

Podstawy animacji Flash

Zajęcia 2 – animacje z użyciem ActionScript 3.0

dr Joanna Sekulska-Nalewajko
Instytut Informatyki Stosowanej Politechniki Łódzkiej



Łódź, 2017

Spis treści

1	Wprowadzenie do zasad stosowania ActionScript 3.0.....	3
1.1	Sterowanie listwą czasową.....	3
1.2	Właściwości obiektów	4
1.3	Zdarzenia i ich nasłuchiwanie	4
2	Zadania do realizacji	7
2.1	Zmiana kursora.....	7
2.2	Aktywny mur	9
2.3	Teleportacja	12
2.4	Lupa	16

1 Wprowadzenie do zasad stosowania ActionScript 3.0

ActionScript 3.0 umożliwia sterowanie animacjami, interakcję z użytkownikiem oraz dynamiczną zmianę zawartości projektu (np. pobieranie treści projektu z zewnętrznych źródeł).

Skrypty można umieszczać w klatkach osi czasu za pomocą panelu Operacji (*Okno > Operacje* lub F9) lub w osobnych plikach o rozszerzeniu .as, które następnie muszą zostać skojarzone z odpowiednimi plikami .swf.

1.1 Sterowanie listwą czasową

Sterowanie listwą czasową polega na umieszczeniu w klatce osi czasu komendy sterującej (metody). Komenda taka zostanie wykonana w momencie, gdy wskaźnik odtwarzania klatek dotrze do klatki ze skrypsem.

stop();

– metoda zatrzymuje odtwarzanie listwy czasowej w miejscu umieszczenia skryptu;

gotoAndStop(nr_klatki); gotoAndStop(„etykieta”); gotoAndStop(nr_klatki, „Scena 2”);

– zatrzymuje odtwarzanie listwy czasowej po uprzednim przemieszczeniu się do wskazanej klatki, zamiast numeru klatki można podać także jej etykietę; a jeśli projekt zawiera kilka scen, to należy wskazać tę scenę, do której chcemy się przekierować wraz z numerem klatki;

gotoAndPlay(nr_klatki); gotoAndPlay(„etykieta”); gotoAndPlay(nr_klatki, „Scena 2”);

– powoduje przekierowanie odtwarzania listwy czasowej we wskazane miejsce;

nextScene(); prevScene();

– powoduje przekierowanie odtwarzania listwy do kolejnej/poprzedniej sceny;

nextFrame(); prevFrame();

– powoduje przekierowanie odtwarzania listwy do kolejnej/poprzedniej klatki.

Istnieją także mechanizmy sterowania listwą czasową na różnych poziomach zagnieżdżeń. Jeśli skrypt umieszczamy na głównej osi czasu i chcemy sterować zagnieżdżonym na niej elementem (**posiadającym nazwę instancji!**), to musimy poprzedzić komendę sterującą nazwą instancji tego obiektu:

this.kolo_mc.gotoAndPlay(5);

Jeśli chcemy sterować listwą nadrzędną z poziomu listwy czasowej zagnieżdżonego na niej klipu możemy odnieść się do listwy nadrzędnej w dwojaki sposób. Pierwszy sposób to sposób tzw. adresowania względnego. Tu słowo kluczowe **parent** wskazuje zawsze na listwę nadrzędną, bezpośrednio gnieźdzącą obiekt wskazujący.

Object(this.parent).gotoAndStop(2);

W związku z tym, przy drugim poziomie zagnieżdżenia (na głównej osi czasu jest zagnieżdżony klip, w którym jest zagnieżdżony kolejny klip), jeżeli chcielibyśmy zawiadować listwą nadrzędną w stosunku do listwy gnieźdzącej nasz obiekt to musielibyśmy użyć zapisu:

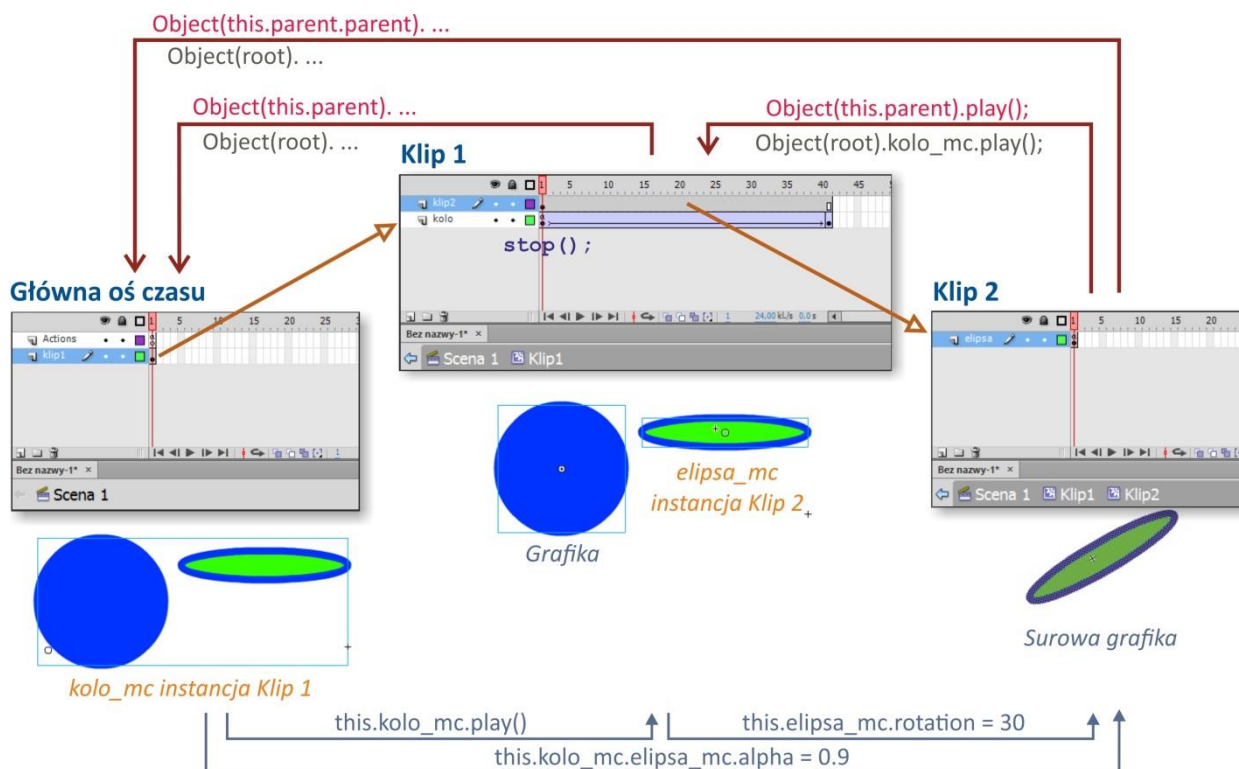
Object(this.parent.parent).gotoAndStop(2);

Z kolei przy adresowaniu bezwzględnym obowiązuje słowo kluczowe **root**, które oznacza główną oś czasu, oraz z zasady potrzebne są nazwy instancji jeśli odwołujemy się do obiektów zagnieżdżonych. Odwołanie do głównej osi czasu, obojętnie z jakiego poziomu zagnieżdżenia, wygląda następująco:

Object(root).gotoAndStop(2);

Jeśli z poziomu klipu znajdującego się na drugim poziomie zagnieżdżenia (w klipie o nazwie instancji *kolo_mc*) chcemy zarządzać osią czasu klipu gnieźdzącego to w sposób bezwzględny zapisujemy to następująco:

Object(root).kolo_mc.play();



Schemat obrazujący sposób odwoływania się do obiektów zagnieżdżonych w celu starowania nimi. Kolor czerwony (na górze rysunku ponad liniami) oznacza adresowanie względne, a pozostałe kolory reprezentują adresowanie bezwzględne.

