Graphics Programming Lecture 2a

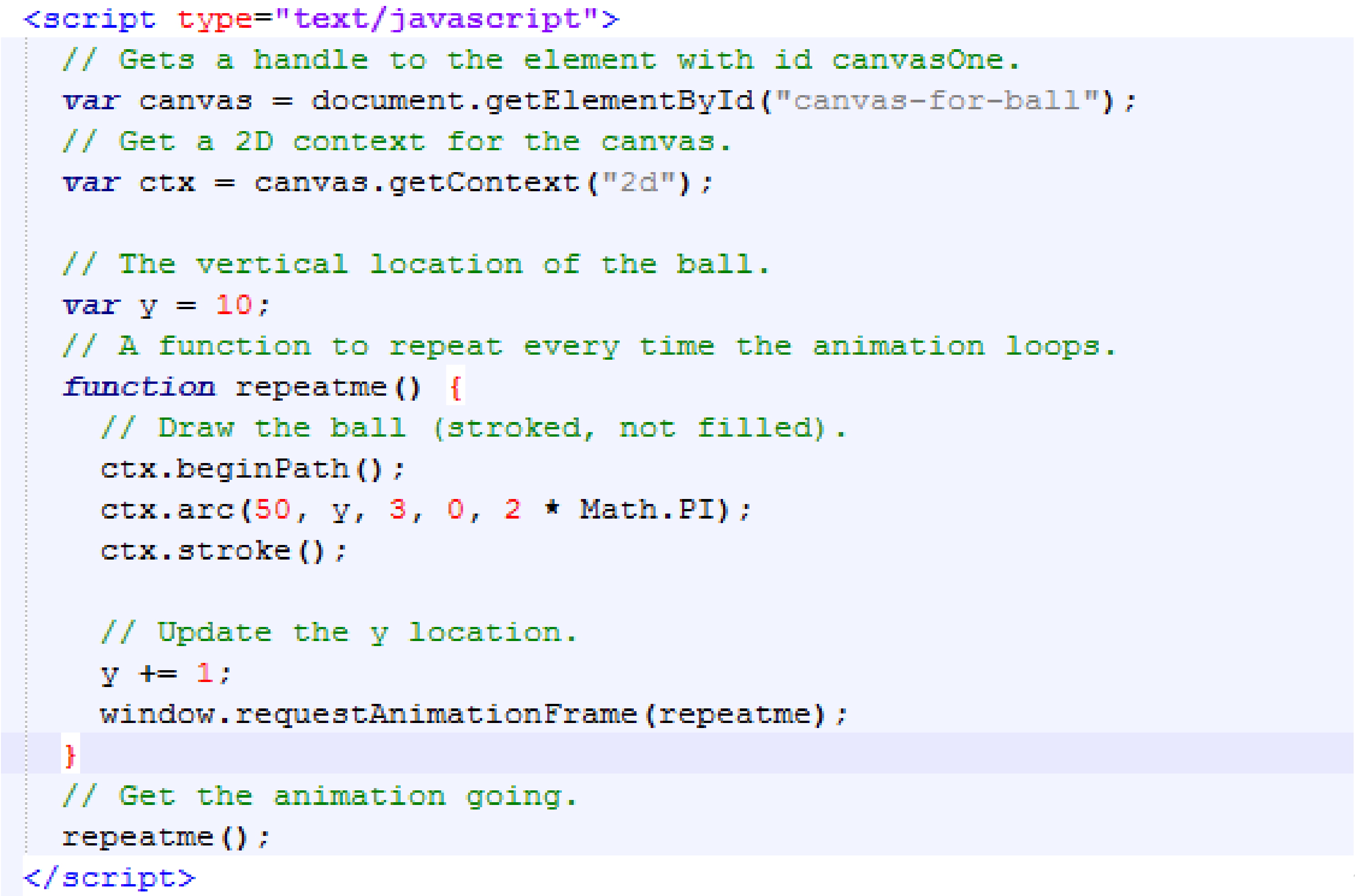
HTML5 Canvas Animation

- Until now, the biggest limitation is, that once a shape gets drawn, it stays that way.
- If we need to move it we have to redraw it and everything that was drawn before it.
- It takes a lot of time to redraw complex frames and the performance depends highly on the speed of the computer
- Since we're using JavaScript to control <canvas> elements, it's also very easy to make animations.

HTML5 Canvas Basic Animation Steps

- Clear the canvas
  - Unless the shapes you'll be drawing fill the complete canvas (for instance a backdrop image), you need to clear any shapes that have been drawn previously. The easiest way to do this is using the clearRect() method.
- Draw animated shapes
  - The step where you do the actual frame rendering.
- Controlling an animation:
  - requestAnimationFrame(callback)
  - Tells the browser that you wish to perform an animation and requests that the browser call a specified function to update an animation before the next repaint.

requestAnimationFrame(callback) – ball.html



```
<script type="text/javascript">
  // Gets a handle to the element with id canvasOne.
  var canvas = document.getElementById("canvas-for-ball");
  // Get a 2D context for the canvas.
  var ctx = canvas.getContext("2d");

  // The vertical location of the ball.
  var y = 10;
  // A function to repeat every time the animation loops.
  function repeatme() {
    // Draw the ball (stroked, not filled).
    ctx.beginPath();
    ctx.arc(50, y, 3, 0, 2 * Math.PI);
    ctx.stroke();

    // Update the y location.
    y += 1;
    window.requestAnimationFrame(repeatme);
  }
  // Get the animation going.
  repeatme();
</script>
```

Lab Instructions

- Put a border around the canvas so that we can see its edges.
- Clear the canvas at each step of the animation, so that only one copy of the ball is visible at a time. You can use the clearRect method for this.
  - ctx.clearRect(left, top, width, height);
- Given ball.html, Stop the ball moving when it hits the bottom of the canvas.
  - You might use an if statement for this purpose.
- Represent the ball as a class/object that includes the ball’s data (position, velocity, size) and methods to move and draw the ball.
- Change the code so that the ball starts moving up the way once it reaches the bottom of the screen.
- Change the code so that the ball bounces from the bottom to the top of the screen, and back again, repeatedly.

Lab Instructions

- Give the ball a horizontal velocity, as well as a vertical one, and have it bounce off the left and right sides of the canvas also.
- Advanced:
  - Look at lab
- Material:
  - Mozilla's docs on Window.requestAnimationFrame() - https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame