

Ćwiczenie 2 – Dynamiczne ładowanie treści wyświetlanych

Do aplikacji Flash można załadować dynamicznie (za pomocą ActionScript lub komponentów-kontenerów) dowolne z poniższych treści graficznych:

- Plik SWF utworzony w języku ActionScript 3.0
- Plik obrazu — pliki: JPG, PNG i GIF.
- Plik SWF zapisany w języku ActionScript 1.0 lub 2.0.

Do ładowania tych zasobów służy klasa **Loader**.



Można także załadować dynamicznie teksty, filmy i dźwięki, do czego służą inne klasy i metody.

Aby załadować plik SWF lub plik graficzny do aplikacji należy stworzyć obiekt **Loader**, wskazać podrzędny obiekt wyświetlany (np. plik .swf), a następnie dodać obiekt **Loader** do listy wyświetlania według poniższego schematu:

```
var movieLoader:Loader = new Loader();
var movieURL:String = "galeria.swf";
var movieURLReq:URLRequest = new URLRequest(movieURL);
movieLoader.load(movieURLReq);
this.addChild(movieLoader);
```

W zasadzie powyższy kod wystarczy, aby załadować plik .swf. Jednak wykonamy ćwiczenie, które zademonstruje także śledzenie procesu ładowania za pomocą tak zwanego paska postępu i dodatkowo w polu tekstowym wskaże na bieżąco ilości załadowanej informacji. Dodamy także wstępną animację, która będzie stanowić tak zwany preloader.

Jest to wstęp do tworzenia wielowątkowych interaktywnych aplikacji takich jak strony internetowe. Na bazie informacji przekazanych w całym tym ćwiczeniu będziesz już mógł zacząć samodzielne działania we Flashu©

Pobierz przykład (<http://jsekulska.kis.p.lodz.pl/studia.htm>).

Krok 1

Tworzenie pliku kontenera

Otwórz nowy plik i ustal rozmiar stołu montażowego na 800X600 pikseli, częstotliwość odtwarzania klatek na 24 kl/s. i tło czarne.

Zaznacz klatkę i w panelu **Operacje** wpisz powyższy kod. Zmień nazwę warstwy na „akcje”.

Zapisz plik pod nazwą kontener.swf w lokalizacji takiej samej jak plik galeria.swf. Teraz przetestuj film (Ctrl+ENTER).

Jak widać kontener od razu wczytuje plik z galerią. Dzieje się to bardzo szybko, gdyż korzystamy z lokalnych zasobów komputera. Gdyby jednak odtwarzanie naszego pliku odbywało się poprzez połączenie sieciowe, to moglibyśmy dłużej oczekiwać na wyświetlenie zawartości, dlatego niezbędne jest wyświetlenie jakiejś informacji, że plik jest właśnie pobierany. Dobrą praktyką jest informowanie odbiorcy w postaci informacji graficznej lub tekstowej, że za jakiś czas nastąpi poprawne wyświetlenie oczekiwanej treści. Przygotujemy teraz elementy, które spełnią taką rolę informacyjną.

Krok 2

Pasek postępu

Utwórz nowy klip filmowy o nazwie „pasek”. Wewnątrz klipu narysuj wąski prostokąt o wypełnieniu białym lub szarym i długości ok. 600 pikseli. Zwróć uwagę, aby punkt zaczepienia paska (środek klipu) znajdował się z lewej strony.

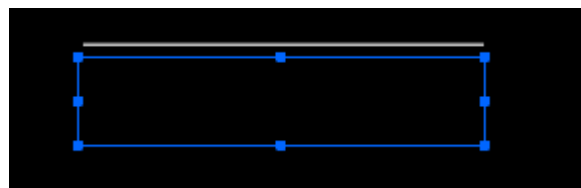


Wróć do głównej sceny, wstaw nową warstwę o nazwie „pasek” i umieść w niej stworzony klip z paskiem w położeniu x około 100 i y około 300; i nadaj mu nazwę instancji **bar_mc**.

