Kod ActionScript 3.0 służący do obsługi galerii zdjęć z ćwiczenia 3

```
import flash.events.MouseEvent;
import flash.display.MovieClip;
import flash.display.DisplayObject;
import flash.utils.Timer;
import flash.events.TimerEvent;

var currentImg:MovieClip;
var oldImg:MovieClip;

function showFoto(photo_):void {
    photo_.gotoAndPlay(2);
}

function hideFoto(photo_):void {
    photo_.gotoAndPlay(21);
}
function onPhotoRelease(myevent:MouseEvent):void {
      currentImg=MovieClip(myevent.target);
      if(oldImg==null){
            showFoto(currentImg);
            oldImg=currentImg;
            }
      else{
            if(currentImg==oldImg){
                  hideFoto(currentImg);
                  oldImg=null;
               }
            else{
            hideFoto(oldImg);
            showFoto(currentImg);
            oldImg=currentImg;
            }
      }
}
var klipy:Array = new Array();
var i:Number=0;

//uzupełnienie tablicy
while(1){
      i++;
      var nazwa:String = "fot"+i+"_mc";
      var obj:DisplayObject = getChildByName(nazwa);

      if (obj == null){
            break;
            }
      klipy.push(obj);
}

//nasłuchiwanie zdarzeń myszki dla odbiorników (klipów)
for each (var element in klipy){
      element.addEventListener(MouseEvent.CLICK,onPhotoRelease);
}
```

```
//autoodtwarzanie galerii

auto_btn.addEventListener(MouseEvent.CLICK, autoPlayGallery);

var intervals:int = 3000;
var klipIlosc:int = klipy.length;
var timerGallery:Timer = new Timer(intervals);
var statusAuto:Boolean;

function autoPlayGallery(evt:MouseEvent):void{
      if(statusAuto==false){
            statusAuto=true; //przestawienie przycisku na możliwość
            zatrzymania trybu auto
            timerGallery.start();
            timerGallery.addEventListener(TimerEvent.TIMER,
timerGalleryHandler);}
      else{timerGallery.stop(); statusAuto=false;}
}

function timerGalleryHandler(event:TimerEvent):void{

      var i:int = event.target.currentCount;

            if(i==1){
                  if(oldImg != null){
                        hideFoto(oldImg)
                        oldImg=null;
                        currentImg=null;}

                  currentImg = klipy[i-1];
                  showFoto(currentImg);
                  oldImg=currentImg;}

            else if(i<klipIlosc+1){
                  currentImg = klipy[i-1];
                  showFoto(currentImg);
                  hideFoto(oldImg);
                  oldImg=currentImg;}
            else{
                  timerGallery.stop();
                  timerGallery.reset();
                  statusAuto=false;}
}
```

Kod ActionScript 3.0 służący do zmiany tła, jak w przykładzie pokazanym na zajęciach.

```
var bgshape:Sprite;
var default_bg_color:uint = 0x000000;
var current_bg_color:uint;
var new_bg_color:uint;
var kkRed:uint;
var kkGreen:uint;
var kkBlue:uint;
var kpRed:uint;
var kpGreen:uint;
var kpBlue:uint;
var delay:Number = 10;
var repeat:uint = 20; //second parameter of Timer;
var myTimer:Timer = new Timer(delay, repeat);
var transformLoop:uint;

bgshape = new Sprite();
    bgshape.graphics.beginFill(default_bg_color);
    bgshape.graphics.drawRect(0-(stage.stageWidth/2),0,
         stage.stageWidth*2, stage.stageHeight);
    addChildAt(bgshape, 0);
    stage.addEventListener(Event.RESIZE, resizeBGWithStage);

red_btn.addEventListener(MouseEvent.CLICK,colorBase);
blue_btn.addEventListener(MouseEvent.CLICK,colorBase);
violet_btn.addEventListener(MouseEvent.CLICK,colorBase);
orange_btn.addEventListener(MouseEvent.CLICK,colorBase);
black_btn.addEventListener(MouseEvent.CLICK,colorBase);

function colorBase(event:MouseEvent):void{

     myTimer.start();
     myTimer.addEventListener(TimerEvent.TIMER, timerHandler);
     myTimer.addEventListener(TimerEvent.TIMER_COMPLETE, bgHandler);

     if(event.target==red_btn){
          new_bg_color = 0xFF0000;
          kkRed=255;
          kkGreen=0;
          kkBlue=0;}
     else if(event.target==blue_btn){
          new_bg_color = 0x0000CC;
          kkRed=0;
          kkGreen=0;
          kkBlue=200;}
     else if(event.target==violet_btn){
          new_bg_color = 0x643264;
          kkRed=100;
          kkGreen=50;
          kkBlue=100;}
     else if(event.target==orange_btn){
          new_bg_color = 0xFF6600;
          kkRed=250;
          kkGreen=100;
```

```
           kkBlue=0;}
     else if(event.target==black_btn){
           new_bg_color = 0x000000;
           kkRed=0;
           kkGreen=0;
           kkBlue=0;}
     else{current_bg_color = default_bg_color;
           kpRed=0;
           kpGreen=0;
           kpBlue=0;}
}


function timerHandler(event:TimerEvent):void{
           transformLoop = event.target.currentCount;
           var myColorTransform = new ColorTransform();
           myColorTransform.color = current_bg_color;
           myColorTransform.redOffset = kpRed + Math.floor((kkRed-
kpRed)/repeat)*transformLoop;
           myColorTransform.blueOffset = kpBlue + Math.floor((kkBlue-
kpBlue)/repeat)*transformLoop;
           myColorTransform.greenOffset = kpGreen + Math.floor((kkGreen-
kpGreen)/repeat)*transformLoop;
           bgshape.transform.colorTransform = myColorTransform;
}

function bgHandler(event:TimerEvent):void{
           myTimer.reset();
           current_bg_color = new_bg_color;
           kpRed=kkRed;
           kpGreen=kkGreen;
           kpBlue=kkBlue;
}

function resizeBGWithStage(e:Event){
    try {
        bgshape.width = stage.stageWidth*2;
        bgshape.height = stage.stageHeight;
    } catch(e){}
}
```