Using Flash to Teach Mathematics

G. Donald Allen
Department of Mathematics
Texas A&M University
College Station, TX 77843-3368
dallen@math.tamu.edu

Flash

Flash is the ultra-hot animation tool from Macromedia that touches our lives almost every time we access the Internet. You have seen it most as the creator of the clever but sometimes annoying animations in your browser window. It can be used to make objects "morph" from one shape to another, it can be used to make real-time clocks; it can also be used to make interactive interfaces with the client. Remarkably, it also has applications to mathematics teaching and in surprisingly deep ways. That is the subject of this paper.

The applications to mathematics education range far beyond the usual razzle-dazzle of animated Riemann sums, secant lines converging or surfaces rotating, the typical fair of Maple and Mathematica. Only a few years ago, these types of animations were regarded as modern breakthroughs, devices that could help students visualize and understand what we have always had so difficult a time teaching. With Flash these devices can be joined with timed events such as appearing and dissolving annotations. As well, branching events can be added with varying interactions according to particular responses. Most importantly the animations can be made to be controlled by user input. One particularly interesting and completely unexplored activity, the creation of multiple-choice quizzes with animations as distracters, is relatively simple with Flash.

The competition

The makers of Maple, Mathematica, and MATLAB furnish the main competition to Flash as an animator. Several important distinctions must be made. First and foremost, Flash is not a computer algebra system. Flash is an animation tool. Thus, symbolic mathematical operations are impossible with Flash as it is shipped. Second, Flash runs on most browsers on most platforms, with no cost (and almost no effort) to the user. Third, Flash can be used to create interactive environments well beyond the capacities of the three computer algebra systems. As it happens the makers of Maple, Mathematica, and MATLAB have developed web interfaces, but they do require client based server extensions. The interface design is quite different between them, thus porting a piece of code from one platform to the other is difficult, requiring dual expertise. However, once learned one can apply the tremendous mathematical power of these tools to teaching mathematics.

What can Flash do?

The simpler question is what can it not do, and this has been answered above. It cannot do symbolic and numerical mathematics in the manner of Maple, Mathematica, and MATLAB. It can evaluate functions and it can graph functions. Below we make a list of some of the important features one can use to develop a Flash movie. The term "movie" is the basically the output Shockwave file that Flash creates.

- Animations
- Annotations
- Timed events
- Branching events
- Input/output boxes
- Object morphing
- Interactivity
- Graphing
- Interactive quizzing
- Create or delete multiple objects as needed

How does Flash work?

Flash is a high level language that works both through a JavaScript-like language called ActionScript and though mouse driven animation tools. The latter is for basic animations and morphing with little or no client interaction possible. The results are much like highly sophisticated animated gifs. These mouse tools are effective at augmenting and annotating animated gifs, while often reducing their size. With few exceptions, the animations created through the automated tools can also be programmed with ActionScript. So, the developer writing ActionScript statements can control objects on the stage that is the window where the animations/interactions take place. Afterwards, the program, the actions, scripts, and objects are compiled into a Shockwave (SWF) file ready for the web page. The final step is to embed the SWF file into the web page. Flash does what it can to make this easy, but it still takes some effort. Here is what Flash gives you: the Shockwave file together with an HTML file that has the required embedding code. This code must be copied and pasted exactly into your HTML file where you want the Flash movie to be, for example in the cell of a table. If you also use Macromedia's Dreamweaver HTML editor, it contains a feature to embed the shockwave file as easily as one inserts a graphic file – completely painless.

A little more detail

There are basically four kinds of objects within Flash: text boxes, graphics, buttons, and movies. Making all these function together is the language ActionScript. Consider the single command that "sits under a button", which when pressed stops the current action.

```
on (press) {
        stop();          //This command stops all animations.
}
```

Not all actions are this simple. Below is the animation that will make the object named 'planet' move about in an ellipse.

```
angle = 6 *i*Math.PI /180;
xpos =  250 +   2*90*Math.cos(angle); //pixelate the position
ypos = 200 +  90*Math.sin(angle);
setProperty(planet ,  _x , xpos);
setProperty(planet ,  _y , ypos);
```

From the 'angle' the rectangular coordinates are computed in pixels from a center. Then the 'setProperty' command is evoked to place the planet where is should be. What isn't shown is the command that increments the next angle "i".

This brings to the fore another important feature of Flash, the Timeline. In Flash, each second is divided into frames. When the frame advances the screen is updated with the commands executed during the (time of the) frame. The Timeline is the ordered collection of the frames. So, for moving the planet above we require two frames. In the first frame the positioning commands above are stored, and in the second frame, we place a single command to update the angle. After the last frame is executed, control reverts to Frame 1. So, in our two-frame movie, the commands in the first frame are above; the second frame contains the command

```
i = (i+1)%60;  //This is the Frame 2 command.
```

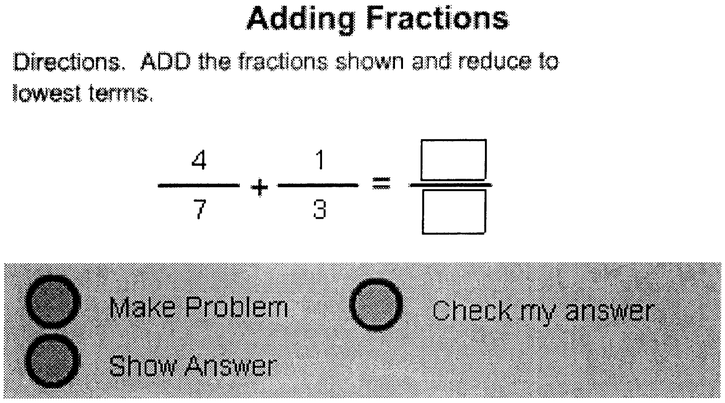
This statement computes the next angle in modulo 60 arithmetic. Note that "i" is multiplied by six in Frame 1. So, one rotation of our planet takes 60 steps of six degrees each.