Krok 3

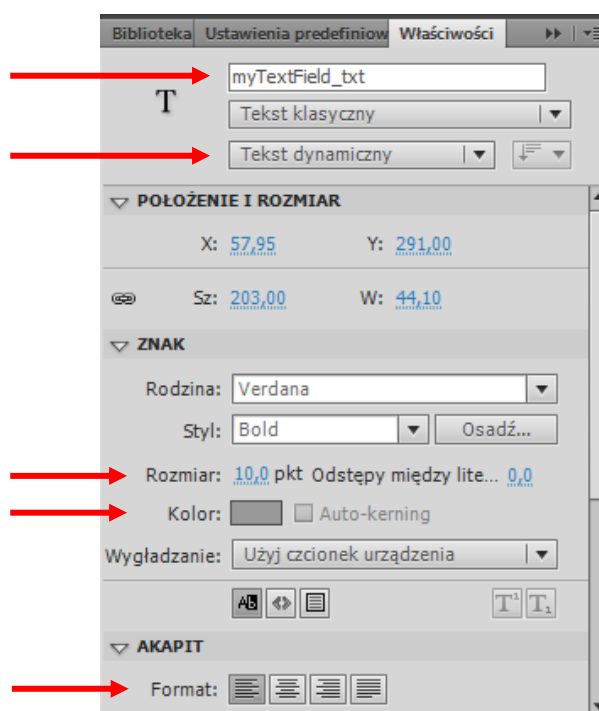
Pole tekstowe

Wstaw nową warstwę o nazwie „pole_txt” i za pomocą narzędzia **tekst** wstaw poniżej paska pole tekstowe.



Ustaw właściwości pola tekstowego:

- nazwa instancji: **myTextField_txt**
- tekst dynamiczny
- rozmiar: 10 pkt
- kolor szary (lub biały)
- użyj czcionek urządzenia
- wyrównanie do lewej



Krok 4**Dodanie elementów nasłuchujących**

Teraz w klatkę skryptu wpisujemy kod służący do nasłuchu ładowania pliku .swf. Otwórz panel **Operacje** i wpisz dodatkowy kod:

Do nasłuchiwanie pobierania danych:

```
movieLoader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS, progressHandler);

function progressHandler(myevent:ProgressEvent):void{
    var myprogress:Number= myevent.target.bytesLoaded/myevent.target.bytesTotal;
    bar_mc.scaleX = myprogress;
    myTextField_txt.text=Math.round(myprogress*100)+"%";
}
```

Do nasłuchiwanie zakończenia pobierania danych:

```
movieLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, finished);

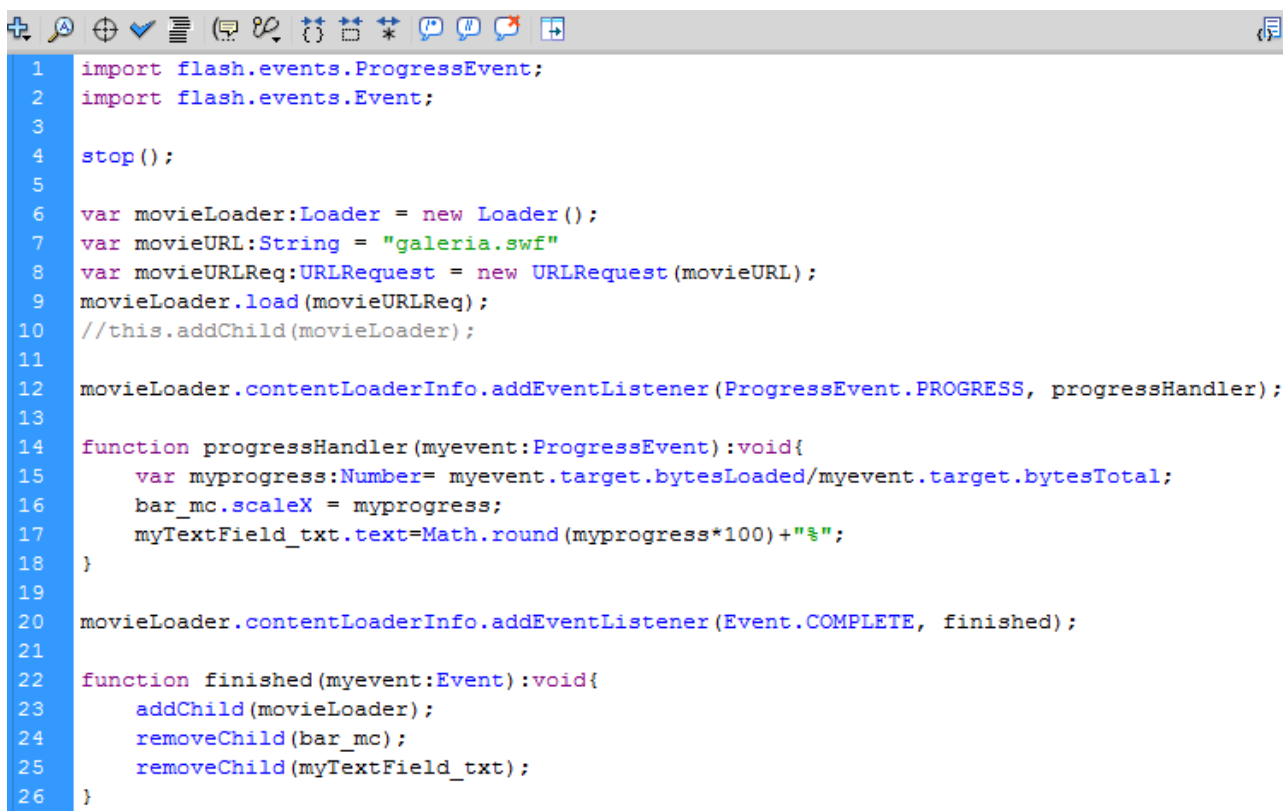
function finished(myevent:Event):void{
    addChild(movieLoader);
    removeChild(bar_mc);
    removeChild(myTextField_txt);
}
```

