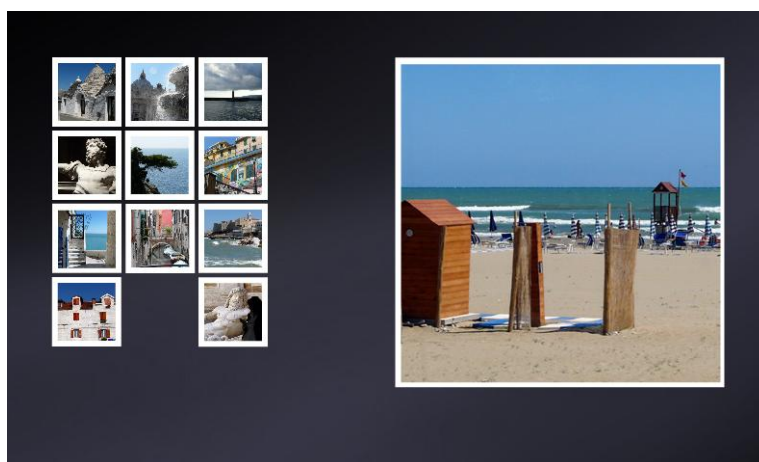


ZADANIE 1

Tworzenie galerii zdjęć z wykorzystaniem stacjonarnych klipów filmowych .

Galeria może posiadać od kilku do kilkunastu zdjęć (klipów), które w postaci miniatur położone są w równych odstępach z lewej strony projektu. W podstawowej postaci tego projektu po naciśnięciu miniatury odtwarzana jest animacja, która płynnie powiększa małe zdjęcie. Po wybraniu kolejnej miniatury powiększone zdjęcie płynnie zmniejsza się i powraca na miejsce. W dalszych etapach pracy projekt będzie modyfikowany. Zostaną dodane funkcje:

- zmniejszanie się zdjęcia po naciśnięciu jego powiększenia;
- opcja Full Screen całej galerii;
- zmiana tła galerii;
- autoodtwarzanie.



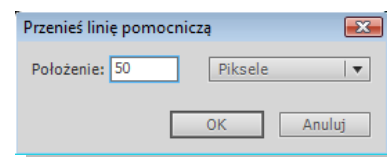
Ustawienie stołu montażowego

Otwieramy nowy plik i nazywamy go (np. galeria.fla). Ustalamy wielkość pola roboczego na 800 x 600 pikseli i ustawiamy ciemny kolor tła.

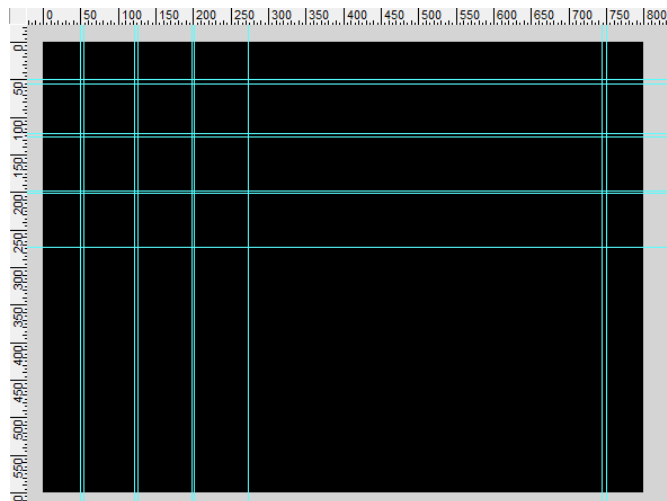
Klip w miniaturze powinien mieć wielkość 72x72 piksele (wielkość umowna). Położenie pierwszego klipu względem stołu montażowego to $x=50$ i $y=50$. Po powiększeniu klip ma rozmiar 342x342 (to też wielkość umowna) i leży 50 px od górnej i 50 px od prawej krawędzi sceny.

Należy na początek zadbać o równe rozmieszczenie miniatur (klipów), które będą stopniowo przygotowywane. Może posłużyć do tego siatka (najwygodniejszy sposób) lub prowadnice lub po prostu operowanie współrzędnymi klipów.

