Zeynep Kınlı *11233*

# Flash Games

## Content

- Our lives B.F (Before Flash)
- How Flash has changed our lives
- The difference between Actionscript 2.0 and Actionscript 3.0
- Examples from successful Flash games
- How to build our own Flash Game?

Zeynep Kınlı *11233*

# Flash Games

## Our lives B.F (Before Flash)

Traditionally, the Web browser has not been kind to online games. The challenge has been to find a solution that will deliver high quality graphics over dial-up connections. Even with the big shift to broadband, nearly 50% of all Internet users still use a dial-up modem to get online.

Graphics, video and audio all stack up to produce large files. The way around this for years has been to sell a CD-ROMs, which install the large media files and then connect you to the Internet so that the system can pass your computer information about the movements of a character, and to update your system with the latest patches.


Zeynep Kınlı *11233*

# Flash Games

## How Flash has changed our lives

Again, the problem here is that you need a CD before you can begin to use the game. Flash is able to help you work through these problems, however, by placing all rich content onto the Web and delivering it to you through your Web Browser.

To begin with, the core competency of Flash has always been to deliver broadband experiences over dial-up connections. Through using Flash's native Vector graphics animation format, you can deliver crisp, stunning graphics in very small files.

Zeynep Kınlı *11233*

# Flash Games

## The difference between Actionscript 2.0 and Actionscript 3.0

Flash also has another trick up its sleeve its own programming language, called ActionScript. ActionScript is based upon JavaScript.

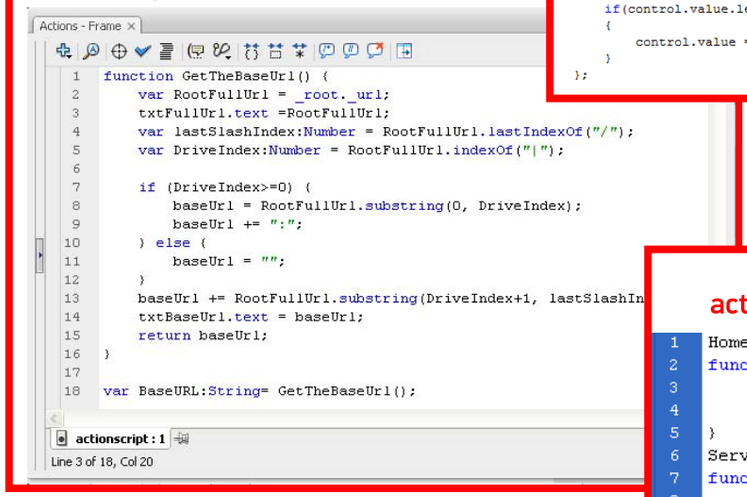
If you are a JavaScript wiz then you will be able to migrate to ActionScript very easily. If you are new to ActionScript and programming in general you will find that ActionScript is both logical and easy to master.

- AS 3.0 is more like a rewrite of AS 2.0 than an incremental upgrade
- AS 3.0 is much faster compared to AS 2.0
- AS 3.0 requires much stricter coding compared to AS 2.0
- AS 3.0 focuses more on OOP compared to AS 2.0
- AS 3.0 is better for big projects while AS 2.0 is easier for small projects
- AS 3.0 is a bit more difficult to learn than AS 2.0

javascript

```javascript
function doBeforePaste(control){
    maxLength = control.attributes["maxLength"].value;
     if(maxLength)
     {
          event.returnValue = false;
     }
}
function doPaste(control){
    maxLength = control.attributes["maxLength"].value;
    value = control.value;
     if(maxLength){
          event.returnValue = false;
          maxLength = parseInt(maxLength);
          var oTR = control.document.selection.createRange();
          var iInsertLength = maxLength - value.length + oTR.text.length;
          var sData = window.clipboardData.getData("Text").substr(0,iInsertLength);
          oTR.text = sData;
     }
}
function LimitInput(control)
{
    if(control.value.length > control.attributes["maxLength"].value)
    {
        control.value = control.value.substring(0,control.attributes["maxLength"].value);
    }
};
```

actionscript 2.0




actionscript 3.0

```actionscript
Home.addEventListener(MouseEvent.CLICK, clickHome);
function clickHome(event:Event):void {
    trace("home");
    gotoAndStop("Home");
}
Services.addEventListener(MouseEvent.CLICK, clickServices);
function clickServices(event:Event):void {
    trace("services");
    gotoAndStop("Services");
}
AboutUs.addEventListener(MouseEvent.CLICK, clickAboutUs);
function clickAboutUs(event:Event):void {
    trace("About Us");
    gotoAndStop("About Us");
}
ContactUs.addEventListener(MouseEvent.CLICK, clickContactUs);
function clickContactUs(event:Event):void {
    trace("Contact Us");
    gotoAndStop("Contact Us");
}
```