Na górze skryptu dodajemy jeszcze akcję `stop()`;

Z pierwszej zaś części skryptu, którą pisaliśmy na początku ćwiczenia usuwamy linijkę `addChild(movieLoader)`;

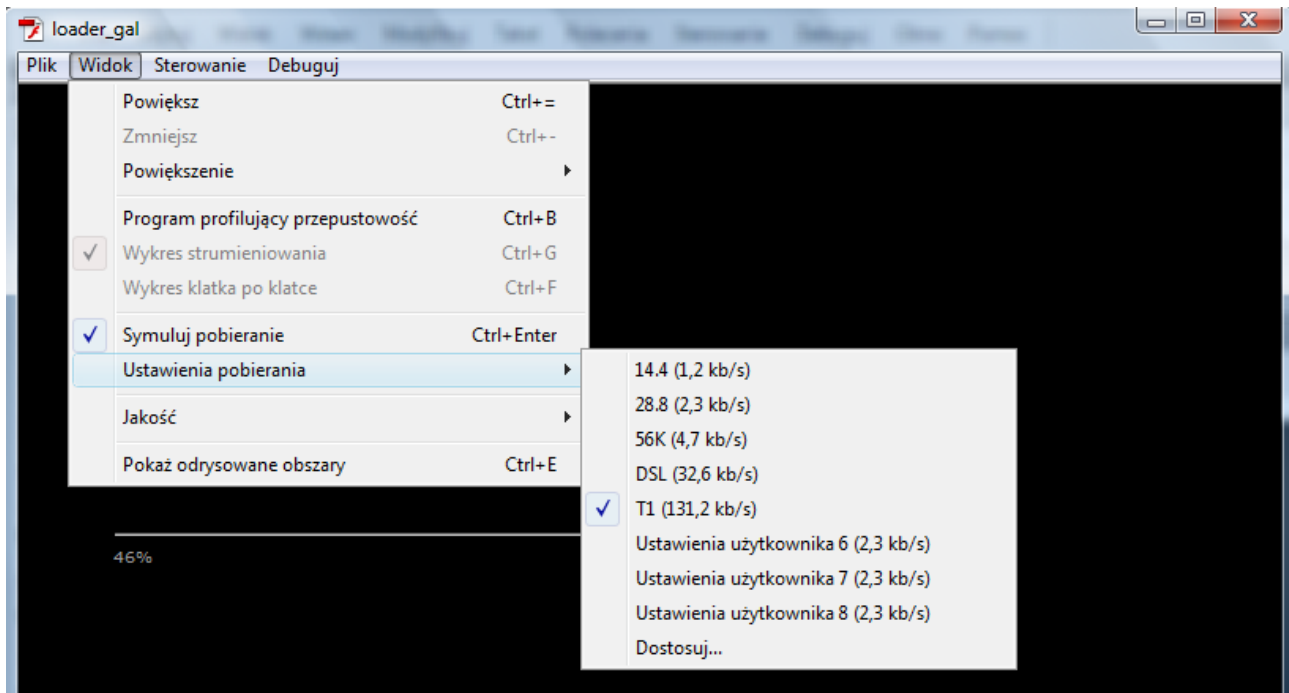
gdyż pojawiła się ona w funkcji `finished`.