Jeśli korzystamy z prowadnic, to ich ustawienie jest dość proste: musimy mieć widoczne linijki (dostęp przez menu Widok), z których ściągamy linie pomocnicze (prowadnice). Klikając dwukrotnie na wprowadzoną linię możemy ustawić jej dokładne położenie, jak pokazane jest na rysunku.



Podstawowe prowadnice pionowe i poziome (dla 9 klipów) umieszczone są w położeniu: 50, 122, 126, 198, 202 i 274. Dodajemy także prowadnice pionowe w położeniu 744 i 750 oraz dodatkową poziomą i pionową prowadnicę 56, które to wyznaczają nam krawędzie, do których będą przylegać duże zdjęcia wyświetlane po kliknięciu miniatur.

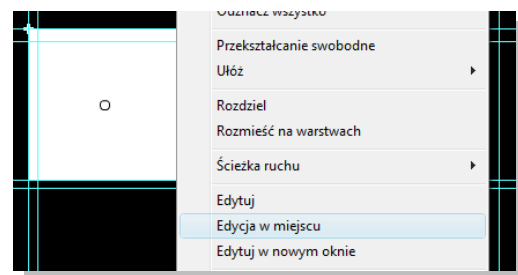


Praca z klipami

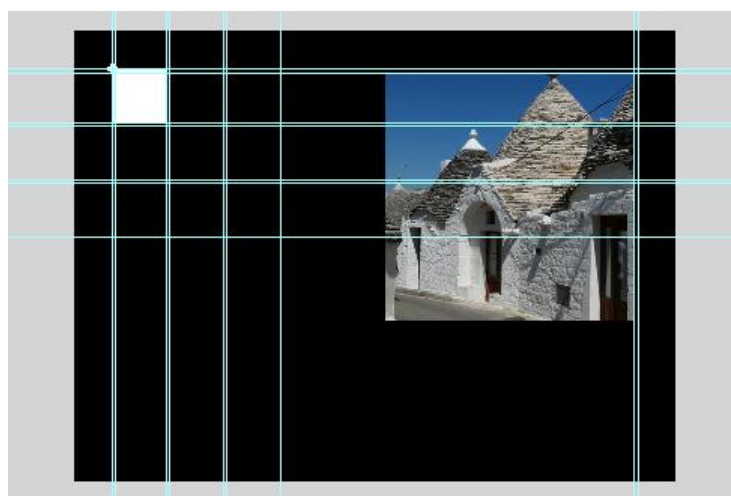
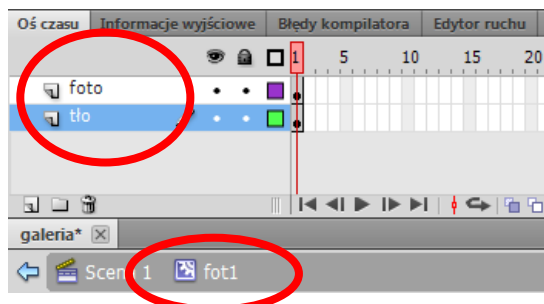
Importujemy zdjęcia do biblioteki (menu Plik > Importuj > Importuj do biblioteki...).

W położeniu $x=y=50$ (czyli w pierwszym „oczku” siatki prowadnic) umieszczamy biały kwadrat o boku 72 px, najlepiej bez konturu. Od razu konwertujemy go do symbolu grafiki (skrót do konwersji F8) a potem jeszcze raz konwertujemy, tym razem do klipa filmowego. Grafikę możemy nazwać „kwadrat” a klip „fot1”.

Edytujemy klip w miejscu – wystarczy kliknąć na klipie dwa razy lub wybrać z menu podręcznego polecenie **Edycja w miejscu**. Dzięki edycji w miejscu będzie widoczna siatka prowadnic ustawioną na stole montażowym oraz pozostałe elementy umieszczone na stole, jeśli takowe istnieją.



