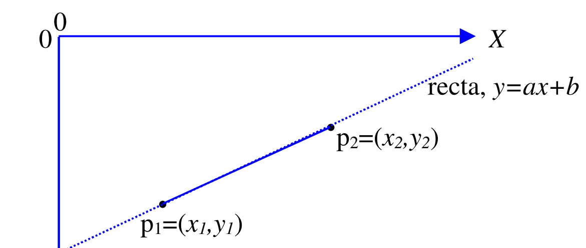
```
function interseccio2Segments(p1,p2,p3,p4){
  return check(p1,p2,p3) != check(p1,p2,p4) && check(p1,p3,p4) != check(p2,p3,p4) ;

}
```

Nota: cal considerar a part el cas p2=p3, és a dir, que els segments formin una polilínia.

5.4.3 Punt intersecció de dos segments.

Ara el problema és trobar el punt d'intersecció de dos segments si aquests intersequen. Per resoldre aquest problema, primer haurem de trobar les rectes que contenen els segments i calcular el punt intersecció de les dues rectes. Finalment, haurem de comprovar que aquest punt d'intersecció pertany als dos segments.

5.4.3.1 Recta que passa per dos punts (conté un segment).



$$\begin{cases} y_1 = ax_1 + b \rightarrow b = y_1 - ax_1 \\ y_2 = ax_2 + b \rightarrow y_2 = ax_2 + (y_1 - ax_1) \end{cases}$$

$$(y_2 - y_1) = a(x_2 - x_1)$$

$$a = \boxed{\frac{y_2 - y_1}{x_2 - x_1}}$$

$$b = y_1 - \left(\frac{y_2 - y_1}{x_2 - x_1}\right)x_1 = \frac{y_1(x_2 - x_1) - (y_2 - y_1)x_1}{x_2 - x_1} = \boxed{\frac{x_2y_1 - x_1y_2}{x_2 - x_1}}$$

per tant,

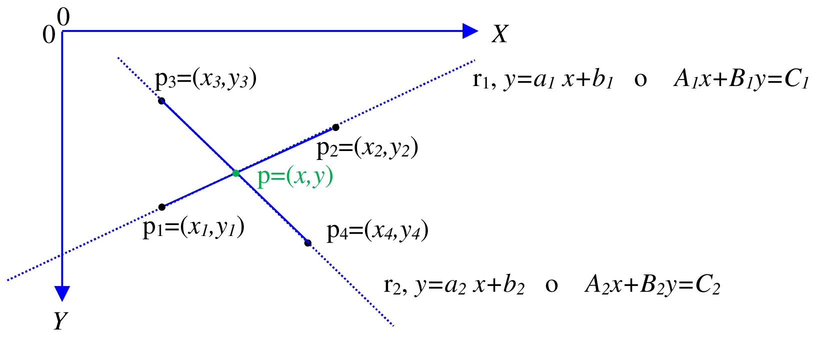
$$y = ax + b = \frac{y_2 - y_1}{x_2 - x_1}x + \frac{x_2y_1 - x_1y_2}{x_2 - x_1}$$

$$(x_2 - x_1)y = (y_2 - y_1)x + (x_2y_1 - x_1y_2)$$
$$(y_2 - y_1)x + (x_1 - x_2)y = (x_1y_2 - x_2y_1)$$

per tant

$$\boxed{Ax + By = C \text{ on } A = y_2 - y_1\,, B = x_1 - x_2 \text{ i } C = x_1y_2 - x_2y_1}$$

Ara, calcular el punt intersecció és fàcil, només cal resoldre un sistema de dues equacions amb dues incògnites,



mètode 1,

$$\begin{cases} y = a_1 x + b_1 \\ y = a_2 x + b_2 \end{cases}$$

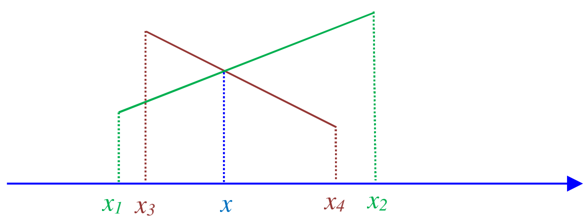
$$a_2 x + b_2 = a_1 x + b_1$$
$$a_2 x - a_1 x = b_1 - b_2$$
$$\boxed{x = \frac{b_1 - b_2}{a_2 - a_1} \quad , \qquad y = a_1 \left(\frac{b_1 - b_2}{a_2 - a_1}\right) + b_1}$$

mètode 2,

$$\begin{cases} A_1 x + B_1 y = C_1 \\ A_2 x + B_2 y = C_2 \end{cases}$$

$$\boxed{x = \frac{\begin{vmatrix} C_1 & B_1 \\ C_2 & B_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}} = \frac{C_1 B_2 - C_2 B_1}{A_1 B_2 - A_2 B_1} \quad , \qquad y = \frac{\begin{vmatrix} A_1 & C_1 \\ A_2 & C_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}} = \frac{A_1 C_2 - A_2 C_1}{A_1 B_2 - A_2 B_1}}$$

