

“Using Flash ActionScript to create a Pong game”

Created by: Stacey Fornstrom

Thomas Jefferson High School - Denver, CO

Student Project Examples: <http://sfornstrom.tjctweb.com/>

Background:

Flash ActionScript is the programming component of Adobe 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 Center for Communication Technology Magnet program at Thomas Jefferson High School. Students learn Flash basics in a previous class. Students have also learned programming basics with Game Maker and VB.net.

Class sessions are 50 minutes long. This unit covers 5 class sessions. The tutorial was completed with Adobe CS6 Flash on Dell PC's. Each student has their own computer and class sizes range from 20 to 29 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 warm up 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.

Objective:

What are you doing? We are working through in-class tutorials to create a Pong game in Flash.

Why are you doing it? To learn programming and ActionScript basics and creating a fun game.

What tools are you using? Adobe Flash CS6

How will you know you are successful? We will **create a Pong game** in Flash that is **suitable for publishing**.

Project Overview: We will use Flash and ActionScript 2.0 to create a basic Pong game. There will be 2 frames on the main timeline. Frame 1 will have instructions and creator information, and a button to start the game. Frame 2 will have the game. The player will control the paddle with arrow keys. A ball will bounce within walls, Score will increase when the paddle hits the ball and decrease when it hits the bottom wall.

Skills Addressed:

- Stage
- Timeline
- Toolbar
- Library
- Symbols – movie clips and buttons
- Instance Name
- Properties
- Frames, Key Frames, and Frame Rate
- Events and Event Handlers (function)
- hitTest

Day 1 – Pong Tutorial

Day 1 Assignment:

- Create a new Flash ActionScript 2.0 file, save as **pd_lastName_Pong**
- Set Stage Size = 800px by 600px
- On Main timeline, create 2 layers named Actions and Graphics
- In Frame 1, create instructions of how to play the game, information about creator and date made, and a button to start the game which will go to Frame 2.
- Frame 1 should have an attractive design that uses an acceptable color scheme and effects to make the design unique and memorable.
- The main timeline should have 2 layers: “**Graphics**” and “**Actions**”.

Day 2 – Pong Tutorial

Quiz:

Create a Blog Post Named: “Flash Basics” → Use your notes & programs to answer the following in the post (remember to publish the post):

- 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.

Day 2 Assignment:

- Blog Post – answer quiz questions
- Cover background information that will be needed to create Pong in Flash ActionScript.
- Create movie clips to use in creating a Pong game:
 - Ball – symbol name: **mcBall**
 - Paddle - symbol name: **mcPaddle**
 - Walls - symbol name: **mcWall**

Quiz Answers:

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

- **Frequently Used Movie Clip Events.**

Event	Event
onEnterFrame	onMouseMove
onMouseDown	onKeyDown
onMouseUp	onKeyUp
onLoad	onRelease
onPress	onRollover

Day 3 – 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.

- Open `pd_lastName_Pong` in Flash. Place movie clips in Frame 2 on the main timeline.
- Place the `mcBall` movie clip on the stage, give it an instance name of `ball`.
- Place the `mcWall` movie clip on the stage, extend across the bottom of the stage, give it an instance name of `ground`.
- 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")  
} // end if
```

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

- The following code should be placed in Frame 2 of the `Actions` layer. Make sure that nothing on the stage is selected when you begin typing the code.

```
xmove = 5 // variable used to control x movement  
ymove = 5 // variable used to control y movement  
ball.onEnterFrame = function() {  
    // move the ball  
    this._x += xmove  
    this._y += ymove  
    // see if the ball hits the ground  
    if (this.hitTest(ground)) {  
        ymove *= -1 // changes direction on the y-axis  
    } // end HitTest ground  
} // end ball.onEnterFrame function
```

Day 4 – Pong Tutorial

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

- Open **pd_lastName_Pong** in Flash.
- 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:

- 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 (replace the Day 3 code that tests: **if (this.hitTest(ground))**)

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

→make sure that the above code is in the **EVENT ball.onEnterFrame**

Day 5 – Pong Tutorial

Question of the Day:

- Open `pd_lastName_Pong` in Flash.
- 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 paddle is not touching a wall 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.