Zeynep Kınlı *11233*

# Flash Games

## Examples of successful Flash Games


Fashion World


Music Challenge

Restaurant City

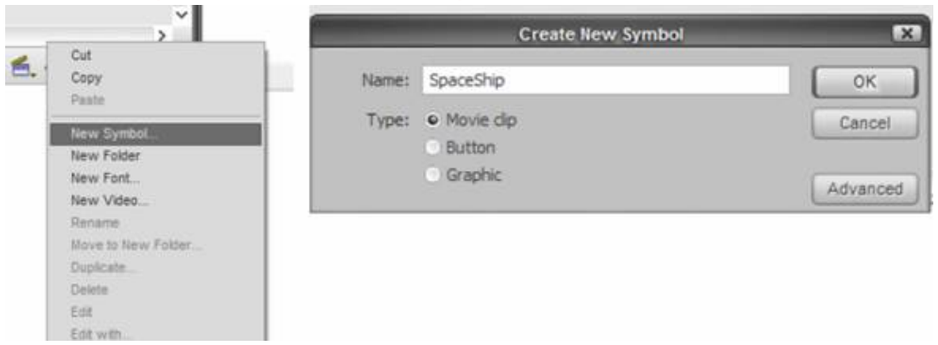

Geo Challenge

Zeynep Kınlı *11233*

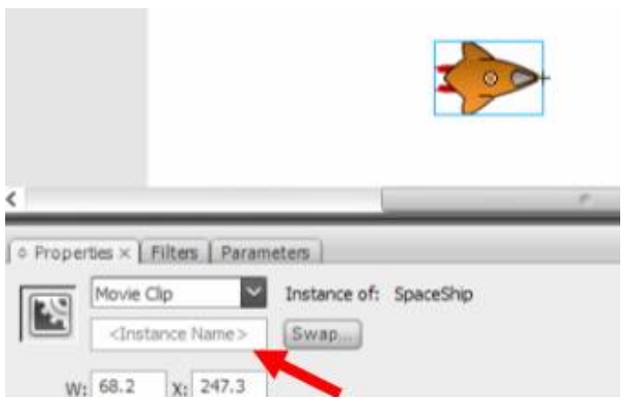
# Flash Games

## How to build our own Flash Game?

**1** In Actionscript 2.0, create a movie clip




**2** Return to main stage and drag your movieclip from the library to the stage and give it an instance name, ‘ship’




**3** To move your movieclip ‘ship’, right click in the 1st frame of your timeline, select actions and type this code:

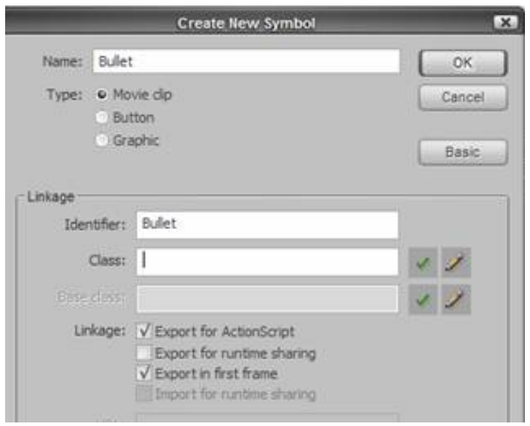
```
this.onEnterFrame = function()
{
	if (Key.isDown(Key.RIGHT))
	{
		Ship._x += 10;
	} else if (Key.isDown(Key.LEFT))
	{
		Ship._x -= 10;
	} else if (Key.isDown(Key.UP))
	{
		Ship._y -= 10;
	} else if (Key.isDown(Key.DOWN))
	{
		Ship._y += 10;
	}
}
```

Zeynep Kınlı *11233*

# Flash Games

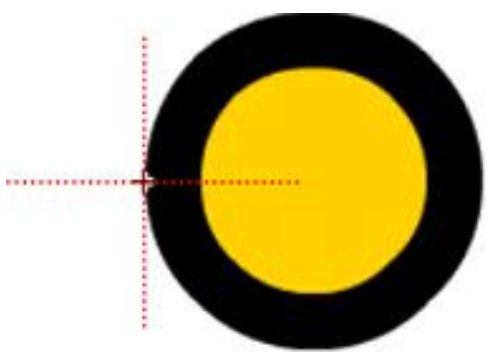
## How to build our own Flash Game?

