Transformations

In this exercise we will look at transformations using matrices.

Introduction

1. A matrix looks like the following. This is called the identity matrix. It has 0's everywhere except for the main diagonal and 1's in the main diagonal. Multiplying another matrix by the identity matrix (of the corresponding size) leaves the matrix as it is.

   $$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Transformations in 2D can be represented as 3x3 matrices. In the matrix below the letters can be replaced by numbers to achieve desired effects such as translating, scaling and rotating, among others.

   $$\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix}$$

3. Translation is the name we give to moving an object from one location to another. An object might be a rectangle, circle, triangle or something more complex. We replace e in the above matrix with the number we want to translate in the direction of the x axis, and f by that of the y axis. If translation is all we want to do, we leave the four other letters as they are in the identity matrix. Below is an example of a matrix that will translate objects by (4,6).

   $$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix}$$

4. To use this matrix on a point (x,y), we map (x,y) to a 3x1 matrix, left multiply that by the translation matrix, and then map the product back to a point. Below we translate the point (1,1) by the matrix.

$$(1,1) \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 5 \\ 7 \\ 1 \end{bmatrix} \rightarrow (5,7)$$

5. Scaling is achieved by replacing the first two numbers in the matrix diagonal, a and d. Replacing a with a number will scale the object along the x axis. Likewise, replacing d will scale along the y axis. For instance to scale by (2,3), we use the following matrix.

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

6. Rotation, counter-clockwise about the origin, is achieved using the following matrix.

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Exercises

1. Create a blank HTML file with a CSS section and a JavaScript section.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Transformations</title>
    <meta charset="UTF-8">
    <style type="text/css"></style>
  </head>
  <body>
    <script type="text/javascript">
    </script>
  </body>
</html>
```

2. Add a canvas element with an id, and get a context for it.

```
<canvas id="transformcan"></canvas>

var canvas = document.getElementById(transformcan);
var ctx = canvas.getContext('2d');
```

3. Create a class for drawing a Pacman (without the eye, for now) on the canvas. Create a “draw()” method within the class. Draw Pacman at the origin. Create an instance of the class and test the “draw()” method.

4. Create a function for clearing the canvas. Note the use of save() and restore() on the context. This allows the resetting of the canvas so that it can be cleared.

```
function clear() {
  ctx.save();
  ctx.setTransform(1, 0, 0, 1, 0, 0);
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  ctx.restore();
}
```

5. Create a repeatme animation function that translates the context by (1,0), and rotates it by π/2 radians every 400 steps.

6. Set the global variable nosteps to 0, and set an initial translation on the canvas to show pacman fully at the start.

```
var nosteps = 0;
ctx.translate(30,30);
```

7. Start your animation by calling the repeatme function.

8. Replace the ctx.translate() in step() with a call to ctx.transform() – using matrix notation described in lectures

9. Replace the ctx.rotate() in step() with a call to ctx.transform().

Advanced exercises.

1. Add an eye and outline (stroke) to Pacman, and have it translate and rotate correctly with Pacman.

2. Draw a rectangular track/grid that Pacman moves through. Hint: Pay attention to how the Transformation matrix is saved, reset and then restored in step 4.

3. Have the text "Rotation" displayed in the centre of the canvas for half a second everytime Pacman rotates, and scale it up in size at each step of the animation. Use a transformation to achieve this scaling