Coder Dojo 2012-03-24

Catch the symbol 01

Starting new and meeting new characters

Download the images from...

https://github.com/willknott/CoderDojo-HTML5-CanvasCatch01

Thanks to...

Symbol from @CoderDojo twitter account

Background image by Tim Buechler via Smashing Magazine

http://www.smashingmagazine.com/texture-gallery-paper/

Code attributes Lost Decade Games

http://www.lostdecadegames.com/

HTML Template... catch.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>HTML5 Catch</title>
    </head>
    <body>
        <script src="catch.js"></script>
    </body>
</html>
```

catch.js

```
//since this is a canvas game, create the
canvas
var canvas = document.createElement
("canvas");
var ctx = canvas.getContext("2d");
canvas.width = 512; //hardcoded for a reason
canvas.height = 480;
document.body.appendChild(canvas);
```

a game needs graphics...

```
// Background image
var bgReady = false;
var bgImage = new Image(); //we have an
image object
bgImage.src = "chpaper.png";
bgImage.onload = function () {
    //different way to create a function
    bgReady = true;
};
```

Now lets display that background

```
// Draw everything
// we will be adding to this later

var render = function () {
    {
        ctx.drawImage(bgImage, 0, 0);
    }
//incomplete there is more
```

loops

```
// The main game loop
var main = function () {
    render();
};

// Let's play this game!

setInterval(main, 1); // Execute as fast as possible
//We will be editing here later...
```

And now...

You'll display the background image and have rendered test on top of it.

Not much of a game.

So far...

SCOOOOOOOOOOOOOOOORE

```
var symbolCaught = 0 // temp, will delete
    // Score
    ctx.fillStyle = "rgb(250, 250, 250)"; //white
text
    ctx.font = "24px serif";
    ctx.textAlign = "left";
    ctx.textBaseline = "top";
    ctx.fillText("Symbols collected: " +
symbolCaught , 32, 32);
};
//end of render
```

Text....

We're pushing out text to the screen.
This allows us to display a score..

and do more

the render function is also where all the animation will happen.

Quick note on the score...

```
ctx.fillText("Symbols collected: " +
symbolCaught , 32, 32);
```

String contationation.

symbolCaught is just a variable, any simple varable (or object property which is a scalar) can be displayed in a string)

We need a hero...

```
//Underneath where you load the bgimage

// Hero image
var heroReady = false;
var heroImage = new Image();
heroImage.onload = function () {
    heroReady = true;
};
heroImage.src = "redl.png";
```

And our hero has properties

```
var hero = {
    speed: 200 // movement in pixels per second
};  //created and edited an aspect of the hero object
hero.x = canvas.width / 2;
hero.y = canvas.height / 2;
```

and change the render...

```
// Draw everything
var render = function () {
        if (bgReady) {
                ctx.drawImage(bgImage, 0, 0);
        }
        if (heroReady) {
                ctx.drawImage(heroImage, hero.x, hero.
y);
        }
        var symbolCaught = 0;
        // Score
        ctx.fillStyle = "rgb(250, 250, 250)";
        ctx.font = "24px serif";
        ctx.textAlign = "left";
        ctx.textBaseline = "top";
        ctx.fillText("Symbols collected: " + symbolCaught , 32, 32);
};
```

Save and run

And now Red Lemonade is sitting on the paper.

Red Lemopnade is the IP I'm creating for these games so as not to get sued.

I'll bring along the evil icecubes, White Lemonade, the evil twins Limone and Lime as well as Cheri Cola.

But we need something else...

A lost symbol, found

```
// symbol image
var symbolsReady = false;
var symbolsImage = new Image();
symbolsImage.onload = function () {
    symbolsReady = true;
};
symbolsImage.src = "logo.png";
```

Symbol properties

```
var symbols = {
    speed: 100};
var symbolCaught = 0;
var direction = -1;

// yes we are going to remove the
symbolCaught from render
```

And add something new...

```
// Reset the game when the player catches a symbols
var reset = function () {
    // Throw the symbol somewhere on the screen randomly
    symbols.x = symbolsImage.width + (Math.random() * (canvas.width - (2 * symbolsImage.width)));
    symbols.y = symbolsImage.height + (Math.random() * (canvas.height - (2 * symbolsImage.height)));
};
```

What was that...

You are setting the position for the hewo and the symbol.

x and y are coordinates for where to display the image.

but the maths!

```
symbols.x = symbolsImage.width + (Math.
random() * (canvas.width - (2 * symbolsImage.
width)));
```

math.random is between 0 and 1

canvas.width - (2 * symbolsImage.width)));
is the available coordinates to display the symbol

Rendering

```
var render = function () {
    if (bgReady) {
        ctx.drawImage(bgImage, 0, 0);
    }
    if (heroReady) {
        ctx.drawImage(heroImage, hero.x, hero.y);
    }
    if (symbolsReady) {
        ctx.drawImage(symbolsImage, symbols.x, symbols.y);
    }
    // Score
    ctx.fillStyle = "rgb(250, 250, 250)";
    ctx.font = "24px serif";
    ctx.textAlign = "left";
    ctx.textBaseline = "top";

    ctx.fillText("Symbols collected: " + symbolCaught , 32, 32);
};
```

but we need to call reset...