```
if(this.hitTest(paddle)) {  
    ymove *= -1  
    score += 1  
} // end hitTest paddle
```

Day 6 – Pong Tutorial

Question of the Day:

- Open `pd_lastName_Pong` in Flash.
- 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 2. 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**. In the **Character** dropdown property, look for “Anti-alias” and choose “Use device fonts” from the dropdown.
 4. Test your project, the display should look like:
Score: 0
 5. In the **Actions** layer, place the cursor in frame 2. In the IF statement `if(ball.hitTest(paddle))`, add the following:
score += 1
 6. In the **Actions** layer, place the cursor in frame 2. In the IF statement `if(ball.hitTest(ground))`, add the following:
score -= 1

Day 7 & 8 – Pong Tutorial

Objectives:

- Over the next 2 days: complete the **basic Pong** program.
- Design **at least 2 additional features** and include them in your game. Examples: additional pong balls, extra paddles, background pictures or colors, additional objects for the ball to bounce off.
- What steps do we need to complete to add **Scoring** to our Pong game?

Today's Assignment → show Mr. Fornstrom your completed Pong game. (**75 points**)

Rubric:

C = Complete the steps of in-class tutorial to create a “basic” Pong game. Design must be attractive and use the capabilities of Flash such as effects and animations.

B = Add 1 additional feature to the game.

A = Add at least 2 additional features to the game.

100% = Add at least 3 additional, *complex* features to the game.

Final Code for the Pong Program

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

```
stop()  
btnPlay.onRelease = function( ) {  
    gotoAndPlay(2)  
}
```

ActionScript in the **Actions** layer, frame 2:

```
stop()  
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
```

EXTRA NOTES – FAQ's:

Q: How do I program movie clips to respond to other keys?

A: The following code provides an example of 2 paddles with instance names of p1 and p2. P1 is programmed to move up and down with W and S keys and p2 is programmed to move up and down with arrow keys. A link to the key codes for Flash is:

<http://sfornstrom.tjcctweb.com/Lessons/General/FlashKeyCodes.pdf>

```
ymove1 = 0 // variable to move p1  
ymove2 = 0 // variable to move p2
```

```
p1.onEnterFrame = function( ) {  
    if(Key.isDown(87)) { // w key  
        ymove1 = -10  
    }  
    else if(Key.isDown(83)) { // s key  
        ymove1 = +10  
    }  
    else {  
        ymove1 = 0  
    }  
    this._y += ymove1  
} // end p1.onEnterFrame
```

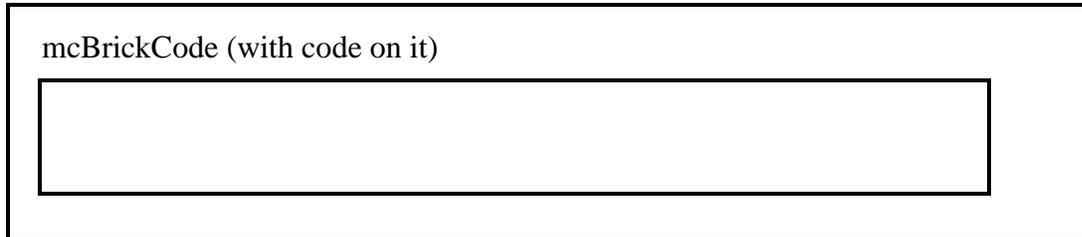
```
p2.onEnterFrame = function() {  
    if(Key.isDown(38)) { // up arrow  
        ymove2 = -10  
    }  
    else if(Key.isDown(40)) { // down arrow  
        ymove2 = +10  
    }  
    else {  
        ymove2 = 0  
    }  
    this._y += ymove2  
}
```

Q: How do I make bricks that disappear when the ball hits them?

A: In a future tutorial, we will learn ActionScript code to remove movie clips from the stage. However, that is beyond the scope of this project so I will show you an alternative way to create this effect.

1. Create a movie clip with a symbol name of **mcBrick** that looks like a brick. We want each instance of mcBrick to contain code that determines when it is hit by the ball, so we will **embed** code in the mcBrick symbol.
2. Double-click on mcBrick so you are in the symbol timeline, highlight the brick graphic and choose **Modify > Convert to Symbol** from the menu. Make the symbol name **mcBrickCode** and Export for ActionScript. What we have done is create a movie clip inside of a movie clip so that we can place code on the embedded movie clip. Think of it this way:

mcBrick



3. Highlight **mcBrickCode** and type the following code in the Actions window:

```
onClipEvent(enterFrame) {  
    if (this.hitTest(_root.ball)) {  
        _parent._y = -100  
        _root.score +=10  
        _root.ymove *= -1  
    }  
} // onClipEvent(enterFrame)
```

4. To place bricks on the stage, drag instances of **mcBrick** from the library to the stage. Each instance of mcBrick has **mcBrickCode** embedded in it which will move mcBrick (the parent) off the screen, add to the score, and switch the ymove direction of the ball.