Skrypt jest już w całości gotowy. Powinien on wyglądać następująco:



```
1 import flash.events.ProgressEvent;
2 import flash.events.Event;
3
4 stop();
5
6 var movieLoader:Loader = new Loader();
7 var movieURL:String = "galeria.swf"
8 var movieURLRequest:URLRequest = new URLRequest(movieURL);
9 movieLoader.load(movieURLRequest);
10 //this.addChild(movieLoader);
11
12 movieLoader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS, progressHandler);
13
14 function progressHandler(myevent:ProgressEvent):void{
15     var myprogress:Number= myevent.target.bytesLoaded/myevent.target.bytesTotal;
16     bar_mc.scaleX = myprogress;
17     myTextField_txt.text=Math.round(myprogress*100)+"%";
18 }
19
20 movieLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, finished);
21
22 function finished(myevent:Event):void{
23     addChild(movieLoader);
24     removeChild(bar_mc);
25     removeChild(myTextField_txt);
26 }
```

Przetestuj film. Jeśli nie widzisz paska postępu ładowania a galeria pojawia się natychmiast to nic złego się nie dzieje. Plik .swf po prostu ładuje się szybko i sprawnie. Możemy podejrzec jednak jak działa nasz efekt postępu ładowania i czy pojawiają się wszystkie elementy. Śledzenie takie wykonuje się za pomocą narzędzi **Ustawienia pobierania** i **Symuluj pobieranie**, które dostępne są w oknie Flash Playera.

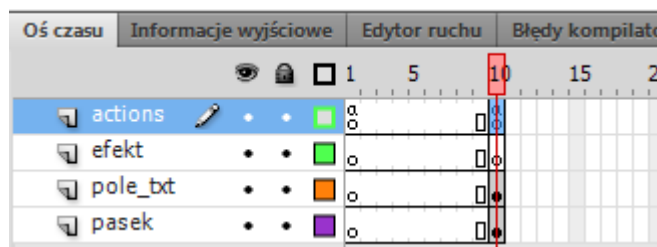


Krok 4

Utworzenia animacji wstępnej

Teraz wykonany drugą część ćwiczenia – animację, która pojawi się zanim jakakolwiek treść pochodząca ze źródeł zewnętrznych zostanie załadowana. Wprowadzimy także element interaktywności – użytkownik sam zadecyduje kiedy przejść do etapu ładowania zawartości z zewnątrz.

Zaczynamy od dodania warstwy, której nadajemy nazwę „efekt”. W każdej warstwie wstawiamy klatkę kluczową w klatce nr 10 i odsuwamy tam dotychczasową zawartość filmu. Do klatki nr 1 w warstwie „action” wstawiamy akcję `stop()` ;



Zaznaczamy klatkę nr 1 warstwy „efekt” i importujemy na stół montażowy zdjęcie. Ustawiamy zdjęcie po środku stołu montażowego.

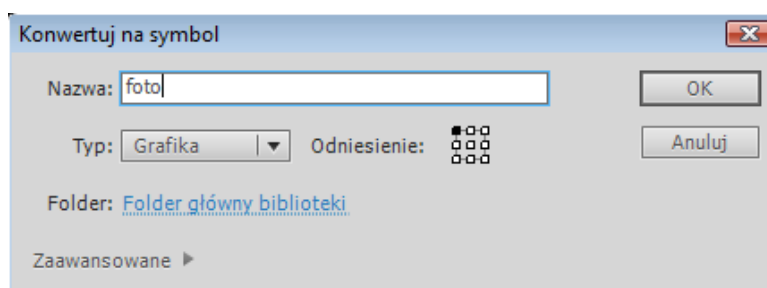
Zdjęcie konwertujemy do klipu (F8) i nadajemy mu nazwę „efekt”.

Krok 5

Praca w klipie

Klikamy dwa razy na utworzony klip w celu edycji go.

