Μέρη κώδικα

Play Sounds

```
package
{
    import flash.media.Sound;
    import flash.media.SoundChannel;
    import flash.media.SoundMixer;
    import flash.media.SoundTransform;
    import flash.utils.*;
    import flash.net.URLRequest;
    import flash.events.*;
    import flash.filters.*;
    import flash.errors.IOError;

    public class audioManager
    {
        private var _sound: Sound;
        private var _sc: SoundChannel;

                private var _index:int;
                private var soundsArray:Array;

                private var _soundsQueue:Array;

            public function audioManager()
        {
                        soundsArray = null;
            _sc = new SoundChannel();
            _sound = new Sound();
            _sound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
                     _sc.addEventListener(Event.SOUND_COMPLETE, onComplete);
                }

                public function onComplete(e:Event):void
                {
                        _sc.stop();
                        _index = 0;
                }

                public function multipleSoundsPlayer(soundsQueue:Array)
                {
                        _sc.stop();
                        _soundsQueue = soundsQueue;
                        playNext();
                }

                private function playNext(event:Event = null)
                {
                        if (_soundsQueue!= null && _soundsQueue.length > 0)
                        {
                                _sound = new Sound();
                                _sound.load(new URLRequest(_soundsQueue.shift()));
                                _sc = _sound.play();
                                _sc.addEventListener(Event.SOUND_COMPLETE, playNext,
false,0,true);
                        }
                }

        private function ioErrorHandler(event: IOErrorEvent): void
                {       }

                public function playSong(iSoundURL:String)
                {
                        soundsArray = null;
                        _sc.stop();
                        _sound = new Sound();
                        _sound.load(new URLRequest(iSoundURL));
```

```
                        _sc = _sound.play();
                        _sc.addEventListener(Event.SOUND_COMPLETE, onComplete);
                        _sound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
                }

                public function playSongs(iSoundURLs:String)
                {
                        soundsArray = iSoundURLs.split(',');

                        _sc.stop();
                        if(soundsArray.length == 1)
                        {
                                this.playSong(iSoundURLs);
                        }
                        else
                        {
                                _sound = new Sound();
                                _sound.load(new URLRequest(soundsArray[_index]));
                                _sc = _sound.play();
                                _sc.addEventListener(Event.SOUND_COMPLETE, onCompleteFirst);
                                _sound.addEventListener(IOErrorEvent.IO_ERROR,
ioErrorHandler);
                        }
                }

                public function onCompleteFirst(e:Event):void
                {
                        _sc.stop();
                        ++_index;
                        _sound = new Sound();
                        _sound.load(new URLRequest(soundsArray[_index]));
                        _sc = _sound.play();
                        _sc.addEventListener(Event.SOUND_COMPLETE, onComplete);
                        _sound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);

                }

        public function stopSong()
        { _sc.stop(); }

        public function pauseSong()
        {  _sc.stop(); }
        }
}
```

Cars Object

```
package
{
	import flash.display.*;

	public class Car extends Sprite
	{
		var _CarSprite:CarSym;
		var _CarCurrDirection:String;
		var _CarColor:String;

		public function Car(iDirection:String, iColor:String)
		{
			_CarSprite = new CarSym();
			this.addChild(_CarSprite);

			_CarCurrDirection = iDirection;
			_CarColor = iColor;

			_CarSprite.gotoAndPlay("fCar" + _CarColor + _CarCurrDirection);

		}

		public function setDirection(iDirection:String)
		{
			_CarCurrDirection = iDirection;
			_CarSprite.gotoAndPlay("fCar" + _CarColor + _CarCurrDirection);
		}

		public function getDirection():String
		{ return _CarCurrDirection; }
	}
}
```

