

## Ćwiczenie 6 - Zmiana kursora

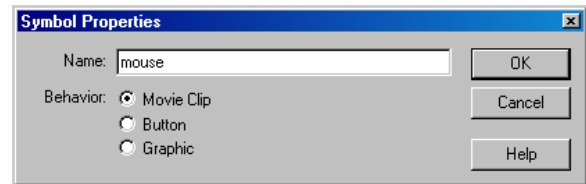
Celem ćwiczenia jest podmiana kursora myszy na dowolny inny obrazek. [Pobierz przykład.](#)

### Krok 1

Na początek tworzymy nowy Obiekt **Movie Clip** pod nazwą np. „mouse” (menu **Insert** → **New symbol**).

### Krok 2

W otwartym oknie dialogowym należy wybrać typ symbolu, jaki zostanie utworzony (czyli *Movie Clip*) oraz wpisać nazwę symbolu (patrz okienko z prawej strony).



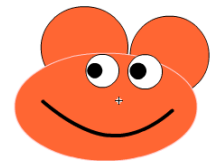
#### Komentarz

Polecenie to otwiera pole robocze symbolu, które posiada jednakowy wygląd jak pole robocze sceny.

### Krok 3

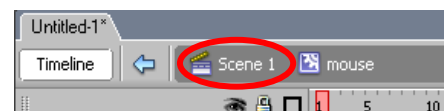
W pole robocze klipu tworzymy **jednoklatkową** animację zawierającą to, co ma zostać wyświetlone zamiast kursora (patrz rysunek obok).

Tworzenie obiektu odbywa się w sposób standardowy za pomocą narzędzi z przybornika zgodnie z technikami wykorzystywanymi na pierwszych zajęciach.



### Krok 4

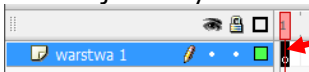
Wracamy do głównej sceny animacji (za pomocą przełącznika wyróżnionego na rysunku obok znajdującego się ponad okienkiem listwy czasowej **Timeline**).



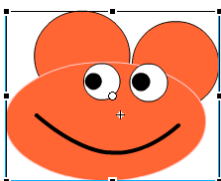
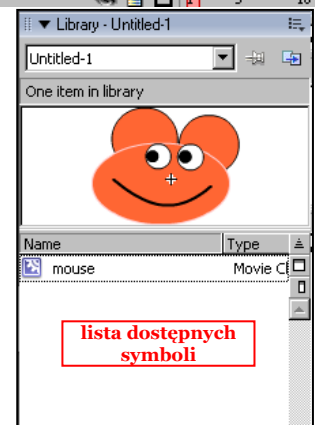
### Krok 5

Do wstawiania obiektów typu **Movie Clip** do animacji służy panel **Library** (Biblioteka) znajdujący się po prawej stronie całej przestrzeni roboczej programu Macromedia Flash (jeśli go nie ma, to jest on dostępny w menu **Window** lub **Ctrl+L**).

Aby wstawić obiekt z biblioteki, wystarczy w głównej animacji zaznaczyć klatkę do której chcemy wstawić obiekt



i z okienka podglądu biblioteki przeciągnąć obiekt w pole robocze sceny.



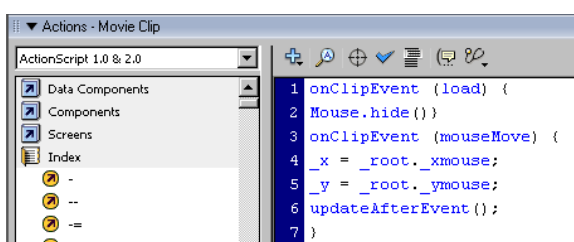
Jeśli obraz po wstawieniu jest zbyt duży, to można zmienić jego proporcje skalując go

narzędziem swobodnej transformacji  z paska przybornika.

### Krok 6

Na wstawiony już **Movie Clip** klikamy prawym klawiszem myszki i wybieramy opcję **Actions**.

W oknie kodu wstawiamy ciąg poleceń, które ukryją aktualny kursor, a w jego miejsce wstawią nasz (zgodnie z rysunkiem poniżej):



#### Kod do bezpośredniego wklejenia:

```
onClipEvent (load) {  
    Mouse.hide();  
}  
onClipEvent (mouseMove) {  
    _x = _root._xmouse;  
    _y = _root._ymouse;  
    updaupdateAfterEvent();  
}
```

**Opis kodu**

Pierwszy fragment ukrywa kursor, drugi zaś mówi, że obrazek ma wodzić za ukrytym kursorem. Jeżeli byśmy pozbyli się fragmentu, który ukrywa kursor był by on widoczny, a obrazek chodziłby za kursorem.