1.2 Właściwości obiektów

Jeśli nadamy nazwy instancji obiektom (klipom, przyciskom, polom tekstu dynamicznego, kontenerom) to mamy możliwość zarządzania nimi z poziomu ActionScript. W pracy z klipami głównie interesuje nas zarządzanie ich osią czasu oraz atrybutami związanymi z ich stroną wizualną. Inne obiekty można np. przemieszczać lub skalować.

Właściwość obiektu można zmienić w prosty sposób. Jeśli chcemy ustanowić położenie obiektu, to posługujemy się jego właściwością **x** i **y**, jeśli chcemy tylko zmienić widoczność, to odnosimy się do właściwości **alpha**. Jeśli obiekt ma być całkiem niewidoczny i niedostępny dla użytkownika to używamy właściwości **visible**. Inne popularne właściwości to **width**, **height**, **rotation**, **scaleX**, **scaleY** (szerokość, wysokość, obrót, skalowanie w osi x i y), a poza tym jest wiele, wiele innych.

```

nazwa_instancji.x = 160;           //położenie obiektu względem osi poziomej
nazwa_instancji.y = 340;          //położenie obiektu względem osi pionowej
nazwa_instancji.alpha = 0.5;      //częściowa widoczność, dozwolone są wartości w zakresie 0 - 1
nazwa_instancji.visible = false;  //jeśli true, to obiekt jest widoczny
    
```

1.3 Zdarzenia i ich nasłuchiwanie

Zdarzenie, to inaczej pretekst do wykonania przygotowanego kodu ActionScript. Zdarzeniem jest na przykład kliknięcie przycisku, załadowanie treści z pliku, upływanie określonego czasu oraz wiele, wiele innych „zjawisk” które mogą mieć miejsce w obrębie naszego projektu. Poznajmy zaletę zdarzeń na przykładzie par: przycisk + kliknięcia myszą oraz klip + odtworzenia klatki sceny. Pierwszy z obiektów tej pary jest tzw. odbiornikiem zdarzenia, a drugi zdarzeniem. Do tych elementów musi dołączyć jeszcze

stan świadomego oczekiwania („czuwania”) na zdarzenie (nie na każde zdarzenie musimy reagować) oraz wreszcie wskazanie co takiego stanie się, gdy to zdarzenie będzie miało miejsce.

Najprościej omówić to na przycisku. Użytkownik stworzył takowy i chce, aby kliknięcie na ten przycisk otwierało przeglądarkę internetową z konkretnym adresem strony www. Przycisk musi zostać umieszczony na osi czasu i musi mieć nadaną **nazwę instancji** (np. *buton_btn*). Teraz kolej na nasłuchiwanie czy przycisk został kliknięty. Metodą nasłuchiwania wszystkich zdarzeń jest **addEventListener**. Obowiązuje tu następujący zapis:

```
Odbiornik_zdarzenia.addEventListener(...typ zdarzenia..., ...co ma nastąpić w rezultacie zdarzenia...);
```

Dla przycisku wyglądałoby to następująco. Typ zdarzenia wskazuje na kliknięcie myszką (możliwe są też inne zdarzenia związane z myszką, co pokazano niżej), a **fl_wwwOpen** jest nazwą funkcji, którą należy zdefiniować osobno. W ramach funkcji określamy, jaka będzie reakcja na wciśnięcie przycisku.

```
buton_btn.addEventListener(MouseEvent.CLICK, fl_wwwOpen);
```

```
var myUrl:URLRequest = new URLRequest("http://www.iis.p.lodz.pl")
function fl_wwwOpen(event:MouseEvent):void{
    navigateToURL(myUrl);
}
```

Inne zdarzenia myszy obsługiwane przez klasę obiektów MouseEvent:

MOUSE_OVER – kursor myszy nad obiektem

MOUSE_DOWN – przycisk myszy jest wciśnięty

MOUSE_UP – przycisk myszy nie jest wciśnięty

MOUSE_MOVE – przesunięcie myszy kursora po ekranie

MOUSE_OUT – kursor opuszcza obszar obiektu docelowego

MOUSE_WHEEL – kółko myszy obraca się

CLICK – naciśnięcie i zwolnienie przycisku myszy

DOUBLE_CLICK – szybkie dwukrotne naciśnięcie i zwolnienie przycisku myszy

Odbiornikiem zdarzeń myszy mogą być także klipy, ramki tekstowe, kontenery i scena.