Stage's Items

```
var _MainStage = new MainStageSym();
this.addChild(_MainStage);
_MainStage.x = 110;
_MainStage.y = 110;


var _MainStageWindow = new MainStageWindowSym();
this.addChild(_MainStageWindow);

GlobalVarContainer.vars.sounds.stopSong();
var step:int = 5;
GlobalVarContainer.vars.sounds.playSong("sound2.mp3");

var light1 = new LightSym();
_MainStage.addChild(light1);
light1.x = 272;
light1.y = 225;
//light1.y = 25;
light1.gotoAndPlay("fLightRed");

var light2 = new LightSym();
_MainStage.addChild(light2);
light2.x = 360;
light2.y = 225;
light2.gotoAndPlay("fLightRed");

var light3 = new LightSym();
_MainStage.addChild(light3);
light3.x = 380;
light3.y = 235;
light3.gotoAndPlay("fLightRed");

var light4 = new LightSym();
_MainStage.addChild(light4);
light4.x = 272;
light4.y = 319;
light4.gotoAndPlay("fLightRed");

var police = new PoliceSym();
_MainStage.addChild(police);
police.x = 320;
police.y = 275;

var myCar = new Car("Right", "Blue");
_MainStage.addChild(myCar);
myCar.x = 110;
myCar.y = 90;

var redCar = new Car("Down", "Red");
_MainStage.addChild(redCar);
redCar.x = 530;
redCar.y = 490;

var yellowCar = new Car("Right", "Yellow");
_MainStage.addChild(yellowCar);
yellowCar.x = 445;
yellowCar.y = 545;

var messages:MessagesSymbol = new MessagesSymbol();
this.addChild(messages);
messages.x = 570;
messages.y = 298;
messages.gotoAndPlay("fPanelOk");

var scoreInt:int;
scoreInt = 0;
var score = new ScoreSymbol();
this.addChild(score);
score.x = 572;
score.y = 200;
```

```
score.scoreText.text = scoreInt;
```

Timers

```
var timerPeriod2 = 2850;
var timerIsStarted2:Boolean;
var scanTimer2:Timer = null;
initTimer2(timerPeriod2);
startTimer2();

function initTimer2(iPeriod:int)
{
        scanTimer2 = new Timer(iPeriod);
        scanTimer2.addEventListener(TimerEvent.TIMER, startScan2_Handler, false, 0, true);
}

function startTimer2()
{
        timerIsStarted2 = true;
        scanTimer2.start();
}

function stopTimer2()
{
        timerIsStarted2 = false;
        scanTimer2.stop();
}

function startScan2_Handler($evt:TimerEvent):void
{
        changeLights();
}

////////////////// Timer ////////////////////
var timerPeriod = 50;
var timerIsStarted:Boolean;
var scanTimer:Timer = null;
initTimer(timerPeriod);
startTimer();

function initTimer(iPeriod:int)
{
        scanTimer = new Timer(iPeriod);
        scanTimer.addEventListener(TimerEvent.TIMER, startScan_Handler, false, 0, true);
}

function startTimer()
{
        timerIsStarted = true;
        scanTimer.start();
}

function stopTimer()
{
        timerIsStarted = false;
        scanTimer.stop();
}
```

Change Lights and Policeman

```
function changeLights()
{
	if(light1.currentLabel != "fLightRed")
	{
		light1.gotoAndPlay("fLightRed");
		light2.gotoAndPlay("fLightRed");
		light3.gotoAndPlay("fLightGreen");
		light4.gotoAndPlay("fLightGreen");
		police.gotoAndPlay("fPoliceLeftRight");
	}
	else
	{
		light1.gotoAndPlay("fLightGreen");
		light2.gotoAndPlay("fLightGreen");
		light3.gotoAndPlay("fLightRed");
		light4.gotoAndPlay("fLightRed");
		police.gotoAndPlay("fPoliceUpDown");
	}
}
```