**Krok 7**

Teraz pozostało tylko zapisanie i opublikowanie animacji – **Ctrl+Enter**.

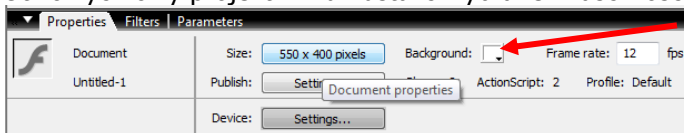
## Ćwiczenie 7 - Duplikowanie klipu

Celem ćwiczenia jest nabycie umiejętności tworzenia i zarządzania pojedynczymi klipami w animacji. [Pobierz przykład.](#)

Scena (animacja) główna składa się z animacji koła po torze i dodatkowych skryptów ActionScript, które ją powtarzają w niewielki odstępach czasu wraz z dodaniem efektu zanikania.

**Krok 1**

Otworzyć nowy projekt filmu i ustalić wybrane właściwości filmu, np. kolor tła – Movie Properties.

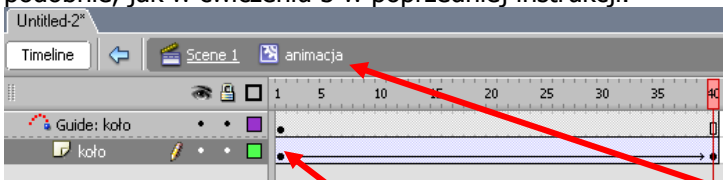


**Krok 2**

Należy utworzyć animację (*Movie Clip*), która zostanie potem wstawiona do sceny głównej. Robimy to za pomocą polecenia z menu **Insert** → **New Symbol** → **Movie Clip**. Klip nazywamy np. „animacja”.

**Krok 3**

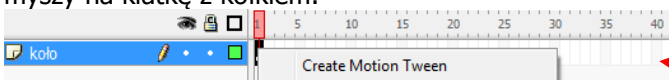
Jako animację można stworzyć coś podobnego do przykładu – czyli kółko animowane po torze. Robimy to podobnie, jak w ćwiczeniu 3 w poprzedniej instrukcji.



Proszę pamiętać, że pracujemy w polu roboczym Movie Clipa – *animacja*. Tu rysujemy kółko, które musi znajdować się w pierwszej warstwie i w pierwszej klatce.

**Krok 4**

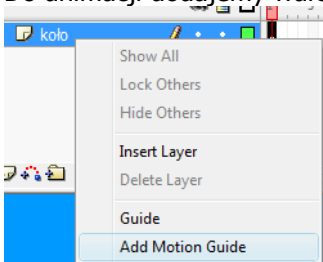
Gdy narysujemy kółko, dodajemy do niego animację - **Create Motion Tween** klikając prawym klawiszem myszy na klatkę z kółkiem.




Teraz w tej samej warstwie do klatki nr 40 wstawiamy klatkę kluczową (**F6**) i w tej klatce przenosimy kółko w inne miejsce sceny.

**Krok 5**

Do animacji dodajemy warstwę typu **Guide**,

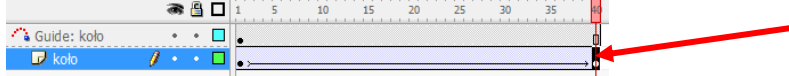



która będzie zawierać narysowany tor ruchu dla koła.

Tor ruchu rysujemy narzędziem pióro lub ołówek  oraz ustawiamy się np. na klatce pierwszej i klawiszem (**F5**) dodajemy puste klatki do 40.

### Krok 6

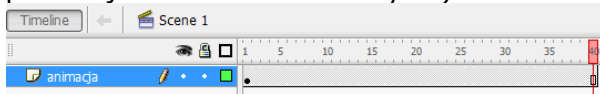
W ostatniej klatce warstwy koła naszej animacji



zmieniamy położenie koła na drugi koniec krzywej. Sprawdzamy, czy w obu klatkach kluczowych (tj. pierwszej i ostatniej na warstwie koło) obiekt koła (a konkretnie jego punkt  leży i styka się z krzywą. Klawiszem **Enter** możemy odtworzyć sobie ruch po torze.

