



4-2003

The Virtual Beta: An Interactive Fish Using Java Script and CSS

Lauren B. Carroll '03
Illinois Wesleyan University

Follow this and additional works at: https://digitalcommons.iwu.edu/cs_honproj



Part of the [Computer Sciences Commons](#)

Recommended Citation

Carroll '03, Lauren B., "The Virtual Beta: An Interactive Fish Using Java Script and CSS" (2003). *Honors Projects*. 4.
https://digitalcommons.iwu.edu/cs_honproj/4

This Article is protected by copyright and/or related rights. It has been brought to you by Digital Commons @ IWU with permission from the rights-holder(s). You are free to use this material in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This material has been accepted for inclusion by faculty at Illinois Wesleyan University. For more information, please contact digitalcommons@iwu.edu.

©Copyright is owned by the author of this document.

The Virtual Betta

An interactive fish using JavaScript and CSS

By Lauren B. Carroll and Prof. Susan Anderson-Freed,

Illinois Wesleyan University,

Computer Science Dept.

April, 2003

Table of Contents

| | |
|-------------------------|----|
| 1. The C Fish | |
| 1.1. Description | 1 |
| 1.2. Code | 2 |
| 2. Drawing Style | |
| 2.1. Description | 11 |
| 2.2. Sketches | 12 |
| 3. Browser Test | |
| 3.1. Description | 14 |
| 3.2. Code | |
| 3.2.1. Browser Test 1 | 15 |
| 3.2.2. Browser Test 2 | 16 |
| 3.3. Illustrations | |
| 3.3.1. Browser Test 1 | 17 |
| 3.3.2. Browser Test 2 | 19 |
| 4. Fish Object | |
| 4.1. Description | 21 |
| 4.2. Illustration | 22 |
| 5. Timers and Rollovers | |
| 5.1. Description | 24 |
| 5.2. Timers | |
| 5.2.1. Code | 25 |
| 5.2.2. Illustrations | 30 |
| 5.3. Rollovers | |
| 5.3.1. Code | 36 |
| 5.3.2. Illustrations | 38 |

| | |
|-----------------------|----|
| 6. Movement | |
| 6.1. Description | 42 |
| 6.2. Layers | |
| 6.2.1. Code | 43 |
| 6.2.2. Illustrations | 44 |
| 6.3. Scaling | |
| 6.3.1. Code | 46 |
| 6.3.2. Illustrations | 47 |
| 6.4. DIV Tags and CSS | |
| 6.4.1. Code | 49 |
| 6.4.2. Illustrations | 56 |
| 7. Frames | |
| 7.1. Description | 58 |
| 7.2. Code | |
| 7.2.1. Frameset | 59 |
| 7.2.2. Picture Frame | 60 |
| 7.2.3. Control Frame | 61 |
| 7.3. Illustrations | 64 |
| 8. The Final Version | |
| 8.1. Code | |
| 8.1.1. Frameset | |
| 8.1.2. Instructions | |
| 8.1.3. Picture Frame | 69 |
| 8.1.4. Control Frame | 73 |
| 8.2. Illustrations | 78 |
| 9. References | 83 |

My first version of the fish was written in C, because it was the language I was studying at the time. I wanted something that more people could view easily, however and one that could manage images more simply than C. I ended up choosing JavaScript. At first I tried to translate my C program directly to JavaScript, but I soon found this was too complex a task.

```

/* Lauren Carroll
2002
  betta.c
The textual Virtual Betta*/

#include <stdio.h>
#include<string.h>
#include<time.h>
#include <stdlib.h> /* for srand */

#define MAX_WORD 25 /*global for string variables*/
#define MAX_FISH 5 /*max number of fish user can have*/
#define MAX_CHOICE 7 /*max number of choices on main menu*/
#define YES 1 /*for clarity*/
#define NO 0

enum color {red, purple, blue, white, green}; /*colors of fish, for
clarity*/

typedef struct { /*where the fish live*/
  char name[MAX_WORD];
  float price;
  float gallons;
  int dirty;
} bowl;

typedef struct { /*the fish */
  char name[MAX_WORD];
  char body[MAX_WORD]; /*body color*/
  char fin[MAX_WORD]; /*fin color*/
  float age;
  int health;
  int hunger;
} betta;

typedef struct { //the user
  char name[MAX_WORD];
  betta fish[MAX_FISH]; /*array of fish owned*/
  bowl kinds[MAX_FISH]; /*array of fishbowls owned*/
  float money;
  int supplies; /*yes or no */
} player;

void make_fish(betta []);
void make_bowl(bowl []);

char print_menu();
char print_store_menu();

void print_fish(betta []);
void print_bowl(bowl []);

void set_fish(betta*, betta);
void set_bowl(bowl*, bowl);

void look_fish(player, int);
void look_bowl(player, int);

void feed_fish(player*, int);
void heal_fish(player*, int);
void make_sick(player*, int);

```

```

void clean_bowl(player*, int);

void change_fish_name(player*, int);

char choose(int);

int main ()
{
    player one; /*make the user*/
    betta first[MAX_FISH]; /*holding for random petstore fish*/
    bowl start[MAX_FISH];
    char hold[MAX_WORD];

    int j, pick, choose; /*counting variables*/
    int choice= print_menu();
    int fish_count, bowl_count; /*more counting variables*/
    fish_count = bowl_count = 0;

    one.supplies = NO; /*user starts with no supplies*/
    one.money = 14.06; /*starting cash*/

    while (choice != MAX_CHOICE) { /*main menu switch-and-loop*/
        switch (choice) {
            case 1: pick = 0; /*go to petstore, get a sub-menu*/
                while(pick !=4){
                    pick = print_store_menu();
                    switch(pick) {
                        case 1: make_fish(first); /*generate random fish*/
                            print_fish(first); /*display random fish*/
                            printf("Which fish would you like? : ");
                            scanf("%d", &choose);
                            if(choose == -1) /*emergency-exit */
                                break;
                            else {
                                printf("%d %1.2f\n",fish_count, one.money);
                                set_fish(&one.fish[fish_count], first[choose-
1]); /*copy chosen fish to player*/
                                fish_count++;
                                one.money=one.money-3.50;
                                printf("%d %1.2f\n", fish_count, one.money);
                            }
                            break;
                        case 2: make_bowl(start); //generate bowls
                            print_bowl(start); //display bowls
                            printf("Which bowl would you like? : ");
                            scanf("%d", &choose);
                            if(choose == -1) //emergency-exit
                                break;
                            else {
                                printf("%d %1.2f\n",bowl_count, one.money);
                                set_bowl(&one.kinds[bowl_count], start[choose-
1]); /*copy chosen bowl to player*/
                                bowl_count++;
                                one.money=one.money-start[choose-1].price;
                                printf("%d %1.2f\n", bowl_count, one.money);
                            }
                            break;
                        case 3: printf("Supplies, check! \n\n");
                            one.supplies = YES;
                            break;
                    }
                }
            }
        }
    }
}

```

```

        } /* end of switch */
    } /* end of while */
    break;
case 2: look_fish(one, fish_count); /*look at the fish user
has*/
        break;
case 3: feed_fish(&one, fish_count); /*feed the fish the user
has*/
        break;
case 4: heal_fish(&one, fish_count); /*heal the fish the user
has*/
        break;
case 5: clean_bowl(&one, fish_count); /*clean dirty bowls the
user has*/
        break;
case 6: make_sick(&one, fish_count); /*make the fish the user
has sick to test healing*/
        break;
    } /* end of switch */
    choice = print_menu();
} /* end of while */

printf("Bye-Bye! \n");
}

void make_fish(betta first[])
{
/* generate random fish */
int i, k, j, q; /*counting variables */

srand(time(NULL)); /*to randomize numbers*/

for(i=0; i < MAX_FISH; i++)
{
    first[i].age=12.5;
    first[i].health=3;
    first[i].hunger=3;
    strcpy(first[i].name, "Fish");
    k=rand()%(5)+1;
    j=rand()%(5)+1;
    printf(" %d\t%d\n", k, j);
    if(k==1)
        strcpy(first[i].fin, "Red");
    if(j==1)
        strcpy(first[i].body, "Red");
    if(k==2)
        strcpy(first[i].fin, "Purple");
    if(j==2)
        strcpy(first[i].body, "Purple");
    if(k==3)
        strcpy(first[i].fin, "Blue");
    if(j==3)
        strcpy(first[i].body, "Blue");
    if(k==4)
        strcpy(first[i].fin, "White");
    if(j==4)
        strcpy(first[i].body, "White");
    if(k==5)
        strcpy(first[i].fin, "Green");
    if(j==5)
        strcpy(first[i].body, "Green");
}

```

```

    } /* done generating fish */
}

void make_bowl(bowl start[])
{
    int m;

    /* Making bowls now */
    strcpy(start[0].name, "Bitty Betta Economy Hex");
    start[0].gallons=0.25;
    start[0].price=2.59;

    strcpy(start[1].name, "Lil' Critter Kaboodle");
    start[1].gallons=0.75;
    start[1].price=4.28;

    strcpy(start[2].name, "Happy Fish House");
    start[2].gallons=1.25;
    start[2].price=5.35;

    strcpy(start[3].name, "Aquafun Fishtank");
    start[3].gallons=2.75;
    start[3].price=7.00;

    strcpy(start[4].name, "Fish's 5th Avenue");
    start[4].gallons=12.00;
    start[4].price=15.43;

    for (m=0; m < MAX_FISH; m++)
        start[m].dirty=0;
    /* done making bowls */
}

char print_store_menu()
{ /*print petstore menu*/
    int choice;
    fflush(stdin);
    printf("\n\t\tTokoz Brand Pets Menu\n");
    printf("1. Buy a Fish\n");
    printf("2. Buy a Bowl\n");
    printf("3. Buy Supplies\n");
    printf("4. QUIT this Menu\n");
    printf("\nEnter your choice:");
    choice = choose(4);
    return choice;
}

void print_fish (betta first[])
{ /*display pet store fishie stats */
    int q;
    printf("\t\tBetta In Stock\n\n");

    printf("\t\tFin Color\tBody Color\tAge\tHealth\n");
    for(q=0; q < MAX_FISH; q++)
    {
        printf("%s %d\t", first[q].name, q+1);
        printf("\t%s\t\t%s\t\t", first[q].fin, first[q].body);
        printf("%1.1f\t%d\t\n", first[q].age, first[q].health);
    }
}

```

```

char print_menu()
{
    /* print the menu and get the user's choice */
    int choice;
    printf("\n\t\t\t\tVirtual Betta Pet Fish Menu\n");
    printf("1. Go To The Petstore\n");
    printf("2. Look at Your Fish\n");
    printf("3. Feed Your Fish\n");
    printf("4. Medicate Fish\n");
    printf("5. Change Fish Water\n");
    printf("6. Filthify Bowl and Fish\n");
    printf("7. QUIT\n");
    printf("\nEnter your choice:");
    choice = choose(MAX_CHOICE);

    return choice; /* convert to number between 1 and 5 */
} /* end of function */

void set_fish(betta* destn, betta source)
{ /* copy user's choice of petstore fish to user's array */
    destn->age = source.age;
    printf("%f\n", destn->age);
    destn->hunger = source.hunger;
    printf("%d\n", destn->hunger);
    destn->health = source.health;
    printf("%d\n", destn->health);
    strcpy(destn->name, source.name);
    printf("%s\n", destn->name);
    strcpy(destn->fin, source.fin);
    printf("%s\n", destn->fin);
    strcpy(destn->body, source.body);
    printf("%s\n", destn->body);
}

void set_bowl(bowl* destn, bowl source)
{ /* copy user's choice of petstore bowls to user's array */

    destn->price = source.price;
    printf("%f\n", destn->price);
    destn->dirty = source.dirty;
    printf("%d\n", destn->dirty);
    destn->gallons = source.gallons;
    printf("%f\n", destn->gallons);
    strcpy(destn->name, source.name);
    printf("%s\n", destn->name);
}

void look_fish(player one, int fish_count)
{ /* display user's fishie's stats */
    int i;
    if(fish_count == 0)
        printf("you have no fish!\n");
    else {
        printf("Name\t\tFin Color\tBody Color\tAge\tStatus\n");
        for(i=0; i < fish_count; i++) {
            printf("%s\t\t%s\t\t", one.fish[i].name, one.fish[i].fin);
            printf("%s\t\t%1.1f\t", one.fish[i].body, one.fish[i].age);
            if(one.fish[i].health >= 3)
                printf("Sick\n");
        }
    }
}

```

```

        else
            printf("Healthy\n");
            if(one.fish[i].hunger >=3)
                printf("\t\t\t\t\tHungry\n");
            else
                printf("\t\t\t\t\tFull\n");
    } /* end of for */
} /* end of else */

} /* end of function */

void look_bowl(player one, int bowl_count)
{ /* display user's fishbowl stats */
    int i;
    if(bowl_count == 0)
        printf("you have no bowls!\n");
    else {
        printf("\tName\t\t\tGallons Held\tPrice\n");
        for(i=0; i < bowl_count; i++)
        {
            printf("%d %s\t", i+1, one.kinds[i].name);
            printf("\t%1.2fgal\t\t $%1.2f\n", one.kinds[i].gallons, one.kinds
[i].price);
            if(one.kinds[i].dirty >= 3)
                printf("Dirty\n");
            else
                printf("Clean\n");
        } /* end of for */
    } /* end of else */

} /* end of function */

void print_bowl(bowl start[])
{ /* display petstore bowls */

    int q; /*counting variable*/

    printf("\t\tBowls In Stock\n\n");

    printf("\tName\t\t\tGallons Held\tPrice\n");
    for(q=0; q < MAX_FISH; q++)
    {
        printf("%d %s\t", q+1, start[q].name);
        printf("\t%1.2fgal\t\t $%1.2f\n", start[q].gallons, start[q].price);
    }
} /* end of function */

void feed_fish(player* one, int fish_count)
{ /* feed user's hungry fish */

    int choice, i, j; /*counting variables*/
    choice = 0;

    if(!fish_count)
        printf("you have no fish!\n");
    else if (one->supplies == NO)
        printf("You have no supplies!\n");
    else {
        look_fish(*one, fish_count);
        printf("Feed which fish? : ");
        scanf("%d", &choice);
    }
}

```

```

    if ((choice > fish_count) || (choice < 0))
        printf("You can't feed a non-existent fish!\n");
    else if (one->fish[choice-1].hunger < 0)
        printf("Fish isn't hungry. Trust me. \n");
    else {
        printf("%d \n", one->fish[choice-1].hunger);
        one->fish[choice-1].hunger -= 2;
        one->kinds[choice-1].dirty += 1;
        printf("%d \n", one->fish[choice-1].hunger);
    } /* end of inner else */

} /* end of outer else */

} /* end of function */

void heal_fish(player* one, int fish_count){
    /* heal user's sick fish with supplies */

    int choice, i, j;
    choice = 0;

    if(!fish_count)
        printf("you have no fish!\n");
    else if (one->supplies == NO)
        printf("You have no supplies!\n");
    else {
        look_fish(*one, fish_count);
        printf("heal which fish? : ");
        scanf("%d", &choice);
        if ((choice > fish_count) || (choice < 0))
            printf("You can't heal a non-existent fish!\n");
        else if (one->fish[choice-1].health < 0)
            printf("Fish isn't sick anymore. Trust me. \n");
        else {
            printf("%d \n", one->fish[choice-1].health);
            one->fish[choice-1].health = one->fish[choice-1].health - 2;
            printf("%d \n", one->fish[choice-1].health);
        } /* end of inner else */

    } /* end of outer else */
} /* end of function */

void make_sick(player* one, int fish_count){
    /* makes fish sick and bowl dirty to test out other functions */
    int i;

    if(!fish_count)
        printf("you have no bowls!\n");
    else {
        printf("begriming... ");
        for(i=0; i < fish_count; i++){
            one->fish[i].health += 3;
            one->fish[i].age += 0.5;
            one->fish[i].hunger += 3;
            one->kinds[i].dirty += 3;
            printf("...\t");
        } /* end of for */
        printf("done!\n");
    } /* end of else */
} /* end of function */

```



```

void clean_bowl(player* one, int bowl_count)
{ /* cleans user's fishbowls */

int choice, i, j;
choice = 0;

if(!bowl_count)
    printf("you have no bowls!\n");
else if (one->supplies == NO)
    printf("You have no supplies!\n");
else {
    look_bowl(*one, bowl_count);
    printf("Clean which bowl? : ");
    scanf("%d", &choice);
    if ((choice > bowl_count) || (choice < 0))
        printf("You can't clean a non-existent bowl!\n");
    else if (one->kinds[choice-1].dirty < 0)
        printf("That bowl is already spotless, trust me. \n");
    else {
        printf("%d \n", one->kinds[choice-1].dirty);
        one->kinds[choice-1].dirty -= 2;
        printf("%d \n", one->kinds[choice-1].dirty);
    } /* end of inner else */

    } /* end of outer else */

} /* end of function */

void change_fish_name(player* one, int fish_count){
char choice;
char newn[MAX_WORD], temp[MAX_WORD];

if(!fish_count)
    printf("you have no fish!\n");
else {
    look_fish(*one, fish_count);
    printf("Change the name of which fish? : ");
    scanf("%d", &choice);
    if ((choice > fish_count) || (choice < 0))
        printf("The non-existent fish doesn't need a new name.\n");
    else {
        printf("Enter fish's new name: ");
        gets(newn);

        strcpy(temp, one->fish[choice-1].name);
        strcpy(one->fish[choice-1].name, newn);

        printf("\nfish was named %s, and is now named %s.\n", temp,
                one->fish[choice-1].name);

    } /* end of inner else */
} /* end of outer else */

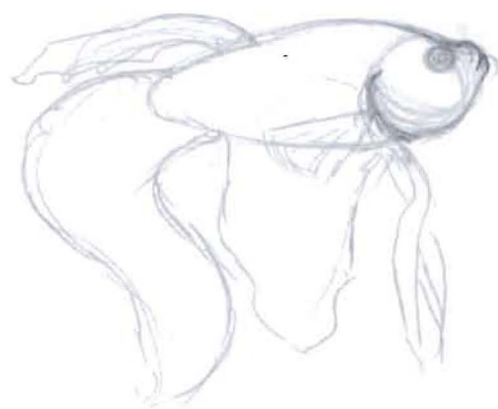
} /* end of function */

char choose(int n)
{
    int choice;
    choice = getchar();
    while ((choice == 10) || (choice == 13)) /*discard CRs and FFs */

```

```
    choice = getchar();
    while ((choice < 49) || (choice > (n+48))) {
        /* check for range errors 48-54 = integers from 1 to n */
        printf("Whoops! I only accept numbers between 1 and %d\n", n);
        printf("Enter your choice: ");
        choice = getchar();
        while ((choice == 10) || (choice == 13))
            choice = getchar();
    } /* end of while */
    return choice - 48; /* convert to number between 1 and 5 */
}
```

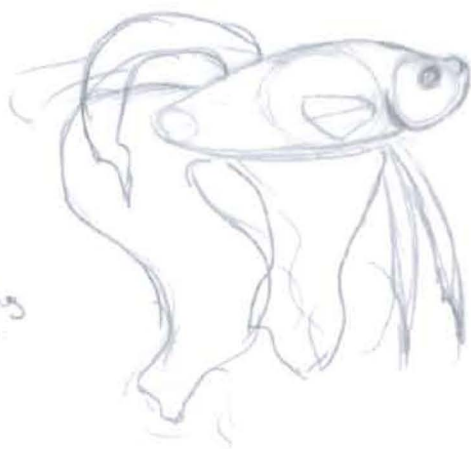
I wanted a picture that conveyed concretely the type of fish I was using, the Siamese Fighting Fish, but one that wasn't so complex that it became hard to manipulate or too large for the web. I went through many versions, but eventually settled on a cartoonish one that was easy for me to draw repeatedly, but still looked like a betta fish. I also drew out designs for the layout of the controls. I knew from the beginning I wanted to use frames.



Sat
↙



kinanuraw



← not so Sat



← saying
we
drawing
him



bottom
view



grr!



