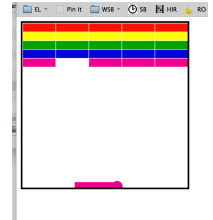


Break Out

Canvas Tutorial continue Simple Game: Break Out

⦿ Animate, Game Elements

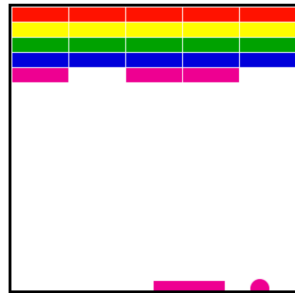
- ✦ Breakout
- ✦ Bill Mill
 - Modified Tutorial
 - Software developer in Maryland



What it Looks Like

⦿ Elements:

- ✦ Color
- ✦ Collision Detection
- ✦ Interaction with User
 - Mouse
 - Keyboard



11.bricks-really-pretty.html

Review: Rough Structure

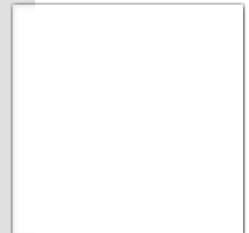
```

<!DOCTYPE html>
<html>

<head>
  <style type="text/css">
    canvas { border: 3px solid black; }
  </style>
  <script type="text/javascript">
    .
    .
    .
  </script>
</head>

<body>
  <canvas id="canvas" width="300" height="300">
    Your browser does not support the canvas element.
  </canvas>
</body>
</html>

```



Structure (Finished Code)

```

// Global variables
// Initialization methods
// Mouse & Keyboard specifications
// Shapes
// draw routines more course grained
// Game logic

```

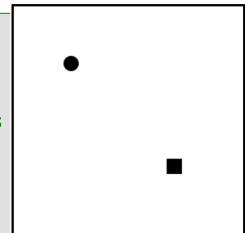
Review: Step 1 : JavaScript: Ball in Break out (a circle)

```

<script type="text/javascript">
var x=200;
var y=200;
window.onload=function()
{
  var c = document.getElementById( "canvas" );
  var ctx = c.getContext( "2d" );

  //draw a circle
  ctx.beginPath();
  ctx.arc(75, 75, 10, 0, Math.PI*2, true);
  ctx.closePath();
  ctx.fill();
}
</script>

```



```

// draw a rectangle
ctx.beginPath();
ctx.rect( x, y, 20, 20);
ctx.fill();
ctx.closePath();
}

```

1.intro.html

Adding Action

Create a function clear and then draws objects in a repeatable manner (e.g., a draw() function).

⦿ Primitive:

- ✦ setInterval(function, timeout) in the init()
 - Using : SetInterval() for now
 - CAVEAT: HW read below links

<http://stackoverflow.com/questions/13935262/settimeout-or-setinterval-or-requestanimationframe>
<http://creativejs.com/resources/requestanimationframe/>

Example Animation

```
var x = 150;
var y = 150;
var dx = 2;
var dy = 4;
var ctx;

function init() {
  ctx = $('#canvas')
  [0].getContext("2d");
  return setInterval(draw, 10);
}

function draw() {
  ctx.clearRect(0,0,300,300);
  ctx.beginPath();
  ctx.arc(x, y, 10, 0, Math.PI*2, true);
  ctx.closePath();
  ctx.fill();
  x += dx;
  y += dy;
}
init();
```

3.action.html

Exercise: How would you make it continue to animate?

How about changing dx, dy?

Modularize

6.bounce-mills.html

```
//BEGIN LIBRARY CODE
var x = 150;
var y = 150;
var dx = 2;
var dy = 4;
var WIDTH;
var HEIGHT;
var ctx;

function init() {
  ctx = $('#canvas')
  [0].getContext("2d");
  WIDTH = $('#canvas').width();
  HEIGHT = $('#canvas').height();
  return setInterval(draw, 10);
}

function circle(x,y,r) {
  ctx.beginPath();
  ctx.arc(x, y, r, 0, Math.PI*2,
true);
  ctx.closePath();
  ctx.fill();
}

function rect(x,y,w,h) {
  ctx.beginPath();
  ctx.rect(x,y,w,h);
  ctx.closePath();
  ctx.fill();
}

function clear() {
  ctx.clearRect(0, 0, WIDTH, HEIGHT);
}
//END LIBRARY CODE
```

Circle Code
Rectangle Code
Clear

```
function draw() {
  clear();
  circle(x, y, 10);
  x += dx;
  y += dy;
}

init();
```

3.library.html

Bounce

- ⦿ Detect when the ball is 'beyond' its boundaries.

```
function draw() {
  clear();
  circle(x,y,10);

  // if outside the width canvas, change direction of ball.
  if (x + dx > WIDTH || x + dx < 0)
    dx = -dx;
  if (y + dy > HEIGHT || y + dy < 0)
    dy = -dy; // -.5;

  x += dx;
  y += dy;
}
```

- ⦿ Linear Motion

- ✦ More realistic, accelerates while descending, (b/c of gravity, and slows down while bouncing up).

