Internet Computing

Advanced Multimedia Development (AMMD)

:::

2004

ActionScript Game Overview

- Project 2 Prototype Report Template
- Creating Instances Dynamically
- Dynamic Event Handlers
- Array Methods
- 2-D Array; multi-D Arrays
- Game - “Match Them Up”
- Game - “Critter Attack” - Space Invader

Internet Computing

Project 2 Prototype Report

Refer to the subject website for the template

Prototype Program & Report
- Due in week 11 Lab
- Game category students will be asked to switch to digital book category if they fail to produce game skeleton (or fail to explain code)

All students to choose game topic by this week's lab session. (fail to do so => digital book)

Creating Instances Dynamically

Reading - P130 to P149
So far… we create instances by dragging them there from the library. (Project 1, mouse chaser)

What would happen if we want to create a game where there were many instances moving about the stage?
Instances explode, disappear, re-appear…

Show space invader game…

Dynamic Movie Clip Instances

Internet Computing

Creating Instances Dynamically

Prepare movie clip – Linkage
Export for ActionScript

Built-in function attachMovie

MovieClip *myMovieClip*.attachMovie (*idName*, *newName*, *depth* [, *initObject*]);

idName: name used exported clip
newName: variable name of new instance
depth: depth the new clip to occupy (like layer)
initObject: populate properties of new instance
e.g. _x, _y, etc (optional)

Creating Instances Dynamically

```
_root.attachMovie ("ball","ball1",1);
_root.attachMovie ("ball","ball2",2);

ball1._x = 100;
ball1._y = 100;

ball2._x = 160;
ball2._y = 100;
```

2 instances created

Internet Computing

Creating Instances Dynamically

```
createEmptyMovieClip ("myBalls",1);   //empty movie
      clip

for (i=1; i<6; i++)
{
      myBalls.attachMovie("ball","ball"+i, i);
      eval ("myBalls.ball"+i)._x += i*100;      //create
      variable names at runtime
}

myBalls._alpha = 50;
```

Empty Clip as container, attach clip to clip

eval function for creating variable name

Array Methods

There are a lot of in-built array methods
e.g. toString, reverse

```
//p192, the reverse method
myArray = new Array();
for (i=0; i<5; i=i+1)
        myArray[i] = i;
trace (myArray.toString());   //print array in string
myArray.reverse();                                    //reverse
array
trace (myArray.toString());
trace ("=======================");
```

2-D Array

```
//p206 multi-dimensional array
//2-D array

myArray = new Array();
for (i=0; i<5; i = i+1)
{
        myArray[i] = new Array();
        for (j=0; j<5; j=j+1)
        {
                myArray[i][j] ="["+i+","+j+"]";
        }
}
```

2-D Array

```
for (i=0; i<5; i = i+1)
{
        myOutputString ="";
        for (j=0; j<5; j=j+1)
        {
                myOutputString = myOutputString +
                        myArray[i][j] + " ";
        }
        trace(myOutputString);
}
```

What is the output?

2-D Array

Output is…

[0,0] [0,1] [0,2] [0,3] [0,4]
[1,0] [1,1] [1,2] [1,3] [1,4]
[2,0] [2,1] [2,2] [2,3] [2,4]
[3,0] [3,1] [3,2] [3,3] [3,4]
[4,0] [4,1] [4,2] [4,3] [4,4]

How about 3-D array? 4-D array?

