# ZADANIE 1

Wzbogacenie galerii zdjęć o efekt dynamicznej zmiany koloru (obiekt ColorTransform)

## Rysowanie obiektu tła

Oto kod, który narysuje prosty obiekt graficzny - prostokąt o wymiarach stołu montażowego:



Można dodać też kod, który będzie dopasowywał wielkość prostokąta do aktualnej wielkości stołu montażowego: stage.addEventListener(Event.RESIZE, resizeBGWithStage);

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

## Zmiana koloru tła

Dodaj do projektu przycisk i nadaj mu nazwę instancji (np. red\_btn). Napisz kod, który będzie powodował zmianę koloru wypełnienia dużego prostokąta na kolor eksponowany na przycisku – w tym przykładzie jest to kolor czerwony. Są dwie możliwości. W obydwu wykorzystywany jest obiekt ColorTransform:

1: Pierwsza możliwość powoduje zmianę koloru z wykorzystaniem tylko właściwości color obiektu ColorTransform.

Kod:

```
function colorBase(event:MouseEvent):void{
    var myColorTransform = new ColorTransform();
```

**Grafika animacyjna**, przedmiot bloku obieralnego **Grafika i Multimedia** dla kierunku Informatyka, semestr VI Prowadzący: dr Joanna Sekulska-Nalewajko, Instytut Informatyki Stosowanej Politechniki Łódzkiej, jsekulska@kis.p.lodz.pl

```
myColorTransform.color = 0xCC0000;
bgshape.transform.colorTransform = myColorTransform;
}
red btn.addEventListener(MouseEvent.CLICK,colorBase);
```

2: Druga możliwość powoduje zmianę koloru z wykorzystaniem właściwości redOffset, blueOffset lub greenOffset obiektu ColorTransform.

```
function colorBase(event:MouseEvent):void{
    var myColorTransform = new ColorTransform();
    myColorTransform.color = default_bg_color;
    myColorTransform.redOffset = 200;
    bgshape.transform.colorTransform = myColorTransform;
}
red btn.addEventListener(MouseEvent.CLICK,colorBase);
```

### Płynna zmiana koloru tła

Dodaj do projektu modyfikację polegającą na <u>płynnej</u> zmianie koloru wypełnienia prostokąta. Zadanie to umożliwi obiekt **Timer()**.

#### Propozycja rozwiązania tego zadania na bazie już istniejącego kodu z poprzednich zadań:

Całkowitej zmianie ulega funkcja colorBase, która określa to co stanie się po kliknięciu przycisku. Ta funkcja nie będzie już bezpośrednio odpowiedzialna za transformację koloru, tak jak dotychczas, tylko będzie uruchamiała obiekt Timer (uruchamiała odliczania kroków, które przebiegają w określonych okresach czasowych) oraz wywoła funkcję timerHandler, która to od momentu kliknięcia przycisku wykona się tyle razy ile przewiduje wyjściowy parametr repeat.

Parametr repeat określa ile interwałów czasowych ma upłynąć do zatrzymania licznika. Ilość interwałów będzie wykorzystana do uzyskania efektu krokowej zmiany koloru – z każdym krokiem zmiana będzie coraz bardziej widoczna. Długość trwania interwału (okresu) określa pierwszy z parametrów wyjściowych obiektu Timer(), podawany jest w milisekundach.

Ilość kroków transformacji koloru wykorzystywana jest w funkcji timerHandler. Wiemy już, że dzięki repeat licznik obiektu Timer zliczy 20 wykonań. Ponieważ zakładamy transformację wypełnienia z koloru czarnego do czerwonego to wiemy też, że przesunięcie koloru musi nastąpić w obrębie składowej czerwonej z wartości 0 na około 200. Dzięki wykorzystaniu obiektu Timer możemy przy każdym okresie trwającym milisekundę (lub dłużej jeśli zwiększymy wartość opóźnienia) zwiększać udział barwy czerwonej np. o 10. Przy 20-tym okresie całkowite przesunięcie w obrębie czerwieni wynosić będzie 200. Pozostaje jeszcze rozwiązać zagadkę: skąd wiadomo kiedy licznik obiektu Timer zakończy odliczanie i jak w ogóle wykorzystać ten obiekt dla naszych celów, czyli do stopniowej zmiany koloru? Tu z pomocą przychodzi właściwość currentCount obiektu zdarzenia TimerEvent. Obiekt Timer wywołuje obiekt zdarzenia TimerEvent za każdym razem po upływie zadanego przedziału czasu, a wtedy wartość currentCount zwiększa się iteracyjnie.

#### Kod do rozwiązania zadania:

//powołanie dodatkowej zmiennej, która zapobiega nieoczekiwanej zmianie koloru
var statusBlack:Boolean;

**Grafika animacyjna**, przedmiot bloku obieralnego **Grafika i Multimedia** dla kierunku Informatyka, semestr VI Prowadzący: dr Joanna Sekulska-Nalewajko, Instytut Informatyki Stosowanej Politechniki Łódzkiej, jsekulska@kis.p.lodz.pl

```
statusBlack = true;
function colorBase(event:MouseEvent):void{
      if(statusBlack==true) {
           myTimer.start();
           myTimer.addEventListener(TimerEvent.TIMER, timerHandler);
            statusBlack=false;}
}
var delay:Number = 1;
var repeat:uint = 20;
var myTimer:Timer = new Timer(delay, repeat);
var transformLoop:Number;
function timerHandler(event:TimerEvent):void{
      transformLoop = event.target.currentCount;
     var myColorTransform = new ColorTransform();
     myColorTransform.color = default_bg_color;
     myColorTransform.redOffset = transformLoop*10;
     bgshape.transform.colorTransform = myColorTransform;
}
```

Zadanie samodzielne: Płynna zmiana koloru z udziałem wielu przycisków.

Dodaj do projektu więcej przycisków, które będą służyć do zmiany koloru prostokąta, oczywiście zmiana koloru tak jak poprzednio ma przebiegać w płynny sposób. Przyciski muszą mieć nazwy instancji. Należy dobrze określić kolory, jakie chcemy wykorzystać i zapamiętać, lub zapisać wartość składowych RGB danego koloru, np. kolor czerwony (250,0,0).