**4** Right click in the library tab, select ‘new symbol’ and give it the name ‘bullet’, click on ‘advanced’ and select ‘Export for Actionscript’ and make sure the identifier is ‘bullet’




We do this because we need to export the Bullet Movie Clip during running time to create many bullets and we will use the name determined in identifier to call it later

**5** Draw a small bullet and position it like this, it will be the registration point



**6** Open the actions panel and write this code within the 1st frame of the bullet movieclip

```
this.onEnterFrame = function()
{
        this._x += 12;
        if (this._x > 550)
        {
                this.removeMovieClip();
        }
}
```

Zeynep Kınlı *11233*

# Flash Games

## How to build our own Flash Game?

**7** Go to the main timeline, select the first frame and open the actions windows, add this code so that we can fire our bullets

```
var i = 0;
this.onEnterFrame = function()
{
      .
      .
      .
      else if (Key.isDown(Key.DOWN))
   {
      Ship._y += 5;
      }
      if (Key.isDown(Key.SPACE))
      {
            i++;
            _root.attachMovie("Bullet", "Bullet" + i, _root.getNextHighestDepth());
            _root["Bullet" + i]._x = Ship._x + 3;
            _root["Bullet" + i]._y = Ship._y;
      }
}
```

**8** Create a new Movie Clip and draw your enemy ship, don't forget to set the identifier to "Enemy" and select "Export to Action-script". Now drag the enemy Movie Clip to the stage, give it an instance name of "Enemy0", right click it and select "Actions". Paste this.

```
onClipEvent(load)
{
      function reset()
      {
            var timer = 12;
            this._y = Math.random() * 300
            this._x = 550
            mySpeed = Math.ceil(Math.random() * 6) + 1;
      }
      reset();
}
```

Zeynep Kınlı *11233*

# Flash Games

## How to build our own Flash Game?

**9** Paste this code to define direction of the ship in various situations

```
onClipEvent(enterFrame)
{
	//in every frame the enemy move left in the speed defined in the reset function.
	this._x -= mySpeed;
	if (this._x < -10)
	{
		//if the enemy is not on the screen, we reset it.
		reset();
	}
	//if our timer is bigger than 12 we get a new
	if (timer >= 12)
	{
		//direction to the Ship.
		var dir = Math.ceil(Math.random() * 2)
		//We get a random number, 1 or 2. 1 is up, 2 is down.
		//Then we set timer to 0, so we only get a new dir when timer is bigger than 12
		timer = 0;
	}
	if (dir == 1)
	{
		//if dir = 1, we move the ship up.
		this._y -= 3;
	} else if(dir == 2)
	{
		//if dir =  2, we move the ship down.
		this._y += 3;
	}
	//increase timer by 1, so the timer gets equal to 12 and we get a new direction.
	timer++
}
```

**10** To create more enemies, paste this code in the main timeline before the rest of the code.

```
var nrEnemies = 3;
for (i = 1; i < nrEnemies; i++)
{
	_root.Enemy.duplicateMovieClip("Enemy" + i, _root.getNextHighestDepth());
}
```

**11** In order to add score to our basic space shooter game, open the main timeline and paste this.

```
var score = 0;
```

Zeynep Kınlı *11233*

# Flash Games

## How to build our own Flash Game?

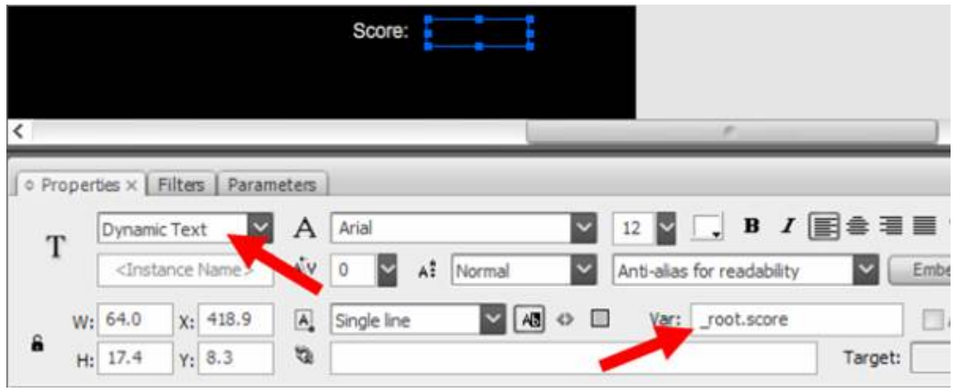
**12** Now open the code of our 'bullet' movieclip and paste this

```
this.onEnterFrame = function()
{
	this._x += 9;
	if (this._x > 550)
	{
		this.removeMovieClip();
	}
	for (i = 0; i < _root.nrEnemies; i++)
	{
		if (this.hitTest(_root["Enemy" + i]))
		{
			_root["Enemy" + i].reset();
			_root.score += 10;
			this.removeMovieClip();
		}
	}
}
```