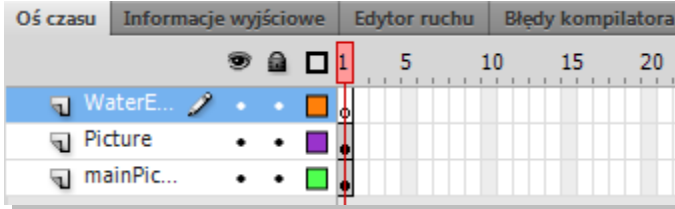
Wewnątrz klipu zaznaczamy zdjęcie i konwertujemy je do grafiki (F8). Możemy nazwać ją „foto”.



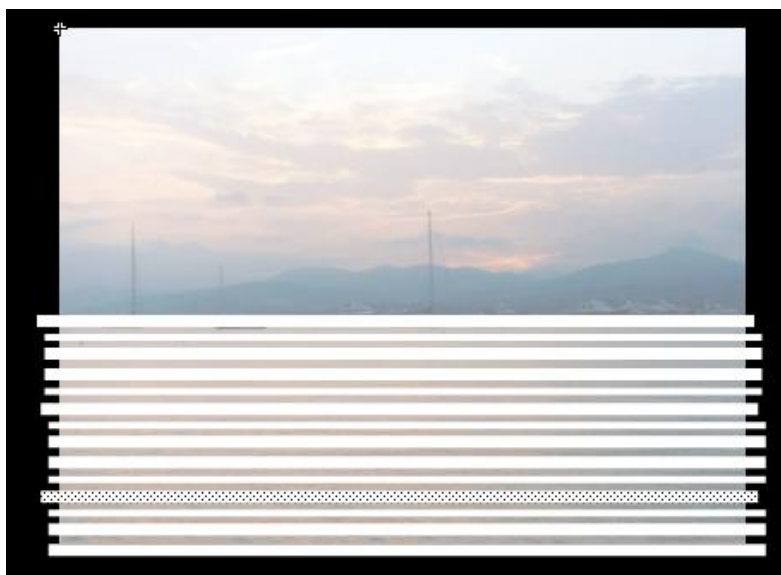
Zmieniamy nazwę warstwy na **mainPicture**. Klikamy na klatkę nr 1 i z menu podręcznego wybieramy polecenie **Kopiuje klatki**.

Wstawiamy nową warstwę i nazywamy ją **Picture**. Klikamy na klatkę nr 1 tej warstwy i wybieramy z menu podręcznego polecenie **Wklej klatki**.

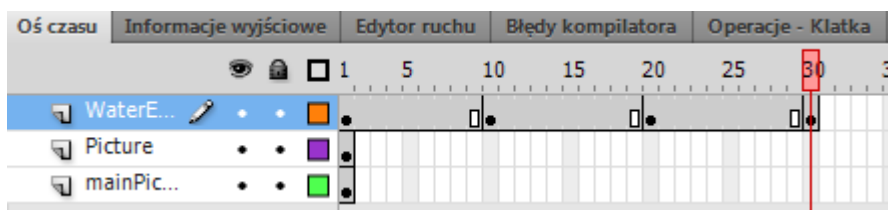
Wstawiamy kolejną nową warstwę i nazywamy ją **WaterEffect**.



Wybieramy narzędzie prostokąt (R) i na warstwie **WaterEffect** rysujemy na dolnej połowie zdjęcia bardzo wąskie prostokąty bez obramowania. Prostokąty muszą być bardzo gęsto położone:

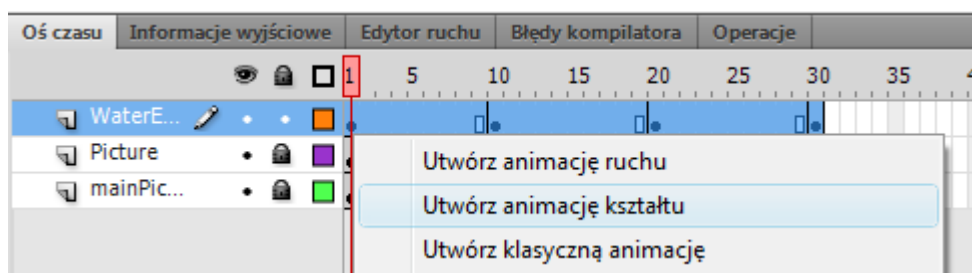


Klikamy na klatkę 10, 20 i 30 na warstwie **WaterEffect** i wstawiamy klatki kluczowe (F6).