Now just what is the 'planet'? It is simply a circle that has been declared to be a movie clip. The circle was made with the simple drawing tools that come with Flash. We could just as well have imported a graphic depicting an actual planet.

Another Example

Let's consider a project and list what needs to be done to accomplish it within the Flash-programming environment. *Problem*: write a Flash code to produce two fractions with instructions to the student to sum the fractions and reduce them to lowest terms. *Solution*: In Figure 1, we show the stage will be displayed to the user. This is just design

Let's make an inventory of what is needed. (1) Six input/output boxes for the fraction parts; (2) Title and directions; (3) Three command buttons; (4) Correct and Incorrect pop-ups (not shown). The input/output boxes are easiest to draw; Flash has this facility. They need to be labeled because they must be referenced to make new fractions or retrieve and check the answer. The directions are placed in a text box, easy to make and easy to position. The buttons are just circles declared to be buttons. Indeed, we create just one button and then use it three times. The ActionScript code consists of several simple functions. First we code the make problem function. This will require a random number generator. Flash supplies this. Second, we need a function that adds and reduces the fractions for the "Show Answer" call. Finally we need a "Check my Answer" function to check the input answer that will compare the correct answer with that submitted. In short, sitting under each button is a small amount of code that carries out the specific tasks required.


Adding Fractions

Directions. ADD the fractions shown and reduce to lowest terms.

$$\frac{4}{7} + \frac{1}{3} = \frac{\square}{\square}$$

Figure 1. Adding Fractions Movie Layout

As a practical matter, it's not quite that easy. When making the fractions, some criteria may be required such as the numerator to be less than the denominator. We may also wish the fractions to be different. This bit of logic is written as part of the make problem function. As well, reducing the fraction to lowest terms requires coding up the Euclidean algorithm or something equivalent. Some programming experience, whether in Basic, Fortran, C, or any web language, is very, very helpful. .

Computing functions and advanced features

For mathematicians using technology to teaching, a traditional plateaus has been the parsing, computation, and graphing of a user input function. This is possible within Flash, together with the browser. While flash will evaluate any preprogrammed function, it will not evaluate a user input function. The reason, as Macromedia describes it, is that the code size of Flash player must be highly restricted because it is a downloadable Plug-in. However, using a feature called LiveConnect, it is possible to directly connect the Flash movie with the browser engine. Since the evaluation command in most browsers is extremely powerful, contains a full math parser, it is possible to pass the function you need evaluated to the browser and ship back the result to the Flash movie. Finally, just as with Java, full server side capabilities such as CGI are possible within Flash. As well XML operations can be carried out.

Flash output options

Besides the Shockwave file, Flash gives you many other output options. Referring to our adding fractions example above, Flash will generate an HTML file with the SWF file embedding code. A link to the HTML file can be placed anywhere you choose. Alternatively, you may edit the HTML file as desired with the embedding still in it. However, Flash will also export an animation as an animated GIF file. For example, one could begin with a Maple generated animated GIF, use Flash to slow it down, and add annotations, pauses, and the like and then re-export it as an animated gifs. Naturally, any controls you have created will not function, as animated files are basically static slide shows. Yet there is more. Flash will export fully executable projector files that will operate without a browser. Thus Flash movies can be stand-alone programs. Remarkably, it will export executables for both PC's and the MAC. Finally, it will export files in the QuickTime format. This will make the buttons and actions fully functional.

All is not perfect. New versions of the Flash player emerge faster that people update their browsers. Thus there can be a compatibility issue. This is compounded by the fact that the newest version of Flash, called MX, is much more powerful than earlier versions. Users need to be cautioned that they may need to update their Flash player. It is helpful to add this statement to the HTML page and add the link to the player – it's free. If the code does not use the newest features, it is possible to save the SWF file in the format of earlier Flash versions.

Using Flash to teach math

To understand better what Flash can do for teaching mathematics, it is best to have a basic inventory of math-like tasks that can be accomplished with Flash. The Flash-specific operation listed above can be used with them

- Graphing data sets
- Turning on or off graphics – deleting graphics
- User input functions for computation and graphics
- Computation of all standard functions at given values
- Full logic control (if, and, or)
- Numerical comparisons ($<$, $\leq$, $>$, $\geq$, $=$, $\neq$)

Here are some simple interactive math applications. Similar to the add fractions above, there are also interactive quizzes for addition, subtraction, multiplication, division, and quadratic factoring. Standard fair such as summing a function at integers over a range, showing the tangent to a function at a given point, showing Riemann sums, and general slide shows are all straightforward. A useful application of a slide show is the unfolding of a proof with accompanying diagrams. In consideration of better understanding of the slope and/or intercept, movies have been made that show the slope of a straight line by numerical input or by selecting an animate button. The line can be set to animate with changing slope, or changing intercept, or both. Numerical values accompany each graph. Similarly, we can imagine a movie for the shapes and positions of quadratics. Statistical examples involving distributions, pie charts, bar charts, and the like are also easy to produce.

In the area of statistics, we explored four types of interactive flash movies. In physics, we have shown the action of a pendulum with correct and incorrect dynamics and also with varying gravity and varying degrees of damping. In addition, we have developed a solar system movie, which shows the respective orbits with variable eccentricities. These movies, while not new, are fairly easy to produce within Flash.

These applications and more are at http://www.math.tamu.edu/~dallen/flash_ex.