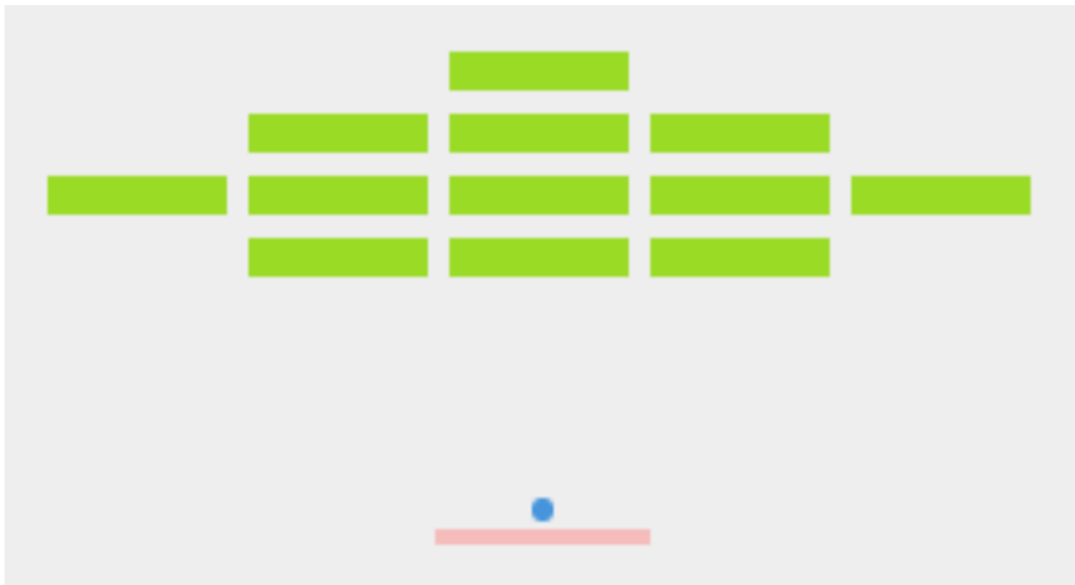


Pràctica 3 de Laboratori Multimèdia.

En aquesta tercera i darrera pràctica desenvoluparem un altre joc web basat en l'element HTML5 <canvas>. Es tracta del famós Atari Breakout. El joc consisteix en eliminar els totxos d'una paret fent xocar una bola amb aquests totxos. El jugador disposa d'una raqueta per tal de redirigir la bola cap al mur de totxos. El jugador perd quan la bola ultrapassa la posició de la raqueta. Es tracta, per tant, d'una mena de frontó.

Interfície Gràfica d'Usuari.



El jugador, en començar, triarà un nivell d'inici i posarà el seu nick per tal d'emmagatzemar el seu record personal de forma permanent a l'ordinador on s'està jugant. El jugador disposa de fins a 4 intents per superar el nivell actual. Si no ho aconsegueix, haurà de tornar a començar des del nivell 1.

La raqueta es podrà moure horitzontalment i dins de la pista a través de les fletxes esquerra i dreta del teclat.

El joc mostra els records dels tres jugadors amb més punts en la capçalera de la interfície.

Hi haurà so associat al xoc amb un totxo, xoc amb la raqueta, superació d'un nivell i final de partida amb èxit o no. El so sempre es podrà desactivar.

Rúbrica d'avaluació

Gestió del mur	Dibuix dels murs de totxo
	Tria de nivell i canvi de nivell per botons
	Xoc amb els totxos del mur
	Xoc amb la pala
	Xoc vora inferior, perd vida i reinicialitza
	Detecta el final del mur, passant al següent nivell
	Gestiona el temps de l'animació, parant quan es canvia de mur.
So	Gestió del so
Classes	Classes Mur i Display.
	Classes Game, Paddle, Ball, totxo
Disseny gràfic i usabilitat	Pantalla inicial de la partida
	Disseny gràfic de l'aplicació
Gestió del display	Control de les vides disponibles
	Control de la puntuació
	Persistència puntuació

Característiques del software.

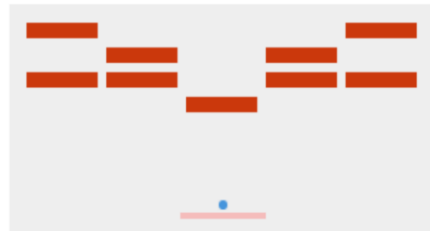
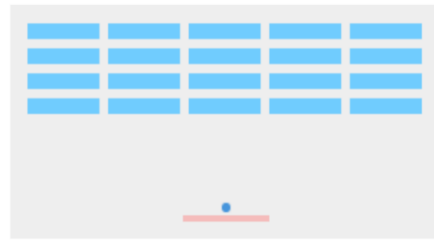
En aquesta pràctica treballarem la creació i manipulació d'objectes. Concretament, s'utilitzaran les "classes" d'objectes ja començades: App, Joc, Pala, Bola, Totxo i les "classes" que vosaltres haureu de dissenyar: Mur i Display. Se us entrega un programa inicial amb les següents funcionalitats que us han de servir de base per elaborar la vostra solució:

- App.js
 - Inicialització i actualització recursiva dels elements del canvas
- Joc.js
 - Instància dels objectes necessaris (bola, pala, totxo)
 - Dibuix al canvas dels objectes anteriors
 - Definició de la captura de tecles esquerra i dreta
- Bola.js
 - Objecte bola i el seu moviment
 - Control del xoc de la bola amb les vores del canvas
 - Control del xoc de la bola amb un totxo

Important, **No estarà permès utilitzar cap llibreria javascript excepte la jQuery.**

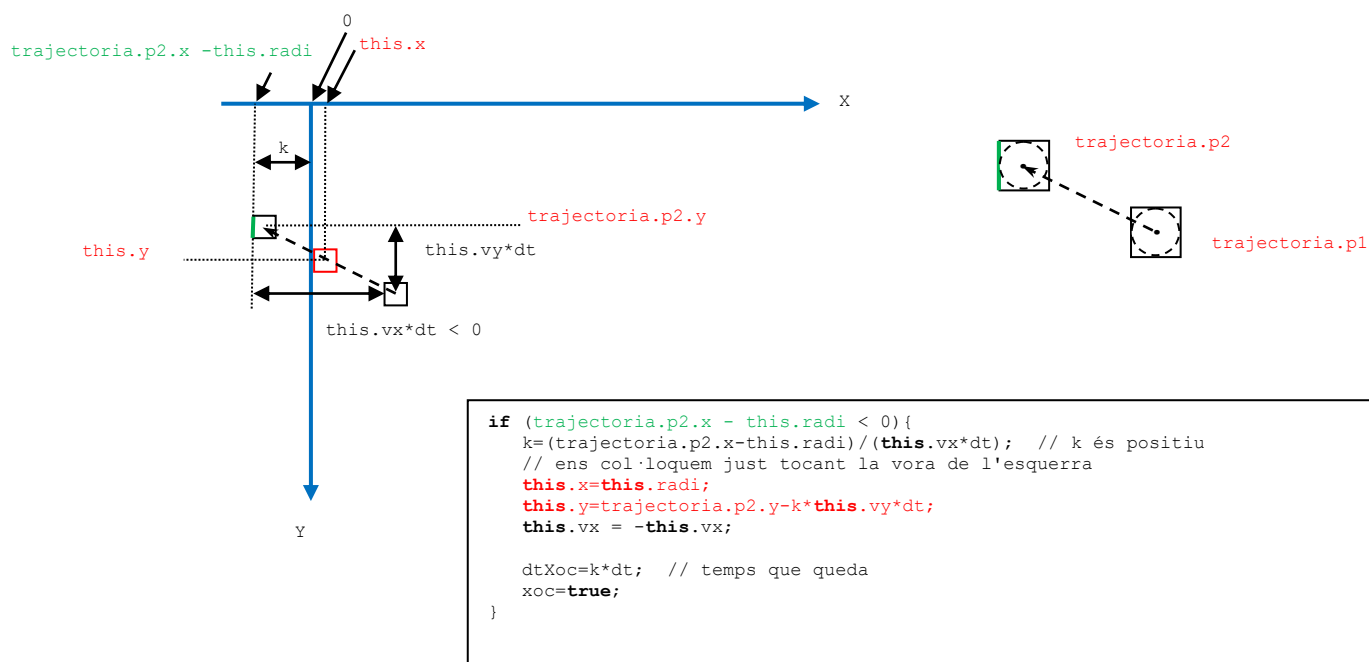
Els murs els llegirem com un objecte JS literal:

```
42
43 defineixNivells(){
44     this.nivells=[
45         {
46             color: "#4CF", // blue cel
47             totxos:[
48                 "aaaaa",
49                 "aaaaa",
50                 "aaaaa",
51                 "aaaaa",
52             ]
53         },
54         {
55             color: "#8D1", // verd
56             totxos:[
57                 " a ",
58                 " aaa ",
59                 "aaaaa",
60                 " aaa ",
61             ]
62         },
63         {
64             color: "#D30", // vermell
65             totxos:[
66                 "a a",
67                 " a a ",
68                 "aa aa",
69                 " a ",
70             ]
71         }
72     ];
73 }
74
75 };
```

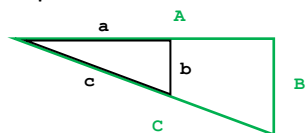


Rebot de la trajectòria de la bola en la paret esquerra de la pista.