Zdarzenie **ENTER_FRAME** oznacza wejście na kolejną ramkę osi czasu. Zdarzenie to obowiązuje zarówno dla osi jednoramkowych, jak i z większą liczbą ramek. Zarówno dla animacji z jedną ramką jak i tych wieloramkowych zdarzenie **ENTER_FRAME** generowane jest z częstotliwością ustaloną w FPS (liczba klatek na sekundę w pliku SWF).

Wykorzystanie tego zdarzenia jest sensowne zazwyczaj wtedy, gdy chcemy poprowadzić animację za pomocą skryptów. Skojarzenie animacji z liczbą klatek na sekundę w pliku SWF jest korzystne, ponieważ jest to częstotliwość, z jaką program Flash Player lub środowisko AIR aktualizuje ekran.

Większość programistów, korzysta ze zdarzenia **ENTER_FRAME** jako sposobu na powtarzanie określonych operacji w czasie. Możliwe jest np. napisanie kodu, który będzie wykrywał zdarzenie **ENTER_FRAME** i przesuwał animowaną kulkę o pewną odległość w każdej klatce, i w miarę odświeżania ekranu (w każdej klatce) kulka będzie ponownie rysowana w nowym położeniu, co będzie powodowało wrażenie ruchu.

Każdy obiekt wyświetlany zawiera zdarzenie **ENTER_FRAME**. Odbiornikiem zdarzenia **ENTER_FRAME** może być też SCENA animacji (obiekt **stage**).

Jeśli chcemy zrezygnować z nasłuchiwania zdarzenia możemy użyć metody **removeEventListener**.

```
function fadeCircle(event:Event):void {  
    circle.alpha -= .05;  
    if (circle.alpha <= 0) {  
        circle.removeEventListener(Event.ENTER_FRAME, fadeCircle);    }  
    }  
circle.addEventListener(Event.ENTER_FRAME, fadeCircle);
```

***Uwaga:** Innym sposobem powtarzania operacji w czasie jest wykorzystanie klasy `Timer`. Instancja `Timer` wywołuje powiadomienie o zdarzeniu za każdym razem po upływie określonego czasu. Możliwe jest napisanie kodu, który będzie wykonywał animację poprzez obsługę zdarzenia licznika czasu klasy `Timer`, i zostanie ustawiony odpowiednio krótki okres czasu (część sekundy).*

Zwróć uwagę na typ zdarzenia `TIMER` w poniższym przykładzie.

```
var myTimer:Timer = new Timer(100,50);  
myTimer.start();  
  
myTimer.addEventListener(TimerEvent.TIMER,ballAlpha);  
  
function ballAlpha(event:Event):void {  
    ball_mc.alpha -= 0.05;  
}
```

2 Zadania do realizacji

2.1 Zmiana kursora

Na pewno nie raz widzieliście, że w niektórych animacjach Flash'a zmieniony jest kursor. Teraz właśnie sprawimy, że podmieniony zostanie on na stworzony przez nas obrazek.

ETAPY PRACY:

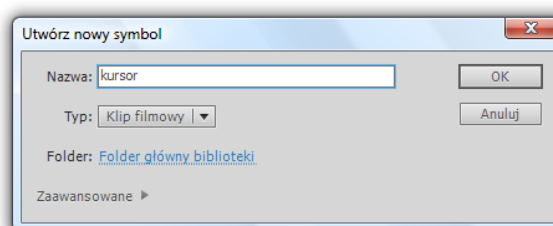
Krok 1 – Tworzenie klipu

Krok 2 – Praca w scenie głównej

Krok 1 – Tworzenie klipu

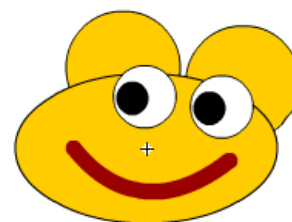
Otwieramy nowy plik i nadajemy mu określone dowolne parametry (tło, wielkość ramki, fps).