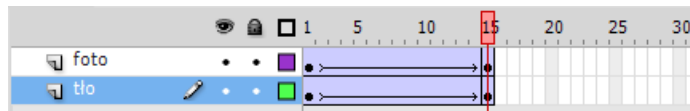
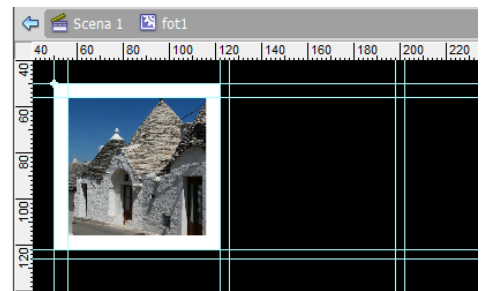
W klipie dodajemy warstwę (warstwy w klipie możemy nazwać „foto” i „tło”), na której umieszczamy pierwsze zdjęcie (pobieramy je z biblioteki). Umieszczamy je z prawej strony pola roboczego przy pionowej prowadnicy „744” i poziomej „56”.



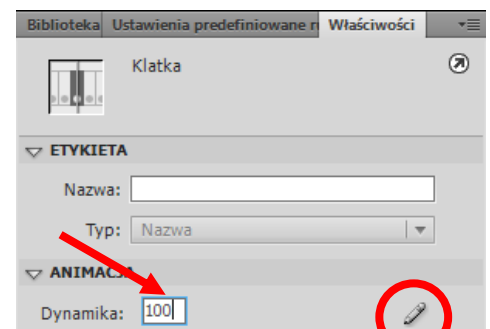
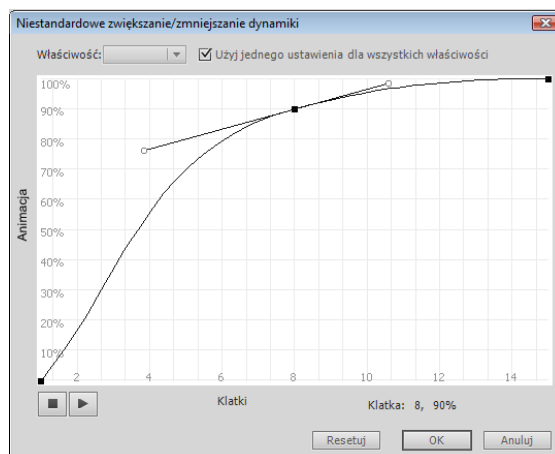
Zdjęcie jest duże (330x330 px) i tak będzie wyglądać po powiększeniu, natomiast musi wystartować jako miniatura (60x60 px). Wobec tego stosujemy animację ruchu (polecenie Utwórz klasyczną animację). Animacja ta powinna przebiegać do klatki nr 15, więc tam na warstwie „foto” dodajemy klatkę kluczową (skrót F6).

Wracamy do klatki nr 1, w której zdjęcie zmniejszamy do wielkości 60x60 pikseli i umieszczamy je dokładnie pośrodku białego kwadratu.

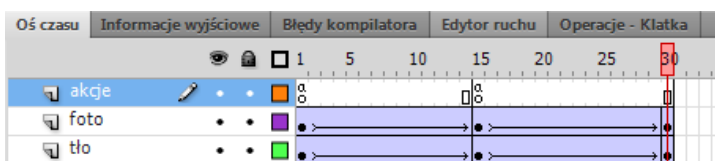
Podobną animację zastosujemy dla białego kwadratu - od klatki 1 do 15. W klatce nr 1 zachowujemy rozmiar 72x72 px, natomiast w klatce nr 15 powiększamy kwadrat do rozmiaru 342x342 px i umieszczamy go przy pionowej prowadnicy „750” i poziomej „50”.



