

## **“Using Flash ActionScript to create a Pong game”**

**Created by: Stacey Fornstrom**

**Thomas Jefferson High School**

**Denver, CO**

**Student Project Examples:** <http://tjhs.dpsk12.org/~softeng/>

### **Background:**

Flash ActionScript is the programming component of Macromedia Flash. ActionScript can be used to add interactivity to web sites. I have found that one of the best ways to introduce students to ActionScript is to create a “Pong” game together as a class.

I use the Pong Tutorial to teach Flash ActionScript to high school students in the Computer Magnet program at Thomas Jefferson High School. Students learn Flash basics in a previous class. Students have also learned programming basics with Lego Mindstorms and VB.net.

Class sessions are 50 minutes long. This unit covers 5 class sessions. The tutorial was completed with Macromedia Flash MX 2004 on Dell PC's. Each student has their own computer and class sizes range from 20 to 23 students.

These lessons provide many opportunities for differentiation. Students can use any design they wish for the walls, balls, or paddle. The version of Pong that we create in class is very simple. Students add complexity by adding additional paddles, players, lives, balls, bricks, and anything else that they think of. The content interests most students enough to use their own time to enhance what we build in class together.

### **Daily Lessons:**

I have a projector that I use to demonstrate each step required to create Pong. Students follow along with me, at the end of the unit they each have their own game of Pong. I assess daily progress by providing an assignment for each class period and asking students to show me their program when it is working correctly. I start each class session with a “Question of the Day” that acts as a warmup and provides a review of important content or a preview for the day's lesson.

Day 1 of the Pong Tutorial is a compilation of background information that we learn before beginning the tutorial. It is actually used over several days as we work on multiple activities to learn about Flash.

## Day 1 – Pong Tutorial

### Quiz → Use your notes to answer and turn in to Fornstrom:

- What is a symbol in Flash?
- How do you create a symbol in Flash?
- How does Flash know the difference between copies of the same symbol?
- What is the basic ActionScript syntax to make a movie clip move?
- List the primary motion scripting properties available for Flash movie clips.
- Provide a list of events that are frequently used with movie clips.

### Objectives:

- Cover background information that will be needed to create Pong in Flash ActionScript.
- Create graphics to use in creating a Pong game:
  - Ball
  - Paddle
  - Walls

### Answer to Question of the Day:

- A **Symbol** is a reusable image, animation, or button in Flash.
- Create a symbol: 1. create an object 2. turn into symbol: **Modify > Convert to Symbol** 3) specify a name 4) check the box “Export for ActionScript”.
- An **Instance** is a copy of any symbol from the Flash library that resides on the stage. **Instance Names** are used by ActionScript to identify each instance. This is what distinguishes copies of the same symbol. Select the instance on the stage and enter the **Instance Name** in the **Property** window.
- What is the basic ActionScript syntax to make a movie clip move?

→ instanceName.event = function( ) { statements }

```
ball.onEnterFrame = function() {  
    this._x += 10;  
} // end of the function
```

- **The primary motion scripting properties available in Flash.**

ActionScript Property	Description
<b>_x</b>	The x, or horizontal, position of the movie clip
<b>_y</b>	The y, or vertical, position of the movie clip
<b>_rotation</b>	The rotation of the movie clip in degrees, based on a 360 degree circle.
<b>_xscale</b>	The percentage of horizontal scaling to apply along the x-axis. 100 is unchanged, 50 is half width, 200 is double width, and so on.
<b>_yscale</b>	The percentage of vertical scaling to apply along the y-axis.
<b>_alpha</b>	The percentage of opacity that the movie clip should have. 100 is completely solid, 50 is half visible, 0 is completely transparent (invisible).
<b>_visible</b>	Determines whether or not to draw the movie clip at all. Can be set to <b>True</b> or <b>False</b> .