Move Other Cars

```
function moveYellowCar()
{
	if(yellowCar.x == 305 && yellowCar.y == 45 && yellowCar.getDirection() != "Down")
		yellowCar.setDirection("Down");
	else if(yellowCar.x == 585 && yellowCar.y == 545 && yellowCar.getDirection() != "Up")
		yellowCar.setDirection("Up");
	else if(yellowCar.x == 305 && yellowCar.y == 545 && yellowCar.getDirection() !=
"Right")
		yellowCar.setDirection("Right");
	else if(yellowCar.x == 585 && yellowCar.y == 45 && yellowCar.getDirection() != "Left")
		yellowCar.setDirection("Left");
	else
	{
		if(yellowCar.getDirection() == "Right")
		{
			if(isValidMovement(yellowCar.x + step + yellowCar.width/2,
yellowCar.y, 1) != false)
				yellowCar.x += step;
		}
		else if(yellowCar.getDirection() == "Up")
		{
			if(isValidMovement(yellowCar.x, yellowCar.y - step -
yellowCar.height/2, 1) != false)
				yellowCar.y -= step;
		}
		else if(yellowCar.getDirection() == "Down")
		{
			if(isValidMovement(yellowCar.x, yellowCar.y + step +
yellowCar.height/2, 1) != false)
				yellowCar.y += step;
		}
		else
		{
			if(isValidMovement(yellowCar.x - step - yellowCar.width/2,
yellowCar.y, 1) != false)
			yellowCar.x -= step;
		}
	}
}

function moveRedCar()
{
	if(redCar.x == 530 && redCar.y == 490 && redCar.getDirection() != "Left")
		redCar.setDirection("Left");
	else if(redCar.x == 100 && redCar.y == 490 && redCar.getDirection() != "Up")
		redCar.setDirection("Up");
	else if(redCar.x == 100 && redCar.y == 295 && redCar.getDirection() != "Right")
		redCar.setDirection("Right");
	else if(redCar.x == 530 && redCar.y == 295 && redCar.getDirection() != "Down")
		redCar.setDirection("Down");
	else
	{
		if(redCar.getDirection() == "Right")
		{
			if(isValidMovement(redCar.x + step + redCar.width/2, redCar.y, 2) !=
false)
				redCar.x += step;
		}
		else if(redCar.getDirection() == "Up")
		{
			if(isValidMovement(redCar.x, redCar.y - step - redCar.height/2, 2) !=
false)
				redCar.y -= step;
		}
		else if(redCar.getDirection() == "Down")
		{
			if(isValidMovement(redCar.x, redCar.y + step + redCar.height/2, 2) !=
false)
				redCar.y += step;
```

```
                }
                else
                {
                        if(isValidMovement(redCar.x - step - redCar.width/2, redCar.y, 2) !=
false)
                                redCar.x -= step;
                }
        }
}

function startScan_Handler($evt:TimerEvent):void
{
        moveRedCar();
        moveYellowCar();
}
```

Check if the movement is Valid

```
function isValidMovement(iDimX:int, iDimY:int, iCarIndex:int):Boolean
{
	if(iCarIndex !=1 && iCarIndex !=2)
	{
		if(iDimX < 27 || iDimX > 610 || iDimY < 27 || iDimY > 860)
			return false;

		if(iDimX > 130 && iDimY > 110 && iDimX < 280 && iDimY < 240)
			return false;

		if(iDimX > 355 && iDimY > 110 && iDimX < 505 && iDimY < 240)
			return false;

		if(iDimX > 130 && iDimY > 310 && iDimX < 280 && iDimY < 460)
			return false;

		if(iDimX > 355 && iDimY > 310 && iDimX < 505 && iDimY < 460)
			return false;

		if(iDimX > 130 && iDimY > 570 && iDimX < 280 && iDimY < 735)
			return false;

		if(iDimX > 355 && iDimY > 570 && iDimX < 505 && iDimY < 735)
			return false;
	}

	if(iCarIndex !=0)
	{
		if(iDimX > myCar.x - myCar.width && iDimY > myCar.y - myCar.height &&
		iDimX < myCar.x + myCar.width && iDimY < myCar.y + myCar.height)
			return false;
	}

	if(iCarIndex !=1)
	{
		if(iDimX > yellowCar.x - yellowCar.width && iDimY > yellowCar.y -
yellowCar.height &&
		   iDimX < yellowCar.x + yellowCar.width && iDimY < yellowCar.y +
yellowCar.height)
		{
			if(iCarIndex ==0)
			{
				scoreInt--;
				score.scoreText.text = scoreInt;

				if(messages.currentLabel != "fPanelOtherCar")
					messages.gotoAndPlay("fPanelOtherCar");
			}
			return false;
		}
	}

	if(iCarIndex !=2)
	{
		if(iDimX > redCar.x - redCar.width && iDimY > redCar.y - redCar.height &&
			iDimX < redCar.x + redCar.width && iDimY < redCar.y + redCar.height)
		{
			if(iCarIndex ==0)
			{
				scoreInt--;
				score.scoreText.text = scoreInt;

				if(messages.currentLabel != "fPanelOtherCar")
					messages.gotoAndPlay("fPanelOtherCar");
			}
			return false;
		}
```

```
        }

        if(iDimX >= 155 && iDimX <= 485 && iDimY >= 254 && iDimY <= 294)
        {
                if(iCarIndex ==0)
                {
                        if( myCar.getDirection() == "Right" && iDimY != 294 ||
                                myCar.getDirection() == "Left" && iDimY != 254)
                        {
                                scoreInt--;
                                score.scoreText.text = scoreInt;

                                if(messages.currentLabel != "fPanelOneway")
                                        messages.gotoAndPlay("fPanelOneway");

                                return false;
                        }
                }
        }

        if(iDimX >= 300 && iDimX <= 338 && iDimY >= 180 && iDimY <= 185 && light1.currentLabel
== "fLightRed")
        {
                if(iCarIndex ==0)
                {
                        scoreInt--;
                        score.scoreText.text = scoreInt;

                        if(messages.currentLabel != "fPanelRedLight")
                                messages.gotoAndPlay("fPanelRedLight");
                }

                return false;
        }

        if(iDimX >= 213 && iDimX <= 217 && iDimY >= 293 && iDimY <= 296 && light4.currentLabel
== "fLightRed")
        {
                if(iCarIndex ==0)
                {
                        scoreInt--;
                        score.scoreText.text = scoreInt;

                        if(messages.currentLabel != "fPanelRedLight")
                                messages.gotoAndPlay("fPanelRedLight");
                }
                return false;
        }

        if(iDimX >= 423 && iDimX <= 426 && iDimY >= 253 && iDimY <= 255 && light3.currentLabel
== "fLightRed")
        {
                if(iCarIndex ==0)
                {
                        scoreInt--;
                        score.scoreText.text = scoreInt;

                        if(messages.currentLabel != "fPanelRedLight")
                                messages.gotoAndPlay("fPanelRedLight");
                }
                return false;
        }

        if(iCarIndex == 0)
        {
                //if(iDimX >= 295 && iDimX <= 345 && iDimY >= 90 && iDimY <= 470 &&
myCar.getDirection() == "Up")
                if(iDimX >= 295 && iDimX <= 345 && iDimY >= 90 && iDimY <= 735 &&
myCar.getDirection() == "Up")
                {
                        scoreInt--;
                        score.scoreText.text = scoreInt;

                        if(messages.currentLabel != "fPanelDontEnter")
```

```
                                messages.gotoAndPlay("fPanelDontEnter");

                        return false;
                }
        }

        if(messages.currentLabel != "fPanelOk" && iCarIndex == 0)
                messages.gotoAndPlay("fPanelOk");

        if(iCarIndex == 0)
        {
                scoreInt++;
                score.scoreText.text = scoreInt;
        }

        return true;
}
```