Do animacji zastosujemy własne ustawienia **Dynamiki**. Zaznaczamy obie klatki nr 1 i w panelu **Właściwości** ustawiamy wartość **Dynamiki** na **100**, a następnie zmieniamy te ustawienia w oknie **Niestandardowe zwiększanie / zmniejszanie dynamiki** – okno to otworzy się po kliknięciu ołówka:

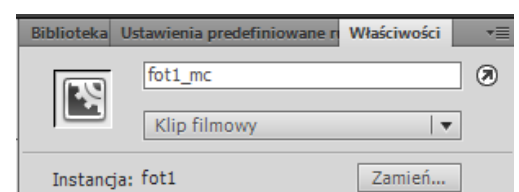


Teraz w obu warstwach kopiujemy klatkę nr 1 (**Kopiuj klatki**) i wklejamy ją w klatce nr 30 (**Wklej klatki**) i tym samym odwracamy animację. W klatkach nr 15 możemy zmienić też dynamikę, tak jak powyżej.



Dodajemy warstwę „akcje”, w której w klatce 1 i 15 wstawiamy akcje stop ().

To koniec pracy z tym klipem. Wracamy więc do głównej sceny. Zaznaczamy umieszczony w niej klip i wprowadzamy nazwę jego instancji: **fot1_mc**.



W podobny sposób przygotowujemy kolejnych 8 klipów ze zdjęciami. Oczywiście dbając o odpowiednie rozmieszczenie zdjęć w siatce. Instancje klipów nazywamy odpowiednio **fot2_mc**, **fot3_mc**, **fot4_mc** itd.

Kod:

Na głównej scenie dodajemy warstwę i nazywamy ją „akcje”. W klatce tej warstwy umieszczamy podstawowy skrypt sterujący galerią. Zawiera on trzy funkcje – dwie sterujące listwą wewnątrz klipów oraz funkcja główna wywołująca te funkcje w zależności od istniejących warunków.

```
import flash.events.MouseEvent;
import flash.display.MovieClip;

var currentImg:MovieClip;
var oldImg:MovieClip;

function onPhotoRelease(myevent:MouseEvent):void {
    currentImg=MovieClip(myevent.target);
    if(oldImg==null){
        showFoto(currentImg);
        oldImg=currentImg;
    }
    else{
        hideFoto(oldImg);
        showFoto(currentImg);
        oldImg=currentImg;
    }
}

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

function hideFoto(photo_):void {
    photo_.gotoAndPlay(21);
}
```

Należy jeszcze określić, które obiekty będą reagowały i wykonywały nasze funkcje (będą odbiornikami zdarzenia myszki - MouseEvent). Są to oczywiście klipy ze zdjęciami umieszczone na scenie głównej i oznaczone nazwami instancji, po to aby były „widoczne” dla skryptów AS.

Napisz kod, który nasłuchuje zdarzeń myszki i wywołuje napisane wyżej funkcje, najlepiej aby korzystał on z tablicy przycisków.

Do wykorzystania:

- metoda **addEventListener** - ma dwa parametry – pierwszym z nich jest typ zdarzenia, które chcemy wykryć (dla nas jest to CLICK), a drugim jest funkcja reagująca na zdarzenie (w naszym przypadku jest to nasza funkcja główna **onPhotoRelease**).
- tablica uzupełniona nazwami instancji klipów: `var klipy:Array = new Array();`
- pętla do przeglądania zawartość tablicy: `for each (var element in klipy){...}`

Tablicę można uzupełnić „ręcznie” wszystkimi nazwami instancji klipów, możemy także spowodować automatyczne dołączenie wszystkich obiektów ekranowych znajdujących się na liście wyświetlania, które spełniają zadane parametry. Musimy wówczas powołać instancję nowego obiektu DisplayObject i przypisać mu instancję obiektu wyświetlanego o podanej nazwie:

```
var obj:DisplayObject = getChildByName(nazwa);
```

Metoda `getChildByName(name:String):DisplayObject` zwraca instancję podrzędnego obiektu wyświetlanego o podanej nazwie. Jeżeli więcej niż jeden podrzędny obiekt wyświetlany nosi tę określoną nazwę, metoda zwraca pierwszy obiekt z listy obiektów podrzędnych.

Do uzupełniania tablicy należy użyć pętli `while` oraz metody `push` tablicy.

ZADANIE 2

Modyfikacja kodu. Zmodyfikuj kod obsługujący wyświetlanie klipów, tak, aby kliknięcie powiększenie klipu spowodowało jego zmniejszenie i powrót do pozycji wyjściowej.