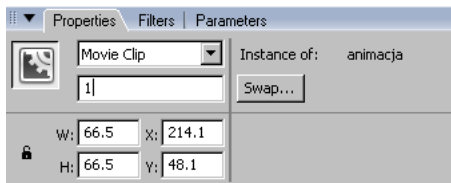
### Krok 7

Tak przygotowaną animację wstawiamy do warstwy pierwszej w naszej głównej scenie (animację pobieramy z biblioteki podobnie jak w ćw. 6 krok 5). Ilość klatek tej sceny zwiększamy do 40 (ustawiamy się na klatce pierwszej i wielokrotnie wciskamy **F5**).



### Krok 8

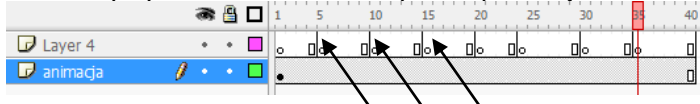
Teraz musimy nadać nazwę animacji kółka. Dzięki niej będzie rozpoznawana w napisanym przez nas skrypcie i będzie można ją skopiować. W tym celu w polu roboczym klikamy na wstawioną animację (kółko) i w panelu **Properties** wpisujemy w polu **Instance Name 1** (patrz rysunek).



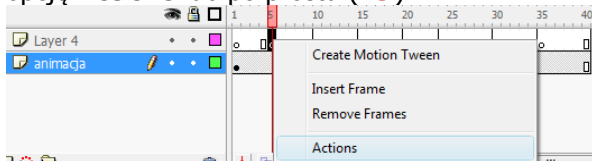
### Krok 9

Dodajemy kolejną warstwę w naszej scenie .

Tu tworzymy kilka klatek kluczowych za pomocą klawisza **F6**.



Pierwszą tak stworzoną klatkę kluczową pomijamy, na kolejną tj. 5 klikamy prawym klawiszem i wybieramy opcję **Actions** lub po prostu (**F9**).



i wstawiamy taką akcję:

```
duplicateMovieClip ("1", "2", 3);  
setProperty ("2", _alpha, "95");
```

Po tym kroku symbol klatki na liście czasowej uzyskał dodatkową literkę *a*.

#### Opis kodu

```
duplicateMovieClip ("1", "2", 3);  
skopiuj MovieClip o nazwie 1, nadaj mu nową nazwę 2 oraz ustal odległość od poprzedniego symbolu na 3.  
  
setProperty ("2", _alpha, "95");  
ustal wartość Alpha (przezroczystość) dla obiektu o nazwie 2 o wartości 95 (widoczność obiektu 95%).
```

Po wstawieniu funkcji wybranej z listy ukaze się podpowiedź z listą argumentów:

```
1 duplicateMovieClip ();  
2  
duplicateMovieClip( target, newname="", depth );
```

oraz

```
1 duplicateMovieClip ("1", "2", "3" );  
2 setProperty ();  
3  
setProperty( target, property, value );
```

### Krok 10

Powtarzamy ten sam skrypt w kolejnych klatkach kluczowych, tylko z tym wyjątkiem, że kopiujemy już obiekt o wcześniej nadanej nazwie.

I tak: dla klatki kluczowej tj. 10 kopiujemy utworzony skryptem w klatce 5 obiekt animacji (klip) o nazwie 2, nadajemy mu teraz nową nazwę 3, ustalamy odległość na 4. Kod będzie wyglądał:

```
duplicateMovieClip ("2", "3", 4);
```

Z kolei w `setProperty` ustalamy dla obiektu 3 zanikanie o 5 mniejsze, czyli Value 90.

```
setProperty ("3", _alpha, "90");
```

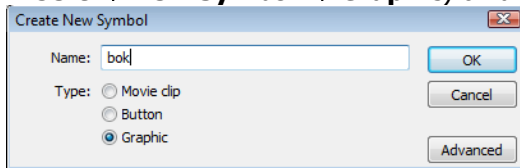
Takich klatek tworzymy od 10 do 20 (sami zorientujecie się ile wam potrzeba) za każdym razem cyklicznie zmieniając odpowiednie parametry. Na koniec zapisujemy i publikujemy naszą animację (**Ctrl+Enter**).

