CSC 240 Computer Graphics
Day 8: Matrix Compositions

Nick Howe
Smith College
On Transform Order

- When we compose transformations, we tend to read left to right: $T_1T_2T_3P$
- Mathematically, they are actually applied from rightmost to leftmost $(T_1(T_2(T_3P)))$
- In code, new transformations are added at the rightmost position $T_1T_2T_3T_4P$

...thus are applied **before** all the previous transformations
Q. Why does the drawing axes shrink when scaling it to 2x? Why do we apply the applications in the opposite order? I don't understand why we had to multiply the second puff translations with $S*T1$?

A. In one visualization, we hold the drawing axes constant, and shrink the canvas when we scale 2x. This means that the drawn objects are twice as big relative to the canvas.

In the second visualization, we hold the canvas steady and adjust the axes. The axes double in size with a 2x scaling, and the size of drawn objects double with them.
Let’s revisit my example with attention to the transform:

<table>
<thead>
<tr>
<th>Command sequence</th>
<th>Transform matrix</th>
<th>Square in viewport</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>drawSquare(&quot;blue&quot;);</code></td>
<td>$I$</td>
<td>$\begin{bmatrix} 80 &amp; 120 &amp; 120 &amp; 80 \ 80 &amp; 80 &amp; 120 &amp; 120 \end{bmatrix}$</td>
</tr>
<tr>
<td><code>graphics.scale(2,2);</code></td>
<td>$S$</td>
<td></td>
</tr>
<tr>
<td><code>drawSquare(&quot;red&quot;);</code></td>
<td>$S$</td>
<td>$\begin{bmatrix} 160 &amp; 240 &amp; 240 &amp; 160 \ 160 &amp; 160 &amp; 240 &amp; 240 \end{bmatrix}$</td>
</tr>
<tr>
<td><code>graphics.scale(2,2);</code></td>
<td>$SS$</td>
<td></td>
</tr>
<tr>
<td><code>drawSquare(&quot;yellow&quot;);</code></td>
<td>$SS$</td>
<td>$\begin{bmatrix} 320 &amp; 480 &amp; 480 &amp; 320 \ 320 &amp; 320 &amp; 480 &amp; 480 \end{bmatrix}$</td>
</tr>
<tr>
<td><code>graphics.rotate(-0.5);</code></td>
<td>$SSR$</td>
<td></td>
</tr>
<tr>
<td><code>drawSquare(&quot;green&quot;);</code></td>
<td>$SSR$</td>
<td>$\begin{bmatrix} 434 &amp; 575 &amp; 651 &amp; 511 \ 127 &amp; 51 &amp; 191 &amp; 268 \end{bmatrix}$</td>
</tr>
</tbody>
</table>
Your Questions

Q. I'm still a little confused about how the train hierarchical model works.

A. For hierarchical modeling, you apply the transform for the parent object(s) first. Then you apply the transform for the subpart.

So puff #2 is the child of puff #1, which is the child of the train body.

\[ T_B T_{S_1} S_s T_{S_2} S_s \]

(For consistency with the code usage, I have now reversed the matrix multiplication order given in the quiz.)

Q. I'm not sure why the translations are written like that in the video answers.

A. (Written like what?)
Your Questions

Q. Why do the smoke puffs move differently? Could you explain visually what this animation will look like?

A. When they first come out, the puffs move upwards. Later the move more towards the back of the train.

Q. The first question is why does the smoke puffs use different transformations? I see them always moving to the left upward so I think the translation is always on the same line.

A. This is a judgment call. As described, the movements differ slightly.
Your Questions

Q. I don't quite understand the questions "What is the full transformation applied to driver wheel #1? What is the full transformation applied to smoke puff #2?"

A. You are supposed to combine the transformation components together.

Q. The second is why does the smoke puff 2 take in Ssts1? Is it because it is evolved from puff 1? And can we move Tb to be at the first but not the last one in the full transformation?

A. Yes, the idea is that we base the position of puff #2 on the position of puff #1. And see my previous comment about the matrix order.
Q. For the questions, why would the transformations applied to smoke puff 1 be applied to puff 2, but not the wheels?

A. The wheels are attached to the main body, as is puff 1. Neither is the parent of the other.

Q. Do you always need to scale, rotate, translate in that order? For applying the applications in reverse order, are you supposed to create a separate matrix multiplying the scaling, rotation, translation matrices, then multiply by the object matrix so the object only encounters "one" matrix?

A. You can apply transforms in any order, but some are less confusing than others. You don’t need to combine matrices yourself; this happens for you.
Q. I have a question about save() and restore() functions. Does the save() function store object drawn after calling the function or before? Does the restore() function print again the saved object?

A. The save() and restore() functions don’t do anything with already-drawn objects. They affect the transformation that is applied to any new drawing.

Q. Does the order of transformation applied to each object matters?

A. Hopefully your investigations in homework 3 have made clear that order is crucial. There are a few transformations that commute with each other, but many important ones do not.
Q. Can you explain what is viewport?

A. Here’s an example.
Your Questions

Q. In hierarchical modeling does having many layers of subparts get laggy? How do people make it speedier?

A. Because multiple transformations can be composed into a single matrix, it takes no longer to draw a part in a hierarchy than one that is separate.

Q. Why would people use drawing axis view instead of screen view?

A. I think that the visualization I called “screen view” is the easiest way to approach the use of transformations. I presented the other view to make the point that it is not the only way to think about transformations.
Other Questions?
Lab 3

Transformations! Animation! Excitement!

*The Horse in Motion*
June 1878

Photos by Eadweard Muybridge
Horse: Annie G.
Rider: unknown

(https://fdiv.net/2015/01/02/who-were-jockeys-muybridges-photographs)