Javascript Turtle Graphics

The Concept

Learn Javascript programming in a graphics environment. Javascript provides action and functionality to web pages. The Javascript Turtle Graphics page at http://bonner-carlson.net/turtle is written in Javascript and it provides an environment for exploring Javascript and its use of graphics using traditional turtle graphics functions.

The Code

```
// first program
write ("Hello World")

// first readable program
right (90)
write ("Hello World")

// simple square function
function square4 () {
  forward (100)
  right(90)
  forward (100)
  right (90)
  forward (100)
  right (90)
  forward (100)
  right (90)
}


// square with repeat
function el () {
  forward (100)
  right (90)
}

function square () {
  repeat (4, el)
}


//simplest form of while:
var i = 0;   // initiator
while (i<4) {  // (condition) {block of instructions}
  i = i + 1; // incrementer
}
```

```
//while form of square
var i = 0;
while (i<4) {
  //write (i + " --> ")
  forward (100)
  right(90)
  i = i + 1;
}


//functional form of square
function square (side) {
  var i=0
  while (i<4) {
    //write (i + " --> ")
    forward( side)
    right (90)
    i=i+1
  }
}


function stackedBoxes (number) {
  var i = 0
  size = 40
  while (i <= number) {
    square( i/number * size )
    penup()
    forward( i/number * size)
    pendown()
    i = i + 1
  }
}


function squareNumbered (side) {
  var i=0
  while (i<4) {
    if (i%2) {
      color("red")
    } else {
      color ("blue")
    }
    write(i)  // want to show 100+i  and i +"--->" and i + "00"
    forward( side)
    right (90)
    i=i+1
  }
}
```

```
function turningSquare () {
  var steps = 100
  var stepSize = 200/steps
  var i = 0;
  for (var i=0; i<steps; i=i+1) {
    square2(stepSize*i);
    right (360/steps)
    i = i+1;
  }
}


star....
zorro gets back to the same point and same direction ... = 360 degrees
360 / 5 = 72... that is a pentagon
720 /5 = 240


function spikey (size,n,revs) {
  var i = 0
  while (i<n) {
    forward (size)
    write (i)
    right (revs*360/n)
    i = i + 1
  }
}

function pentagon (size) {
  spikey (size, 5, 1)
}

function star (size) {
  backward (size/2)
  spikey (size, 5, 2)
}

function polygon (size, sides) {
  spikey (size, sides, 1)
}

function starN (size, points) {
  backward (size/2)
  spikey (size, points, Math.floor(points/2) )
}

//spikey(200,39,19)
//spikey(200,39,19)
//spikey(200,45,19)
//spikey(200,49,27)
//n must be odd
//revs is best about rev/2
```

What is the Next Step???

- Learn about the for() loop instruction
- Set up a demo of finding pi with a random number generator. Hint: use a square that is 1 unit by 1 unit and a quarter of a circle with a radius of 1 unit. Remember the Pythagorean theorem.
- Play with random colors or color around a color wheel, hint: color (random(15)) or color("hsl("+i/n*360+", 100%, 50%")
- Investigate fractiles and draw them
- Investigate tessellations and draw them
- Do an animated graphics demonstration
- Make the page web accessible
    - add to a server, perhaps on a Raspberry Pi with Apache.
- Learn more about Javascript, HTML, and CSS using resources:
    - Read a book from it-ebooks.info,
    - Take a course from Khan Academy
    - Get hands on experience with Code.org
    - Find a particular feature at W3School
- Learn about code development tools
    - Browser based debugging tools
    - “lint” programs to check CSS and HTML syntax
    - “minify” programs to make your final code smaller

Possible Careers in Information Technology

- help desk / computer support
- system administrator
- system analyst
- coder
- front-end web developer (HTML, CSS, Javascript and many more)
- back-end web developer (PHP and many more)
- web designer (heavy CSS with HTML and Javascript)
- product developer/engineer
- software engineer
- system engineer
- network engineer
- protocol engineer
- engineering management
- chief information office