## Ćwiczenie 8 – Aktywny mur

Celem ćwiczenia jest nabycie umiejętności tworzenia i zarządzania pojedynczymi przyciskami w animacji. [Pobierz przykład](#). Pokazowy element przycisku można zastosować np. w menu. Po najechnaniu np. na sześcian może pokazać się jakiś napis sugerujący łącze do jakiejś strony.


### Krok 1

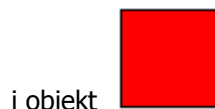
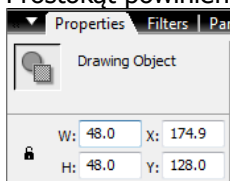
Na początek otwieramy nowy film, a w nim kolejno tworzymy trzy elementy graficzne (z menu **Insert** → **New Symbol** → **Graphic**) o nazwach odpowiednio: „przód”, „górze”, „bok”.



Powstałe elementy stanowią będą trzy ściany kostki.

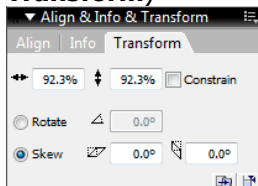
### Krok 2

Do utworzenia ścianki przedniej wystarczy użyć narzędzia prostokąt . Prostokąt powinien mieć wymiary 48x48 pt (paleta **Properties** na dole).



### Krok 3

Kolejne ściany tworzymy na bazie tego prostokąta. W odpowiednich symbolach tj. „bok” i „górze” wstawiamy symbol ściany przedniej, z tym, że za pomocą palety **Transform** po prawej (z menu **Window** → **Transform**)

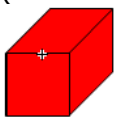


spłaszczamy ścianę górną (*Skrew* horizontally 45°) i boczną (*Skrew* vertically -45°)



**Krok 4**

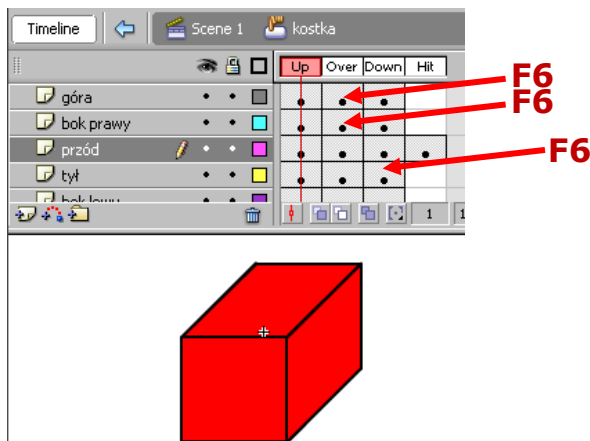
Teraz łączymy ściany w jeden obiekt wstawiając je do nowego przycisku (z menu **Insert** → **New Symbol** → **Button**) – o nazwie „kostka”.



Składamy ładną kostkę. Element „bok” możemy wstawić dwukrotnie budując w ten sposób ścianę prawą i lewą. Oczywiście **WSZYSTKIE** elementy kostki pobieramy z panelu **Library**.

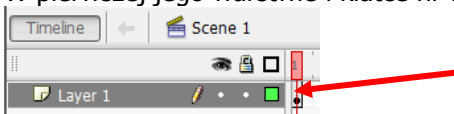
**Krok 5**

Teraz należy skopiować wszystkie elementy przycisku do klatek *Over*, *Down* i *Hit*. W tym celu wystarczy w omawianych klatkach użyć klawisza (**F6**). W polu *Hit* przycisku możemy pozostawić tylko jedną ścianę - jest to ściana przednia.



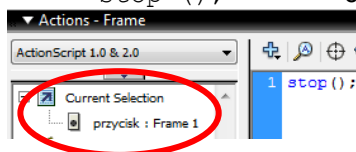
**Krok 6**

Następnie tworzymy Movie Clip'a (z menu **Insert** → **New Symbol** → **Movie Clip**) o nazwie „animacja”. W pierwszej jego warstwie i klatce nr 1 wstawiamy stworzony przycisk (z panelu **Library**).



Do w/w klatki dodajemy akcję klikając prawym klawiszem i wybieramy opcję **Actions** lub po prostu (**F9**). Po upewnieniu się, że akcja definiowana jest właśnie dla klatki 1 (świadczy o tym wpis z nazwą warstwy i numerem klatki pod pozycją „*Current selection*”) piszemy:

`stop ();` - oznacza to zatrzymanie animacji i oczekiwanie na zdarzenie.