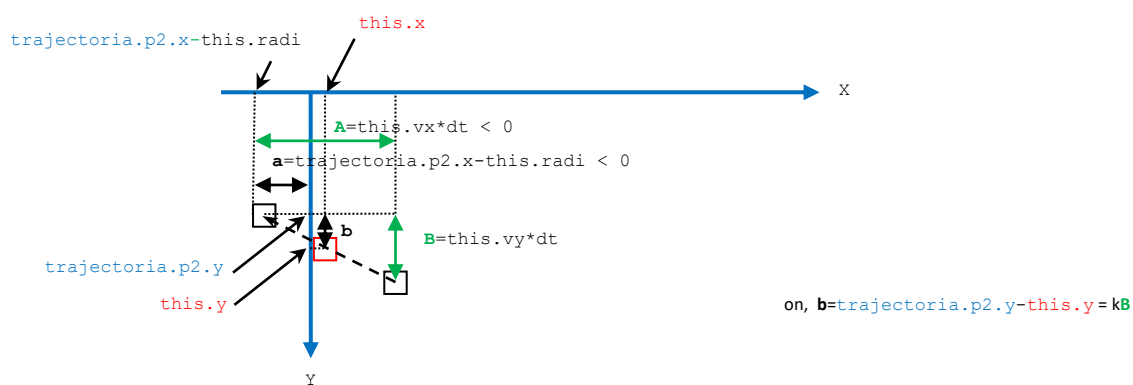
Cal estudiar quan hi ha col·lisió entre la bola (un cercle) i la paret esquerra de la pista (o el costat dret d'un rectangle). La trajectòria és un segment, determinat pels punts extrems.



Observeu que hem fet ús del Teorema de Tales:



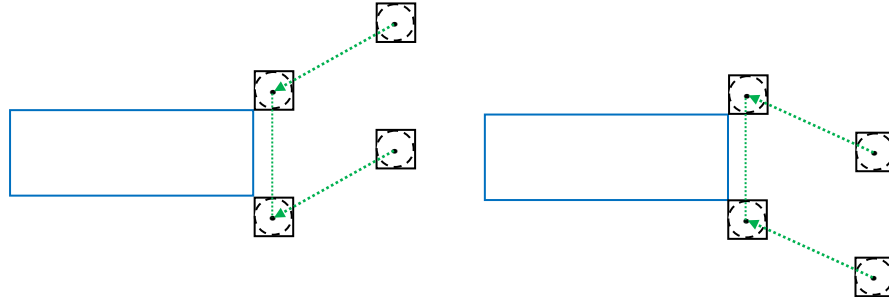
$$k = \frac{a}{A} = \frac{b}{B} = \frac{c}{C}$$



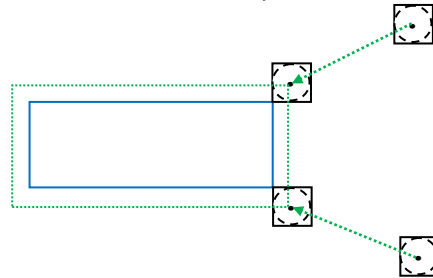
Rebot de la trajectòria de la bola en una de les quatre parets d'un rectangle.

En aquest cas, el problema serà trobar el punt d'intersecció d'una trajectòria (segment amb direcció) i un rectangle, tot indicant en quina vora ha estat la intersecció.

Anem a analitzar el cas de la vora de la dreta. Els límits d'intersecció amb la vora de la dreta els tenim representats en el següent dibuix, segons si baixem o pugem (sempre anem a l'esquerra),



Així, en els dos casos, tenim intersecció amb la vora de la dreta si hi ha intersecció entre la trajectòria i la vora de la dreta d'un rectangle engruixat amb la mida del radi de la bola. Justament, el punt d'intersecció serà la nova posició del centre de la bola perquè, just, toqui la vora de la dreta,



Ens ajudarem d'una funció que ens diu el punt d'intersecció d'un segment amb un rectangle:

```
Utilitats.interseccioSegmentRectangle = function(seg,rect){ // seg={p1:{x:,y:},p2:{x:,y:}}
// rect={p:{x:,y:},w:,h:}

var pI, dI, pImin, dImin=Infinity, vora;
// vora superior
pI=Utilitats.puntInterseccio(seg.p1, seg.p2,
{x:rect.p.x,y:rect.p.y}, {x:rect.p.x+rect.w, y:rect.p.y});

if(pI){
dI=Utilitats.distancia(seg.p1, pI);
if(dI<dImin){
dImin=dI; pImin=pI;
vora="superior";
}
}
// vora inferior
pI=Utilitats.puntInterseccio(seg.p1, seg.p2,
{x:rect.p.x+rect.w, y:rect.p.y+rect.h},{x:rect.p.x, y:rect.p.y+rect.h});

if(pI){
dI=Utilitats.distancia(seg.p1, pI);
if(dI<dImin){
dImin=dI; pImin=pI;
vora="inferior";
}
}
// vora esquerra
pI=Utilitats.puntInterseccio(seg.p1, seg.p2,
{x:rect.p.x, y:rect.p.y+rect.h},{x:rect.p.x,y:rect.p.y});

if(pI){
dI=Utilitats.distancia(seg.p1, pI);
if(dI<dImin){
dImin=dI; pImin=pI;
vora="esquerra";
}
}
// vora dreta
pI=Utilitats.puntInterseccio(seg.p1, seg.p2,
{x:rect.p.x+rect.w, y:rect.p.y}, {x:rect.p.x+rect.w, y:rect.p.y+rect.h});

if(pI){
dI=Utilitats.distancia(seg.p1, pI);
if(dI<dImin){
dImin=dI; pImin=pI;
vora="dreta";
}
}
if(vora) return {p:pImin,vora:vora}; // retorna el punt d'intersecció mínim i en quina vora ha succeït.
}
```