Swimming
down
whoosh!



he's
looking
at you



My first JavaScript function was a small one to determine the type of browser being used and put the results on a new page, one that was generated by the script. I changed it so that the results were put on the bottom of the current page. I included the browser checking function in my project, though the fish is only viewable by IE at the moment. Internet Explorer adheres more closely than Netscape to the standards of JavaScript put out by the 3WC.

```
<html>

<head>

<title>

Betta Fish test of JavaScript
</title>
<SCRIPT LANGUAGE = "JavaScript">
bName = navigator.appName;

if (bName == "Netscape") ver="n";
    else if (bName == "Microsoft Internet Explorer") ver="ie";
        else ver="otr";

if (ver == "n") { window.location.href="netscape.html"}
else if (ver=="ie") {window.location.href="ie.html"}
else if (ver=="otr") {window.location.href="other.html"}

</script>

<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<h2> Stub</h2>
<hr>

</body>
</html>
```

```

<html>

<head>

<title>

Betta Fish test of JavaScript
</title>
<SCRIPT LANGUAGE = "JavaScript">
bName = navigator.appName;

if (bName == "Netscape") ver="n";
    else if (bName == "Microsoft Internet Explorer") ver="ie";
        else ver="otr";
</script>


<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<h2> BROWSER DETECTION TEST 2 -- no reloading pages</h2>
<hr>
Checking your browser type....
<br><br><br>

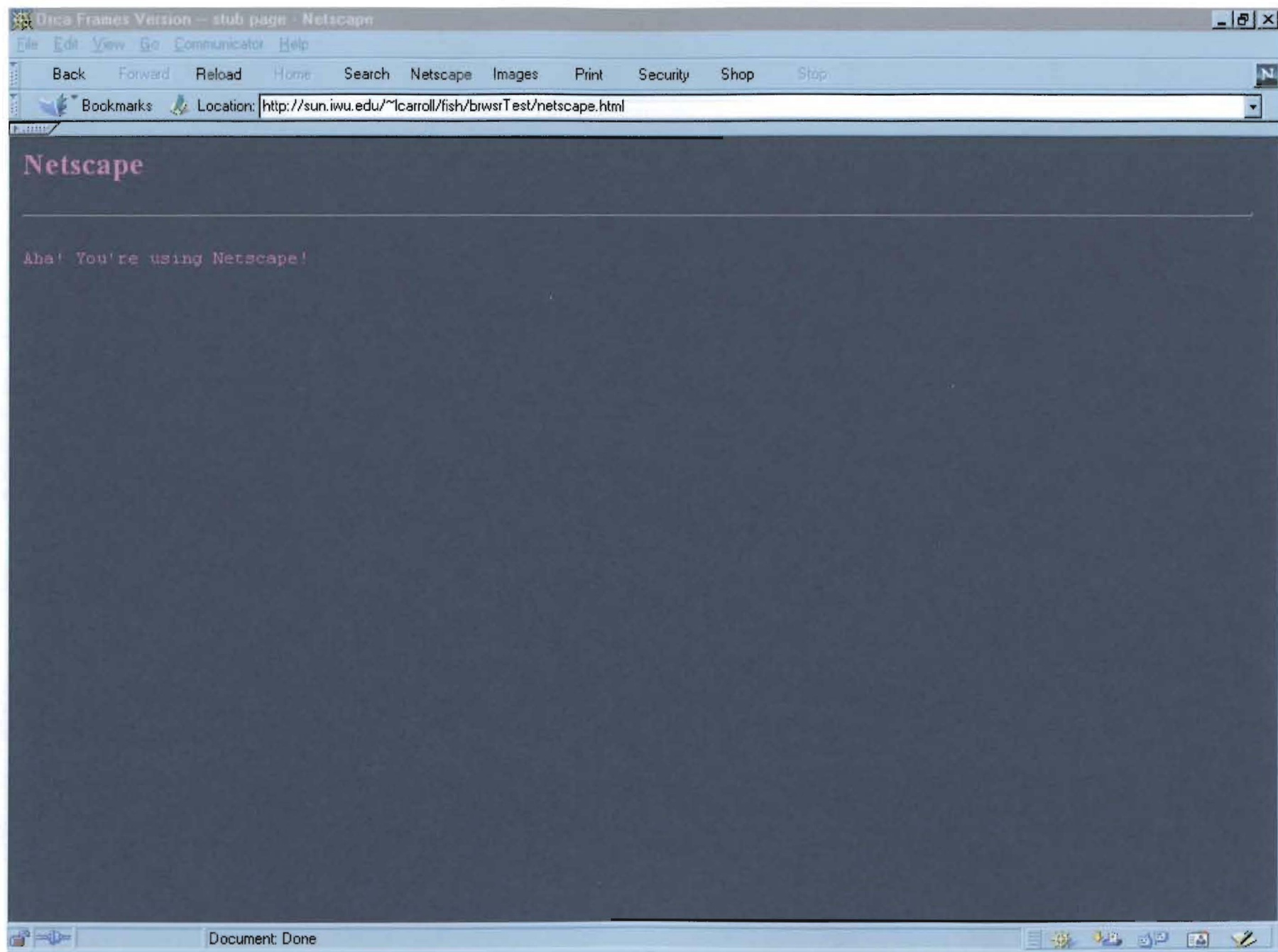
<script language="JavaScript">
var page = "" //start next part of page...?

if (ver == "n") { page += "Aha! You're using Netscape! <br>" }
else if (ver=="ie") {page += "Aha! You're using Internet Explorer!
<br>"}
else if (ver=="otr") { page += "Aha! You're... You're using... Something
" +
                        "that's not Internet Explorer or Netscape... Right?
<br>"}

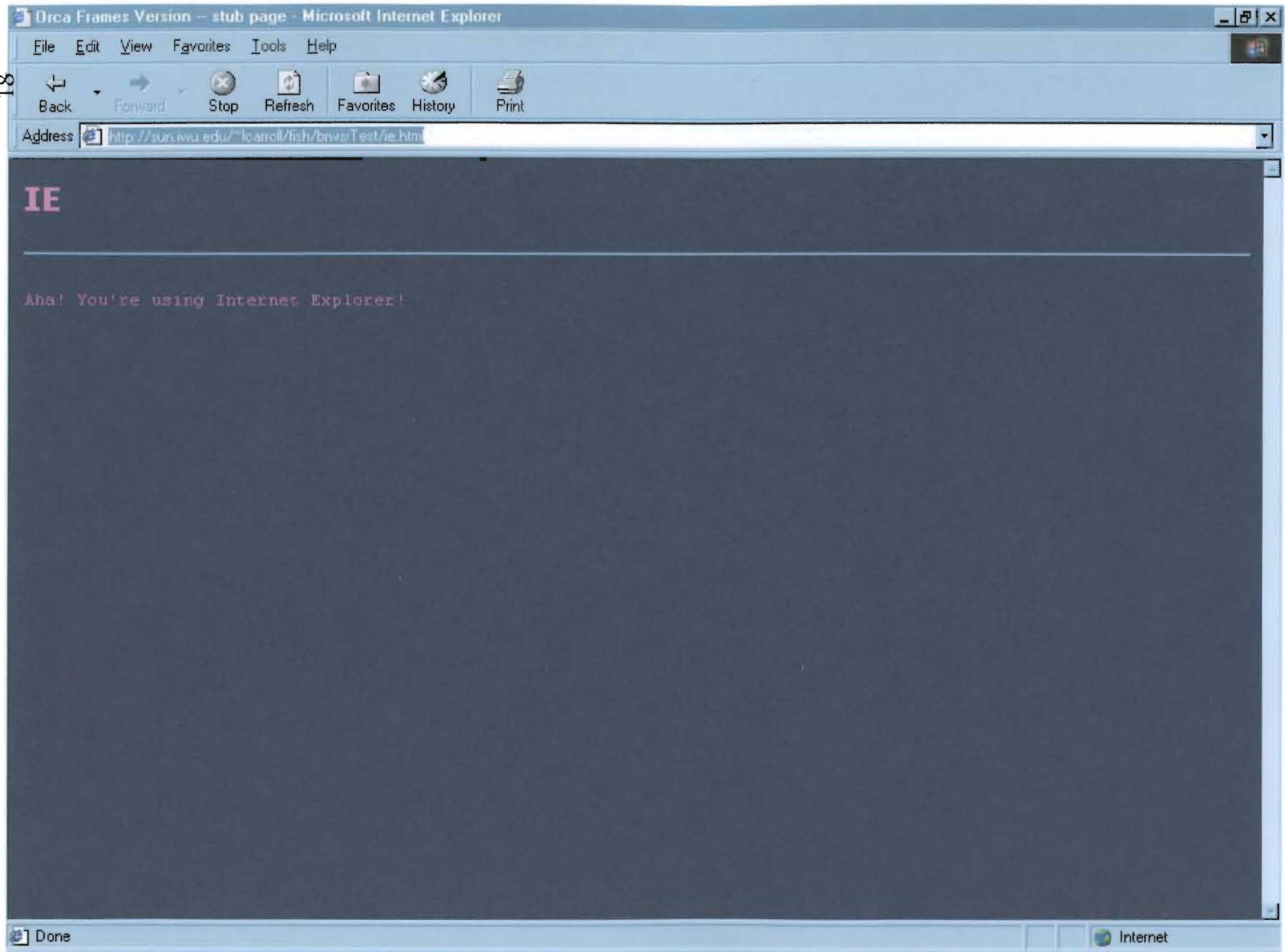
document.write(page)
</script>

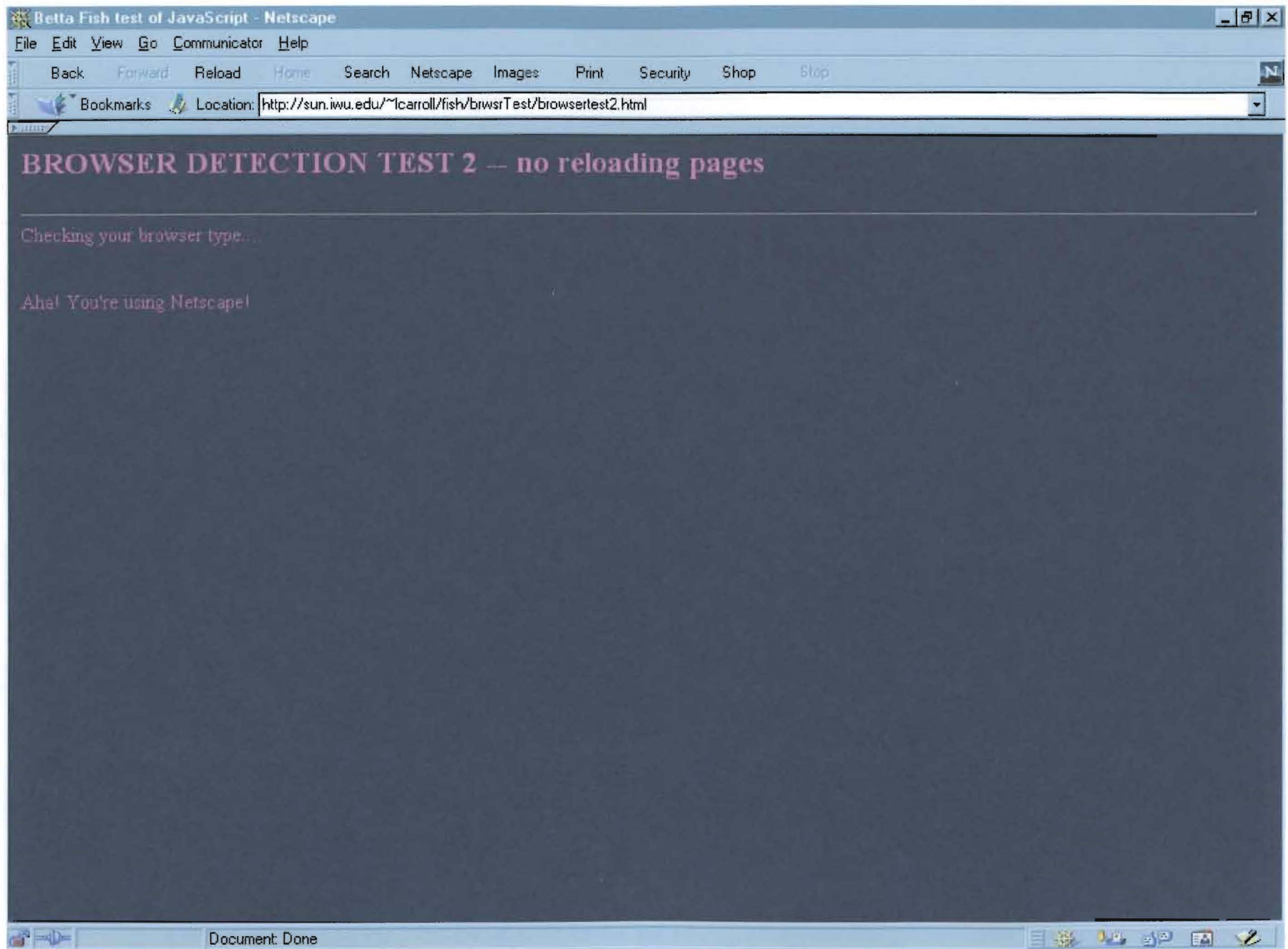
</body>
</html>

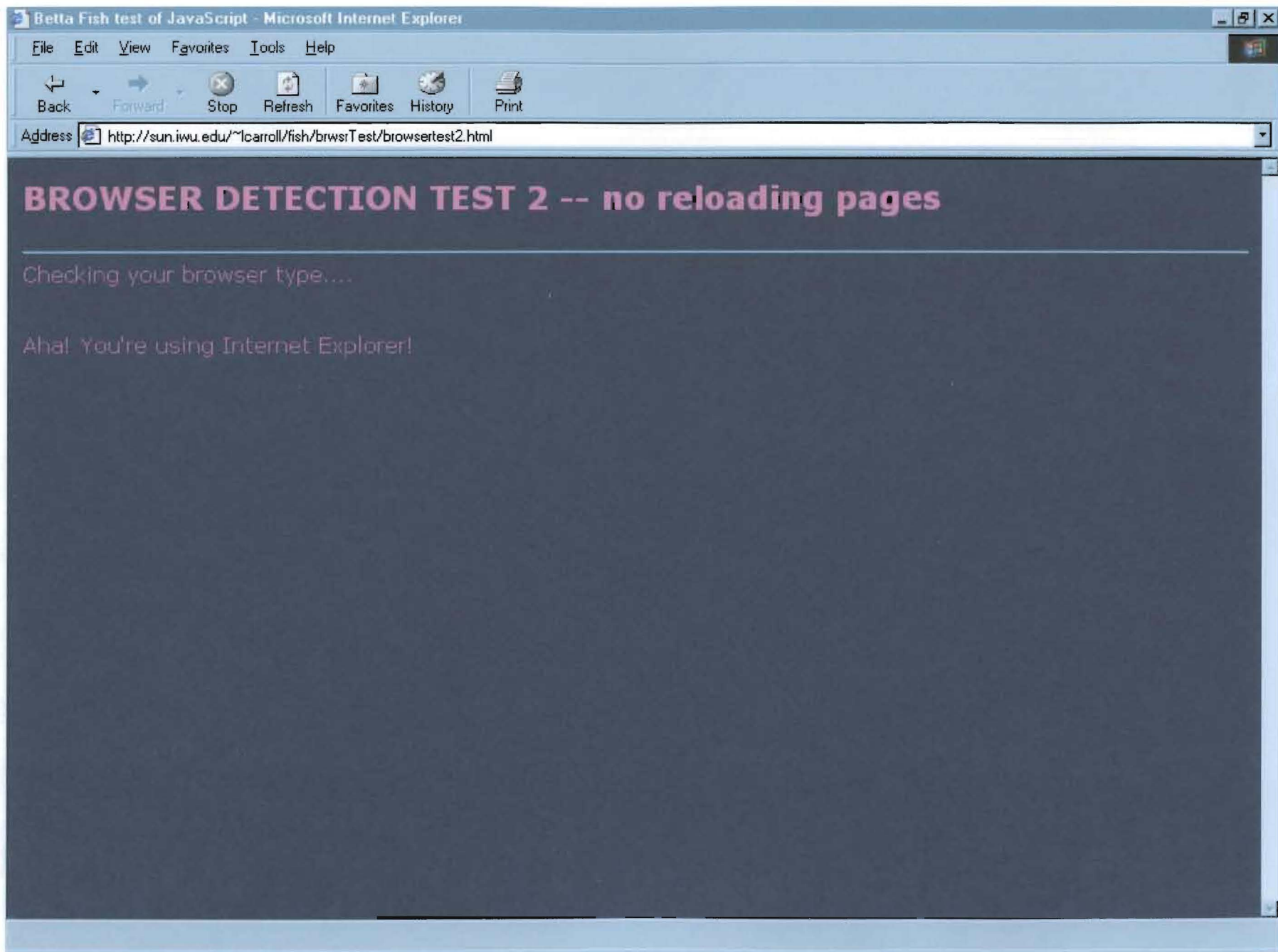
```

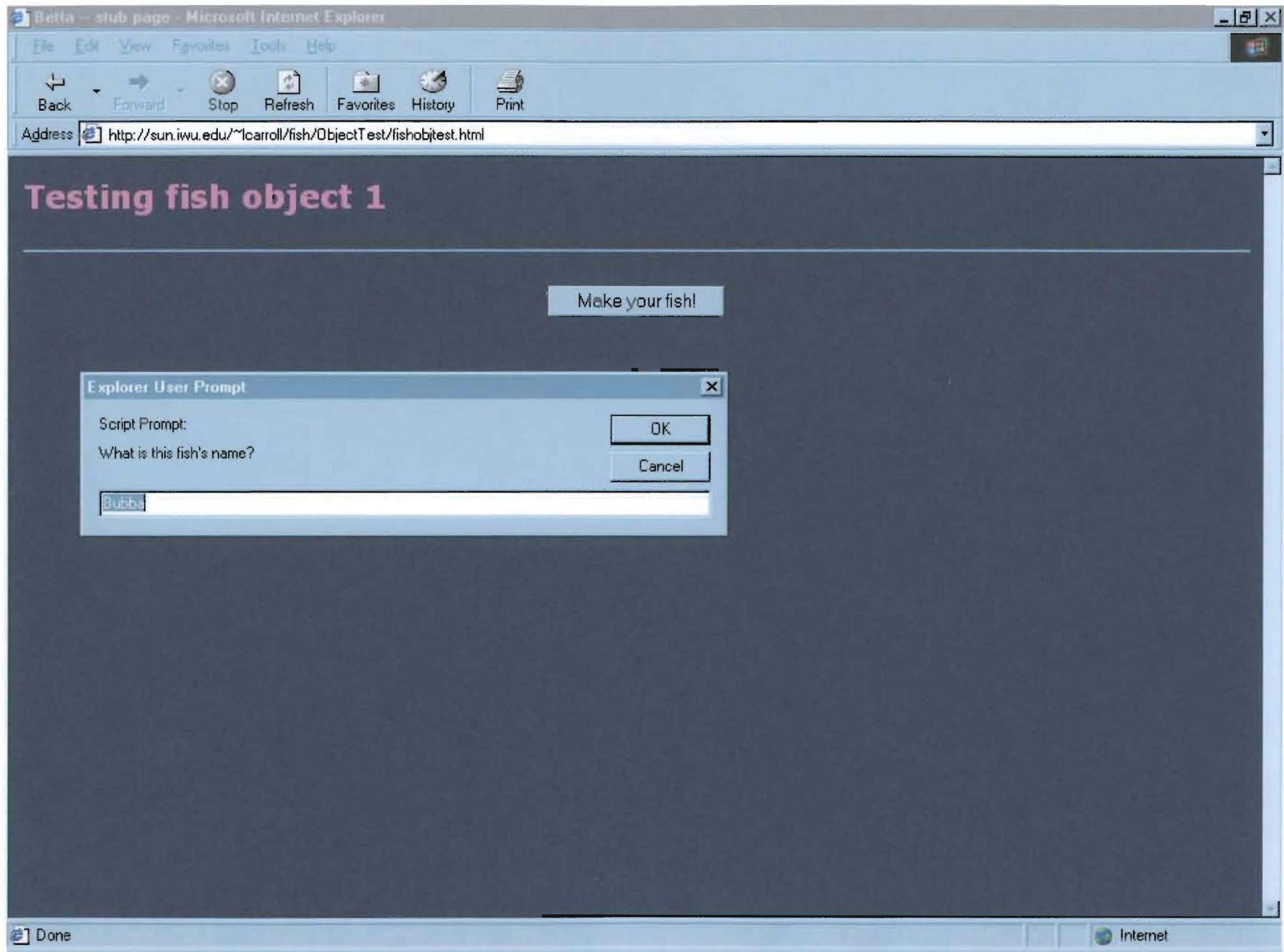
18

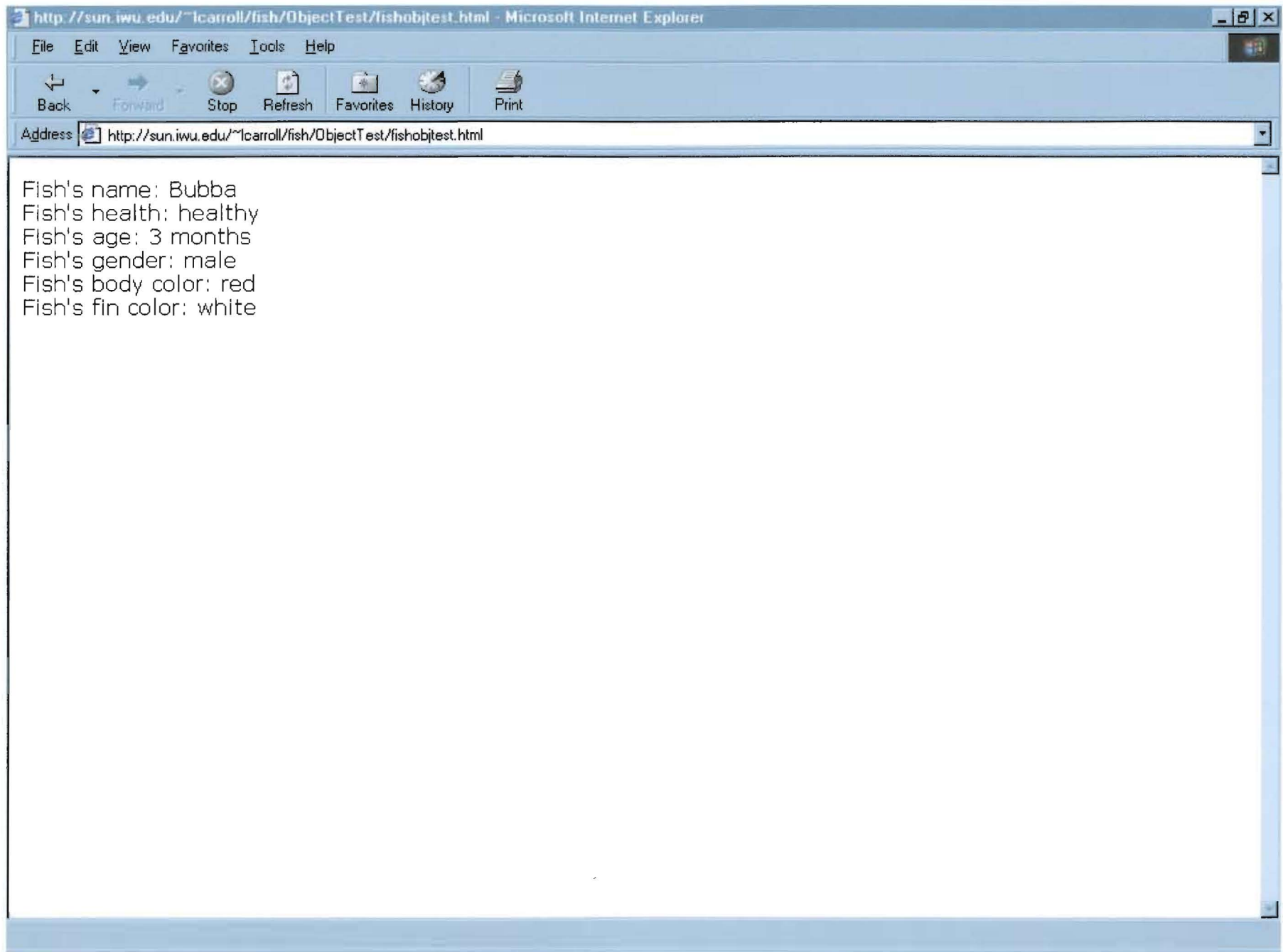






Next, I made an object to hold the variables describing the fish. I modeled it off my C program, in which I made a struct for the fish. Popup windows prompted the user to fill in data, and default values were provided for all boxes. The results were displayed on a new page. I ended up not using the object in my project.





I made pages that defined a countdown timer. The whole amount of time was set by the user through radio buttons and the countdown display was shown in a text box. This was managed internally through the use of an increment to tell a built-in JavaScript function when to update the display. I made the increment adjustable by the user as well, even though nothing changed in this display.

Next I made image rollover. I had a fish image that, when the user moved the mouse over it, changed into another image. With a little bit of modification, this function became a cycling function, which allowed the user to cycle through a set of images when a button was pushed.

