



## Programming

- Why is programming fun?
  - Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning. The programmed computer has all the fascination of the pinball machine or the jukebox mechanism, carried to the ultimate.

Source: Frederick P. Brooks, Jr. *The Mythical Man-Month Essays on Software Engineering*.



## Recovery

- I have been cleared to return to work
- I'll see you in lecture on Monday!



## Plagiarism

- Presenting someone else's work as your own
- Not giving them credit for their work
- Stealing someone else's work
- Plagiarism is cheating



## Plagiarism

- We have finished grading Project 1B:
  - \* We found cases of plagiarism
  - \* This course has a zero-tolerance policy for cheating
  - \* It will go better for you if you confess
    - Report to any of the TA's or DA in our office hours.



## Project 2

- Project 2: Creating an Online Quiz
  - Create the GUI in HTML
  - Part A: Write the questions
    - Choose any topic you know well
    - Write a list of fill-in-the-blank questions
    - Write a list of multiple-choice questions
    - Create the GUI
  - Parts B,C, D:
    - Score the quiz with JavaScript
    - Add Mouseover effects



## Animation

*JavaScript can be used for animating images on a web page*

© 2004, Lawrence Snyder



## The Plan

An animation is the rapid display of a series of still images ... like cartoons

There are three steps to animation

- 1) Place first still image(s) on web page
- 2) Prefetch the series of images and store them
- 3) Setup a timer to cycle through the images

**new0.gif, new1.gif, new2.gif, new3.gif**

New New New New

Smooth motion requires 30 times/sec display



## Creating GIFs

GIF files for animation are progressively different ... make them w/Photoshop

- The series should all have the same size
- Begin with an initial GIF and build all others from it
- Getting the motion to be smooth may take a bit of fiddling

Animated GIFs -- GIFs that automatically cycle use a special format and software



## 1. Place Still Image(s)

Placing the image uses a standard `<img src=...>` tag

```
<html><head><title>Test Page</title></head>
<body>

<script language="JavaScript"> Code here
</script>
</body>
</html>
```



## The document.images

When HTML draws a page, the images go in an array: **document.images**

- Recall, arrays are names w/ indexes, like A[1]
- Each element of **document.images** array holds one image
- Pictures are put into document.images in the order encountered on page build ... so for Test Page, **document.images[0]** ↔ new0.gif
- Changing the .src property of the array changes the picture

But the images must be prefetched first



## 2. Prefetch Images I

"Prefetch" means to get the images and save them in (our own) array so they are handy to assign to doc.im

- We must declare an array (and probably an index variable, too):

```
var i, pref = new Array(4);
```

- Then we set it up to hold images:

```
for (i=0; i<4; i++) {
  pref[i] = new Image;
}
```




## Prefetch Images II

Once the array is declared and setup, get the images and assign them to the .src field of the array:

```
for (i=0; i<4; i++) {
  pref[i].src = "new" + i + ".gif";
}
```

- Notice that the names of the images, **new0.gif, new1.gif, new2.gif, new3.gif** are constructed using the index variable



### Test It



```

<body>


<script language="JavaScript">
var i, pref = new Array(4);
for (i=0; i<4; i++){
  pref[i] = new Image;
}
for (i=0; i<4; i++){
  pref[i].src="new"+i+".gif";
}
document.images[0].src=pref[1].src
</script>
</body>

```

Place two "0" pix then change the first to "1"

### 3. Change Image

Once Web page is drawn, nothing happens unless you cause an event


- To animate a series of stills you must cause the computer to "wake-up" and change to the next image 30 times a second
- Set a timer to cause the wake-up

```

timerID=setTimeout("animate()",30);

```

↑ Milliseconds to wait  
 ↑ Function to change picture  
 ↑ JS Timer Setting Function  
 ↑ Handle holder -- to be explained




### Animate Function

**animate()** must advance the frame counter, update the image and schedule the next timer ...

```

var frame=0, timeID;
function animate(){
  frame=(frame+1)%4; //advance
  document.images[0].src
    = prefetch[frame].src; //update
  timerID=setTimeout("animate()",30);
}

```





### Watch It Go

```

<h1>My</h1>

<script language="JavaScript">
var i, pref = new Array(4);
var frame=0, timerID;
for (i=0; i<4; i++){
  pref[i] = new Image;
}
for (i=0; i<4; i++){
  pref[i].src="new" + i + ".gif";
}
timerID=setTimeout("animate()",2000); ← Start action
function animate(){
  frame = (frame+1)%4;
  document.images[0].src=pref[frame].src;
  timerID = setTimeout("animate()",30);
}
</script><h1>Page</h1>

```

### Changes ...

Suppose we want "new" to revolve once ever 2 seconds ...


- animate()** sets timer for two different times
- When animating, 30 ms
- When waiting, 2000ms
- Use an **if**-statement

It's a little more subtle

```

if (frame == 0)
  setTimeout("animate()",2000);
else
  setTimeout("animate()",30);

```




### Watch It Go


```

<h1>My</h1>

<script language="JavaScript">
var i, pref = new Array(4);
var frame=0, timerID;
for (i=0; i<4; i++){
  pref[i] = new Image;
}
for (i=0; i<4; i++){
  pref[i].src="new" + i + ".gif";
}
timerID=setTimeout("animate()",2000);
function animate(){
  frame = (frame+1)%4;
  document.images[0].src=pref[frame].src;
  if (frame == 0)
    timerID = setTimeout("animate()",2000);
  else
    timerID = setTimeout("animate()",30);
}
</script><h1>Page</h1>

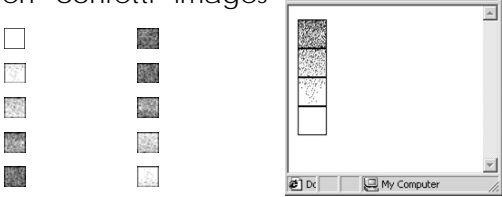
```






## Another Example

Ten "confetti" images




Names: Confett<number>.gif



## Chapter 22

Chapter 22 illustrates solving a large problem -- an animated page

- The main topic is how to decompose a large problem into pieces and reassemble
- Project 2 is a large project (that was divided for you to be an assignment)
- When you have time -- end of term? -- try creating your own animations




## Summary

Animation requires a 3-step process

- 1) Place the initial image(s)
- 2) Prefetch the series of images that will be the animation
- 3) Setup the animation function to draw the next item in the series

When creating your own GIFs make sure that the sizes are all the same



## Reflection

Write for five minutes on this topic:

- \* How do for loops and arrays work together?