Niezwłocznie z menu na górze wybieramy polecenie **Wstaw > Nowy symbol** i tworzymy symbol typu Klip filmowy o nazwie np. „*kursor*”.



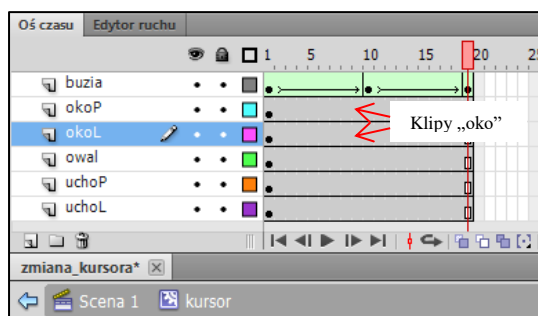
Po naciśnięciu *OK* otwiera się okno robocze symbolu, które posiada taki sam wygląd jak pole robocze sceny.

W pole robocze wstawiamy elementy graficzne naszego kursora. Należy zadbać, o to aby zachować odpowiednią kolejność warstw lub kolejność rysowania elementów na jednej warstwie (tych elementów, których nie zamierzamy poddać animacji).

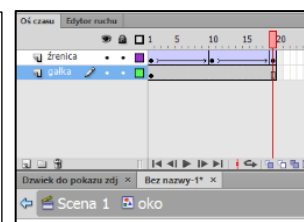


Tworzenie obiektu odbywa się w sposób standardowy za pomocą narzędzi z przybornika.

Jeśli chcemy stworzyć obiekt animowany wymagane jest wtedy utworzenie tylu warstw ile mamy elementów graficznych, które chcemy animować.



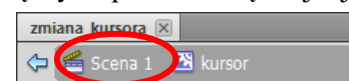
Oto przykładowy wygląd warstw w klipie „kursor”. W tym projekcie animowane są także oczy. Oko jest także klipem, w którym wykonano prostą animację klasyczną ruchu gałki ocznej. Klip wstawiono dwukrotnie (oko lewe i oko prawe). - klipy te nie muszą być na osobnych warstwach, jeśli nie chcemy ich animować.



Krok 2 – Praca w scenie głównej

Po narysowaniu obrazka wracamy do naszej głównej sceny - przełącznik między klipem a sceną znajduje się tuż ponad stołem montażowym.

Musimy teraz umieścić nasz klip „*kursor*” w scenie głównej. Do wstawiania gotowych symboli służy panel **Biblioteka** (powinien znajdować się np. z prawej strony okna roboczego. Jeśli Biblioteka nie jest otwarta to, jak każdy inny panel, dostępna jest w menu **Okno**).

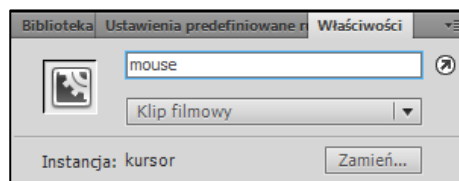


Aby wstawić obiekt z biblioteki wystarczy zaznaczyć klatkę, do której chcemy wstawić obiekt i z okienka podglądu biblioteki przeciągnąć obiekt w pole robocze sceny.



Jeśli klip po wstawieniu jest zbyt duży, to można zmienić jego proporcje skalując go narzędziem **Przekształcanie swobodne**.

Wstawionemu klipowi nadajemy **nazwę instancji** (np. *mouse*) – dzięki niej nasz klip będzie rozpoznawany przez kod ActionScript – nazwę instancji wpisujemy w oknie **Właściwości** klipu.



