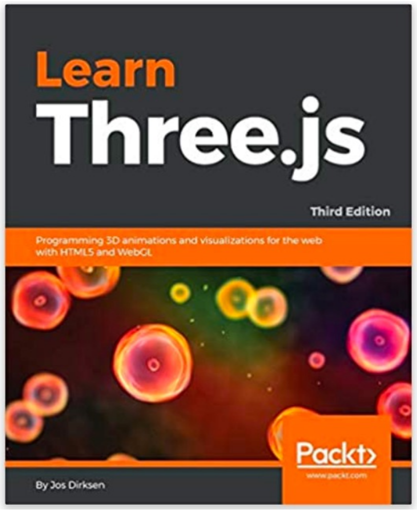
COMPUTER GRAPHICS


Learn
Three.js
Third Edition
Programming 3D animations and visualizations for the web with HTML5 and WebGL
Packt>
www.packt.com
By Jos Dirksen

* Adapted from CISC 3326 lecture by Michael Mandel

INTRODUCTION TO CANVAS DRAWING

Based on CS 307 lecture 2a

Topics for today

- Covered last time
  - The HTML5 <canvas> element
  - Drawing rectangles
  - Drawing paths
- Drawing arcs in paths
- Setting properties of path segments
- Transformations and saving and restoring state

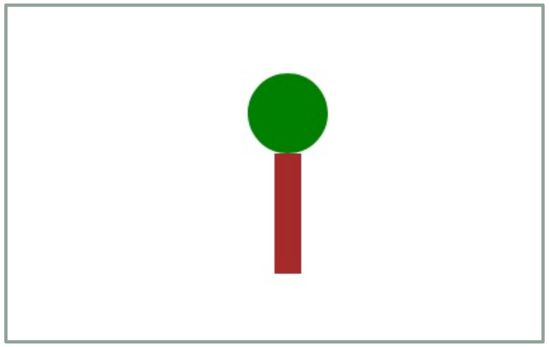
Topics for today

- Exercises
    - draw a tree
    - draw a house
    - draw a village

EXERCISES

Exercise 1: Drawing a tree

- Start from this codepen
  - Note that there is a function flipY() that changes the coordinate so the positive y direction is up
- Write the function drawTreeAt(ctx,x,y,height,width,radius)
  - drawTreeAt(ctx,200,50,90,20,30); should draw



flipY

- What does this do?
  - function flipY(ctx) {
    - var canvas = ctx.canvas;
    - var w = canvas.clientWidth;
    - var h = canvas.clientHeight;
    - ctx.translate(0,h);
    - ctx.scale(1,-1);
  - }

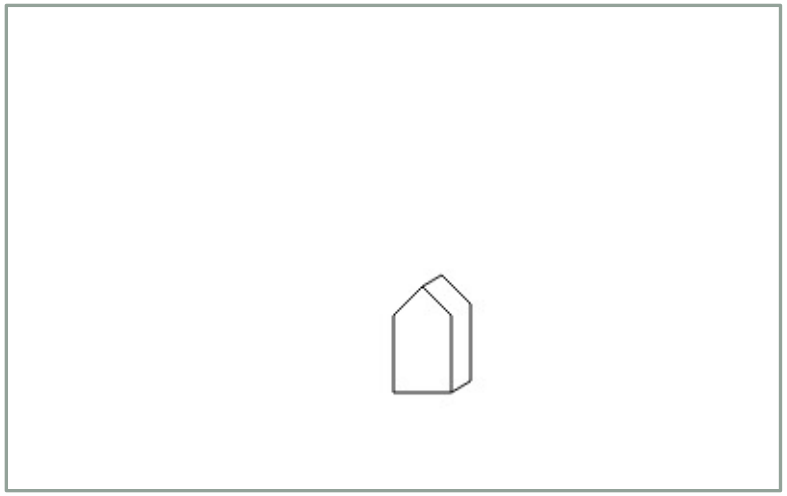
flipY

- What does this do?
  - What are w and h?
  - Translate moves original to (0 , h)
  - Scale paints from top to bottom.

```
function flipY(ctx) {
    var canvas = ctx.canvas;
    var w = canvas.clientWidth;
    var h = canvas.clientHeight;
    ctx.translate(0,h);
    ctx.scale(1,-1);
}
```

Exercise 2: A Projected House

- Click on the tiny fork button on the bottom right to make a copy of your codepen
- Add a function drawHouseAt(ctx, x, y, width, height, dx, dy)
  - drawHouseAt(ctx,200,50,30,40,10,6);
  - should draw:

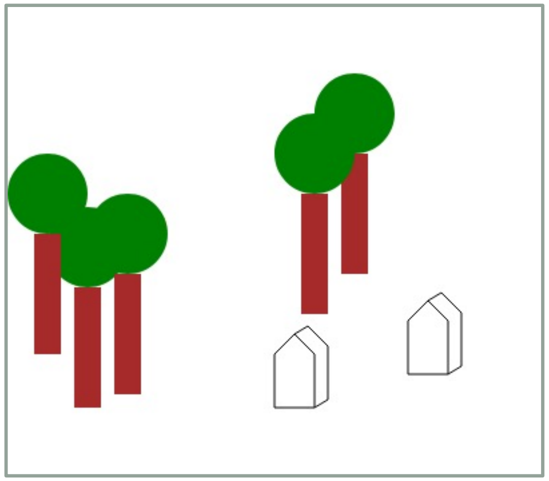


Exercise 2: A Projected House

- See if you can use a translate(x,y) transformation
    - Write a function drawHouse(ctx, width, height, dx, dy) that draws a house at the origin

Exercise 3: Houses in a Forest

- Click on the tiny fork button on the bottom right to make a copy of your codepen
- Use the drawTreeAt() and drawHouseAt() functions
  - draw many houses and trees scattered about
    - like this:



Summary

- Drawing in a 2D coordinate system isn’t so bad.
- Achieve effects using methods on a *context* object.
- Code using functions to achieve higher-level effects
- Parameterize the functions
  - e.g. a function to draw a tree with a given width and height.

Summary

- Make the functions be *generic* , e.g. a house with its origin at the lower left.
- Use transformations to translate the generic objects.

Questions?