Applied Matrix Algebra
Computer Graphics

Objective The objective of this project is to illustrate how Matrices can be used in computer graphics using *Mathematica*.

Narrative In this project we illustrate the geometry of some simple transformations of the plane by applying them to a simple test pattern. We will then use some *Mathematica* commands to animate several transformations.

First load the test pattern and graph it by typing the following lines into *Mathematica*.

```
House = Transpose[{{0, 6, 1}, {3, 8, 1}, {6, 6, 1}, {0, 6, 1},
        {0, 0, 1}, {5, 0, 1}, {5, 3, 1}, {4, 3, 1}, {4, 0, 1},
        {6, 0, 1}, {6, 6, 1}}];
MatrixForm[House]
Strip2d = {{1, 0, 0}, {0, 1, 0}};
MatrixForm[Strip2d]
ListPlot[Transpose[Strip2d.House], Joined -> True,
        PlotStyle -> Black, PlotRange -> {{-1, 10}, {-1, 10}}]
```

Recall in class that we talked about transformations on the plane by using 3 by 3 matrices. We are embedding points from $\mathbb{R}^2$ into $\mathbb{R}^3$. The matrix **Strip2d** is used to project back into the plane.

Now define the basic transformations that we discussed in class. While typing these into *Mathematica* note the form of the functions and try to interpret the variables used. Also recall that the symbol \[Theta] is really θ.

```
Rot[M_, \[Theta]_] := {{Cos[\[Theta]], -Sin[\[Theta]], 0},
        {Sin[\[Theta]], Cos[\[Theta]], 0}, {0, 0, 1}}.M
Reflect[M_, a_] := {{(-1)^a, 0, 0}, {0, (-1)^(a + 1), 0}, {0, 0, 1}}.M
Dilate[M_, r_, s_] := {{r, 0, 0}, {0, s, 0}, {0, 0, 1}}.M
ShearH[M_, k_] := {{1, k, 0}, {0, 1, 0}, {0, 0, 1}}.M
ShearV[M_, k_] := {{1, 0, 0}, {k, 1, 0}, {0, 0, 1}}.M
Trans[M_, h_, k_] := {{1, 0, h}, {0, 1, k}, {0, 0, 1}}.M
```

Lets test the first of these transformations by graphing our test pattern after a rotation by 30°.

```
ListPlot[Transpose[Strip2d.Rot[House, Pi/6]], Joined -> True,
     PlotStyle -> Black, PlotRange -> {{-10, 10}, {-10, 10}},
     AspectRatio -> 1]
```

Notice that we use radians instead of degrees for our rotation.

Now let's have some fun. The **Manipulate** command in *Mathematica* is used to animate graphics. The following series of commands allows us to see the results of rotating the pattern through a wide range of angles.

```
Manipulate[
 Show[
  ListPlot[Transpose[Strip2d.House], Joined -> True,
   PlotStyle -> {Black, Thickness[0.005]},
   PlotRange -> {{-10, 10}, {-10, 10}}, AspectRatio -> 1],
  ListPlot[Transpose[Strip2d.Rot[House, \[Theta]]], Joined -> True,
   PlotStyle -> Blue, PlotRange -> {{-10, 10}, {-10, 10}},
   AspectRatio -> 1]
  ],
 {\[Theta], 0, 2*Pi, Pi/12}
 ]
```

Ok. Notice that we use the **Show** command to display both the original pattern and the rotated pattern. The parameter of the **Manipulate** command is θ. We have the command set up so that θ can range over the values of $0 \leq \theta \leq 2\pi$ in increments of $\pi/12$.

Here is an example of animating reflections:

```
Manipulate[
 Show[
  ListPlot[Transpose[Strip2d.House], Joined -> True,
   PlotStyle -> {Black, Thickness[0.005]},
   PlotRange -> {{-10, 10}, {-10, 10}}, AspectRatio -> 1],
  ListPlot[Transpose[Strip2d.Reflect[House, a]], Joined -> True,
   PlotStyle -> Blue, PlotRange -> {{-10, 10}, {-10, 10}},
   AspectRatio -> 1]
  ],
 {a, 1, 2, 1}
 ]
```

Here is an example of animating a horizontal shear:

```
Manipulate[
 Show[
  ListPlot[Transpose[Strip2d.House], Joined -> True,
   PlotStyle -> {Black, Thickness[0.005]},
   PlotRange -> {{-20, 20}, {-20, 20}}, AspectRatio -> 1],
  ListPlot[Transpose[Strip2d.ShearH[House, k]], Joined -> True,
   PlotStyle -> Blue, PlotRange -> {{-10, 10}, {-20, 20}},
   AspectRatio -> 1]
  ],
 {k, -2, 2, 0.1}
 ]
```

Finally, here is an example with two "controls".

```
Manipulate[
 Show[
  ListPlot[Transpose[Strip2d.House], Joined -> True,
   PlotStyle -> {Black, Thickness[0.005]},
   PlotRange -> {{-20, 20}, {-20, 20}}, AspectRatio -> 1],
  ListPlot[Transpose[Strip2d.Trans[House, h, k]], Joined -> True,
```

```
    PlotStyle -> Blue, PlotRange -> {{-10, 10}, {-20, 20}},
    AspectRatio -> 1]
  ],
 {h, -10, 10, 0.1}, {k, -10, 10, 0.1}
 ]
```

Play with these and the other transformations until you feel comfortable with the basics, then do the following problems.

Task Some of these problems will require performing tasks that do not lend themselves to output that can be printed. In these cases print out the code that generates the animations. The instructor may ask you to demonstrate your working code at some future time.

You may need to change ranges of values in the **PlotRange** command in order to keep the patern in the picture.

(1) Generate an animation for the above pattern dilated in the y-direction by a factor that ranges from 0.5 to 1.5 and then rotated by an angle that ranges from $0 \leq \theta \leq 2\pi$. Now reverse the operations and determine if order of operation matters for these two transformations.

(2) Generate an animation for the above pattern with a shear in both the x and y-directions. Play with different ranges for the shear variable(s).

(3) Generate an animation for the above pattern with a translation along a line segment from the point (0,0) to the point (5,3). Now pick a curve that goes through the origin and "fly" the pattern along the curve.