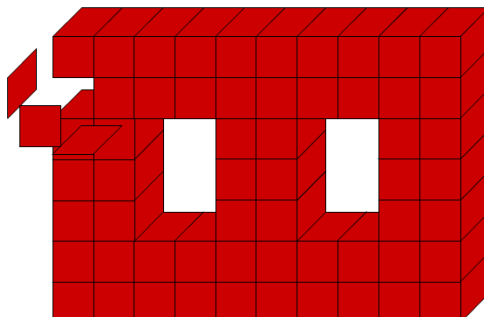
Zaznaczamy klatkę z klipem i wybieramy opcję **Operacje** (Windows: **F9**). Otworzy się okno do pisania kodu, w którym wstawiamy ciąg poleceń, które ukryją aktualny kursor a w jego miejsce wstawią nasz:

```
1 import flash.events.MouseEvent;
2
3 Mouse.hide();
4
5 function mouseMoving(myevent:MouseEvent):void{
6     mouse.x = stage.mouseX;
7     mouse.y = stage.mouseY;
8
9     myevent.updateAfterEvent();
10 }
11 stage.addEventListener(MouseEvent.CLICK, mouseMoving);
```

Pierwszy fragment kodu ukrywa mysz systemową, 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.

Teraz testujemy projekt. Jeśli wszystko działa to pozostało tylko zapisanie i opublikowanie animacji.

2.2 Aktywny mur



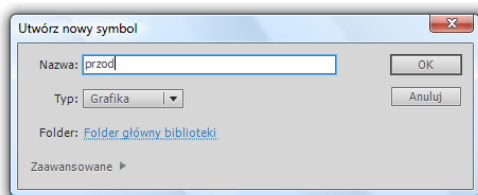
ETAPY PRACY:

- Krok 1: Tworzenie symboli graficznych
- Krok 2: Tworzenie przycisku
- Krok 3: Tworzenie klipu
- Krok 4: Praca w scenie głównej

Krok 1 – Tworzenie symboli graficznych

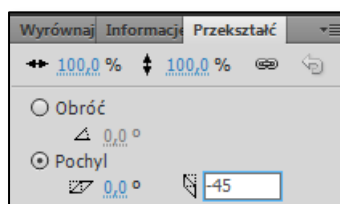
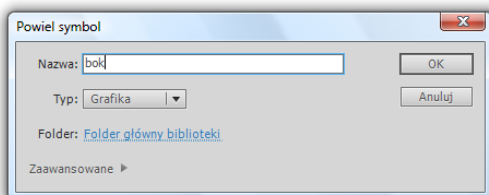
Otwórz nowy plik Flasha. Nie zmieniaj właściwości dokumentu.

Utwórz kolejno trzy elementy graficzne reprezentujące ściany kostki, przy czym tylko pierwszy element utwórz od podstaw (Wstaw > Nowy symbol) – nazwij go „przod”:

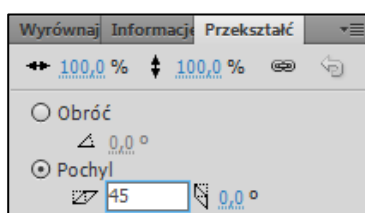


Wewnątrz symbolu grafiki narysuj przednią ścianę - kwadrat o wymiarach 50x50 pt.

Kolejne ściany można utworzyć na bazie tego prostokąta. W bibliotece duplikuj istniejący symbol (polecenie **Powiel**) i kopię nazwij „bok”. Edytuj symbol „bok” i za pomocą palety **Przekształć** pochyl kwadrat „w pionie” o kąt -45° :

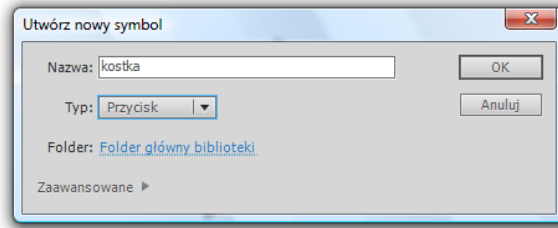


Jeszcze raz duplikuj symbol „przod” i duplikat nazwij „gora”. Edytuj ten symbol i za pomocą palety **Przekształć** spłaszcz kwadrat tak, aby uzyskać górną ściankę kostki (Pochyl „w poziomie” o kąt 45°):