- **Frequently Used Movie Clip Events.**

Event	Event
<b>onEnterFrame</b>	<b>onMouseMove</b>
<b>onMouseDown</b>	<b>onKeyDown</b>
<b>onMouseUp</b>	<b>onKeyUp</b>
<b>onLoad</b>	<b>onRelease</b>
<b>onPress</b>	<b>onRollover</b>

## Day 2 – Pong Tutorial

### Question of the Day:

- What is `movieClip.hitTest()` used for?
- How can we use `hitTest()` in our projects?

### Today's Assignment → show Mr. Fornstrom upon completion.

- Create a new Flash file, save the file as “**Pong**”.
- Create a movie clip on the stage, give it an instance name of “**ball**”.
- Create a second movie clip that extends across the bottom of the stage, give it an instance name of “**ground**”.
- Rename Layer1 to “**Graphics**”.
- Create a new layer and name it “**Actions**”.
- Together we will add code to make the ball move and bounce off the ground.

### Objectives:

- Introduce the `hitTest()` command.
- Begin building **Pong** game in Flash.

### Answer to Question of the Day:

- `hitTest()` is used to determine if the boundary boxes of 2 movie clips intersect at any point. The following example uses `hitTest()` to determine if the movie clip **ball** collides with the movie clip **ground**.

```
if(ball.hitTest(ground)) {  
    trace("ball hit the ground")  
    ymove = -5 // makes the ball move up  
} // end if
```

### Sample Code for Today's Assignment: (in:" BallHitsGrnd")

- The following code should be placed in Frame 1 of the “actions” layer. Make sure that nothing on the stage is selected when you begin typing the code.

```
xmove = 10; // variable used to control x movement  
ymove = 10; // variable used to control y movement  
ball.onEnterFrame = function() {  
    // move the ball  
    ball._x += xmove  
    ball._y += ymove  
    // see if the ball hits the ground  
    if (ball.hitTest(ground)) {  
        ball._alpha = "30"  
        ball._width = 400  
        ball._height = 5  
        xmove = 2  
        ymove = -2  
    } // end HitTest ground  
} // end ball.onEnterFrame function
```

## **Day 3 – Pong Tutorial**

### **Question of the Day:**

- Open the “**Pong**” Flash file that you started yesterday.

### **Today’s Assignment** → show Mr. Fornstrom upon completion.

- Open the “**Pong**” Flash file.
- Add 3 more walls to contain the ball on the stage. The instance names for the 4 walls should be named: **LeftW, TopW, RightW, ground**
- Add ActionScript code that checks to see when the ball hits a wall.
- Make the ball change directions after hitting a wall.

### **Objectives:**

- Begin creating a Pong program.
- Pong: create 4 walls and ActionScript code to contain the ball on the stage.

### **Sample Code for Today’s Assignment:** (fn:“Pong-Day1”)

The following should be entered in the **ball.onEnterFrame** function:

```
if(ball.hitTest(LeftW)) {
    xmove *= -1          // change x direction
} else if(ball.hitTest(RightW)) {
    xmove = Math.random() * -10 // change x direction
} else if(ball.hitTest(TopW)) {
    ymove *= -1          // change y direction
} else if(ball.hitTest(ground)) {
    ymove *= -1          // change y direction
    Score -= 1;
} // end if
```

## Day 4 – Pong Tutorial

### Question of the Day:

- Open the “**Pong**” Flash file.
- To this point, we have placed the ActionScript on a layer that we named **Actions**. Where else can ActionScript be placed? How would this change the way that the ActionScript is written?
- What is **\_root** used for in ActionScript?

Today’s Assignment → show Mr. Fornstrom upon completion.

- Add a paddle movie clip, name it **paddle**.
- Use the right and left arrow keys to make the paddle move left and right across the screen.
- Add a hitTest between the paddle and the ball so that the ball bounces off the paddle.

### Objectives:

- Learn about other locations to place ActionScript code.
- Learn what **\_root** and **\_parent** keywords are used for in ActionScript.
- Pong program: add a paddle, make it move, make the ball bounce off the paddle.