ZADANIE 3

Opcja pełnego ekranu.

Wykonaj przycisk obsługujący opcję Full Screen, umieść go w projekcie galerii i nadaj mu nazwę instancji.

W klatkę głównej sceny wpisz kod obsługujący ten efekt:

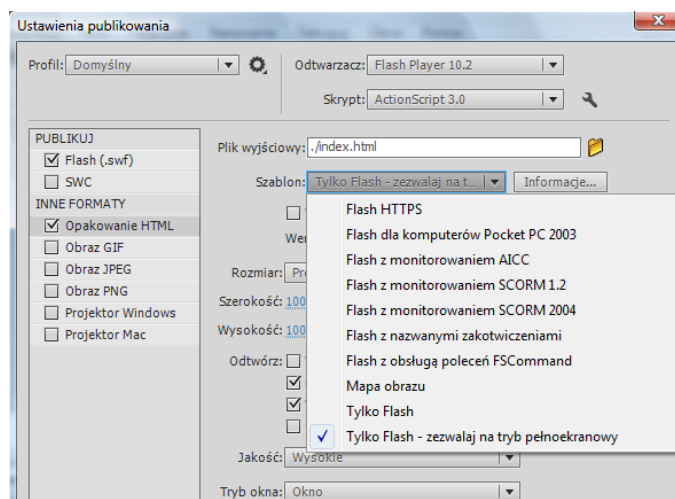
```
import flash.events.MouseEvent;

fullScreen_btn.addEventListener(MouseEvent.CLICK, fsMode);

var statusFS:Boolean = false;

function fsMode(event:MouseEvent):void{
    if(statusFS==false){
        stage.displayState = StageDisplayState.FULL_SCREEN;
        statusFS=true;
    }
    else
    {
        stage.displayState = StageDisplayState.NORMAL;
        statusFS=false;
    }
}
```

W sekcji **Ustawienia Publikowania** dla formatu HTML użyj szablonu Tylko Flash - zezwalaj na tryb pełnoekranowy i opublikuj projekt.



ZADANIE 4

Autoodtwarzanie galerii.

Galerię utworzoną na wcześniejszych ćwiczeniach zmodyfikuj tak, aby można było włączyć automatyczne odtwarzanie zdjęć. Potrzebny jest przycisk, który będzie odpowiedzialny zarówno za rozpoczęcie odtwarzania jak i jego zatrzymanie w dowolnym momencie. Poza tym galerię można też przeglądać ręcznie tak jak do tej pory.

W tym zadaniu do wykorzystania jest obiekt **Timer()** z interwałem czasowym trwającym ok. 3-4 sekundy. Uwaga, interwał jako parametr podawany jest w milisekundach.

Przykładowe elementy do zadania:

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

Obiekt Timer umożliwia uruchamianie kodu w określonym momencie. Powyższy konstruktor tworzy nowy obiekt Timer o określonym opóźnieniu. Drugim parametrem poza opóźnieniem (opcjonalnym) może być ilość powtórzeń.

[Timer](#)(delay:[Number](#), repeatCount:[int](#))

W celu uruchomienia obiektu Timer należy użyć metody `start()`.

Do zatrzymania `stop()`.

W celu skonfigurowania kodu, jaki ma zostać wykonany dla przedziału czasowego obiektu timer, należy dodać detektor zdarzeń dla zdarzenia `TIMER` (jest to zdarzenie dysponowane za każdym razem, gdy obiekt Timer osiąga przedział czasowy określony za pomocą właściwości `Timer.delay`).

Drugim zdarzeniem jest `TIMER_Complete` wywoływane za każdym razem, gdy zakończono liczbę żądań określonych za pomocą właściwości `Timer.repeatCount`.

Inne użyteczne właściwości obiektu Timer:

`currentCount` - całkowita liczba uruchomień czasomierza z zerowym czasem początkowym. Jeśli licznik zdarzeń timer został wyzerowany, zliczane są tylko uruchomienia od momentu zerowania.

`reset()` - zatrzymuje czasomierz (jeśli działa) i ustawia jego właściwość `currentCount` z powrotem na 0.