Lab 9

Problem to solve

Continuing with our breakout game, this week's problem is getting bricks into the game and being able to break them. You are also going to add sound effects when the ball bounces on the paddle and the bricks. If that wasn't enough, you also need to solve the problem of keeping score (1 point per brick

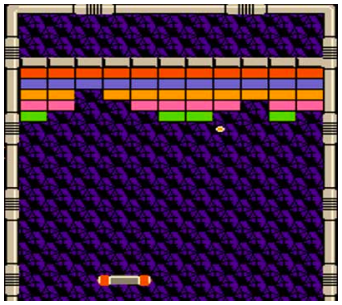

Figure 1 - Arkanoid, 1986

Last week's star level was to add movement to the paddle and make the ball bounce off it. See *help-scripts.js* in this week's download for a solution.

1. Download and unzip *Lab9-files.zip* for sound effects and help scripts.
2. Add at least one row of bricks, containing at least 8 bricks.
3. Check each brick for collision with ball (see help script).
4. If the ball collides with a brick,
    a. the brick should be removed, leaving a hole in the wall.
    b. a score variable should be incremented
    c. brick.mp3 should be played
5. If the ball collides with the paddle,
    a. paddle.mp3 should be played
6. Add text to canvas showing the current score.

Star level

1. Add a game-over-screen (one for winners and one for losers) using canvas text that will be displayed when the ball is missed by the paddle or all bricks have been broken.
2. Make it possible to steer the ball with the paddle. See figure 2 below.

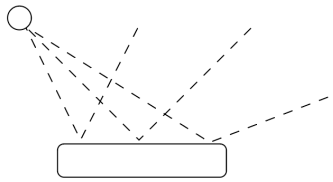

Figure 2 Steering the ball

To get the extra point, your solution must work at first upload and be handed in no later than 4 days after your lab.
Add a comment to the upload “aiming for a star”

Upload your solution

Zip your folder (yes, again, with the complete game) and name your file *Lab9-firstname_lastname.zip*
Upload it on pingpong.

Canvas 2D context properties and methods

++
Text
++
font = "[font style][font weight][font size][font face]" e.g "bold 24px Arial"
fillText(message, x, y, *maxwidth*) -*maxwidth is optional*
strokeText(message, x, y, *maxwidth*)
measureText(message).width -returns the width of the message (based on your font setting)
textAlign = [left] [center] [right]
textBaseline = [top] [hanging] [middle] [**alphabetic**] [ideographic] [bottom]
top -The text is aligned based on the top of the tallest glyph in the text.
hanging -The text is aligned based on the line the text seems to hang from.
middle -The text is aligned according to the middle of the text.
alphabetic -The bottom of vertically oriented glyphs (default)
ideographic -The bottom of horizontally oriented glyphs (asian)
bottom -The text is aligned based on the bottom of the glyph in the text

++
Setting color and style
++
`fillStyle = color` - change fill color
`strokeStyle = color` - change stroke color
`lineWidth = number` - width in pixels
`lineCap = "butt" / "round" / "square"`
`lineJoin = "miter" / "bevel" / "round" –corners`
`miterLimit = integer` - higher numbers allow sharper corners
`save()` saves the current stroke and color state
`restore()` resets any changes made to stroke and color, to last saved state

++
Drawing rectangles
++
`fillRect(startX, startY, width, height)` -a filled rectangle
`strokeRect(startX, startY, width, height)` -an outlined rectangle
`clearRect(x, y, w, h)` -erases everything within

++
Animating canvas
++
Canvas Animation is all about drawing, erasing, drawing, erasing....

Erase a rectangle on canvas:

```
ctx.clearRect(x, y, w, h)
```

If you have a reference to the canvas element you can find its width and height like this:

```
ctx.clearRect(0, 0, canvas.width, canvas.height);
```

Animating:
There are two new methods in HTML5 for animating:
-requestAnimationFrame
-cancelAnimationFrame
Usage:

```
ID = requestAnimationFrame(functionName);
cancelAnimationFrame(ID)
```

For an example see this pen: http://codepen.io/jkohlin/pen/ZeMRBx?editors=0010

++
Audio
++
Add new Audio object and set it's source file:

```
var paddleAudio = new Audio();
paddleAudio.src = "paddle.mp3";
```

Play the sound:

```
paddleAudio.play();
```