By combining the functions on these two pages, I made a function to animate my fish. A timer counted down a user-chosen number of seconds, and at a user-chosen interval, the image of the fish updated itself. The images were all preloaded, so there was no waiting for the pictures to change. If the interval was fast enough, it tricked the eye into seeing the fish wave its fins like it was swimming. Here, the chosen interval controlled how fast the fish images updated.


```

<html>

<head>

<title>

Betta -- stub page
</title>
<script language="JavaScript1.2">
<!--
var running = false
var end = null
var timerID = null
var count = .5 //default 30 sec countdown

function startTimer() {
    running = true
    now = new Date()
    now = now.getTime()

    end = now + (1000 * 60 * count) //set number of minutes for countdown
    (the 1)

    showCountDown()
}

function showCountDown() {
    var now = new Date()
    now = now.getTime()

    if (end - now <= 0) { //time reached
        stopTimer()
        document.forms[0].timerDisplay.value = "Timer has stopped"
    }
    else {
        var delta = new Date(end - now)
        var mins = delta.getMinutes()
        var secs = delta.getSeconds()
        var times = mins

        //checks to see if seconds are less than 10, to display them properly
        times += ((secs <10) ? ":0" : ":") + secs

        //writes the timer to a text window on the page
        document.forms[0].timerDisplay.value = times

        //waits a second (1000 msec) before doing calculations again
        if (running) {
            timerID = setTimeout("showCountDown()", 1000)
        } // end running if
    } //end else
} //end function

function stopTimer() {
    clearTimeout(timerID)
    running = false
    document.forms[0].timerDisplay.value = "0:00"
}

```

```

} // end function

//-->
</script>

</head>

<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<h2> timer test 2 -- countdown timer with variables</h2>
<hr>

Please set countdown length first<br>

<form>

<input type="radio" name="countdown" value="One (1) minute"
      onClick="count = 1"> One (1) minute <br>
<input type="radio" name="countdown" checked value="One half (0.5)
minute"
      onClick="count = .5"> One half (0.5) minute <br>
<input type="radio" name="countdown" value="One quarter (0.25) minute"
      onClick="count = .25"> One quarter (0.25) minute <br>
<input type="radio" name="countdown" value="One tenth (0.10) minute"
      onClick="count = .1"> One tenth (0.10) minute <br>
<br>

<input type="button" name="startTime" value="Start countdown"
      onClick = "startTimer()">

<input type="button" name="clearTime" value="reset countdown"
      onClick = "stopTimer()">

<br><br>

<input type = "text" name="timerDisplay" value=" ">

</form>

</body>
</html>

```

```

<html>

<head>

<title>

Betta -- stub page
</title>
<script language="JavaScript1.2">
<!--
var running = false
var end = null
var timerID = null
var count = .5 //default 30 sec countdown
var duratn = 1000 //count in one second increments

var fishGIFs = new Array("colorfish1.gif", "colorfish2.gif",
"colorfish3.gif",
                        "colorfish4.gif");
var fishpics = new Array(fishGIFs.length);
var pictNo = 0; // global picture counter

    for (var i = 0; i < fishpics.length; i++) { //stuff pics into array and
preload
        fishpics[i] = new Image();
        fishpics[i].src = fishGIFs[i];
    }

function startTimer() {
    running = true
    now = new Date()
    now = now.getTime()

    end = now + (1000 * 60 * count) //set number of minutes for countdown
(the 1)

    showCountDown()
}

function showCountDown() {
    var now = new Date()
    now = now.getTime()

    if (end - now <= 0) { //time reached
        stopTimer()
        document.forms[0].timerDisplay.value = "Timer has stopped"
    }
    else {
        var delta = new Date(end - now)
        var mins = delta.getMinutes()
        var secs = delta.getSeconds()
        var times = mins

        //checks to see if seconds are less than 10, to display them properly
        times += ((secs <10) ? ":0" : ":") + secs

        //writes the timer to a text window on the page

```

```

    document.forms[0].timerDisplay.value = times

    //change image every second
    displayFish()
    document.forms[0].test.value = pictNo

    //waits a second (1000 msecs) before doing calculations again
    if (running) {
        timerID = setTimeout("showCountDown()", duratn)
    } // end running if
    } //end else
} //end function

function stopTimer() {
    clearTimeout(timerID)
    running = false
    document.forms[0].timerDisplay.value = "0:00"
} // end function

function displayFish()
{
    //display the picture with each button click
    if (pictNo == fishpics.length) {
        pictNo = 0;
        document.Fish.src = fishpics[0].src;
    }
    else {
        document.Fish.src = fishpics[pictNo].src;
        pictNo++;
    }
} // end of function

//-->
</script>

</head>

<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<h2> timer test 4 -- countdown with definable increment and duration,
and picture</h2>
<hr>

<IMG SRC = "clear.gif" NAME = "Fish" align = "right">

Please set countdown length first<br>

<form>

<input type="radio" name="countdown" value="One (1) minute"
    onClick="count = 1"> One (1) minute <br>
<input type="radio" name="countdown" checked value="One half (0.5)
minute"
    onClick="count = .5"> One half (0.5) minute <br>
<input type="radio" name="countdown" value="One quarter (0.25) minute"
    onClick="count = .25"> One quarter (0.25) minute <br>
<input type="radio" name="countdown" value="One tenth (0.10) minute"
    onClick="count = .1"> One tenth (0.10) minute <br>
<br>

<input type="button" name="startTime" value="Start countdown"
    onClick = "startTimer()">

```

```

<input type="button" name="clearTime" value="reset countdown"
      onClick = "stopTimer()">

<br><br>

<input type = "text" name="timerDisplay" value=" "><br>

<input type = "text" name="test" value=" ">

<br><br>
Please set the increments<br>

<input type="radio" name="increment" checked value="One (1) sec
increments"
      onClick="duratn = 1000"> Count in one (1) second increments <br>

<input type="radio" name="increment" value="Two (2) sec increments"
      onClick="duratn = 2000"> Count in two (2) second increments <br>

<input type="radio" name="increment" value="half sec increments"
      onClick="duratn = 500"> Count in half second increments <br>

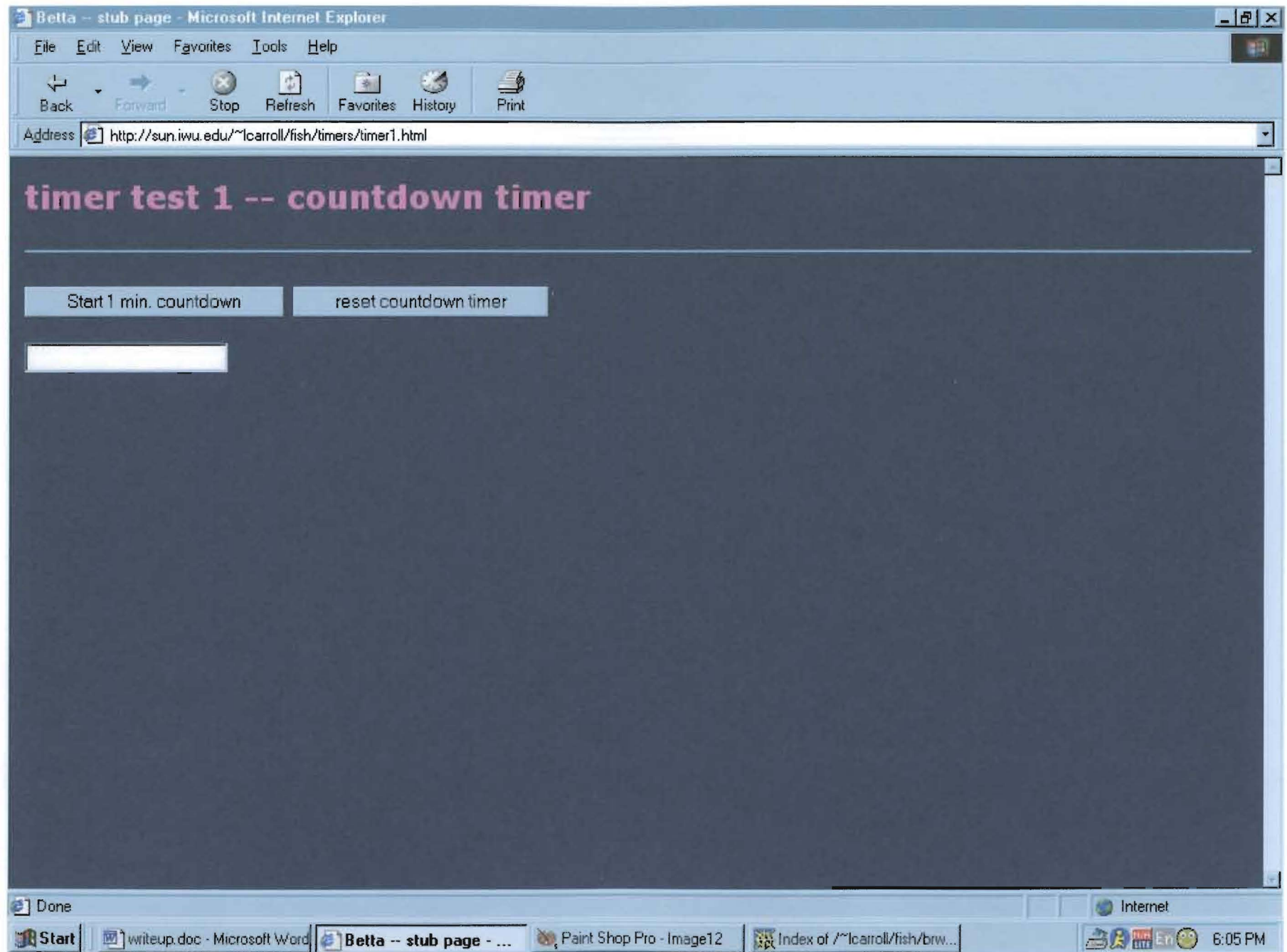
<input type="radio" name="increment" value="quarter sec increments"
      onClick="duratn = 250"> Count in quarter second increments <br>

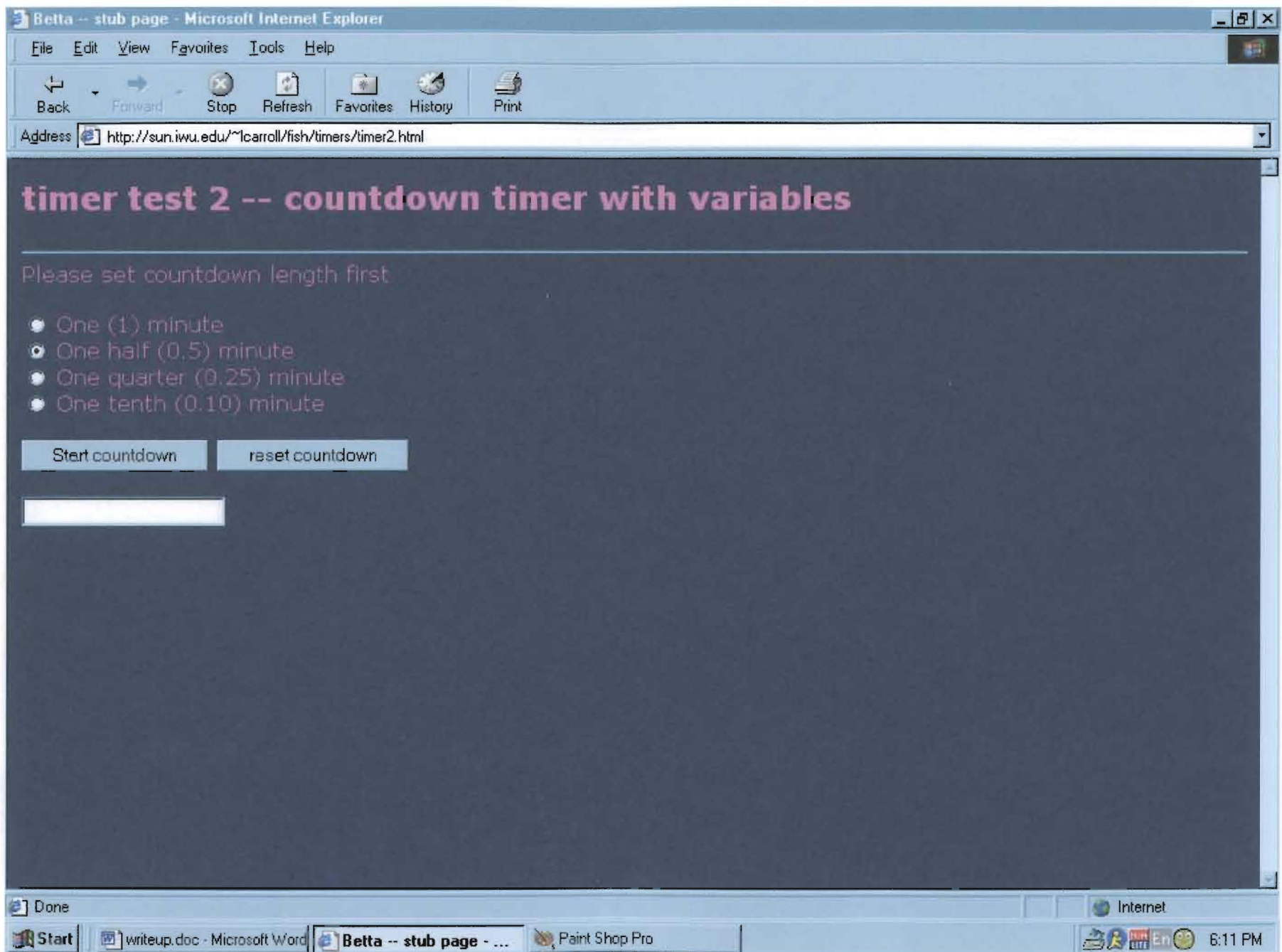
<input type="radio" name="increment" value="tenth sec increments"
      onClick="duratn = 100"> Count in tenth second... erm, count really
fast <br>
<br>

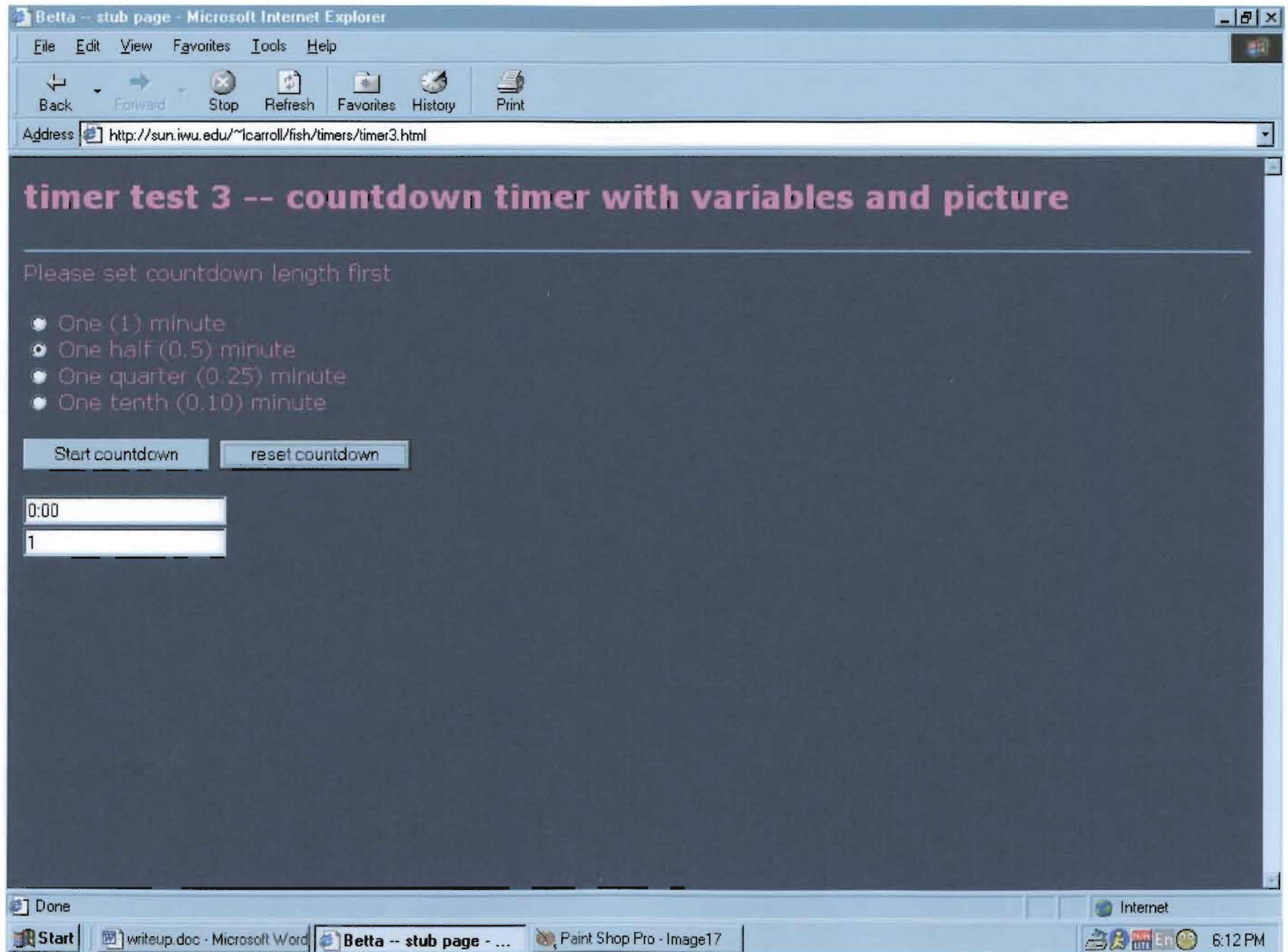
</form>

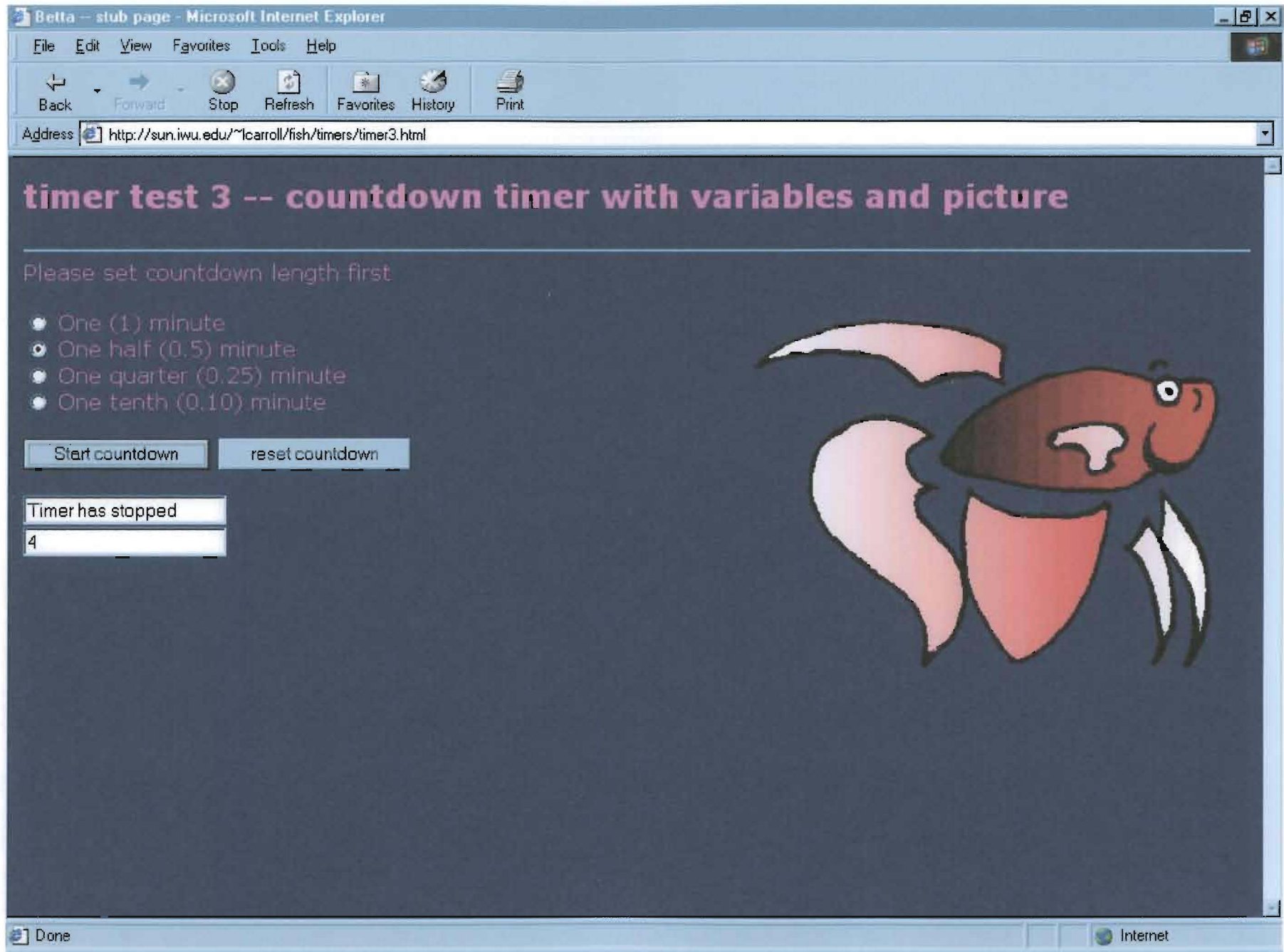
</body>
</html>

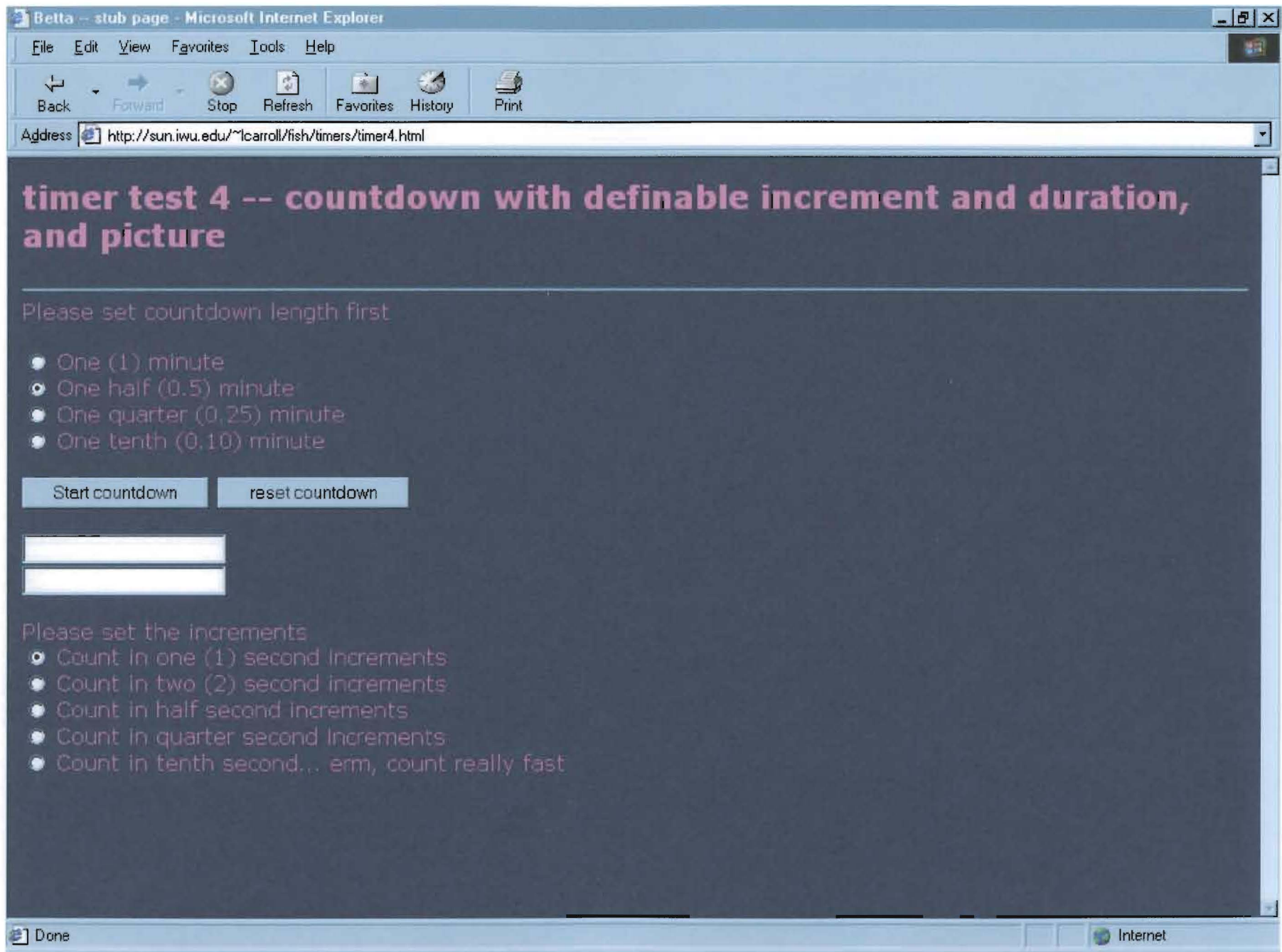
```