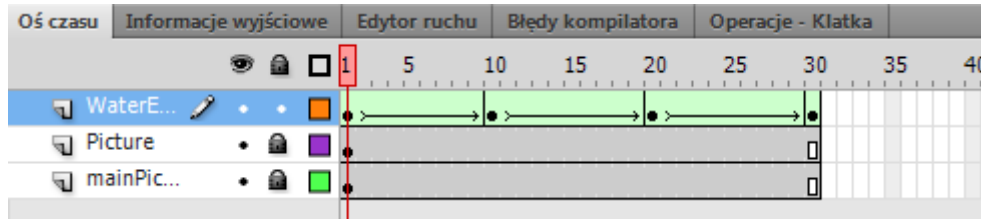


Wracamy do klatki 10, i używając strzałki na klawiaturze przenosimy nasz "efekt" nieznacznie w dół. Potem klikamy na klatkę 30, przenosimy nasz "efekt" nieznacznie w górę.

Zaznaczamy warstwę **WaterEffect**, i będąc na klatce prawym klawiszem myszy uruchamiamy menu podręczne, w którym wybieramy opcję **Utwórz animację kształtu**.



W dwóch niższych warstwach wstawiamy klatki zwykłe (F5) w klatce nr 30.



Zaznaczamy warstwę **WaterEffect** i nadajemy jej właściwość maska (opcja dostępna w menu podręcznym).

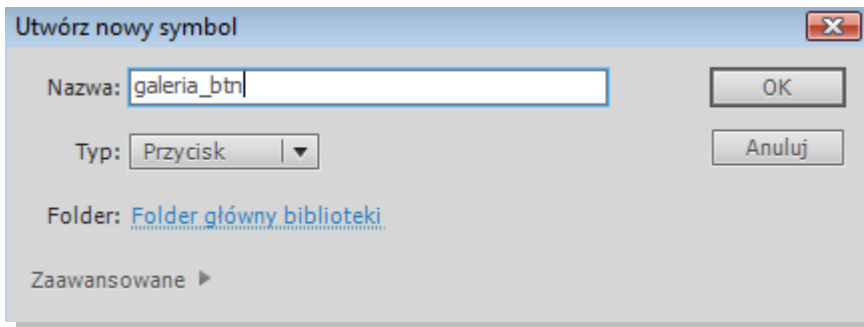
Na koniec pracy z klipem zablokuj i ukryj warstwy **WaterEffect** i **Picture**, a zdjęcie które znajduje się na warstwie **mainPicture** przesunij nieznacznie w dół (używając np. strzałki na klawiaturze). Przetestuj film.

Krok 6
Przycisk

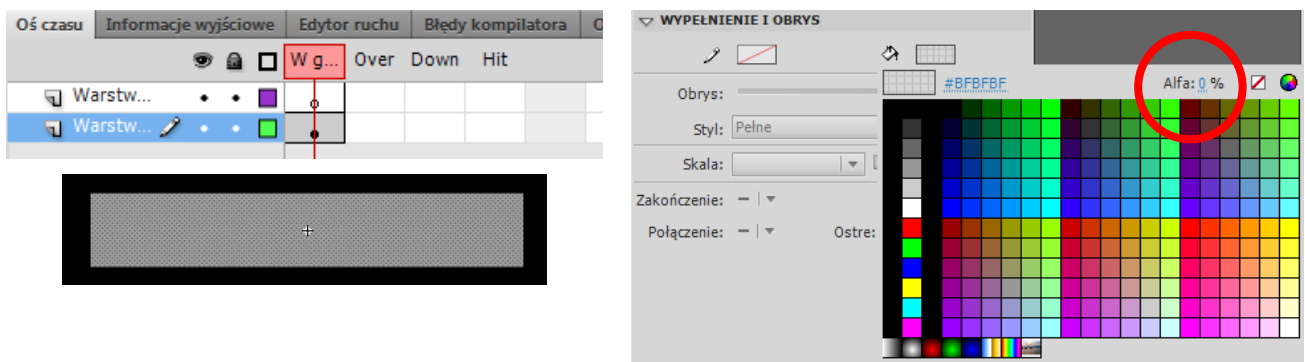
Wracamy do głównej sceny. Na początek dodajemy warstwę „napis”, w której możemy umieścić napis GALERIA.