**Krok 7**

W kolejnym kroku dodajemy akcje DLA PRZYCISKU (kostki). W tym celu zaznaczamy go i klawiszem (**F9**) definiujemy nową akcję:

	<p><u>Kod do bezpośredniego wklejenia:</u></p> <pre>on (rollOver, dragOver) {     gotoAndPlay(5); }</pre>
--	---

**Opis kodu**

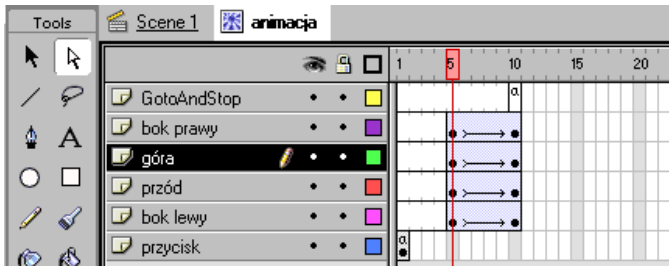
`on (rollOver, dragOver)`  
 uruchamia proces obsługi zdarzenia w wyniku pojawienia się kursora myszy nad obiektem kostki.

`gotoAndPlay(5);`  
 funkcja skacze do klatki nr 5 i od tego miejsca uruchamia animację.

**Krok 8**

Wstawiamy nowe 4 warstwy (kolejność i nazewnictwo zgodne z rysunkiem poniżej).

W każdej ich 5. klatce (oprócz warstw *przycisk* i *GotoAndStop*) umieszczamy kolejne obiekty graficzne składające się na elementy kostki (tj. przód, boki i góra). Umieszczamy je dokładnie na kostce. Aby było łatwiej, można klatkę 1 warstwy *przycisk* skopiować (**Copy Frames**) → (**Paste Frames**) do klatki 5 w tej samej warstwie. Na końcu tego kroku będzie trzeba tę klatkę potem skasować (**Remove Frames**). Dodatkowo, w tych warstwach klikając na klatkę 5 prawym klawiszem myszy tworzymy animację ruchu (**Create Motion Tween**) i potem w klatkach 10 klatki kluczowe (**F6**). W klatkach tych tworzymy efekt rozpadającego się pudełka poprzez rozsuniecie elementów pudełka.



Przypominam o skasowaniu klatki 5 z warstwy *przycisk* (**Remove Frames**).

### Krok 9

W ostatniej klatce Clip'a w warstwie *GotoAndStop* wstawiamy taki skrypt (**F9**):

```
gotoAndStop (1);
```

który zatrzyma i przewinie animację do początku i będzie czekać na jej ponowne uruchomienie po najechaniu kursorem myszy na przycisk (kostkę) – (patrz krok 7).

### Krok 10

Teraz wracamy do głównej naszej sceny. Tu możemy wstawić jakieś tło animacji (warstwa nr 1), dodać jakieś napisy oraz inne elementy, które chcemy, aby się tam znalazły. Dodajemy nową warstwę i do niej wstawiamy nasz Movie Clip „animacja” z animowaną kostką (z panelu **Library**).

Możemy wstawić takich klipów na stronę kilka tworząc z nich jakiś fajny murek.

Po uruchomieniu skryptu (**Ctrl + Enter**) jego cegiełki będą się efektownie rozpadać.

## Ćwiczenie 9 – Aktywne menu

Ten tutorial pokazuje jak wykonać aktywne menu na stronie www.

[Pobierz przykład.](#)

Menu nie jest trudne, ale trzeba wykonać do niego sporo elementów w formie przycisków.

### Krok 1

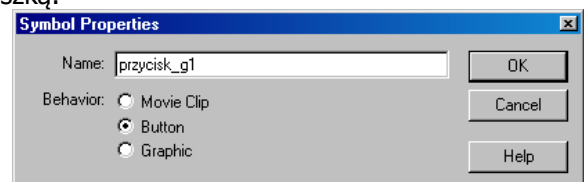
Na początek tworzymy przycisk menu głównego tak jak na przykładzie obok (dla potrzeb tego ćwiczenia wystarczy jeden przycisk, więcej przycisków można stworzyć samodzielnie, aby rozbudować główne menu).

Jest to zwykły przycisk, który zmienia się po najechaniu myszką.