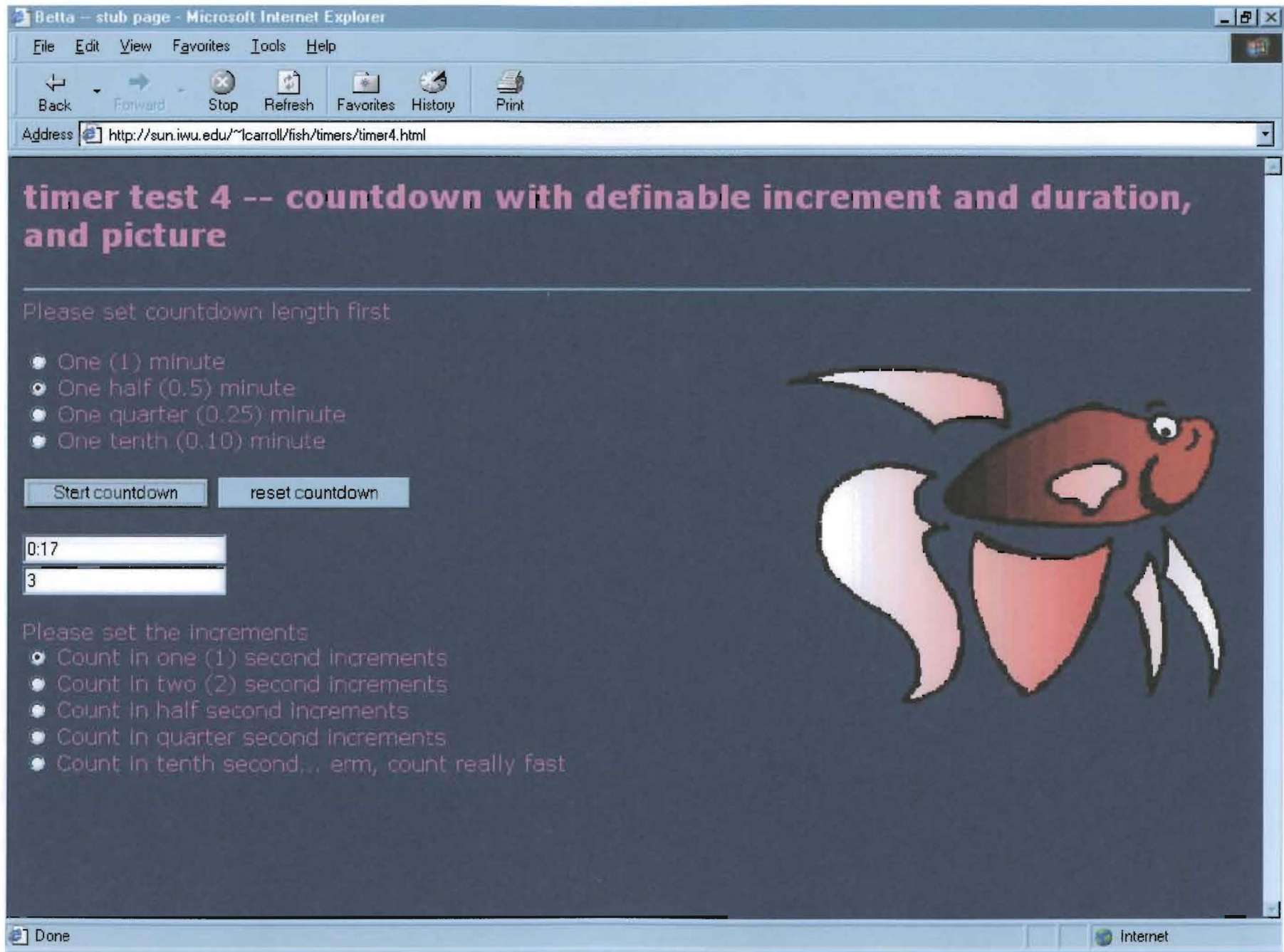












```

<html>

<head>

<title>

Betta Fish test of JavaScript2
</title>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    if(document.images){
        imglon = new Image();
        imglon.src = "fish1.gif";

        imgloff = new Image();
        imgloff.src = "fish2.gif";
    }

function imgOn(imgName) {
if (document.images) {
    document[imgName].src = eval(imgName + "on.src");
}}

function imgOff(imgName) {
if (document.images) {
    document[imgName].src = eval(imgName + "off.src");
}}

//-->
</script>

<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<h2> Image rollover1</h2>
<hr>

<a href="stub.html" onMouseOver="imgOn('img1')"
    onMouseOut="imgOff('img1')">
 </a>
</body>
</html>

```

```

<html>

<head>

<title>

Betta Fish test of JavaScript2
</title>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    if(document.images){
        imglon = new Image();
        imglon.src = "fish1.gif";

        imgloff = new Image();
        imgloff.src = "fish2.gif";

        imgltest = new Image();
        imgltest.src = "femfish1.gif";
    }

function imgOn(imgName) {
if (document.images) {
    document[imgName].src = eval(imgName + "on.src");
    document["holder"].src = eval(imgName + "test.src");
}}

function imgOff(imgName) {
if (document.images) {
    document[imgName].src = eval(imgName + "off.src");
    document["holder"].src = "clear.gif";
}}

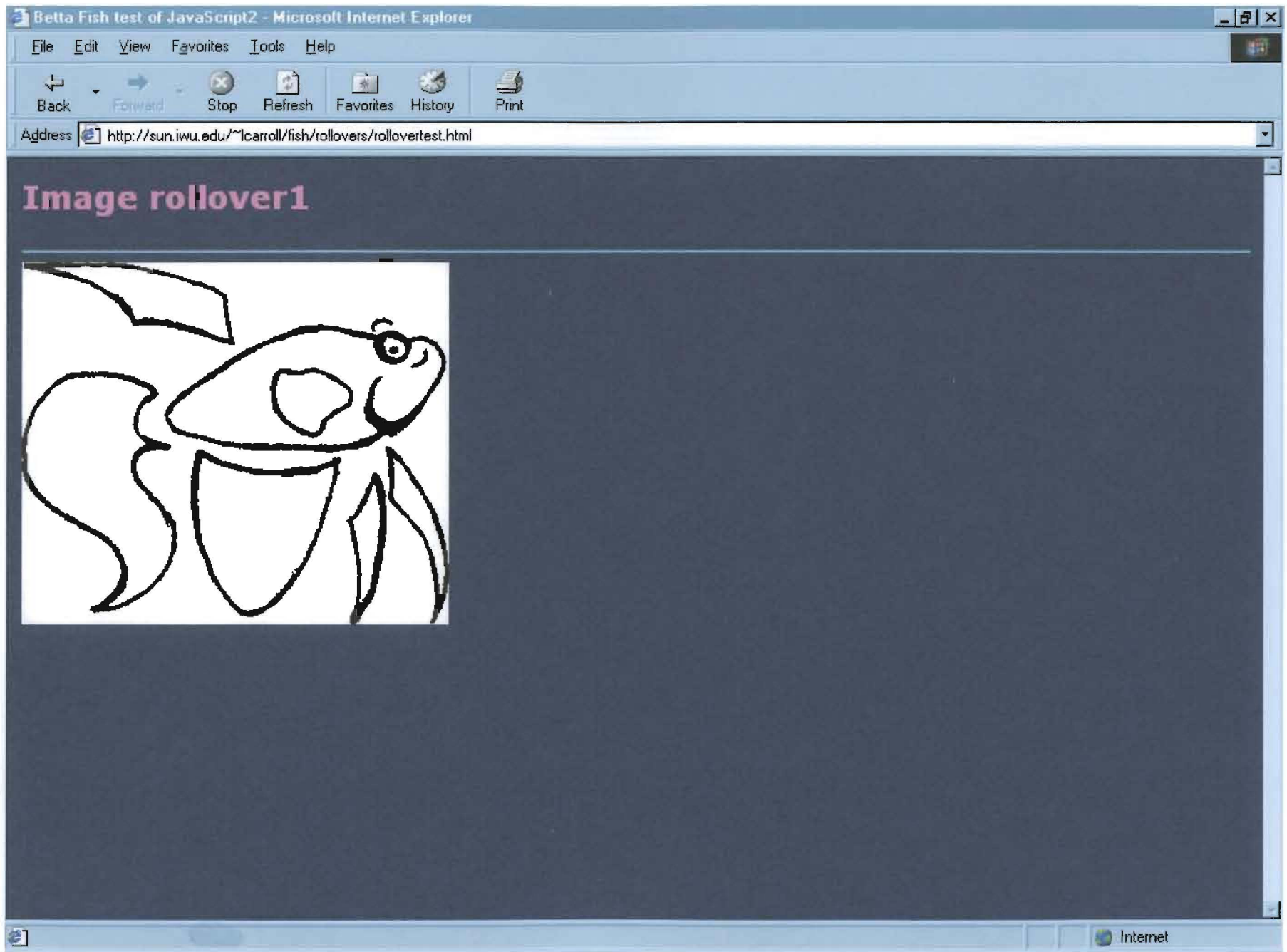
//-->
</script>

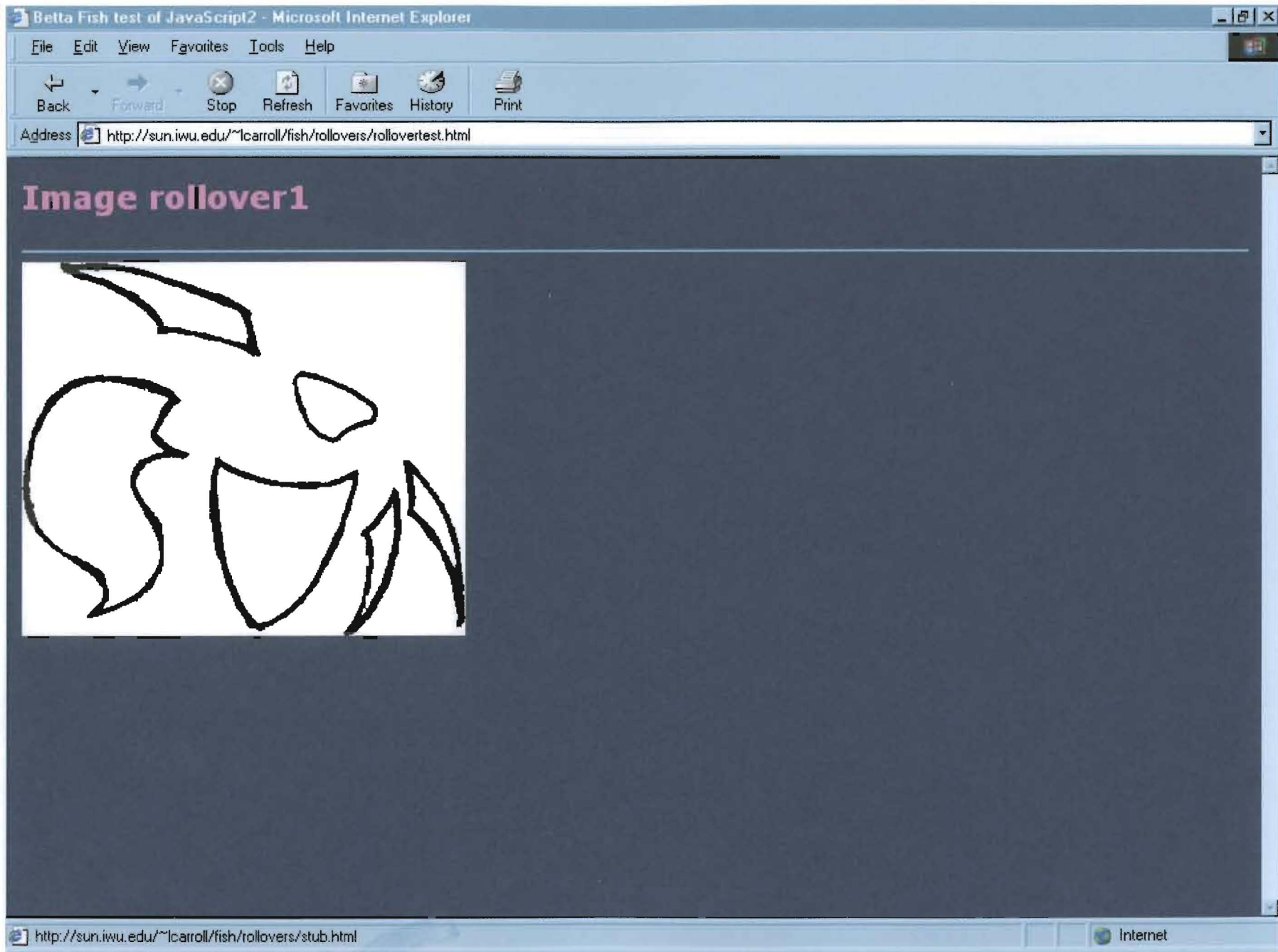
<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<h2> Image rollover2 -- 2 images at once</h2>
<hr>

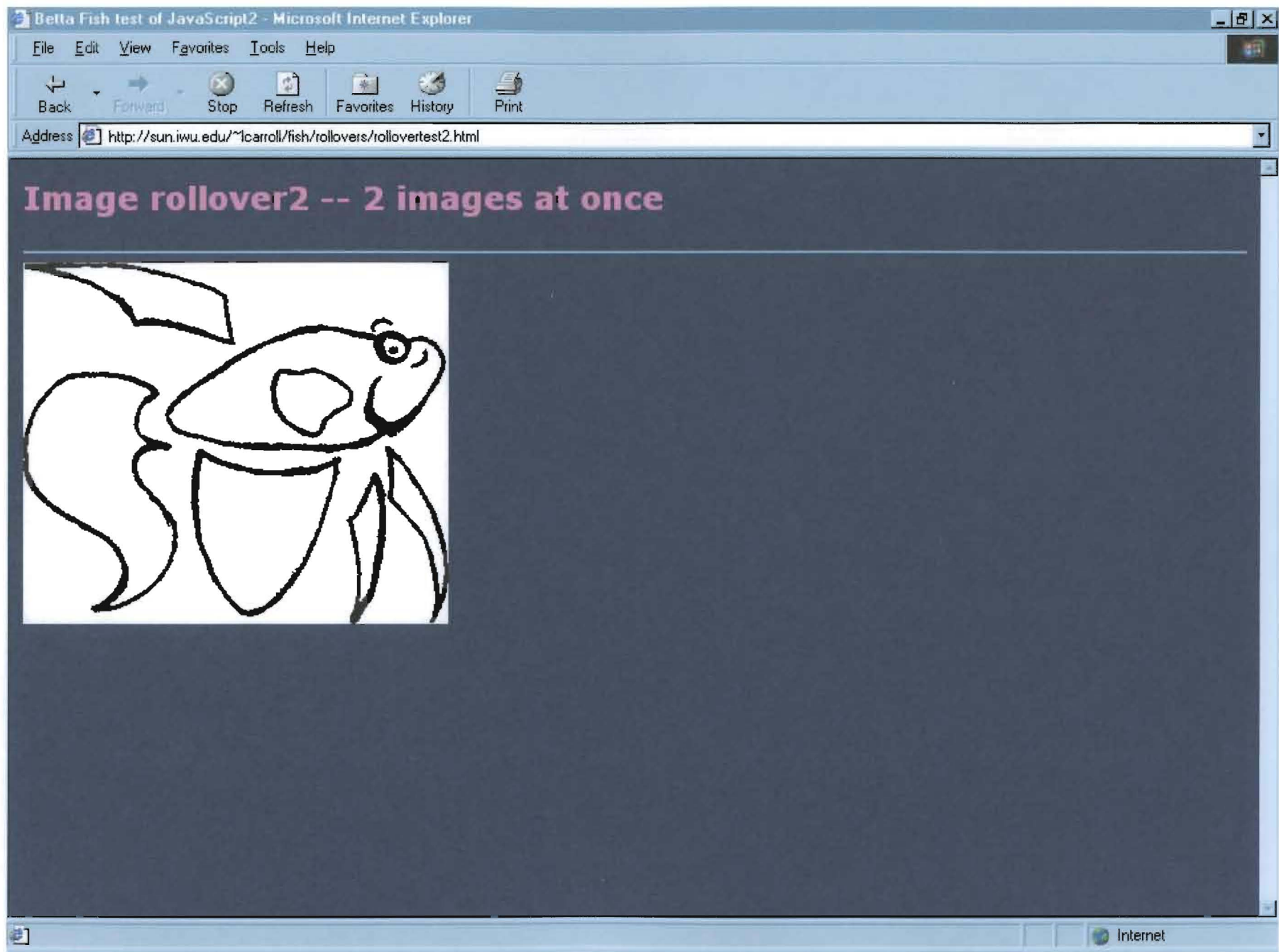
<a href="../stub.html" onMouseOver="imgOn('img1')"
    onMouseOut="imgOff('img1')">
 </a>

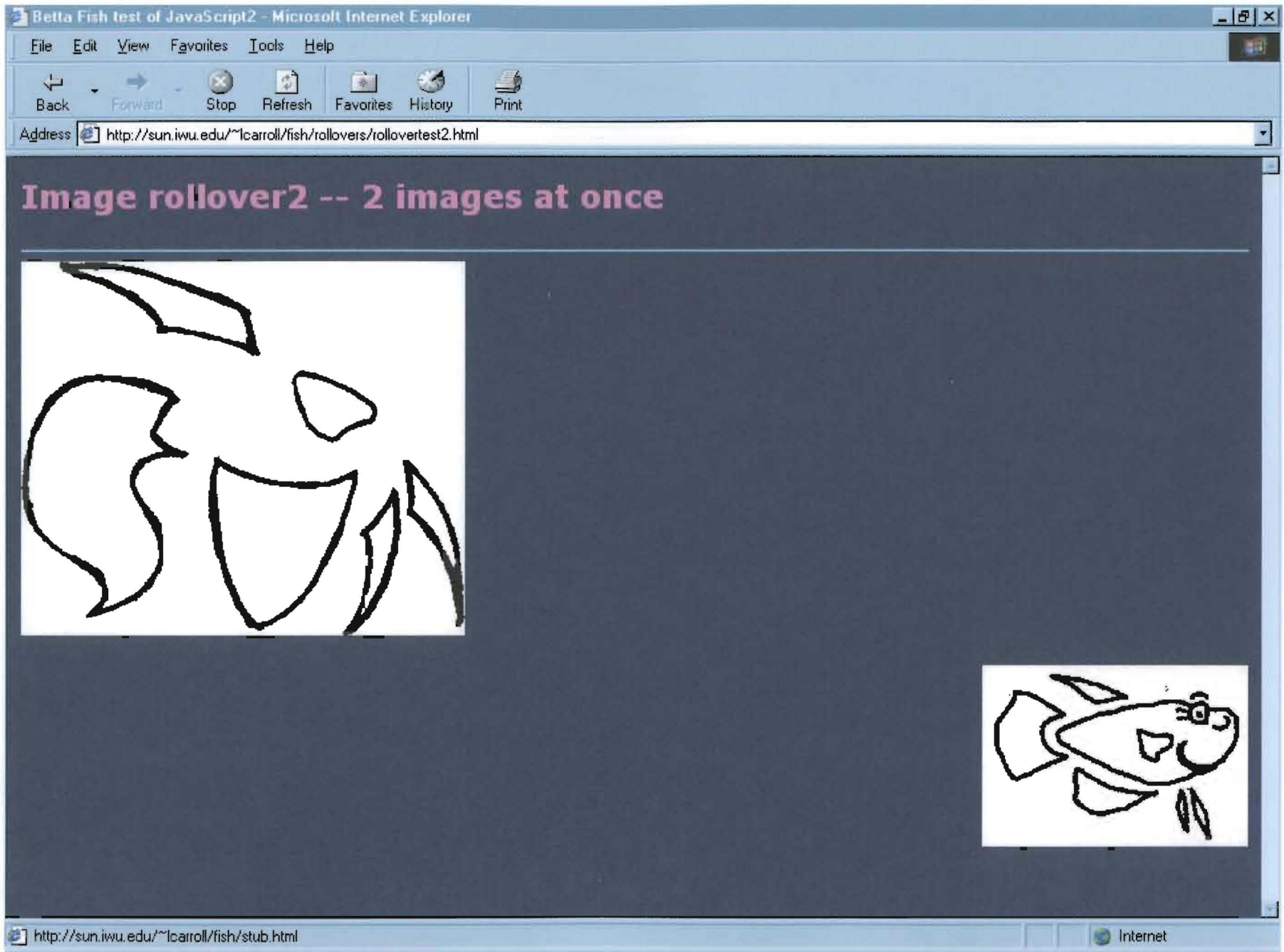
<br><br>
<p align="right"> 
</body>
</html>

```







Having the fish images cycle on an interval proved to be too complex to use in conjunction with other JavaScript functions. I switched to an animated GIF image that did the basically the same thing, but without the timing functions. GIF images also let me make the background of the image transparent, furthering the illusion that the fish was swimming on the page.