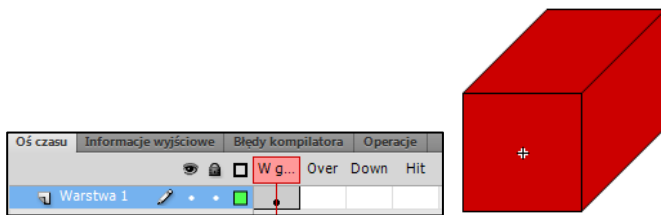


Krok 2 – Tworzenie przycisku

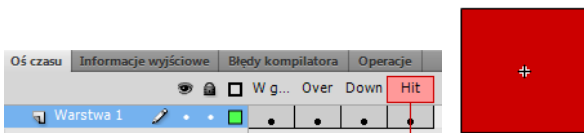
Wstaw nowy symbol – **przycisk** - o nazwie „kostka”:



Wewnątrz przycisku w klatce **W górze** wstaw z biblioteki gotowe ściany kostki i ułóż je w jeden obiekt – kostkę:

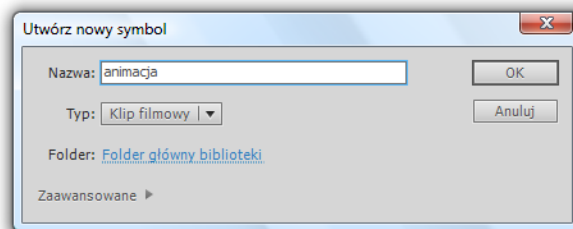


Skopiuj wszystkie elementy przycisku do klatek **Over**, **Down** i **Hit** - wystarczy w omawianych klatkach użyć klawisza F6. W polu **Hit** przycisku należy pozostawić tylko przednią ścianę kostki:

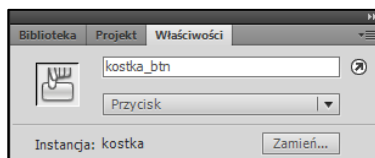


Krok 3 – Tworzenie klipu

Tworzymy pusty klip o nazwie „animacja”:

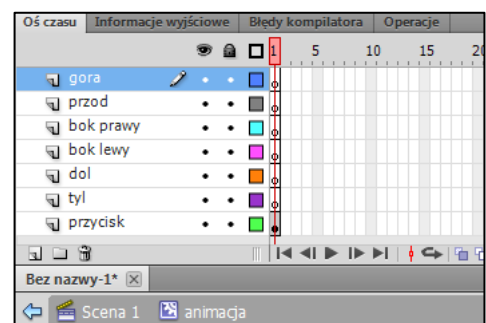


W klatce nr 1 na liście czasowej klipu wstawiamy stworzony przycisk (z **Biblioteki**) i nadajemy mu nazwę instancji „kostka_btn”



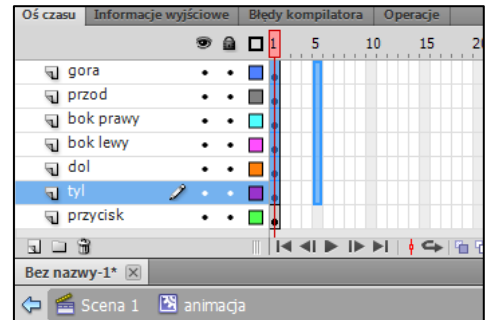
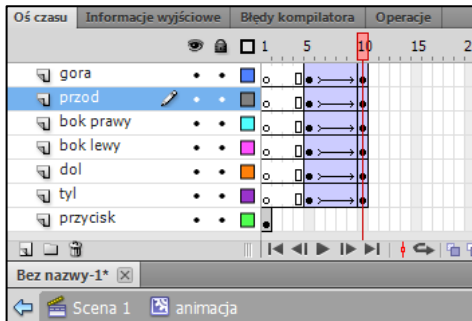
Wstawiamy 6 nowych warstw i nazywamy je odpowiednio do ścianek kostki.