This is the code we used before to know if a bullet hits an enemy, so we can put the score for killing an enemy there.

If you want the player to lose points for getting killed, just add "_root.score -= PointsToLose" to the code that checks if the player was killed.

**13** In order to show the player his/her score, select text tool and draw in on stage and write 'score' in it. Now draw another text box next to it and change these properties.




**14** To make our ship have lives, add another variable in the main code:

```
var lives = 3;
```

Zeynep Kınlı *11233*

# Flash Games

## How to build our own Flash Game?

**15** This is the last part of our game and we are going to add sound and a pause key.

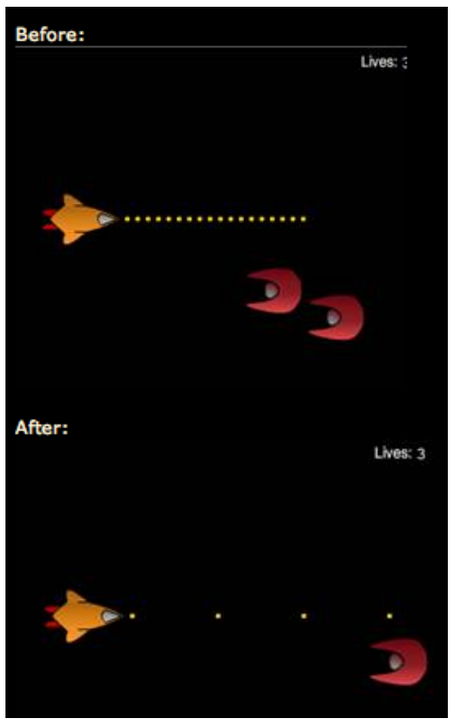
Before we start with sound we need to make some changes. You may have noticed that our ship shoots bullets as much as the player press the space bar, that's not right as it makes the game much easier.
To prevent that, we are going to add a timer and the ship will shoot as much as WE want.
First add a new variable:

```
var timer = 8;
```

**16** Now make these changes

```
this.onEnterFrame = function()
{
      timer++;
      .
      .
      .
      if (Key.isDown(Key.SPACE))
      {
            i++;
            if(timer >= 8)
            {
                  _root.attachMovie("Bullet", "Bullet" + i, _root.getNextHighestDepth());
                  _root["Bullet" + i]._x = Ship._x + 3;
                  _root["Bullet" + i]._y = Ship._y;
                  timer = 0;
            }
      }
}
```

**17** The code runs like this:
1. If timer >= 8 then shoot a bullet and set timer to 0
2. Now that timer = 0, the bullet wont shoot, so the player has to wait all the code to run at least 8 timer to shoot again (timer++ will increase timer by 1 every frame)
3. Timer will be bigger than 8 again and the player can shoot.



Zeynep Kınlı *11233*

# Flash Games

## How to build our own Flash Game?

**18**

We can add the sound now. You need to have an mp3 file that's going to be the sound played when you fire. Simply download the source file that contains sound file.

Now that you have your sound, click File > Import > Import to library...

Select your mp3 file and click open, a new file should appear on the library. Right click it and click "Linkage..." Check "Export to Actionscript..." and set the Identifier to "shoot".

This is the same thing we did previously with the Bullet Movie Clip, we need the linkage to call this on the code.

Add this to the shooting code:

```
if (Key.isDown(Key.SPACE))
{
      i++;
      if(timer >= 8)
      {
            _root.attachMovie("Bullet", "Bullet" + i, _root.getNextHighestDepth());
            _root["Bullet" + i]._x = Ship._x + 3;
            _root["Bullet" + i]._y = Ship._y;

            var shoot_sound = new Sound();
            shoot_sound.attachSound("shoot");
            shoot_sound.start();

            timer = 0;
      }
}
```

Zeynep Kınlı *11233*

# Flash Games

**I hope this presentation and the tutorial have been helpful, thanks for paying attention!**