Next I wanted to make the image of the fish travel across the page, in addition “swimming.” I experimented with some browser-specific implementations – Netscape’s layers and scaling – but in the end decided on Cascading Style Sheets (CSS), since most browsers had some kind of support for them. I enclosed the fish image in DIV tags. DIV tags tell the web browser that everything enclosed in the tag should be treated as one unit. DIV tags are positioned using CSS and are easier to manipulate than IMG tags alone. Through CSS, JavaScript increments the location of the DIV tags containing the fish, every tenth of a second. This results in the illusion of smooth movement, similar to traditional cel animation.

```

<html>

<head>

<title>

Betta -- stub page
</title>
<script language="JavaScript">
<!--
    function moveLayer(lyr, xadd, yadd, xend, timer) {
        lyr.top += yadd
        lyr.left += xadd
        if(((xadd > 0) && (lyr.left < xend)) ||
            ((xadd < 0) && (lyr.left > xend))) {
            setTimeout('moveLayer(document.layers
["'+lyr.name+'"],'+xadd+', '+yadd+', '+xend+', '+timer+'),' , timer)

        } // end of if
    } // end of function

//-->
</script>

</head>

<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4"
    onload="moveLayer(document.layers['ie_gear'], -4, -4, 4, 20)">
<h2> Moving with layers</h2>
<hr>

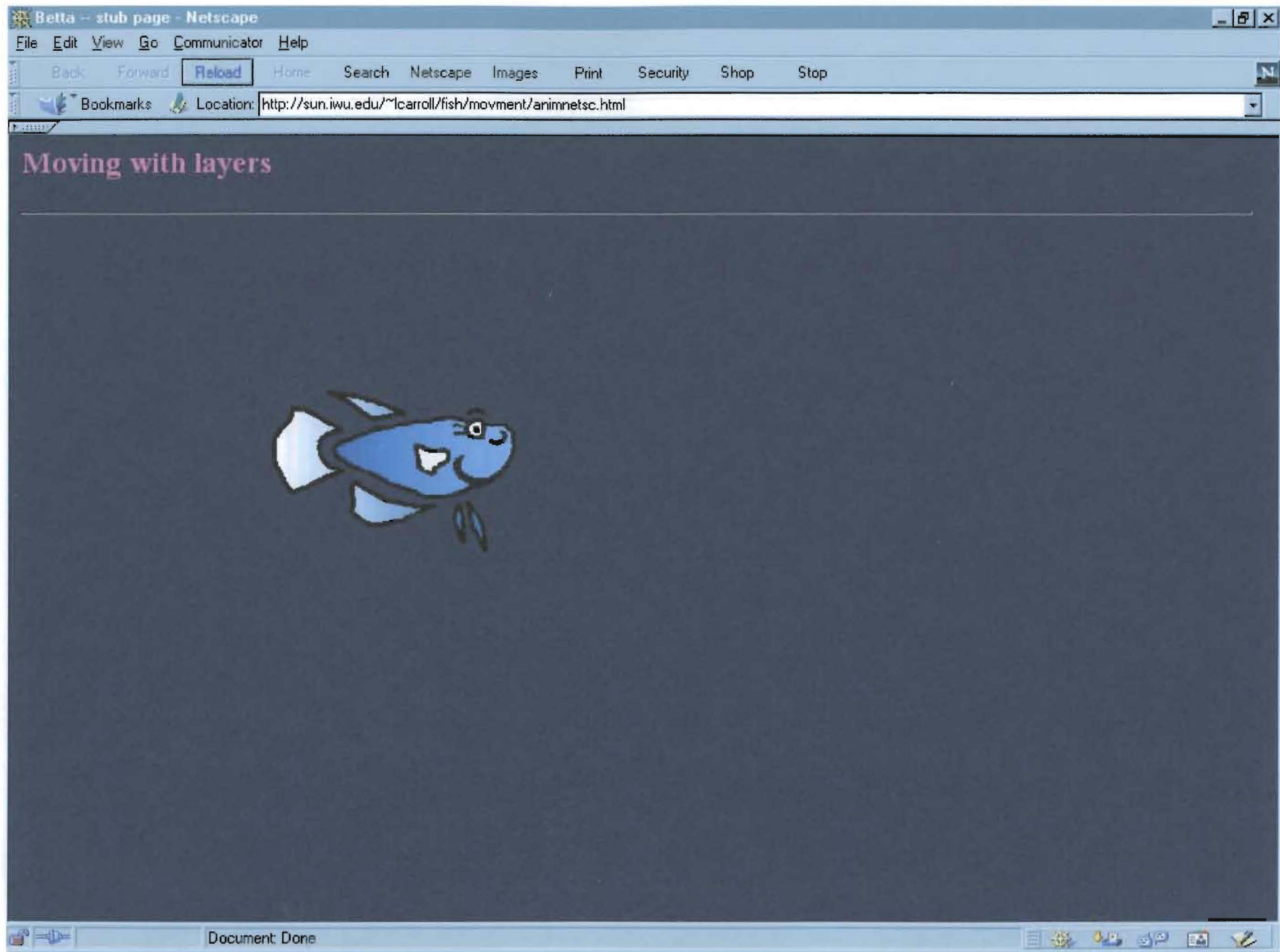
<br>

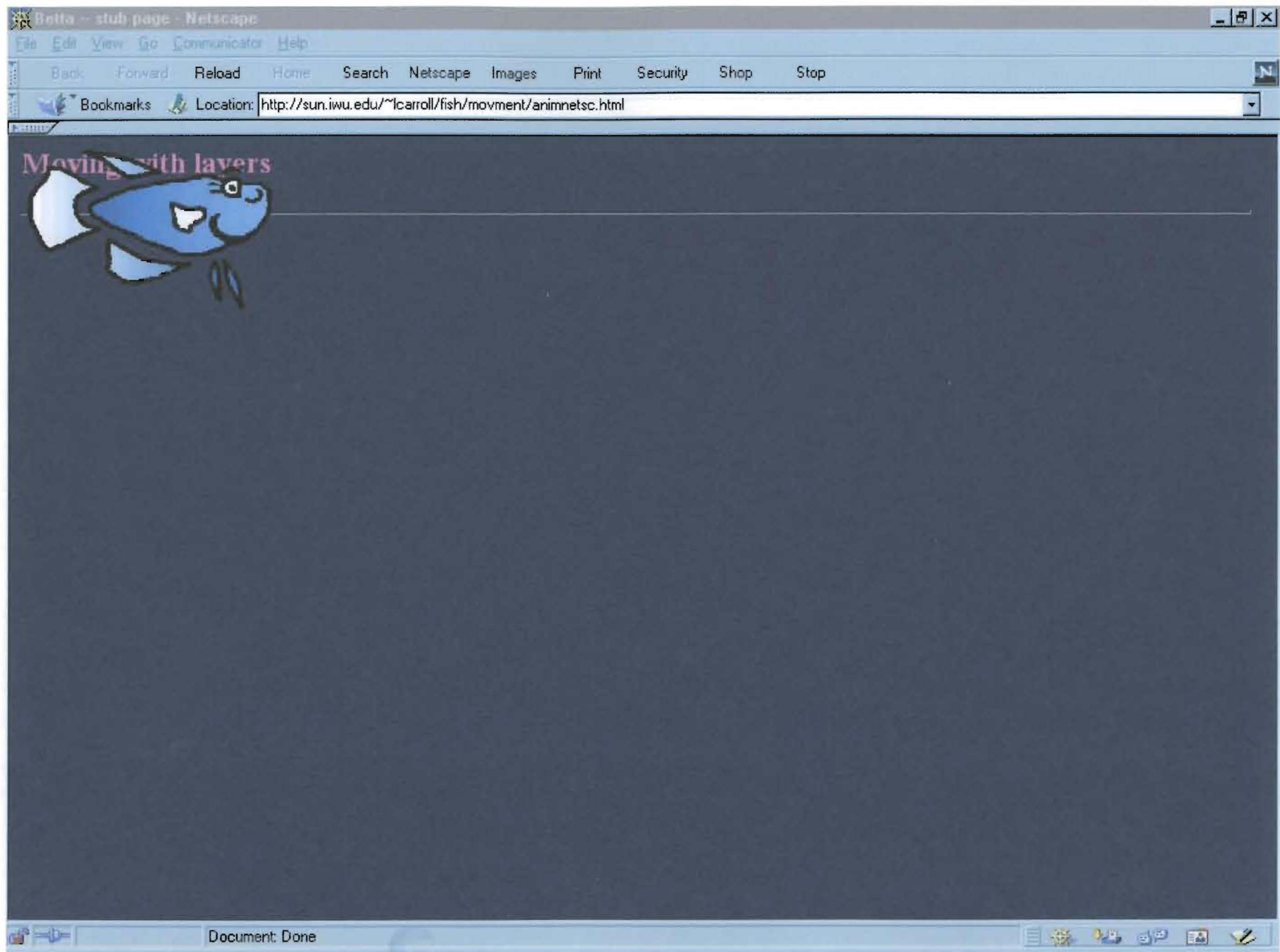
<layer name="ie_gear" top=200 left=200 visibility="show">


</layer>

</body>
</html>

```





```

<html>

<head>

<title>

Betta -- stub page
</title>

<script language="JavaScript">
<!--
    var m=0;

    function scale(){
        m++;
        if(m<30){
            x=window.setTimeout('scale()', 100);
            scaleGraphic(10);
            moveGraphic(5);

        } // end of if
    } // end of function

    function moveGraphic(increment){
        graphic.style.pixelLeft = graphic.style.pixelLeft + increment;
    } //end of function

    function scaleGraphic(increment){
        graphic.width= graphic.width + increment;

    } //end of function

//-->
</script>

<style>

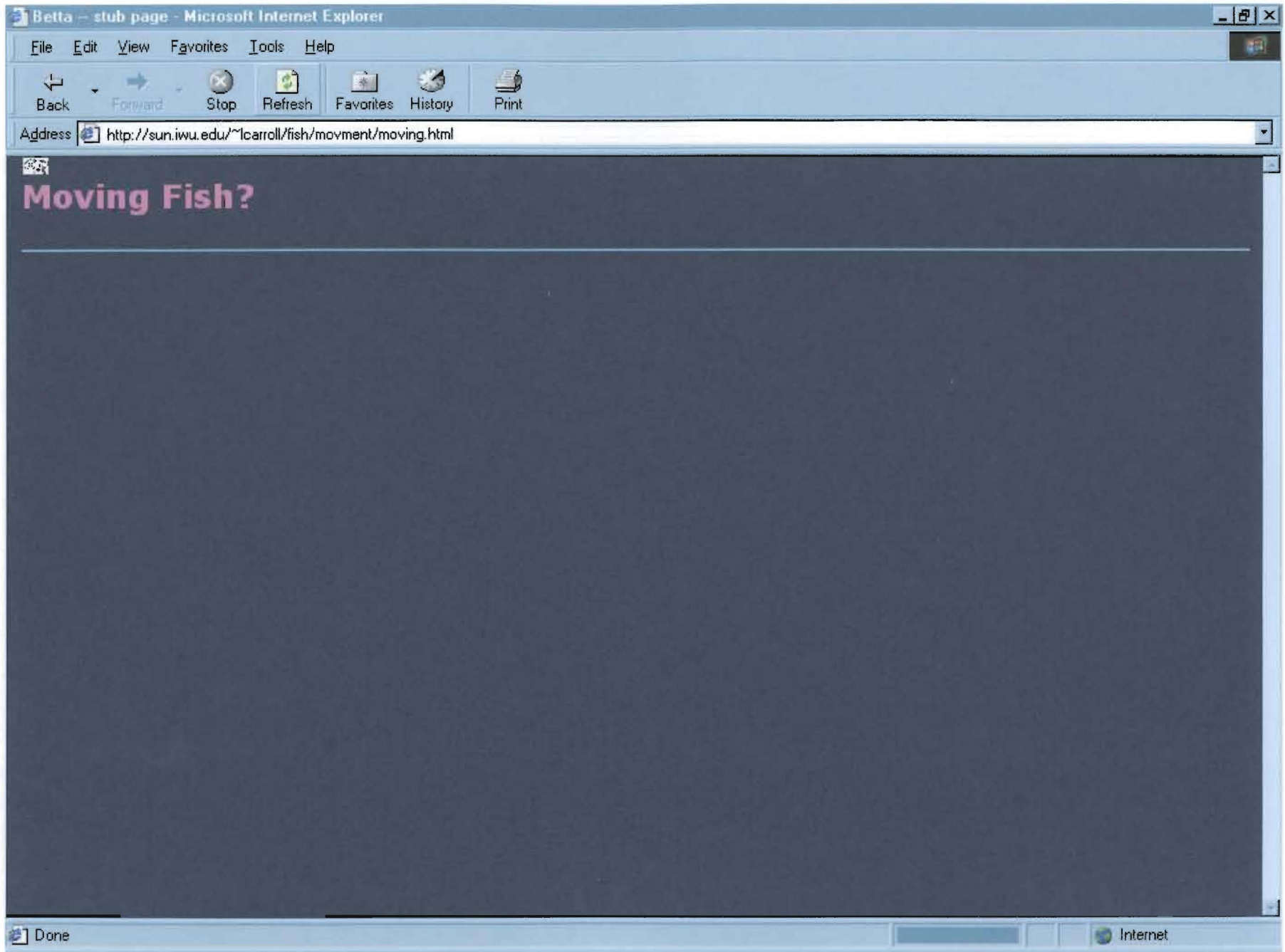
</style>

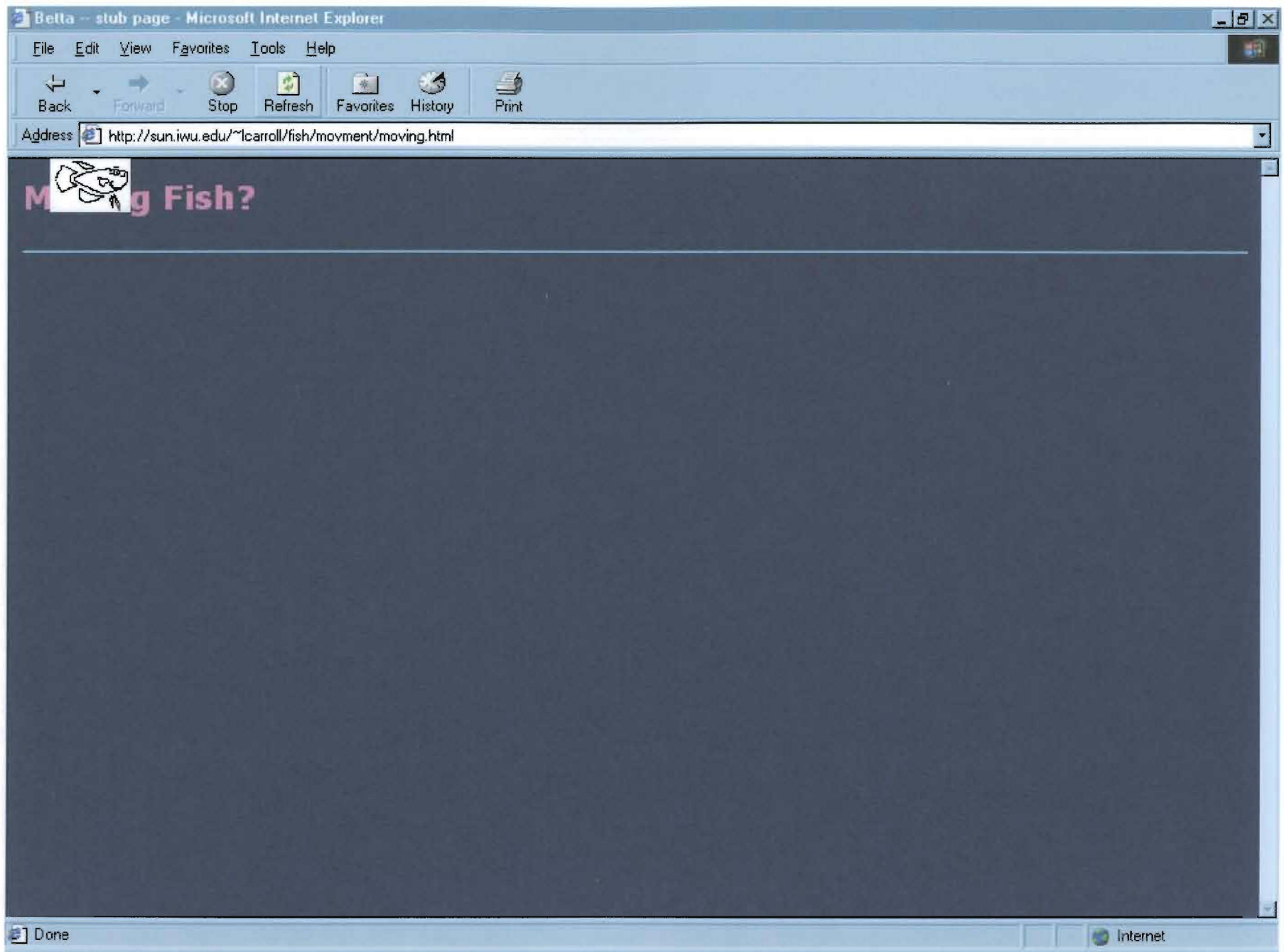
</head>

<body bgcolor="#252130" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4"
onload=scale()>
<h2> Moving Fish? </h2>
<hr>


</body>
</html>

```






```

<html>

<head>
<title> Betta -- stub page </title>
<script language="JavaScript">
<!--
var bName = navigator.appName;
if (bName == "Netscape") ver="n";
    else if(bName == "Microsoft Internet Explorer") ver="ie";

function init(){
    if(ver == "n"){
        pic = document.wholeFish2
        pic.xcor = pic.left
        pic.ycor = pic.top
        } // end of netsc if
    else if(ver == "ie") {
        pic = wholeFish2.style
        pic.xcor = pic.pixelLeft
        pic.ycor = pic.pixelTop
        } //end of ie if

pic.pathcor = 0
pic.active = 1
pic.pathloop = 1
speed = .1
pic.pathx = new Array()
pic.pathy = new Array()
pic.pathx[0] = pic.xcor
pic.pathy[0] = pic.ycor
maxnumbers = 100
increment = 4
counter = 1

for(var i=0; i< maxnumbers; i++){
    if(i< (maxnumbers/2) ){
        pic.pathx[i+1] = pic.pathx[i]+increment
        pic.pathy[i+1] = pic.pathy[i]+increment
        }
    else{
        pic.pathx[i+1] = pic.pathx[i]-increment
        pic.pathy[i+1] = pic.pathy[i]-increment
        }
    }

picpath()
} //end of function

function picpath() {
    if(pic.active && pic.pathcor< pic.pathx.length) {
        pic.xcor = pic.pathx[pic.pathcor]
        pic.ycor = pic.pathy[pic.pathcor]
        pic.left = pic.xcor
        pic.top = pic.ycor
        pic.pathcor+=1
        setTimeout("picpath()", speed*1000)
        } // end of picpath if
    else {

```

```

        if(pic.active && pic.pathloop) {
            pic.pathcor=0
            picpath()
        } // end of inner if
        else pic.active=0
        } // end of else
    } //end of function

function reset() {
    pic.pathcor = maxnumbers;
    pic.pathloop = 0;
    pic.active = 1;

    picpath();

    pic.active= 0;
    pic.pathcor = 0;
    pic.pathloop = 1;

} //endof function
//-->
</script>

<style type="text/css">
<!--
.mtopFin {
    overflow: hidden;
    position: absolute;
    visibility: visible;
    z-index: auto;
    height: auto;
    width: auto;
    left: 30px;
    top: 63px;
}
.mlegFin {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 230px;
    top: 214px;
}
.mbodyFin {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 135px;
    top: 210px;
}
.mfishBody {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 113px;
    top: 120px;
}

```

```

}

.mtailFin {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 4px;
    top: 150px;
}

#wholeFish2 {position:absolute;left:100px;top:37px;
width:357px;height:395px;}
.wholeFish2 {position:absolute;left:100px;top:114px;
width:300px;height:300px;}

-->
</style>
</head>

<body bgcolor="#48BCB4" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4"
onLoad="init()" >
<br>
<form name="testBox">
<input type = "text" name="checker" value=" ">
</form>
<p><br>

<div id="wholeFish2" >
    <p>
    
    </p>
    <p>
    </p>
</div>

</body>
</html>

```

```

p<html>

<head>

<title>

Betta -- stub page
</title>
<script language="JavaScript">
<!--
    bName = navigator.appName;

    if (bName == "Netscape") ver="n";
        else if(bName == "Microsoft Internet Explorer") ver="ie";

function init(){
    if(ver == "n"){
        pic = document.picDiv
        pic.xcor = pic.left
        pic.ycor = pic.top
        } // end of netsc if
    else if(ver == "ie") {
        pic = picDiv.style
        pic.xcor = pic.pixelLeft
        pic.ycor = pic.pixelTop
        }//end of ie if

    pic.pathcor = 0
    pic.active = 1
    pic.pathloop = 1
    speed = .1
    pic.pathx = new Array()
    pic.pathy = new Array()
    pic.pathx[0] = 60
    pic.pathy[0] = 60
    //xend = 0
    maxnumbers = 100
    increment = 4
    counter = 1

for(var i=0; i< maxnumbers; i++){
    if(i< (maxnumbers/2) ){
        pic.pathx[i+1] = pic.pathx[i]+increment
        pic.pathy[i+1] = pic.pathy[i]+increment
        }
    else{
        pic.pathx[i+1] = pic.pathx[i]-increment
        pic.pathy[i+1] = pic.pathy[i]-increment
        }
    }

picpath()
} //end of function

function picpath() {
    if(pic.active && pic.pathcor< pic.pathx.length) {
        pic.xcor = pic.pathx[pic.pathcor]
        pic.ycor = pic.pathy[pic.pathcor]
        pic.left = pic.xcor
    }
}

```

```

    pic.top = pic.ycor
    pic.pathcor+=1
    setTimeout("picpath()", speed*1000)
  } // end of picpath if
else {

  if(pic.active && pic.pathloop) {
    pic.pathcor=0
    picpath()
  } // end of inner if
  else pic.active=0
  } // end of else
} //end of function

//-->
</script>
<style type="text/css">
<!--
  #picDiv {position:absolute; left:60; top:60; width:200; height:200;}
-->
</style>

</head>

<body bgcolor="#48BCB4" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4"
  onload="init()">
<h2> Stub</h2>
<hr>

<div id="picDiv"> 
</div>

</body>
</html>

```

```

<html>

<head>

<title>

Betta -- stub page
</title>
<script language="JavaScript">
<!--
    bName = navigator.appName;

    if (bName == "Netscape") ver="n";
        else if(bName == "Microsoft Internet Explorer") ver="ie";

function init(){
    if(ver == "n"){
        pic = document.picDiv
        pic.xcor = pic.left
        pic.ycor = pic.top
        } // end of netsc if
    else if(ver == "ie") {
        pic = picDiv.style
        pic.xcor = pic.pixelLeft
        pic.ycor = pic.pixelTop
        }//end of ie if

pic.pathcor = 0
pic.active = 1
pic.pathloop = 1
speed = .1
pic.pathx = new Array()
pic.pathy = new Array()
pic.pathx[0] = 60
pic.pathy[0] = 60
//xend = 0
maxnumbers = 100
increment = 4
counter = 1

for(var i=0; i< maxnumbers; i++){
    if(i< (maxnumbers/2) ){
        pic.pathx[i+1] = pic.pathx[i]+increment
        pic.pathy[i+1] = pic.pathy[i]+increment
        }
    else{
        pic.pathx[i+1] = pic.pathx[i]-increment
        pic.pathy[i+1] = pic.pathy[i]-increment
        }
    }

picpath()
} //end of function

function picpath() {
    if(pic.active && pic.pathcor< pic.pathx.length) {
        if (counter == 1){
            pic.xcor = pic.pathx[pic.pathcor]
            pic.ycor = pic.pathy[pic.pathcor]