Game – “Match Them Up”

Play the game…

Need to read/try textbook Pg210 to 229

Structure… 2-D Array x 2
1 2-D Array for Tiles
1 2-D Array for blockers

Internet Computing

Game – “Match Them Up”

Internet Computing

.tile_type=0 [0][0] 75, 62.5 .solved ._visible	.tile_type=0 [0][1] 225, 62.5 .solved ._visible	.tile_type=1 [0][2] 375, 62.5 .solved ._visible	.tile_type=1 [0][3] 525, 62.5 .solved ._visible
.tile_type=2 [1][0] 75, 187.5 .solved ._visible	.tile_type=2 [1][1] 225, 187.5 .solved ._visible	.tile_type=3 [1][2] 375, 187.5 .solved ._visible	.tile_type=3 [1][3] 525, 187.5 .solved ._visible
.tile_type=4 [2][0] 75, 312.5 .solved ._visible	.tile_type=4 [2][1] 225, 312.5 .solved ._visible	.tile_type=5 [2][2] 375, 312.5 .solved ._visible	.tile_type=5 [2][3] 525, 312.5 .solved ._visible
.tile_type=6 [3][0] 75, 437.5 .solved ._visible	.tile_type=6 [3][1] 225, 437.5 .solved ._visible	.tile_type=7 [3][2] 375, 437.5 .solved ._visible	.tile_type=7 [3][3] 525, 437.5 .solved ._visible

Game – “Match Them Up”

1 Create two of each tile instance on the board
2 Create 16 tile blocker instances on the board
3 Lay out the tiles in a 4x4 block on the stage
4 Shuffle the tiles randomly
5 Flip all the tiles face down (use blockers)
6 Wait for the player to make a move by flipping two tiles
7 If the flipped tiles match and all the tiles are flipped, the player wins. Go to step 4
8 If the flipped tiles match but not all the tiles are flipped, go to step 6
9 If the flipped tiles do not match, flip them back over and go to step 6

Internet Computing

Game – “Match Them Up”

Random function

Math.random() => value from 0 to 1

Math.random() * 10 => value from 0 to 10

Math.floor (Math.random()*11) => integer 0 to 10

Internet Computing

Game – “Match Them Up”

Internet Computing

Detecting the User’s Mouse Clicks
There are several methods

1. Test for onMouseUp event on the _root timeline, hit test all the clips
2. onRelease handler for each tile and let them handle it with a function (Mouse Chaser)

For this game… method 1

```
this.onMouseUp = function ()
{for …. //2-D array
    tile_array[i][j].hitTest (_root._xmouse, _root._ymouse)
}
```

Game – “Match Them Up”

Internet Computing

Pausing After Second Choice

number getTimer ();
Tells us the number of milliseconds that passed since the main movie begin to play.

```
If (getTimer() > time_at_last_pick+1000) …

…..
time_at_last_pick = getTimer();
```

Game – “Critter Attack”

Play the game…

Need to read/try textbook Pg333 to 367

Structure…
Ship movie clip has its onEnterFrame handler
Aliens have their own onEnterFrame handler
Use Stage object to get width and height of stage

Game – “Critter Attack”

Player controls the spaceship with arrow keys
Aliens move as large block back and forth
Each Time the aliens get to the edge, they move down
Spaceship fire lasers using space bar
Aliens can fire lasers back
When players shoot an alien, it dies and disappears
When the aliens shoot the player, ship becomes redder
If player dies, game ends
If player kills all the aliens, more faster aliens

Internet Computing

Game – “Critter Attack”

```
//Detect Keypress
shipDepth = 100; ……..
attachMovie("ship","ship",shipDepth,{_x:Stage.width/2,_
      y:Stage.height-65,_xscale:18,_yscale:18});
ship.onKeyDown=shipKeyDown;
function shipKeyDown()
{
      k=Key.getCode();   //get keyboard code when a key
      is pressed
      if (k==37)
      { trace ("left arrow pressed");        }
      else if (k==39)
      { trace ("right arrow pressed");     }
      else if (k==32)
      { trace ("spacebar pressed, fire ..." + k);  }
}
```

Conclusion

Internet Computing

Project 2 Prototype Report Template
Creating Instances Dynamically
Dynamic Event Handlers
Array Methods
2-D Array; multi-D Arrays
Game - “Match Them Up”
Game - “Critter Attack” - Space Invader