### Answer to Question of the Day:

- To this point, we have placed the ActionScript on a layer that we named **Actions**. Where else can ActionScript be placed? How would this change the way that the ActionScript is written?
  - ActionScript can also be placed directly on the symbol or instance.
  - When placing ActionScript on an instance, you do not need to specify the name of the instance.
  - **\_root** – The **\_root property** refers to the root of the current movie clip or object. If the currently executing movie has multiple levels, *the current level is the \_root reference*. Use **\_root** to move up to the top level from where you are currently. **\_parent** moves up 1 level.

### Sample Code for Today’s Assignment:

→ This is ActionScript code that moves the paddle and checks to see if we have run into the walls. This code is placed on the paddle (instance name = “**paddle**”).

```
onClipEvent (enterFrame) {  
    if (!this.hitTest( _root.LeftW) && Key.isDown(Key.LEFT)) {  
        this._x -= 10  
    } // end left key  
    if (!this.hitTest( _root.RightW) && Key.isDown(Key.RIGHT)) {  
        this._x += 10  
    } // end right key  
} // end enterFrame
```

The “&&” means “and”, and the “!” means “Not”,

so the statement actually says that if the pad is not touching the side and an arrow key is pressed, it should move left or right.

→ This is ActionScript code that checks to see when the **ball** contacts the **paddle**. This code is placed in the **Actions** layer within the **ball.onEnterFrame** function.

## **Day 5 – Pong Tutorial**

### **Question of the Day:**

- Open your “**Pong**” program.
- What steps do we need to complete to add **Scoring** to our Pong game?

### **Today’s Assignment** → show Mr. Fornstrom upon completion.

- Add **Scoring** to your Pong program.
- Each time the ball hits the **paddle**, add 1 to a variable named **Score** and display it on the screen. Add a hitTest between the paddle and the ball so that the ball bounces off the paddle.
- Show Mr. Fornstrom when finished.

### **Objectives:**

- Variables: declaring, incrementing, and displaying variables.
- Pong program: Add **Scoring**, each time the ball hits the **paddle**, add 1 to a variable named **Score** and display it on the screen.

### **Answer to Question of the Day:**

- What steps do we need to complete to add **Scoring** to our Pong game?
  1. Create a variable named **Score** and set its initial value to **0**.  
In the **Actions** layer, place the cursor in frame 1. At the top of the ActionScript add this line:  
**Score = 0**
  2. Create a *static text box* that says “**Score**”.
  3. Create a *dynamic text box*, in the **Var:** property, enter **Score**.
  4. Test your project, the display should look like:  
**Score: 0**
  5. In the **Actions** layer, place the cursor in frame 1. In the IF statement `if(ball.hitTest(paddle))`, add the following:  
**Score += 1**
  6. In the **Actions** layer, place the cursor in frame 1. In the IF statement `if(ball.hitTest(ground))`, add the following:  
**Score -= 1**

## Final Code for the Pong Program

ActionScript in the **Actions** layer, frame 1:

```
xmove = 5  
ymove = 5  
Score = 0
```

```
ball.onEnterFrame = function( ) {  
    // move the ball  
    this._x += xmove  
    this._y += ymove  
    // test if the ball hits the paddle  
    if(ball.hitTest(paddle)) {  
        ymove *= -1  
        Score += 1  
    } // end hitTest paddle  
    // test if ball hits ground  
    if(ball.hitTest(LeftW)) {  
        xmove *= -1 // change x direction  
    } else if(ball.hitTest(RightW)) {  
        xmove = Math.random() * -10 // change x direction  
    } else if(ball.hitTest(TopW)) {  
        ymove *= -1 // change y direction  
    } else if(ball.hitTest(ground)) {  
        ymove *= -1 // change y direction  
        Score -= 1  
    } // end if  
} // end ball onEnterFrame
```

ActionScript on the paddle symbol, instance name = **paddle**:

```
onClipEvent (enterFrame) {  
    if (!this.hitTest(_root.LeftW) && Key.isDown(Key.LEFT)) {  
        this._x -= 10  
    } // end left key  
    if (!this.hitTest(_root.RightW) && Key.isDown(Key.RIGHT)) {  
        this._x += 10  
    } // end right key  
} // end enterFrame
```