```

```

pic.left = pic.xcor
pic.top = pic.ycor
pic.pathcor+=1
setTimeout("picpath()", speed*1000)
}
else if(counter == 2){ //x
pic.xcor = pic.pathx[pic.pathcor]
pic.left = pic.xcor
pic.pathcor+=1
setTimeout("picpath()", speed*1000)
}
else if (counter==3){ //y
pic.ycor = pic.pathy[pic.pathcor]
pic.top = pic.ycor
pic.pathcor+=1
setTimeout("picpath()", speed*1000)
}
else {counter == 1;
      pic.pathcor+=1
      setTimeout("picpath()", speed*1000)
}
} // end of picpath if
else {

```

```

    if(pic.active && pic.pathloop) {
        pic.pathcor=0
        picpath()
        counter++
        }// end of inner if
    else pic.active=0
    } // end of else
} //end of function

```

```

//-->
</script>
<style type="text/css">
<!--
    #picDiv {position:absolute; left:60; top:60; width:200; height:200;}
-->
</style>

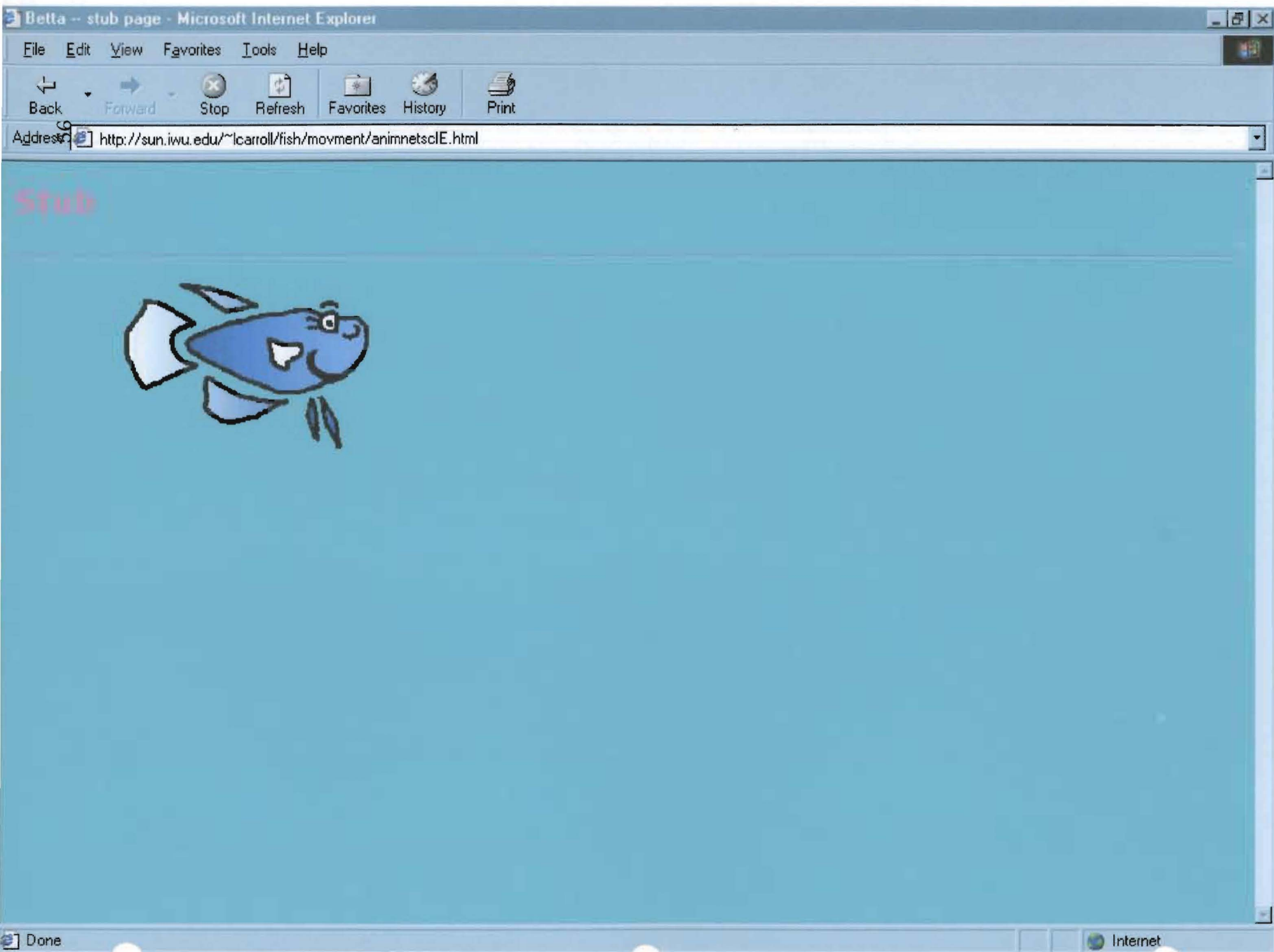
</head>

<body bgcolor="#48BCB4" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4"
    onload="init()">
<h2> Stub</h2>
<hr>

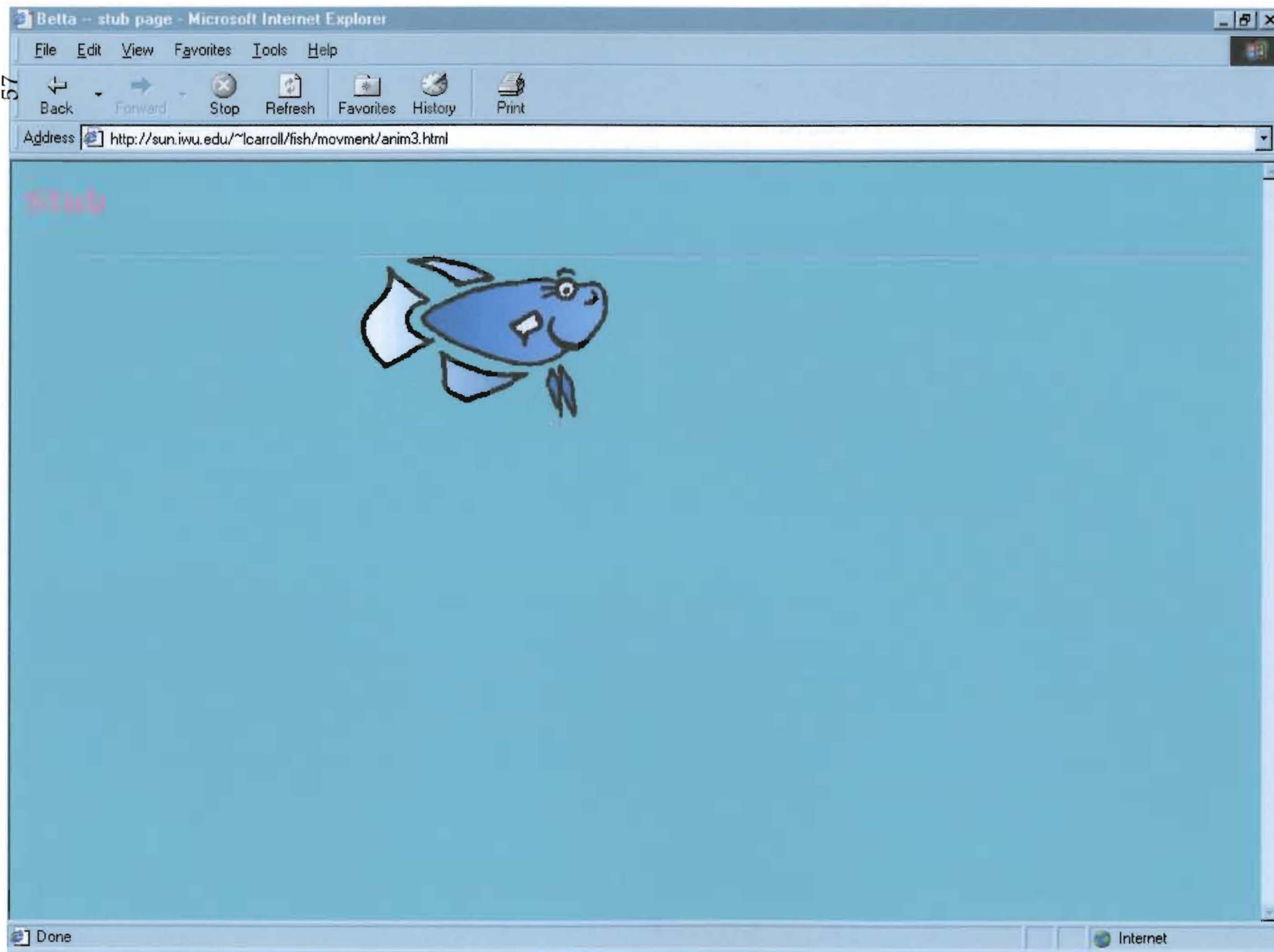
<div id="picDiv"> 
</div>

</body>
</html>

```



57



I wanted the user to be able to choose the color of the fish's fins, but to choose each fin separately, the fins had to be separate images. I sectioned my fish image into pieces and saved several versions of each fin in different colors. I used CSS to put the fish back together on the web page, since CSS allows for very precise control and let me overlap the square GIF images to give the illusion of a whole fish. The pieces I enclosed in a DIV tag so they could be moved in concert. I used JavaScript similar to the roll-over function to replace the current piece of the fish with one chosen by the user from a selection list in another frame.

I started with three colors, red, blue, green, and the outline of the fish. Once that was working, I incorporated the movement function to make the fish move across the page. Then I adjusted the image replacement function to switch an animated GIF of the fish's body for the current static image whenever the user pushed the "begin motion" button. The fish now appeared to swim and move across the page at the same time.

I added a fourth color, purple, and a female version of the fish. The user chose which fish they wanted to change by a selection list and JavaScript immediately updated the images. This JavaScript function controlled through CSS the visibility of two DIV tags – one containing the pieces of the male fish and one containing the pieces of the female fish. These two DIV tags were enclosed in another DIV tag, whose location was controlled by the movement function.

Finally I added a popup window to the main frameset page to contain the instructions for the fish.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript" type="text/JavaScript">

</script>
</head>

<frameset rows="*" cols="140,*" framespacing="0" frameborder="yes"
border="0" bordercolor="#CCCCCC">
  <frame src="complex2.html" name="leftFrame" scrolling="yes" noresize>
    <frame src="pic2.html" name="mainFrame">
  </frameset>
</frameset>
<noframes><body>
sorry, you need to have frames
</body></noframes>
</html>

```

```

<html>

<head>
<title> Betta -- stub page </title>
<script language="JavaScript">
<!--
  //-->
</script>

<style type="text/css">
<!--
.mtopFin {
    overflow: hidden;
    position: absolute;
    visibility: visible;
    z-index: auto;
    height: auto;
    width: auto;
    left: 30px;
    top: 63px;
}
.mlegFin {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 230px;
    top: 214px;
}
.mbodyFin {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 135px;
    top: 210px;
}
.mfishBody {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 113px;
    top: 120px;
}

.mtailFin {
    overflow: hidden;
    visibility: visible;
    position: absolute;
    height: auto;
    width: auto;
    left: 4px;
    top: 150px;
}

```

```

#wholeFish2 {position:absolute;left:100px;top:37px;
width:357px;height:395px;}
.wholeFish2 {position:absolute;left:100px;top:114px;
width:300px;height:300px;}

-->
</style>
</head>

<body bgcolor="#48BCB4" text="#CE70A1" vlink="#A89CCC" link="#7EC9F4">
<br>
<form name="testBox">
<input type = "text" name="checker" value=" ">
</form>
<p><br>

<div id="wholeFish2" >
  <p>
    
      </p>
  <p>
    </p>
</div>

</body>
</html>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Untitled Document</title>
<script language="Javascript">

var MaleGIFs = new Array(
    new Array("noclrbody1.gif", "noclrtaill1.gif",
"noclrlegs1.gif", "noclrtopfin1.gif", "noclrbodyfin1.gif"),
    new Array("redbody1.gif", "redtaill1.gif",
"redlegs1.gif", "redtopfin1.gif", "redbodyfin1.gif"),
    new Array("bluebody1.gif", "bluetail1.gif",
"bluelegs1.gif", "bluetopfin1.gif", "bluebodyfin1.gif"),
    new Array("greenbody1.gif", "greentaill1.gif",
"greenlegs1.gif", "greentopfin1.gif", "greenbodyfin1.gif"));

var maleAnims = new Array("mPurpleAnim.gif", "mRedAnim.gif",
"mBlueBodyAnim.gif", "mGreenAnim.gif");
var fishAnims= new Array();

var fishpics = new Array(new Array(), new Array(), new Array(), new
Array());
//must have a "new Array()" in this for each one above

var pictNo = 0; // global picture counter maleGIFs.length maleGIFs
[i].length

for (var i = 0; i < MaleGIFs.length; i++) {
    fishAnims[i] = new Image();
    fishAnims[i].src = maleAnims[i];
    for (var j = 0; j < 5; j++) {
        //stuff pics into array and preload
        fishpics[i][j] = new Image();
        fishpics[i][j].src = MaleGIFs[i][j];
    } //end of outer for
} //end of inner for

var male = 1; //fish gender 0 is fem, 1 is male

function changeFish(form){
    //alert("I'm in change fish222222");
    // alert("Index: " + fishForm.color.selectedIndex);
    // alert("Index 2: " + fishForm.part.selectedIndex);
    var fishColor = fishForm.color.selectedIndex;
    var fishPart = fishForm.part.selectedIndex;
    // alert("Color: " + fishColor);
    // alert("Part: " + fishPart);

    switch (fishPart)
    {
        case 0: parent.frames[1].mfishBody.src = fishpics[fishColor]
[0].src; break;
        case 1: parent.frames[1].mfishTail.src = fishpics[fishColor]
[1].src; break;
        case 2: parent.frames[1].mlegFin.src = fishpics[fishColor]
[2].src; break;
        case 3: parent.frames[1].mtopFin.src = fishpics[fishColor]
[3].src; break;
        case 4: parent.frames[1].mbodyFin.src = fishpics[fishColor]
[4].src; break;
    }
}

```

```

        case 5: parent.frames[1].mfishBody.src = fishpics[fishColor]
[0].src;
        parent.frames[1].mfishTail.src = fishpics[fishColor][1].src;
        parent.frames[1].mlegFin.src = fishpics[fishColor][2].src;
        parent.frames[1].mtopFin.src = fishpics[fishColor][3].src;
        parent.frames[1].mbodyFin.src = fishpics[fishColor][4].src;
break;
    } // end of switch
    parent.frames[1].document.forms[0].checker.value = "Color: " +
fishColor + "Body: " + fishPart;

} //end of function

function startMoving(){
var fishColor = fishForm.color.selectedIndex;
var fishPart = fishForm.part.selectedIndex;

switch (fishPart)
{
    case 0: parent.frames[1].mfishBody.src = fishAnims
[fishColor].src; break;
    } // end of switch
} // end of function

//-->
</script>
</head>

<body bgcolor="#48BCB4">
<form NAME = "fishForm">
    <p>
        <I>Pick Color</I>
        <select name="color">
            <option value="noclr">Colorless!
            <option value="red" >Red
            <option value="blue">Blue
            <option value="green">Green
        </select>
    </p>
    <p>
        <I>Pick Body Part</I>

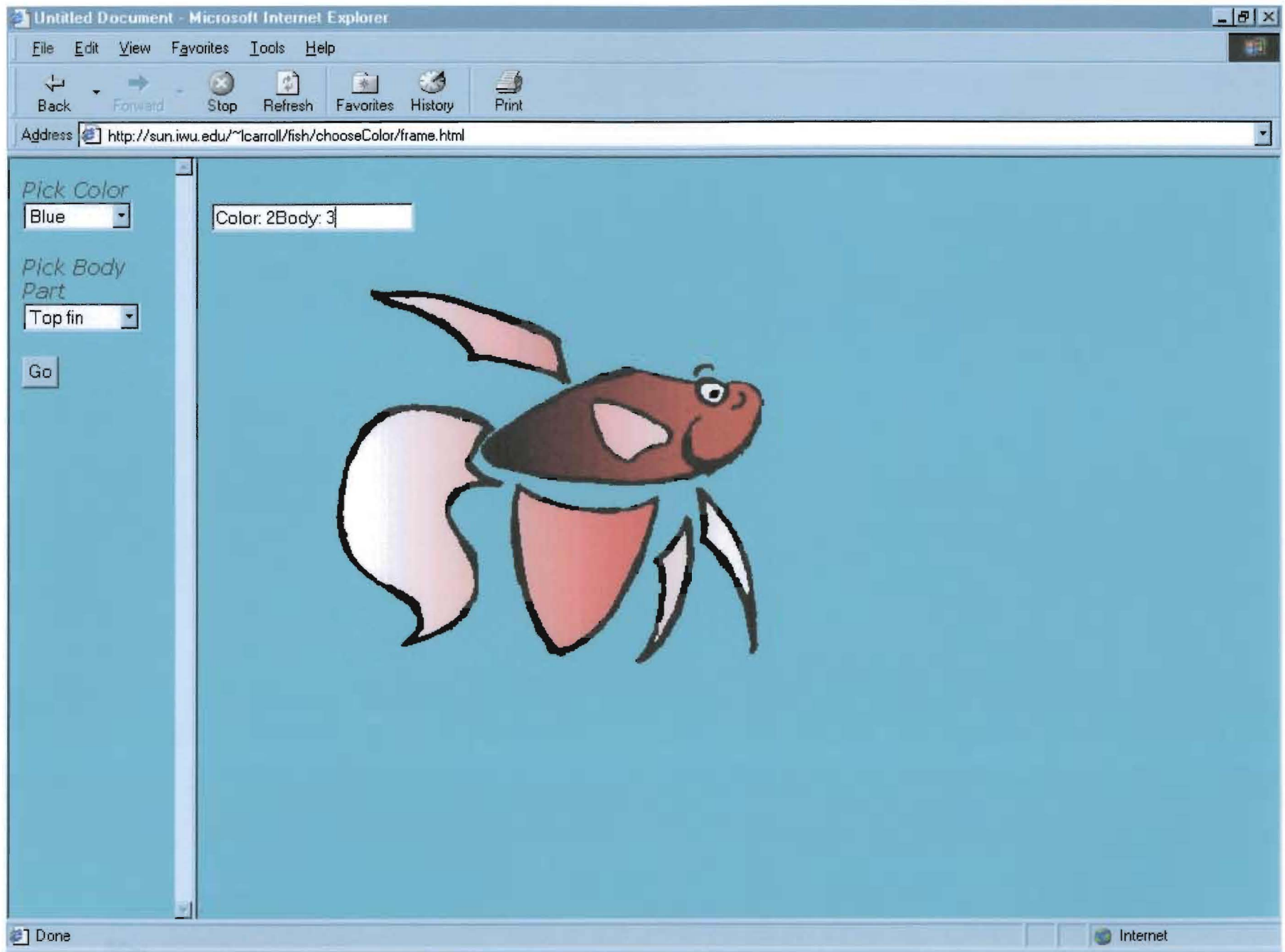
        <select name="part">
            <option value="body">Body
            <option value = "tail"> Tail
            <option value = "legs"> "Leg" fins
            <option value = "topfin"> Top fin
            <option value = "bodyfin"> Bottom fin
            <option value = "all"> Everything!
        </select>
    </p>
    <p>
        <INPUT TYPE = "button" NAME = "PickFish" VALUE = "Go" onClick =
"changeFish(this.form);return true">
    </p>
</form>

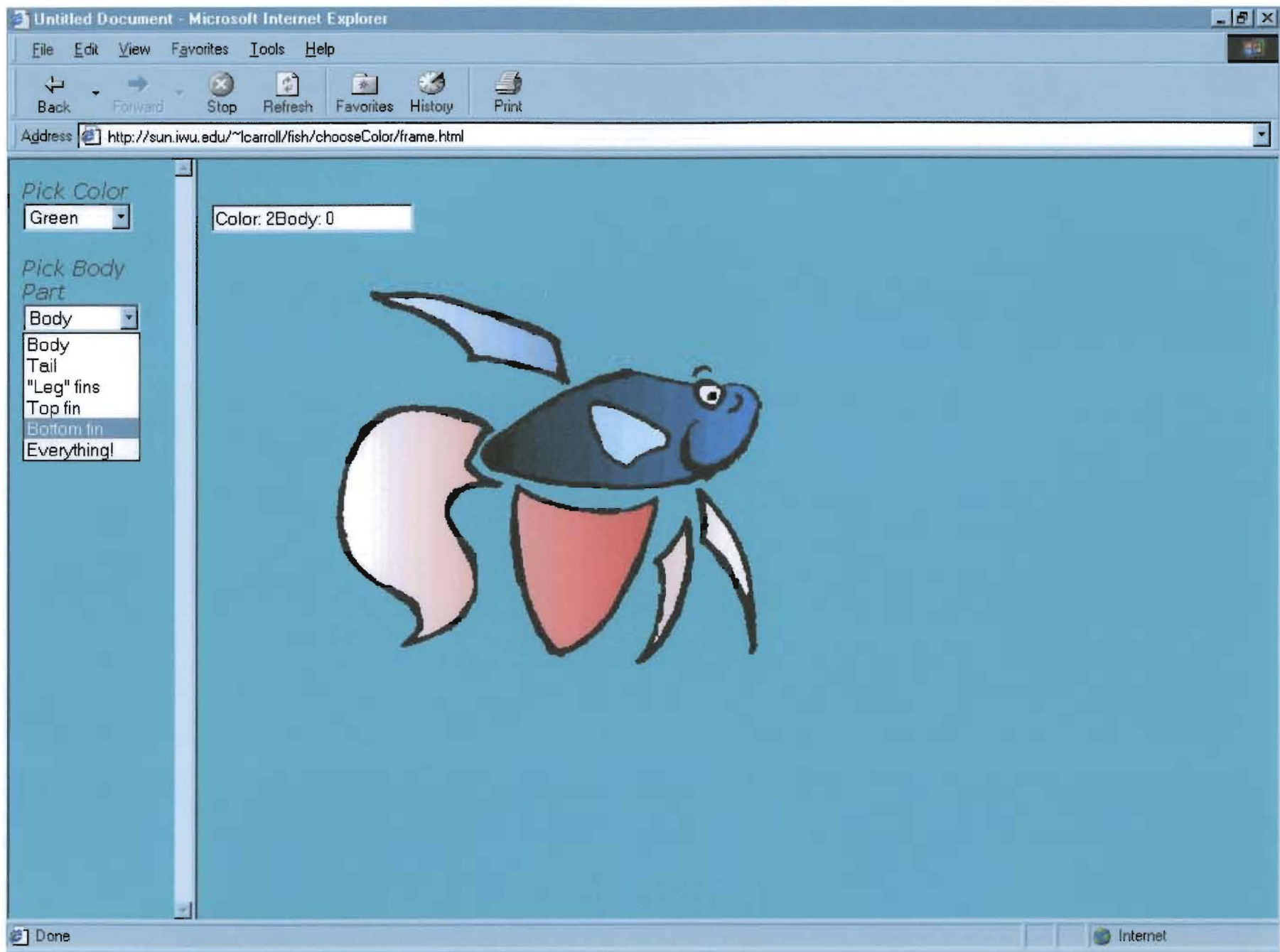
<br><br>
Use these buttons to make the fish move
<form name="ctrlMove">
<INPUT TYPE = "button" NAME = "start" VALUE = "Start" onClick =