Informacje o firmie

Tworzenie przycisku odbywa się poprzez polecenie **Insert** → **New Symbol**. Proszę pamiętać o nadaniu odpowiedniej nazwy przycisku i wybraniu atrybutu przycisku.



**Od tej pory pracujemy już tylko w polu roboczym edycji przycisku.**

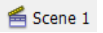
### Krok 2

Tu umieszczamy projekt rysowany za pomocą narzędzia *prostokąt*  i narzędzia *tekst* .

Jednocześnie należy pamiętać, aby odpowiednio zmienić wygląd przycisku dla zdarzenia związanego z najechaniem kursora (w polu **Over** listwy czasowej) lub wciśnięciem przycisku (w polu **Down**). Do tych

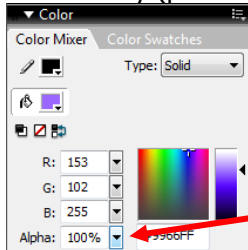
klatek kopiujemy przycisk klawiszem (**F6**). A zmiana, o której mowa, to np. zmiana koloru tekstu, zmiana koloru tła przycisku lub nieznaczne przesunięcie przycisku w dół i w prawo (tzw. „spłaszczenie”).

### Krok 3

Utworzony przycisk od razu umieszczamy w animacji. W tym celu przechodzimy do głównej sceny  i przenosimy przycisk z biblioteki (panel **Library**). Jedyłą warstwę nazywamy jako „menu\_g”.

### Krok 4

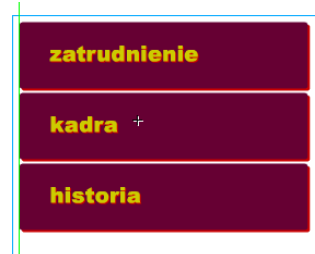
Teraz musimy stworzyć przyciski do podmenu w liczbie czterech z tym wyjątkiem, że czwarty ma być niewidoczny (panel **Mixer**, **Alpha** = 0%) z panelu **Color** po prawej



i powinien mieć rozmiar trochę większy niż elementy podmenu razem wzięte (patrz rysunek poniżej – ramka wokół przycisków). Przyciski znów tworzymy osobno jako **Insert** → **New Symbol**.

### Krok 5

Tworzymy MovieClip (**Insert** → **New Symbol**) o nazwie „podmenu”, do którego wstawiamy wcześniej stworzone trzy przyciski. Ustawienie przycisków powinno wyglądać tak, jak na przykładzie z prawej.



### Krok 6

Teraz określamy akcję każdego przycisku: klikamy na przycisk prawym klawiszem i z menu wybieramy opcję **Actions** lub klawisz (**F9**). Dodajemy skrypt, który spowoduje np. otwarcie strony www lub innej podstrony naszej witryny, np.:

```
on (release) {  
  getURL ("http://www.kis.p.lodz.pl", "_self");  
}
```

lub

```
on (release) {  
  getURL ("zatrudnienie.htm", "_self");  
}
```

### Krok 7

Czwarty przycisk umieszczamy w innej, najniższej warstwie tak, by był TYLKO niewidocznym tłem dla rozwiniętego podmenu.

W ostatniej klatce tej warstwy z przyciskiem - tło wstawiamy akcję:

```
stop ();
```

Przygotowaliśmy właśnie animację, która się rozwinie po najechnaniu na przycisk menu głównego.

### Krok 8

#### **Wracamy do głównej sceny animacji.**

Tu  dodajemy warstwę „podmenu” i w jej drugiej klatce wstawiamy stworzony Clip *podmenu* (pobieramy go z panelu **Library**). Ustawiamy go w pobliżu przycisku, z którego ma się rozwinąć podmenu.

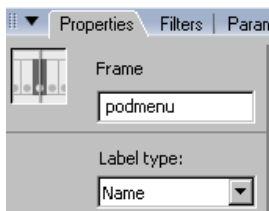
W pierwszej klatce też dodajemy akcję:

```
stop ();
```

Dodatkowo, klatkę tę nazywamy *podmenu*. Robimy to w polu **Frame** w panelu **Properties** przy zaznaczonej klatce w listwie czasowej (patrz rysunek poniżej).