```
// The main game loop
var main = function () {
    render();
};

// Let's play this game!
reset();
setInterval(main, 1); // Execute as fast as
possible
```

Save and load

All the characters are on the screen...

But not much is going on.

Lets get things moving....

moving the symbol

```
//after the reset...

// Update game objects called from the main function
// delta = number of miliseconds between calls
var update = function (delta) {

    symbols.x += (symbols.speed * direction * delta);
```

its moving but...

```
    // Setting boundaries
    if(symbols.x < 0) {
        direction = -direction;
    }

    if(symbols.x > canvas.width    -
symbolsImage.width ) {
        direction = -direction;
    }
};
```

but delta?

Yes we need to change the main and calling functions...

new main()

```
// The main game loop
var main = function () {
    var now = Date.now();
    var delta = now - then;

    update(delta / 1000);
    render();

    then = now;
};
```

new "bottom"

```
/ Let's play this game!
reset();
var then = Date.now();
setInterval(main, 1); // Execute as fast as
possible
```

What was that...

The key is delta...

```
var delta = now - then;
```

The time between calls...

```
update(delta / 1000);
```

Call update with a time based number...

and with that number delta...

symbols.x += (symbols.speed * direction * delta);

speed = pixels per, er, something

delta = time

speed * time = distance between calls

The direction is left or right....

save and load

And we have animation!

And out first logic bug...
How would you fix it?

Logic bug?

2 types of bug...
Syntax ; nothing works.
e.g. no fule in the car

logic; something works but something is wrong.
car runs but the fire is flat

OK, we have something moving, but its not a game

So we need to get or hero moving too....

Add before reset()

```
// Handle keyboard controls
var keysDown = {};

addEventListener("keydown", function (e) {
    keysDown[e.keyCode] = true;
}, false);

addEventListener("keyup", function (e) {
    delete keysDown[e.keyCode];
}, false);
```

Eventlistener

Without this the javascript ignores, well, all input.

This way it listens to the keyboard,

but we only want it listening to certain keys

new update()

```
var update = function (delta) {
    if (37 in keysDown) { // Player holding left
        hero.x -= hero.speed * delta;
    }
    if (38 in keysDown) { // Player holding up
        hero.y -= hero.speed * delta;
    }
```

continue update()

```
    if (39 in keysDown) { // Player holding right
        hero.x += hero.speed * delta;
    }
    if (40 in keysDown) { // Player holding down
        hero.y += hero.speed * delta;
    }

    symbols.x += (symbols.speed * direction *
delta);
```

save and load

And Red now moves...

All over the place...

Still no game (he ignores the symbol) but...
Let's keep him in his box.

Bounding Red (further down in update)

```
// Setting boundaries
if(hero.x < 0) {
    hero.x = 0;
}
if(hero.y < 0) {
    hero.y = 0;
}
if(symbols.x < 0) {
    direction = -direction;
}
```

And below that again...

```
    if(hero.x > canvas.width-heroImage.width ) {
        hero.x = canvas.width-heroImage.width;
    }
    if(hero.y > canvas.height-heroImage.height)
{
        hero.y = canvas.height-heroImage.height;
    }
    if(symbols.x > canvas.width -symbolsImage.
width ) {
        direction = -direction;
    }
```

save and load

Red now stays in the box...

But why isn't he touching the edges?

Make your own Red...

I used gimp (GNU Image Manulipaton Program)

http://www.gimp.org/

To create Red, to fix the bug, either change the maths, or edit the Red image

GET A PARENT TO GO TO THE SITE

More than the box... when they touch

We don't have a game yet. How to we tell in they are touching.

We don't have sprite collision (yet) so what do we know...

more Maths... in Update

```
symbols.x += (symbols.speed * direction * delta);
// Are they touching?
    if (
        hero.x <= (symbols.x + symbolsImage.
width)
        && symbols.x <= (hero.x + heroImage.
width)
        && hero.y <= (symbols.y +
symbolsImage.height)
        && symbols.y <= (hero.y + heroImage.
height)
    ) {
        ++symbolCaught;
```

continued...

```
symbols.x += (symbols.speed * direction * delta);

// Are they touching?
if (
    hero.x <= (symbols.x + symbolsImage.width)
    && symbols.x <= (hero.x + heroImage.width)
    && hero.y <= (symbols.y + symbolsImage.height)
    && symbols.y <= (hero.y + heroImage.height)
) {
    ++symbolCaught;
    reset();
}
```

What was that...

Well, reset puts the symbol in a new random location (and the animation loop kicks in)

++symbolCaught;

Add one to the score (and the render automatically displays it)

And the maths...

```
hero.x <= (symbols.x + symbolsImage.width)
&& symbols.x <= (hero.x + heroImage.width)
&& hero.y <= (symbols.y + symbolsImage.height)
&& symbols.y <= (hero.y + heroImage.height)
```

&& means AND

If the calculated coordinates of the sprites overlap... it counts.

Remember the "bug" in Red not touching the screen....

Technically its wrong, but its close

Your turn...

How would you add to it?

What would you change?