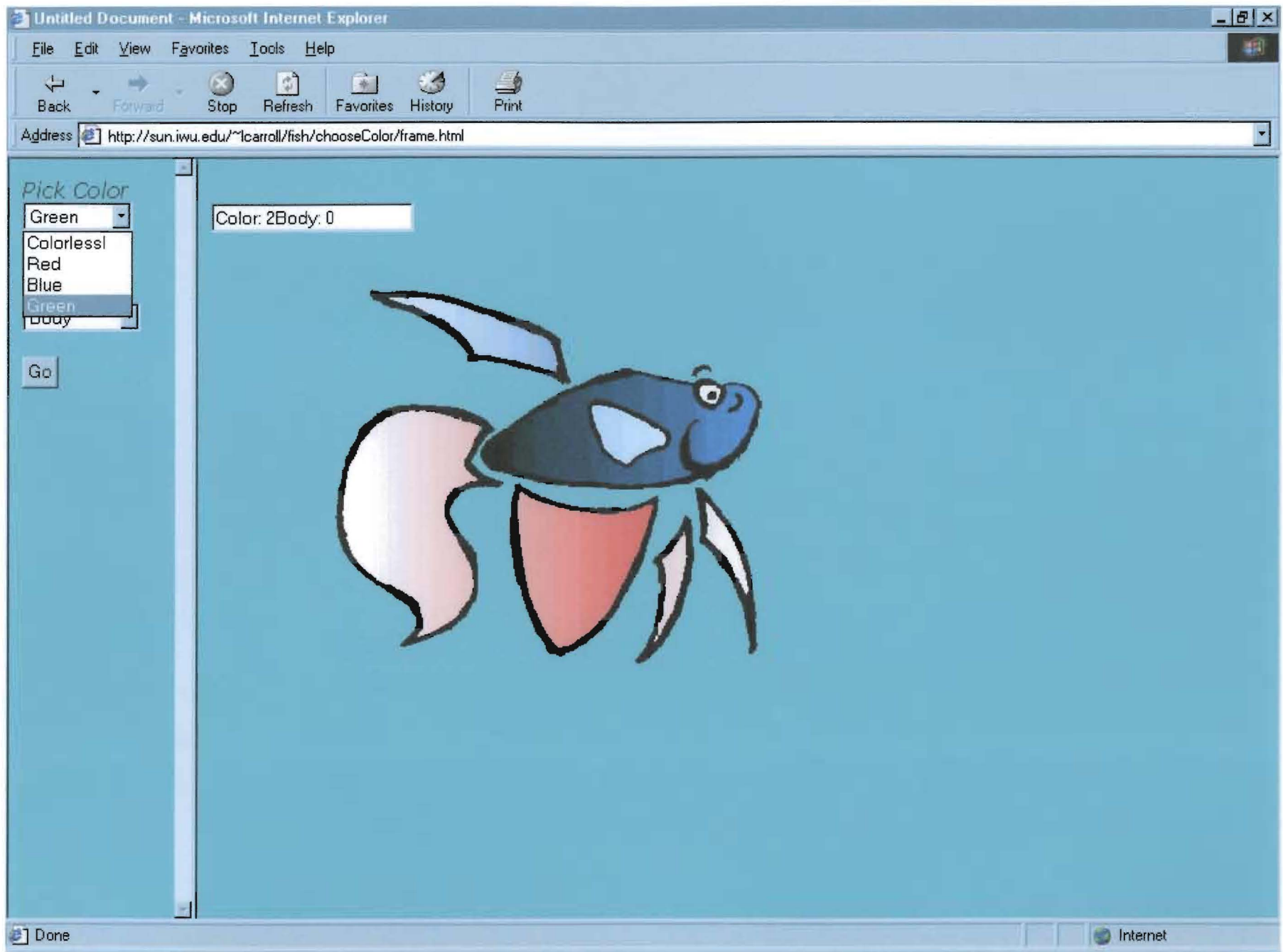
```

```
"parent.frames[1].pic.active=1; parent.frames[1].picpath();return true">
<INPUT TYPE = "button" NAME = "stop" VALUE = "Stop" onClick =
"parent.frames[1].pic.active=0;return true">
<INPUT TYPE = "button" NAME = "reset" VALUE = "Reset" onClick =
"parent.frames[1].reset();return true">

</form>
</body>
</html>
```





```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Untitled Document</title>
<script language="Javascript">

var MaleGIFs = new Array(
    new Array("noclrbody1.gif", "noclrtaill1.gif",
"noclrlegs1.gif", "noclrtopfin1.gif", "noclrbodyfin1.gif"),
    new Array("redbody1.gif", "redtaill1.gif",
"redlegs1.gif", "redtopfin1.gif", "redbodyfin1.gif"),
    new Array("bluebody1.gif", "bluetaill1.gif",
"bluelegs1.gif", "bluetopfin1.gif", "bluebodyfin1.gif"),
    new Array("greenbody1.gif", "greentaill1.gif",
"greenlegs1.gif", "greentopfin1.gif", "greenbodyfin1.gif"));

var fishpics = new Array(new Array(), new Array(), new Array(), new
Array());
//must have a "new Array()" in this for each one above

var pictNo = 0; // global picture counter maleGIFs.length maleGIFs
[i].length

for (var i = 0; i < MaleGIFs.length; i++) {
    for (var j = 0; j < 5; j++) {
        //stuff pics into array and preload
        fishpics[i][j] = new Image();
        fishpics[i][j].src = MaleGIFs[i][j];
    } //end of outer for
} //end of inner for

var male = 1; //fish gender 0 is fem, 1 is male

function changeFish(form){
    //alert("I'm in change fish222222");
    // alert("Index: " + fishForm.color.selectedIndex);
    // alert("Index 2: " + fishForm.part.selectedIndex);
    var fishColor = fishForm.color.selectedIndex;
    var fishPart = fishForm.part.selectedIndex;
    // alert("Color: " + fishColor);
    // alert("Part: " + fishPart);

    switch (fishPart)
    {
        case 0: parent.frames[1].mfishBody.src = fishpics[fishColor]
[0].src; break;
        case 1: parent.frames[1].mfishTail.src = fishpics[fishColor]
[1].src; break;
        case 2: parent.frames[1].mlegFin.src = fishpics[fishColor]
[2].src; break;
        case 3: parent.frames[1].mtopFin.src = fishpics[fishColor]
[3].src; break;
        case 4: parent.frames[1].mbodyFin.src = fishpics[fishColor]
[4].src; break;
        case 5: parent.frames[1].mfishBody.src = fishpics[fishColor]
[0].src;
            parent.frames[1].mfishTail.src = fishpics[fishColor][1].src;
            parent.frames[1].mlegFin.src = fishpics[fishColor][2].src;
            parent.frames[1].mtopFin.src = fishpics[fishColor][3].src;
            parent.frames[1].mbodyFin.src = fishpics[fishColor][4].src;
    }
}

```

```

break;
    }
    parent.frames[1].document.forms[0].checker.value = "Color: " +
fishColor + "Body: " + fishPart;

} //end of function

//-->
</script>
</head>

<body bgcolor="#48BCB4">
<form NAME = "fishForm">
    <p>
        <I>Pick Color</I>
        <select name="color">
            <option value="noclr">Colorless!
            <option value="red" >Red
            <option value="blue">Blue
            <option value="green">Green
        </select>
    </p>
    <p>
        <I>Pick Body Part</I>

        <select name="part">
            <option value="body">Body
            <option value = "tail"> Tail
            <option value = "legs"> "Leg" fins
            <option value = "topfin"> Top fin
            <option value = "bodyfin"> Bottom fin
            <option value = "all"> Everything!
        </select>
    </p>
    <p>
        <INPUT TYPE = "button" NAME = "PickFish" VALUE = "Go" onClick =
"changeFish(this.form);return true">
    </p>
</form>
</body>
</html>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Untitled Document</title>
<script language="JavaScript" >
<!--
/* written by Lauren Carroll
April 2003
*/

//make a popup with instructions for fish, to load when page first loads
function openNewWindow() {
NewWindow=window.open("new.html", "awindow", "width=350, height=300,
toolbar=0,menubar=0, scrollbars=0,resizable=0, status=0,location=0,left=
30,top=30");
}
//-->
</script>
</head>

<frameset rows="*" cols="140,*" framespacing="0" frameborder="yes"
border="0" bordercolor="#CCCCCC" onload="openNewWindow();">

    <frame src="complex8.html" name="leftFrame" scrolling="yes"
noresize>

    <frame src="pic6.html" name="mainFrame">
</frameset>

<noframes>
<body>
sorry, you need to have frames
</body>
</noframes>
</html>

```

```
<html>

<head>

<title>

Instructions for the Virtual Betta
</title>
<script language="JavaScript">
<!--
/* written by Lauren Carroll
April 2003
*/

//-->
</script>

</head>

<body bgcolor="#48BCB4">
<font size="-1">
Hi! This is the Virtual Betta, an online,
interactive, betta fish by Laurie Carroll.
<br><br>

Instructions: <br>

<ul>
<li> The buttons down the lefthand side of
      the screen control the fish and
      change its color. <br>
<li> Select the part you want to change
      and the color you would like it
      to be and then click the 'Go'
      button to view your changes<br>
<li> Please bring the fish to a full and
      complete stop before changing
      its color or gender.<br>

</ul>

<br><br><br>Enjoy!

</font>
</body>
</html>
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Virtual Betta -- picture page</TITLE>

<SCRIPT language=JavaScript>
<!--
/* written by Lauren Carroll
April 2003
*/

var bName = navigator.appName; //checks for browser type
if (bName == "Netscape") ver="n";
    else if(bName == "Microsoft Internet Explorer") ver="ie";

function init(){

    //sets pic to browser-specific CSS controls
    if(ver == "n"){ //netscape
        pic = document.wholeFish2
        pic.xcor = pic.left
        pic.ycor = pic.top
    } // end of netscape if
    else if(ver == "ie") { //internet explorer
        pic = wholeFish2.style
        pic.xcor = pic.pixelLeft
        pic.ycor = pic.pixelTop
    } //end of ie if

    pic.pathcor = 0 //counter to keep track of where in array pic is
    pic.active = 0 //tells whether or not movement is activated
    pic.pathloop = 1 //tells whether or not to loop the motion function
    speed = .1 //increment for timer so that it's easier to use
    pic.pathx = new Array() // array of x-coords for pic
    pic.pathy = new Array() //array of y-coords for pic
    pic.pathx[0] = pic.xcor
    pic.pathy[0] = pic.ycor
    maxnumbers = 100 //max numbers for the arrays
    increment = 4 //how much to add to each number when making arrays

    for(var i=0; i< maxnumbers; i++){ //put numbers into coordinate arrays
        if(i< (maxnumbers/2) ){
            pic.pathx[i+1] = pic.pathx[i]+increment
            pic.pathy[i+1] = pic.pathy[i]+increment
        }
        else{
            pic.pathx[i+1] = pic.pathx[i]-increment
            pic.pathy[i+1] = pic.pathy[i]-increment
        }
    }

    picpath() //call movement function to start moving
    } //end of function

    function picpath() { //moving function

        if(pic.active && pic.pathcor< pic.pathx.length) {
            //if moving is activated and the current coordinate is less than
the

```



```

//the length of the entire coordinate array
pic.xcor = pic.pathx[pic.pathcor]
pic.ycor = pic.pathy[pic.pathcor]
pic.left = pic.xcor
pic.top = pic.ycor
pic.pathcor+=1
//call function for the next movement, but wait for a bit first
setTimeout("picpath()", speed*1000)
} // end of picpath if
else {

    if(pic.active && pic.pathloop) {
        //if moving is activated and looping is activated
        pic.pathcor=0 //reset the coordinates
        picpath() //call function for next movement
        }// end of inner if
    else pic.active=0 //otherwise, looping isn't activated, so stop
moving
    } // end of else
} //end of function

function reset() {
    pic.pathcor = maxnumbers; //set coordinates to end
    pic.pathloop = 0; //run through moving only once
    pic.active = 1; //activate function

    picpath();

    pic.active= 0; //deactivate function
    pic.pathcor = 0; //reset coordinates
    pic.pathloop = 1; //turn looping back on for next time
} //endof function
//-->
</SCRIPT>

<STYLE type=text/css>
<!--
.mtopFin {position: absolute; height: 54px; width: 113px; left: 27px;
top: 6px;}
.mlegFin {position: absolute; height: 100px; width: 69px; left:
182px; top: 125px;}
.mbodyFin {position: absolute; height: 99px; width: 85px; left:
107px; top: 125px;}
.mfishBody {position: absolute; height: 73px; width: 161px; left:
95px; top: 52px;}
.mtailFin {position: absolute; height: 136px; width: 94px; left:
10px; top: 80px;}

.ftopFin {position: absolute; height: 20px; width: 58px; left: 32px;
top: 17px;}
.flegFin {position: absolute; height: 40px; width: 27px; left: 131px;
top: 93px;}
.fbodyFin {position: absolute; height: 33px; width: 60px; left: 57px;
top: 78px;}
.ffishBody {position: absolute; height: 68px; width: 128px; left: 52px;
top: 23px;}
.ftailFin {position: absolute; height: 67px; width: 53px; left: 3px;
top: 21px;}

```

```
#wholeFish2 {position:absolute; left:0px; top:0px; width:277px;
height:243px;}
```

```
#wholeMale {
    position:absolute;
    visibility: visible;
    z-index: 1;
    left:0px;
    top:0px;
    width:262px;
    height:232px;
}
```

```
#wholeFem {
    position:absolute;
    visibility: hidden;
    z-index: 0;
    left:0px;
    top:0px;
    width:190px;
    height:139px;
}
```

```
-->
```

```
</STYLE>
```

```
</HEAD>
```

```
<BODY text=#000000 vLink=#a89ccc link=#7ec9f4 bgColor=#48bcb4
onload=init()>
```

```
<P align="right">
```

```
<table align="right" width=150 border=0 hspace=0 vspace=0>
```

```
<td>
```

```
<font size="-1">
```

```
Please remember to bring the fish to a full and
complete stop before changing its color or gender
```

```
</font>
```

```
</td>
```

```
</table>
```

```
</p>
```

```
<DIV id="wholeFish2">
```

```
<DIV id="wholeMale">
```

```
<IMG class="mtailFin" src="noclrtaill1.gif"
```

```
name="mfishTail">
```

```
<IMG class="mfishBody" src="noclrbody1.gif"
```

```
name="mfishBody">
```

```
<IMG class="mbodyFin" src="noclrbodyfin1.gif"
```

```
name="mbodyFin">
```

```
<IMG class="mlegFin" src="noclrlegs1.gif"
```

```
name="mlegFin">
```

```
<IMG class="mtopFin" src="noclrtopfin1.gif"
```

```
name="mtopFin">
```

```
</DIV>
```

```
<DIV id="wholeFem">
```

```
<IMG class="ftailFin" src="../../../femColorBits/noclrtaill2.gif"
```

```
name="ffishTail">
```

```
<IMG class="ffishBody" src="../../../femColorBits/noclrbbody2.gif"
```

```
name="ffishBody">
    <IMG class="fbodyFin"
src="../../femColorBits/noclrbodyfin2.gif" name="fbodyFin">
    <IMG class="flegFin" src="../../femColorBits/noclrlegs2.gif"
name="flegFin">
    <IMG class="ftopFin"
src="../../femColorBits/noclrftopfin2.gif" name="ftopFin">
    </DIV>

</DIV>

</BODY>
</HTML>
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Virtual Betta -- controls</title>
<script language="Javascript">
/* written by Lauren Carroll
April 2003
*/

//list of the gif names, as text arrays
var MaleGIFs = new Array(
    new Array("noclrbody1.gif", "noclrtaill1.gif",
"noclrlegs1.gif", "noclrtopfin1.gif", "noclrbodyfin1.gif"),
    new Array("redbody1.gif", "redtail1.gif",
"redlegs1.gif", "redtopfin1.gif", "redbodyfin1.gif"),
    new Array("bluebody1.gif", "bluetail1.gif",
"bluelegs1.gif", "bluetopfin1.gif", "bluebodyfin1.gif"),
    new Array("greenbody1.gif", "greentail1.gif",
"greenlegs1.gif", "greentopfin1.gif", "greenbodyfin1.gif"),
    new Array("purplebody1.gif", "purpletail1.gif",
"purplelegs1.gif", "purpletopfin1.gif", "purplebodyfin1.gif"));

var FemGIFs = new Array(
    new Array("noclrbody2.gif", "noclrtaill2.gif",
"noclrlegs2.gif", "noclrtopfin2.gif", "noclrbodyfin2.gif"),
    new Array("redbody2.gif", "redtail2.gif",
"redlegs2.gif", "redtopfin2.gif", "redbodyfin2.gif"),
    new Array("bluebody2.gif", "bluetail2.gif",
"bluelegs2.gif", "bluetopfin2.gif", "bluebodyfin2.gif"),
    new Array("greenbody2.gif", "greentail2.gif",
"greenlegs2.gif", "greentopfin2.gif", "greenbodyfin2.gif"),
    new Array("purplebody2.gif", "purpletail2.gif",
"purplelegs2.gif", "purpletopfin2.gif", "purplebodyfin2.gif"));

//arrays of male and female animation names
var maleAnims = new Array("mNoclrBodyAnim.gif", "mRedBodyAnim.gif",
"mBlueBodyAnim.gif", "mGreenBodyAnim.gif", "mPurpleBodyAnim.gif");
var femAnims = new Array("fNoclrBodyAnim.gif", "fRedBodyAnim.gif",
"fBlueBodyAnim.gif", "fGreenBodyAnim.gif", "fPurpleBodyAnim.gif");

//holding for image objects, one combining maleGifs and femGifs,
//the other combining the animations
//must have a "new Array()" for each array in femGIFS and maleGIFS above
var fishAnims= new Array(new Array(), new Array());
var fishpics = new Array(new Array(new Array(), new Array(), new Array
(), new Array(), new Array()),
    new Array(new Array(), new Array(), new Array(), new
Array(), new Array()) );

var mCurBody = 0; //global current color for male fish's body
var fCurBody = 0; //global current color for female fish's body

//stuff pics into array and preload
for(var w = 0; w < 2; w++){ //outer for loop for gender. 0 is male, 1 is
female
    for (var i = 0; i < MaleGIFs.length; i++) { //middle for loop for
number of colors
        fishAnims[w][i] = new Image();
        if(w)
            fishAnims[w][i].src = femAnims[i];
        else

```

```

        fishAnims[w][i].src = maleAnims[i];

        for (var j = 0; j < 5; j++) { //inner for loop for different
body parts
            fishpics[w][i][j] = new Image();
            if(w)
                fishpics[w][i][j].src = "../femColorBits/" + FemGIFs[i][j];
            else
                fishpics[w][i][j].src = MaleGIFs[i][j];
            } //end of way inner for
        } //end of inner for
    } // end of outer for

function changePics(form){
//function to change which fish is visible, male or female

    //sees which gender is selected. 0 is male, 1 is female
    var fishType = fishForm.gender.selectedIndex;

    if(!fishType) { //i.e., fishType is 0, hence male.

        parent.frames[1].wholeFem.style.visibility = "hidden";
        parent.frames[1].wholeMale.style.visibility = "visible";

    }
    else { //fishType must be 1. therefore, female

        parent.frames[1].wholeMale.style.visibility = "hidden";
        parent.frames[1].wholeFem.style.visibility = "visible";

    } //end of else
} //end of function

function changeFish(form){
//function to change the colors of the static fish images

    /* fishColor is selected color. 0 is no color, 1 is red,
       2 is blue, 3 is green, 4 is purple
       fishPart is selected part. 0 is body, 1 is tail,
       2 is "leg" fins, 3 is top fin, 4 is bottom fin,
       5 is everything
       fishType is selected gender. 0 is male, 1 is female */

    var fishColor = fishForm.color.selectedIndex;
    var fishPart = fishForm.part.selectedIndex;
    var fishType = fishForm.gender.selectedIndex;

    //checking to see if gender swapping is working
    if( ((fishType == 0) && (parent.frames[1].wholeMale.style.visibility
== "hidden"))) )
        changePics(this.form); //male selected (0), but not visible --
change pics
    if( ((fishType == 1) && (parent.frames[1].wholeFem.style.visibility ==
"hidden"))) )
        changePics(this.form); //female selected (1), but not visible --
change pics

    switch (fishType) {
        case 0: //male
            switch (fishPart)
            {

```

```

        case 0:    parent.frames[1].mfishBody.src = fishpics[0]
[fishColor][0].src;
        mCurBody=fishColor; //records the current color for
other functions
        break;
        case 1: parent.frames[1].mfishTail.src = fishpics[0]
[fishColor][1].src;
        break;
        case 2: parent.frames[1].mlegFin.src = fishpics[0]
[fishColor][2].src;
        break;
        case 3: parent.frames[1].mtopFin.src = fishpics[0]
[fishColor][3].src;
        break;
        case 4: parent.frames[1].mbodyFin.src = fishpics[0]
[fishColor][4].src;
        break;
        case 5: parent.frames[1].mfishBody.src = fishpics[0]
[fishColor][0].src;
        mCurBody=fishColor; //records the current color for
other functions
        parent.frames[1].mfishTail.src = fishpics[0]
[fishColor][1].src;
        parent.frames[1].mlegFin.src = fishpics[0][fishColor]
[2].src;
        parent.frames[1].mtopFin.src = fishpics[0][fishColor]
[3].src;
        parent.frames[1].mbodyFin.src = fishpics[0][fishColor]
[4].src; break;
    } // end of male switch
    break;
    case 1: //female
        switch (fishPart)
        {
            case 0:    parent.frames[1].ffishBody.src = fishpics[1]
[fishColor][0].src;
            fCurBody=fishColor;
            break;
            case 1: parent.frames[1].ffishTail.src = fishpics[1]
[fishColor][1].src;
            break;
            case 2: parent.frames[1].flegFin.src = fishpics[1]
[fishColor][2].src;
            break;
            case 3: parent.frames[1].ftopFin.src = fishpics[1]
[fishColor][3].src;
            break;
            case 4: parent.frames[1].fbodyFin.src = fishpics[1]
[fishColor][4].src;
            break;
            case 5: parent.frames[1].ffishBody.src = fishpics[1]
[fishColor][0].src;
            fCurBody=fishColor;
            parent.frames[1].ffishTail.src = fishpics[1]
[fishColor][1].src;
            parent.frames[1].flegFin.src = fishpics[1][fishColor]
[2].src;
            parent.frames[1].ftopFin.src = fishpics[1][fishColor]
[3].src;
            parent.frames[1].fbodyFin.src = fishpics[1][fishColor]
[4].src; break;

```