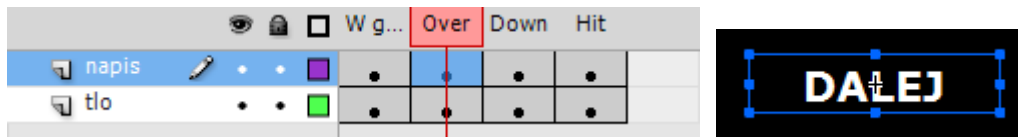
Teraz przystępujemy do projektowania przycisku. Wstawiamy nowy symbol typu przycisk:



I projektujemy go według własnego pomysłu:

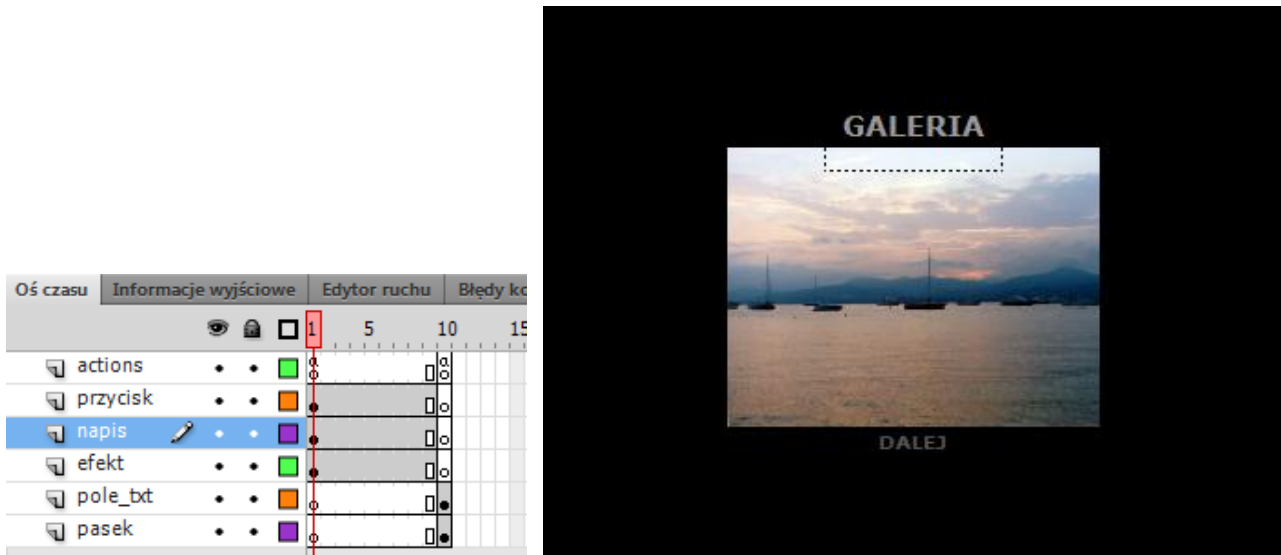


Ja użyłam do jego wykonania prostokąta z niewidocznym wypełnieniem (alfa=0) i bez obramowania. Widoczny będzie tylko sam napis „DALEJ”.



Przycisk ten będzie przełączał użytkownika do klatki nr 10, w której następuje ładowanie pliku z galerią.

Umieszczamy więc go na głównej scenie. Możemy wstawić dla niego osobną warstwę ale nie jest to konieczne. Może znaleźć się w tej samej warstwie co napis galeria. Instancji przycisku nadajemy nazwę „dalej_btn”.



**Krok 7
Akcje przycisku**

W warstwie “**actions**” w klatce nr 1 otwieramy panel **Operacji**. Poniżej akcji `stop()`; wpisujemy kod:

```
dalej_btn.addEventListener(MouseEvent.CLICK,goGallery);

function goGallery(event:MouseEvent):void{
    gotoAndPlay(10);
}
```

Całość wyglądać będzie następująco:

```
1 import flash.events.MouseEvent;
2
3 stop();
4
5 dalej_btn.addEventListener(MouseEvent.CLICK,goGallery);
6
7 function goGallery(event:MouseEvent):void{
8     gotoAndPlay(10);
9 }
```

I to koniec pracy. Przetestuj aplikację.