Gravity

- ⦿ Read (coming back to)

- ✦ <http://codetheory.in/basics-of-implementing-gravity-with-html5-canvas/>

... add Paddle

- ⦿ Add Non-Moving Paddle (rectangle)

- ✦ Allow Ball only to bounces off paddle, otherwise ball is out of bound.
 - (only 'beyond floor')

... KeyBoard Control

- ⦿ Allow Paddle to move
 - ✦ Left Arrow Input
 - ✦ Right Arrow Input
 - ✦ http://www.w3schools.com/jsref/event_key_charcode.asp
 - ✦ <http://www.asquare.net/javascript/tests/KeyCode.html>
- ✦ 8.keyboard-pxaddle.html

... Mouse Control

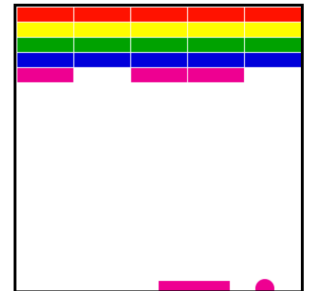
- ✦ mousemove event to an onMouseMove function, see if the mouse is within the borders of the game, and move the paddle if it is.

... Brick and Collisions

- ✦ See code, simple collision detection (looks for overlaps)

What it Looks Like

- ⦿ Elements:
 - ✦ Color
 - ✦ Collision Detection
 - ✦ Interaction with User
 - Mouse
 - Keyboard



11.bricks-really-pretty.html

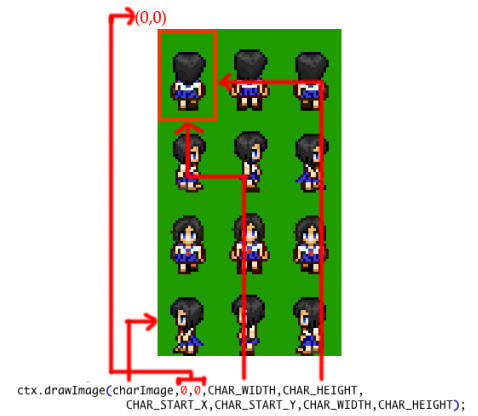
Animating the Sprite

- ⦿ Sprite movements.
 - ✦ Load image
 - ✦ Load a sprite sheet
 - ✦ Animate it
- ⦿ Parallax

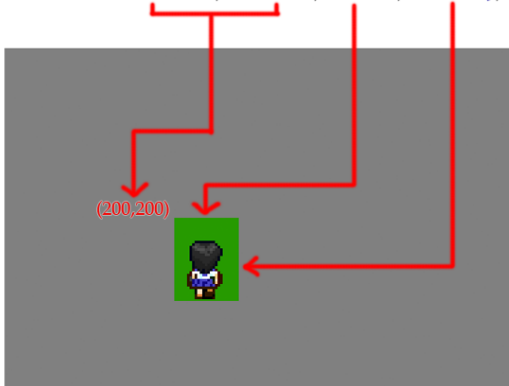
- ⦿ Load Image
- ⦿ Draw Image onto Canvas
- ⦿ Animate Sprite

Sprite Magic

```
ctx.drawImage(charImage, // sprite sheet
currX, currY, // top left corner of sprite sheet
CHAR_WIDTH, CHAR_HEIGHT, // size one instant
CHAR_START_X, CHAR_START_Y, // game canvas location
CHAR_WIDTH, CHAR_HEIGHT); // size on canvas (enables sizing
// up or e down
```

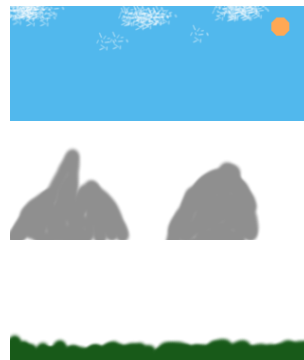


```
ctx.drawImage(charImage, 0, 0, CHAR_WIDTH, CHAR_HEIGHT,
CHAR_START_X, CHAR_START_Y, CHAR_WIDTH, CHAR_HEIGHT);
```



Parallax

- © http://en.wikipedia.org/wiki/Parallax_scrolling
- © <http://javacoffee.de/?p=866>



```
function Layer(
s, // path to image
x, y)
{
this.img = new Image();
this.img.src = s;
this.x = x;
this.y = y;
}
```

Next week

- © More about collision
- © Modularizing data with Javascript