```

        } // end of female switch
        break;
    } // end of big ol' switch

} //end of function

function startMoving(){
    // changes from the static image to the animated one & starts fish
    moving
    if(parent.frames[1].wholeFem.style.visibility == "visible") //fem is
    visible
        parent.frames[1].ffishBody.src = fishAnims[1][fCurBody].src;
    else
        parent.frames[1].mfishBody.src = fishAnims[0][mCurBody].src;

    parent.frames[1].pic.active=1; //tells moving function on picture
    page to activate
    parent.frames[1].picpath(); //calls moving function on picture page
} // end of function

function stopMoving(){
    //changes from animated image to static one and stops fish moving

    if(parent.frames[1].wholeFem.style.visibility == "visible") //fem is
    visible
        parent.frames[1].ffishBody.src = fishpics[1][fCurBody][0].src;
    else
        parent.frames[1].mfishBody.src = fishpics[0][mCurBody][0].src;

    parent.frames[1].pic.active=0; //tells moving function on picture page
    to deactivate

} // end of function

//-->
</script>
</head>

<body bgcolor="#48BCB4">
<form NAME = "fishForm">
    <i>Pick Fish</i>
    <!-- using both onClick and onChange because onChange didn't work in
    some macs -->
    <select name="gender" onChange="changePics(this.form)"
    onClick="changePics(this.form)">
        <option value="0">Male<br>
        <option value="1"> Female<br>
    </select>
    <p>
    <I>Pick Color</I>
    <select name="color">
        <option value="noclr">Colorless
        <option value="red" >Red
        <option value="blue">Blue
        <option value="green">Green
        <option value="purple">Purple
    </select>

    <p>
    <I>Pick Body Part</I>

```

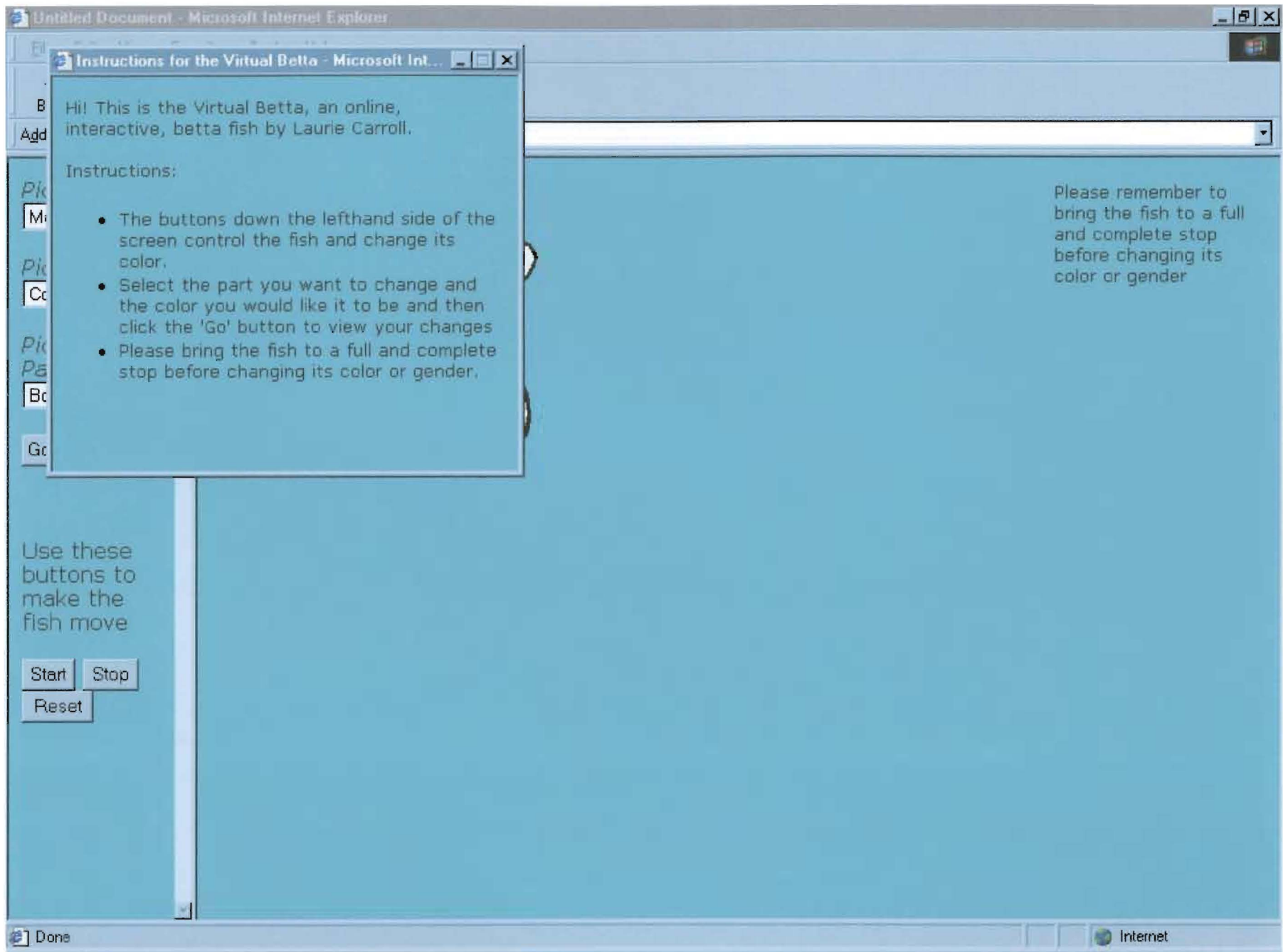
```

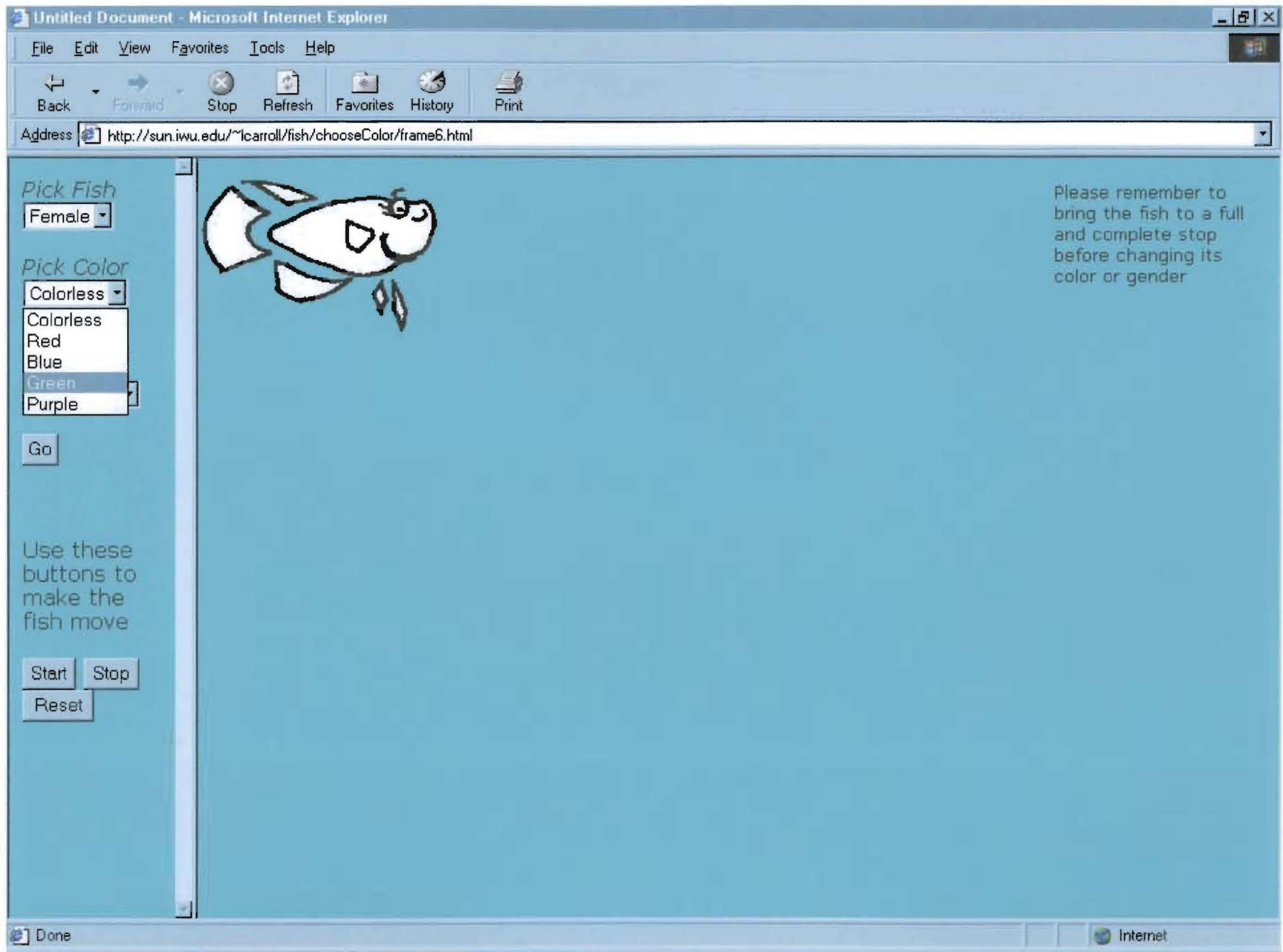
    <select name="part">
      <option value="body">Body
      <option value = "tail"> Tail
      <option value = "legs"> "Leg" fins
      <option value = "topfin"> Top fin
      <option value = "bodyfin"> Bottom fin
      <option value = "all"> Everything!
    </select>
  </p>
  <p>
    <INPUT TYPE = "button" NAME = "PickFish" VALUE = "Go" onClick =
"changeFish(this.form);return true">
  </p>
</form>

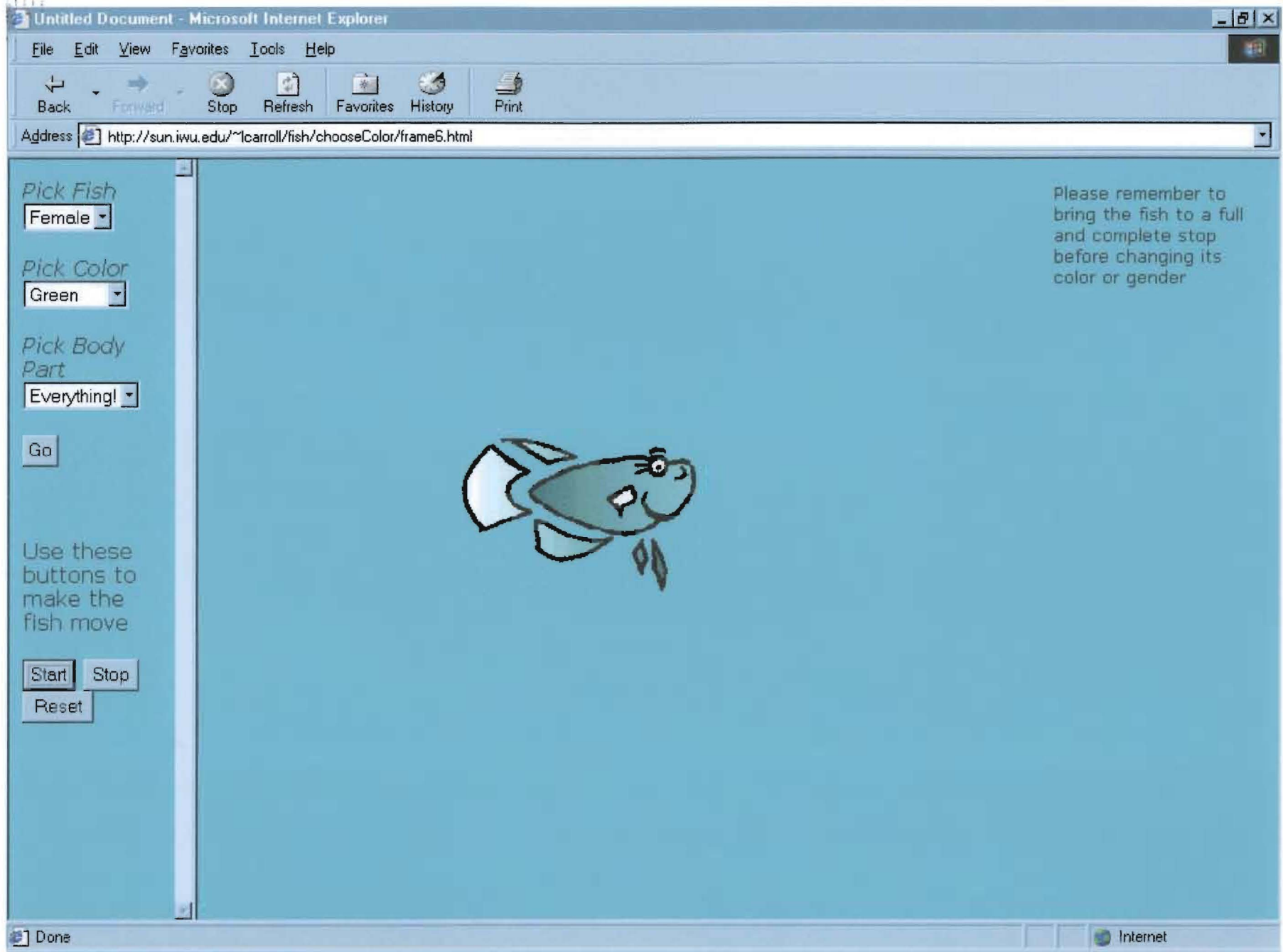
<br><br>
Use these buttons to make the fish move
<form name="ctrlMove">
  <INPUT TYPE = "button" NAME = "start" VALUE = "Start" onClick =
"startMoving();return true">
  <INPUT TYPE = "button" NAME = "stop" VALUE = "Stop" onClick =
"stopMoving();return true">
  <INPUT TYPE = "button" NAME = "reset" VALUE = "Reset" onClick =
"stopMoving();parent.frames[1].reset();return true">

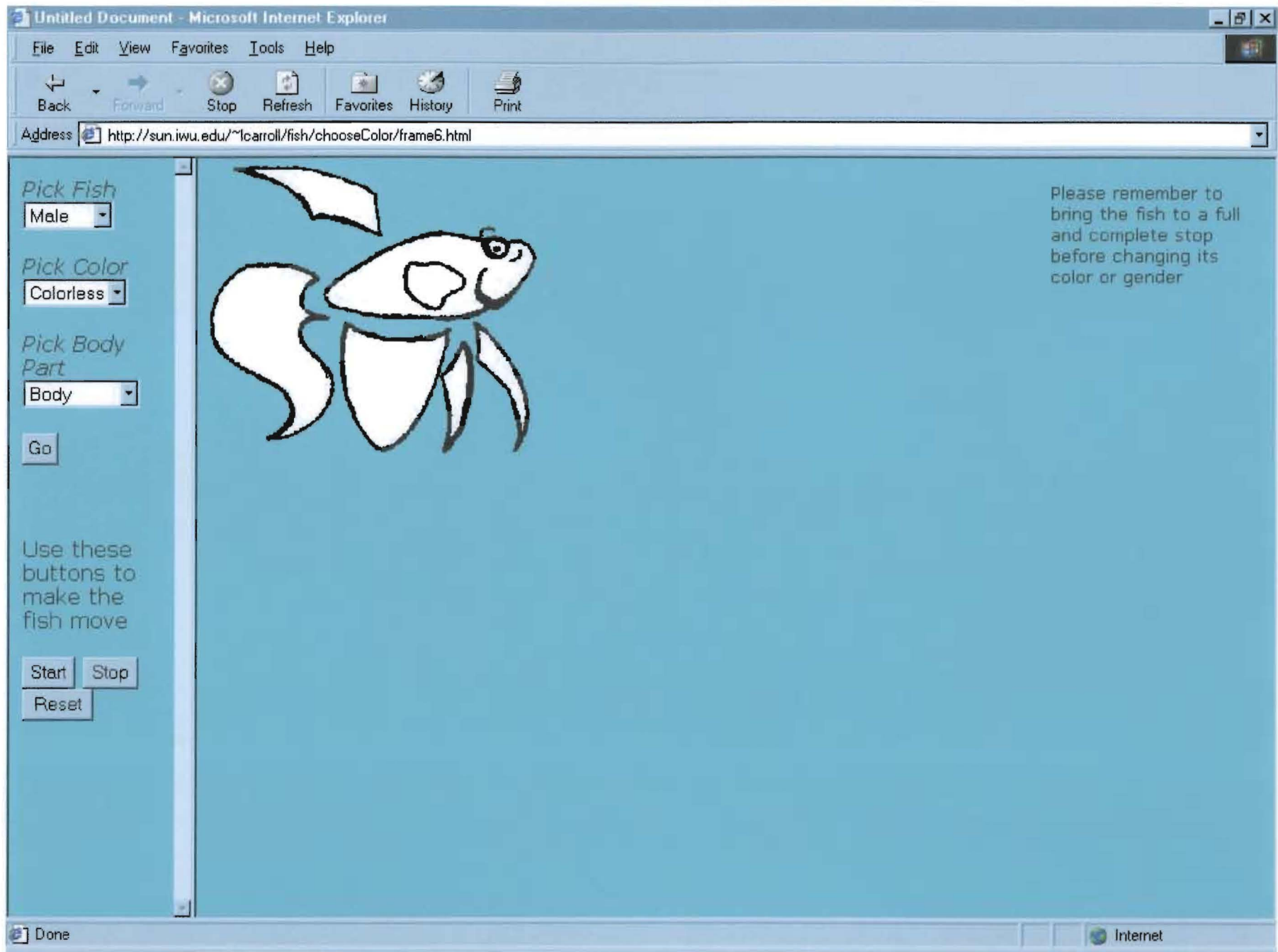
</form>
</body>
</html>

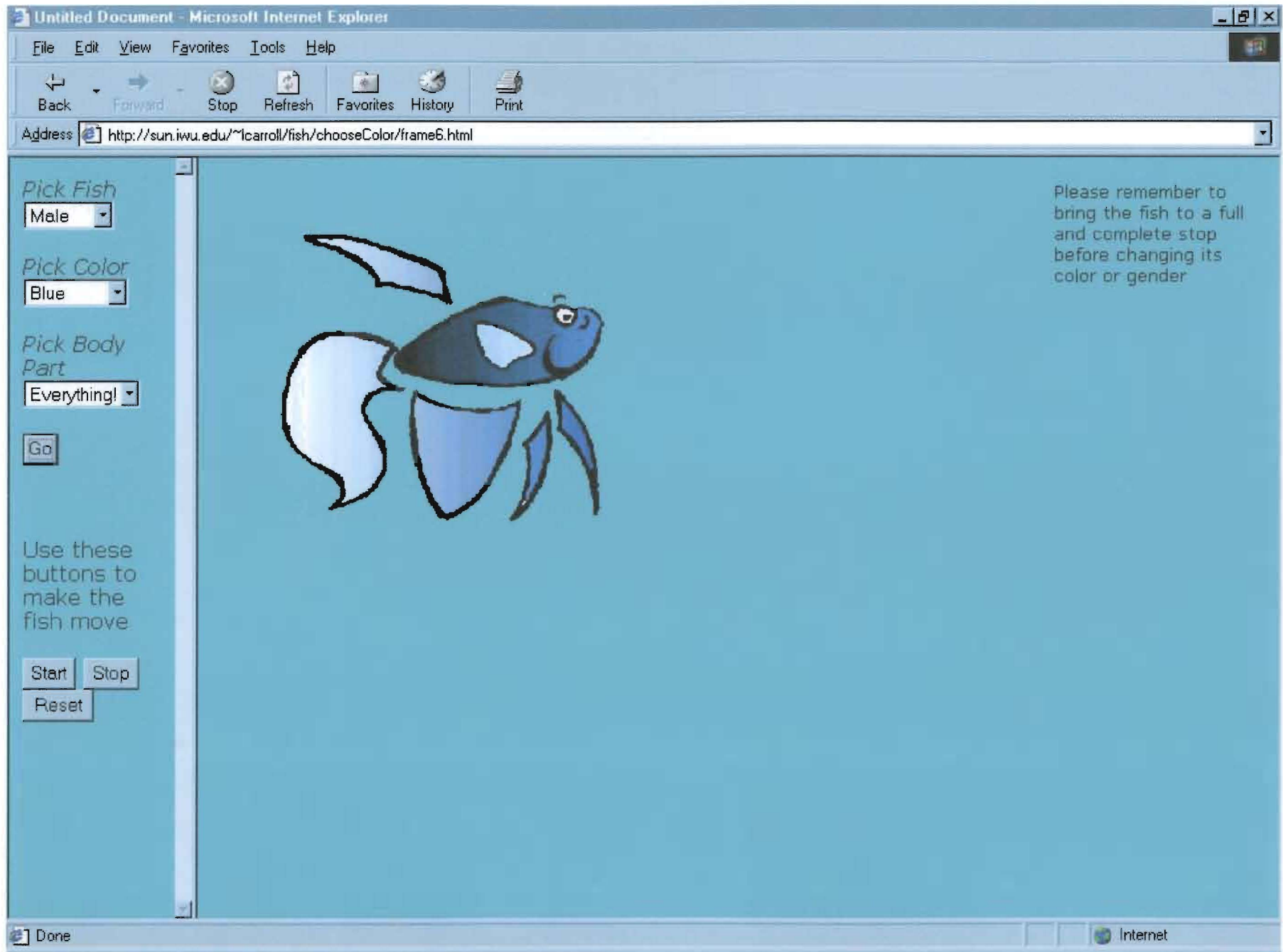
```









References

Goodman, Danny. JavaScript Bible. 2nd ed. Foster City, CA: IDG Books Worldwide, Inc. 1996.

Musciano, Chuck and Bill Kennedy. HTML: The Definitive Guide. 3rd ed. Sebastopol, CA: O'Reilly & Associates, Inc. 1998.

Rule, Jeff. Dynamic HTML: The HTML Developer's Guide. Reading, MS: Addison-Wesley – Addison Wesley Longman, Inc. 1999.