Move our Car

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, keyDownHandler);
function keyDownHandler(event:KeyboardEvent):void
{
	//trace(event.keyCode);
	switch (event.keyCode)
	{
		case Keyboard.RIGHT:
			if( myCar.getDirection() == "Right")
			{
				if(isValidMovement(myCar.x + step + myCar.width/2, myCar.y,
0) != false)
				{
					myCar.x += step;
					_MainStage.x -= step;
				}
			}
			else
				myCar.setDirection("Right");
			break;
		case Keyboard.LEFT:
			if( myCar.getDirection() == "Left")
			{
				if(isValidMovement(myCar.x - step - myCar.width/2, myCar.y,
0) != false)
				{
					myCar.x -= step;
					_MainStage.x += step;
				}
			}
			else
				myCar.setDirection("Left");
			break;
		case Keyboard.UP:
			if( myCar.getDirection() == "Up")
			{
				if(isValidMovement(myCar.x, myCar.y - step - myCar.height/2,
0) != false)
				{
					myCar.y -= step;
					_MainStage.y += step;
				}
			}
			else
				myCar.setDirection("Up");
			break;
		case Keyboard.DOWN:
			if( myCar.getDirection() == "Down")
			{
				if(isValidMovement(myCar.x, myCar.y + step + myCar.height/2,
0) != false)
				{
					myCar.y += step;
					_MainStage.y -= step;
				}
			}
			else
				myCar.setDirection("Down");
			break;
		case 83: //s
			_MainStage.y += 50;
			break;
		case 68: //d
			_MainStage.x += 50;
			break;
		case 65: //a
			_MainStage.x -= 50;
			break;
		case 87: //w
			_MainStage.y -= 50;
```

```
                        break;
                default:
                        break;
        }
}
```

Move People

```
var peopleDirection = 1;
function movePeople():void
{
        //people1.x = 827;
        if(people1.x == 735)
                peopleDirection = 1;
        else if(people1.x == 830)
                peopleDirection = 2;

        if(people2.x == 1210)
                peopleDirection = 2;
        else if(people2.x == 1300)
                peopleDirection = 1;


        if(peopleDirection == 1)
        {
                people1.x += 1;
                people2.x -= 1;
        }
        else
        {
                people1.x -= 1;
                people2.x += 1;
        }
}
```