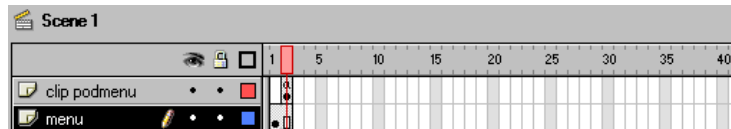






### Krok 9

Listwa czasowa sceny głównej na tą chwilę powinna wyglądać następująco:



### Krok 10

Teraz spowodujemy, że po najechaniu na główny przycisk menu pojawi się nam wstawione **podmenu**. Klikamy na główny przycisk i dodajemy do niego akcję:

```
on (release, rollOver) {  
gotoAndStop ("podmenu");  
}
```

Spowoduje to przejście do klatki z naszym podmenu. Można uruchomić animację (**Ctrl+Enter**).

### Krok 11

Niestety menu rozwija się, ale nie znika.

Musimy wstawić jeszcze jeden przycisk, który pomoże powrócić do klatki nr 1 i przy tym schować niepożądane podmenu.

Przycisk ten powinien być w oddzielnej i najniższej warstwie. Powinien obejmować całe pole animacji (przycisk główny i podmenu) oraz (jak ten w naszym MovieClip'ie dla podmenu) powinien być przezroczysty. Można skorzystać z gotowego symbolu z panelu **Library**.

### Krok 12

Po wstawieniu tego przycisku do głównej sceny animacji dodajemy do niego akcję:

```
on (rollOver) {  
gotoAndStop (1);  
}
```

To sprawi, że gdy z menu i podmenu zjedziemy na pole tego przycisk (czyli na tło sceny) cała animacja cofnie się do klatki nr 1 i tam się zatrzyma. Pamiętajmy przecież, że w klatce 1 podmenu jest niewidoczne.



### Krok 13

#### **Komentarz**

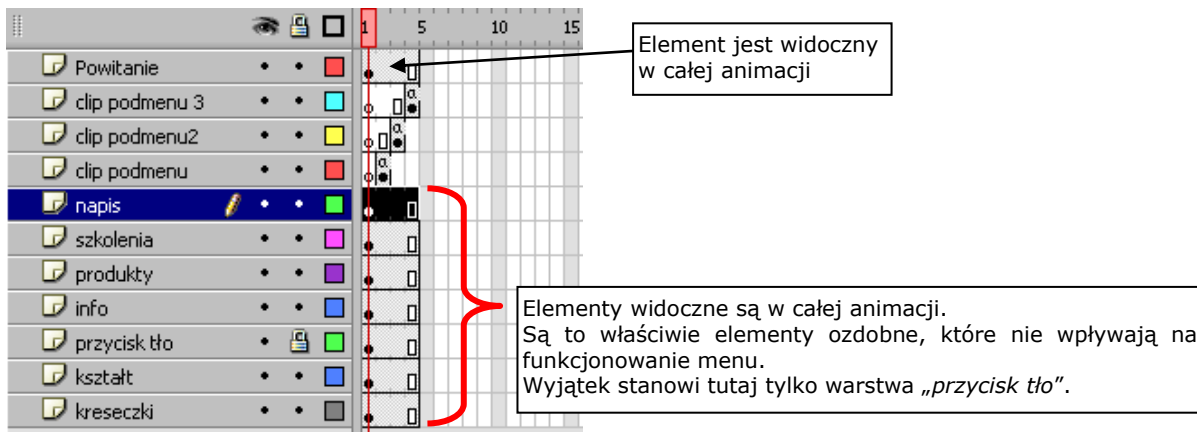
Należy pamiętać o tym, że gdy wstawimy więcej przycisków menu głównego (i to najlepiej każdy w nowych warstwach), MovieClipy zawierające kolejne podmenu musimy wstawiać w nowych warstwach i to kolejnych klatkach (3, 4 itd.). Klatki te muszą być także nazwane odpowiednio „podmenu2”, „podmenu3” itp. (patrz pierwszy rysunek w kolejnym ćwiczeniu)



## Ćwiczenie 10 – Pierwsza strona www

Celem ćwiczenia jest sprawdzenie własnych umiejętności nabytych w ramach poprzednich ćwiczeń. Instrukcja w tym ćwiczeniu podaje tylko pomysły a raczej stroni od podawania ich rozwiązań. Zachęcam do samodzielnego opracowania projektu własnego menu i jego realizacji.

Listwa czasowa sceny głównej całej strony internetowej powinna wyglądać mniej więcej podobnie, jak na rysunku poniżej. Używając przycisku i podmenu z poprzedniego ćwiczenia można stworzyć kompletne menu (zgodnie z komentarzem w ostatnim kroku poprzedniego ćwiczenia).