Ara només cal comprovar que el punt trobat pertany als dos segments. Això ho resolem projectant els segments i el punt intersecció en un dels eixos, el X, per exemple,

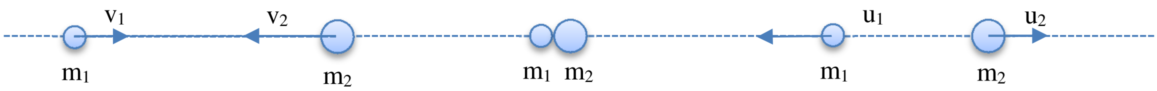


interseqüen si $x \in (x_1,x_2)$ i $x \in (x_3,x_4)$

http://ssh.eupmt.tecnocampus.cat/~jou/LM/exercicis/exercici_4-2b.html

5.5 Xoc elàstic entre dues boles (2D).

Primer resoldrem el problema en una dimensió (1D),



abans del xoc xoc elàstic després del xoc

si el xoc és elàstic, es conserva la quantitat de moviment,

$$m_1v_1 + m_2v_2 = m_1u_1 + m_2u_2$$
$$m_1(v_1 - u_1) = m_2(u_2 - v_2)$$

i l'energia cinètica,

$$\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_1v_2^2 = \frac{1}{2}m_1u_1^2 + \frac{1}{2}m_1u_2^2$$
$$m_1(v_1^2 - u_1^2) = m_2(u_2^2 - v_2^2)$$
$$m_1(v_1 + u_1)(v_1 - u_1) = m_2(u_2 + v_2)(u_2 - v_2)$$
$$m_1(v_1 + u_1)(v_1 - u_1) = m_1(u_2 + v_2)(v_1 - u_1)$$

per tant,

$$v_1 + u_1 = v_2 + u_2$$
$$v_1 - v_2 = u_2 - u_1$$
$$m_2(v_1 - v_2) = m_2(u_2 - u_1)$$
$$m_2(v_1 + u_1 - v_2) = m_2u_2$$

que substituit en la primera equació,

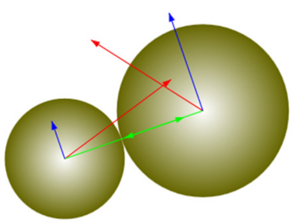
$$m_1v_1 + m_2v_2 = m_1u_1 + m_2(v_1 + u_1 - v_2)$$
$$m_1v_1 + m_2v_2 - m_2v_1 + m_2v_2 = m_1u_1 + m_2u_1$$
$$(m_1 - m_2)v_1 + 2m_2v_2 = (m_1 + m_2)u_1$$

$$\boxed{u_1 = \frac{(m_1 - m_2)}{(m_1 + m_2)}\,v_1 + \frac{2m_2}{(m_1 + m_2)}v_2}$$

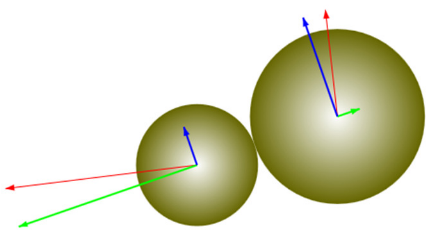
i de forma anàloga,

$$\boxed{u_2 = \frac{(m_2 - m_1)}{(m_1 + m_2)}\,v_1 + \frac{2m_1}{(m_1 + m_2)}v_2}$$

Per resoldre el problema en 2D caldrà descomposar el vector velocitat (en vermell a les imatges següents) en la component normal (en verd) i la component tangencial (en blau) en el moment del xoc:



velocitats abans del xoc



velocitats després del xoc

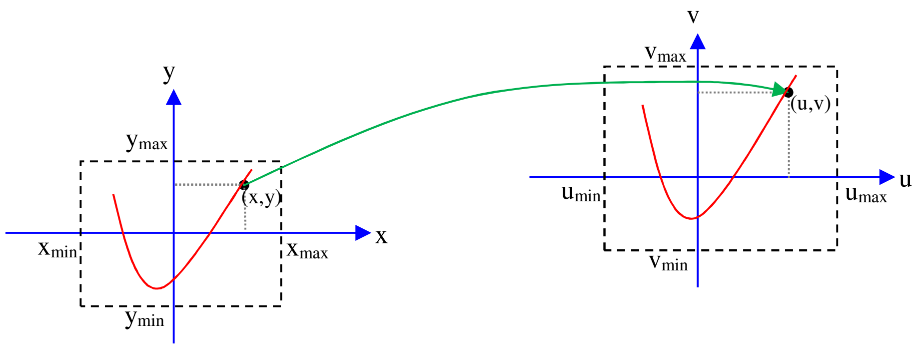
Segons les components normals, el xoc es resolt com en el cas 1D. Les components tangencials no canvien en el xoc i les velocitats després del xoc serà la composició de les noves components normals i les tangencials conservades.

```
function testCol_lisio(bola1, bola2){
   var dist = bola1.pos2D.vRestar(bola2.pos2D);
   if (dist.modul() < (bola1.radius + bola2.radius) ) {  // detecció del xoc
      // component normal de la velocitat abans del xoc
      var normalVelo1 = bola1.velo2D.vProjeccio(dist);
      var normalVelo2 = bola2.velo2D.vProjeccio(dist);
      // component tangential de la velocitat
      var tangentVelo1 = bola1.velo2D.vRestar(normalVelo1);
      var tangentVelo2 = bola2.velo2D.vRestar(normalVelo2);

      // movem les boles just en el moment del impacte asegurant que no hi hagi xoc
      var L = bola1.radius + bola2.radius-dist.modul();
      var vrel = normalVelo1.vRestar(normalVelo2).modul();
      bola1.pos2D = bola1.pos2D.vSumarEscalat(normalVelo1,-(L/vrel+epsilon)); // epsilon=0.0001
      bola2.pos2D = bola2.pos2D.vSumarEscalat(normalVelo2,-(L/vrel+epsilon));

      // càlcul de les components normals de la velocitat després del xoc
      var m1 = bola1.mass;
      var m2 = bola2.mass;
      var v1 = normalVelo1.projeccio(dist);
      var v2 = normalVelo2.projeccio(dist);
      var u1 = ((m1-m2)*v1+2*m2*v2)/(m1+m2);
      var u2 = ((m2-m1)*v2+2*m1*v1)/(m1+m2);
      normalVelo1 = dist.vParallel(u1);
      normalVelo2 = dist.vParallel(u2);

      // velocitat final després del xoc
      ball1.velo2D = normalVelo1.vSumar(tangentVelo1);
      ball2.velo2D = normalVelo2.vSumar(tangentVelo2);
}
```

http://ssh.eupmt.tecnocampus.cat/~jou/LM/Canvas/boles/index.html

5.6 <u>Mapping.</u>

Quan vulguem passar informació d'un domini x=[x_{min}, x_{max}], y=[y_{min}, y_{max}] al domini del canvas u=[0,w], v=[0,h] (per exemple, per dibuixar una funció) hem de fer la següent transformació,



traslat i escalat (no gir) de domini,

$$\begin{cases} u = ax + b \\ v = cy + d \end{cases}$$

per tant,

$u_{min} = ax_{min} + b \qquad v_{min} = cy_{min} + d$
$u_{max} = ax_{max} + b \qquad v_{max} = cy_{max} + d$

és a dir,

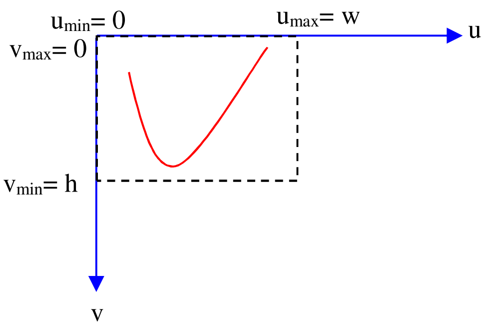
$u_{min} - u_{max} = a(x_{min} - x_{max}) \qquad v_{min} - v_{max} = c(y_{min} - y_{max})$

per tant,

$$a = \frac{(u_{min} - u_{max})}{(x_{min} - x_{max})}, \quad b = u_{min} - \frac{(u_{min} - u_{max})}{(x_{min} - x_{max})} x_{min}$$

$$c = \frac{(v_{min} - v_{max})}{(y_{min} - y_{max})}, \quad d = v_{min} - \frac{(v_{min} - v_{max})}{(y_{min} - y_{max})} y_{min}$$

Ara, concretant per al cas del canvas tenim,



$u_{min} = 0, u_{max} = w \quad i \quad v_{min} = h, v_{max} = 0$

per tant,

$$a = \frac{-w}{(x_{min} - x_{max})}, \quad b = \frac{w}{(x_{min} - x_{max})} x_{min}$$

$$c = \frac{h}{(y_{min} - y_{max})}, \quad d = \frac{-h}{(y_{min} - y_{max})} y_{max}$$

$$\begin{cases} u = (x - x_{min}) \dfrac{w}{(x_{max} - x_{min})} \\ v = (y - y_{max}) \dfrac{h}{(y_{min} - y_{max})} \end{cases}$$