### Krok 1

Warstwa „Powitanie” zawiera tylko statyczny napis:



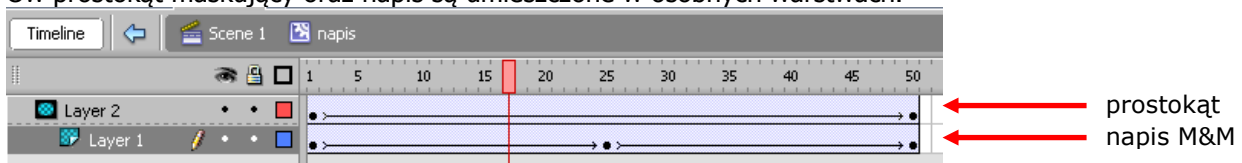
### Krok 2

Warstwa „napis” zawiera MovieClip z animowanym napisem M&M.



Napis M&M maskowany jest zwykłym prostokątem.

Ów prostokąt maskujący oraz napis są umieszczone w osobnych warstwach.

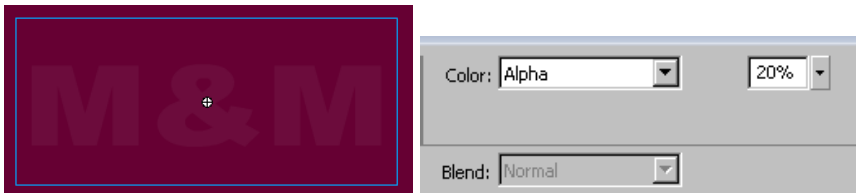


Do obu elementów zastosowano animację **Motion Tween**

– dla prostokąta jest to powolny ruch z lewej strony na prawo,

– dla napisu - zmiana przezroczystości **alpha**.

Zmiana **alpha** polega ona na tym, że napis w klatkach 1 i 50 (czyli na początku i na końcu animacji) jest jeszcze najslabiej widoczny. W tych klatkach ustalamy na pasku **Properties** zmianę koloru: **Alpha** na 20%.



W klatce 25 współczynnik **alpha** pozostaje bez zmian na poziomie 100%.

**Uwaga**

Maskując napis należy oczywiście pamiętać o ustaleniu właściwości warstw: warstwa z napisem jest typu „masked”, a warstwa z prostokątem – „mask”.

**Krok 3**

Warstwy „clip podmenu”, „clip podmenu2” i „clip podmenu3” zawierają kolejne rozwijane podmenu z przyciskami przygotowane zgodnie z instrukcjami z poprzedniego ćwiczenia.

Warstwy „szkolenia”, „produkty” i „info” zawierają główne przyciski menu. Całość może wyglądać podobnie, jak na rysunku poniżej.



Warstwa „przycisk tło” zawiera duży przezroczysty przycisk służący do powrotu do sceny 1.

Warstwy „kształt” i „kreseczki” zawierają dodatkowe elementy graficzne, które uatrakcyjniają wygląd strony.



**Komentarz**

Kilka słów wyjaśnień wymaga jeszcze budowanie kodu Action Script dotyczącego odsyłaczy dla przycisków. Kod zazwyczaj wygląda następująco:

```
on (release) {  
    gotoURL("jakaStrona.html", "_self");  
}
```

Polecenie `on` oznacza wykonanie zestawu poleceń w momencie wciśnięcia przycisku, dla którego projektowana jest dana akcja.

Funkcja `gotoURL` uruchamia nową stronę o adresie podanym jako parametr. Jeżeli adres strony rozpoczyna się fragmentem `http://...` oznacza to, że wczytywana jest strona z zasobów internetowych. A jeżeli będzie to po prostu nazwa pliku `html`, wówczas otwierany jest plik z katalogów na naszym komputerze.

Kolejny parametr tj. opcja `_self` oznacza, że plik podstrony zostanie otwarty w aktualnym oknie przeglądarki. Dostępne są także opcje:

- `_blank` – otwarcie pliku nastąpi w nowym oknie przeglądarki,
- `_parent` – otwarcie nowego pliku w oknie rodzicielskim,
- `_top` – otwarcie pliku w ramce najwyższego poziomu, w aktualnym oknie przeglądarki.

Na bazie powyższych informacji proszę wykonać projekt strony z dużą liczbą podstron i ciekawą grafiką. Powodzenia:-)