Umieszczamy w nich obiekty graficzne składające się na elementy kostki (są to wszystkie ściany, nawet te niewidoczne!). Układamy je dokładnie na przycisku, który znajduje się w najniższej warstwie.

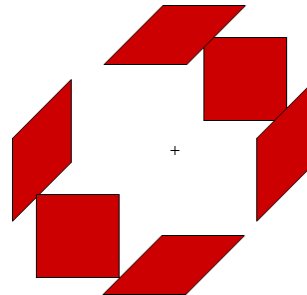


Następnie zaznaczamy wszystkie klatki ze ściankami i przenosimy je do klatki nr 5.

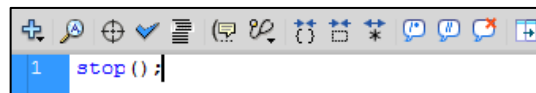
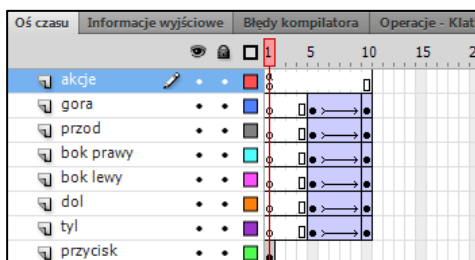
Tam utworzymy efekt otwieranego pudełka. Na zaznaczonych klatkach otwieramy menu podręczne i wybieramy polecenie **Utwórz klasyczną animację**.



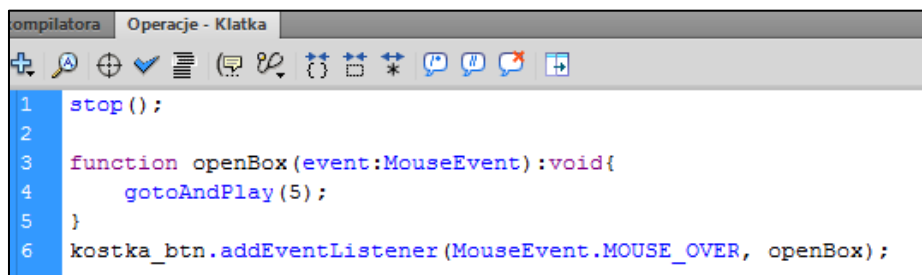
Zaznaczamy teraz klatki nr 10 i wstawiamy tam klatki kluczowe (F6). W klatce 10 każdy element należy przesunąć tak, aby kostka się rozpadła.



Dodajemy jeszcze jedną warstwę i nazywamy ją „akcje”. Otwieramy panel **Operacje** i wstawiamy na początek akcję **stop()** ;



Następnie poniżej akcji **stop()** ; wpisujemy funkcję, która uruchomi dalszą część tworzonego klipa:



Krok 4 – Praca w scenie głównej

Teraz wracamy do naszej głównej sceny. Wstawiamy klip z animowaną kostką. Możemy wstawić takich klipów na stronę kilka tworząc z nich jakiś fajny murek (patrz obrazek na początku tego ćwiczenia) - po uruchomieniu filmu jego cegiełki będą się on efektownie rozpadać.

2.3 Teleportacja

W tym ćwiczeniu stworzymy ciekawy efekt teleportacji.



ETAPY PRACY:

Krok 1: Ustawienie środowiska roboczego

Krok 2: Wstępne ustawienie zawartości warstw na scenie głównej

Krok 3: Praca wewnątrz klipu

Krok 4: Dodanie skryptu wewnątrz klipu

Krok 5: Końcowe ustawienie zawartości warstw na scenie głównej

Krok 1 – Ustawienie środowiska roboczego

Otwórz plik *obrazy_teleportacja fla* z lokalizacji podanej przez prowadzącego i ustaw właściwości filmu: Rozmiar stołu montażowego: 450 x 500 px, 33 Kl./s, AS 3.0 i wersja Flash Player nie mniej niż 10.

Krok 2 – Wstępne ustawienie zawartości warstw na scenie głównej

Zmień nazwę warstwy 1 na „tło”. Przenieś na tę warstwę z biblioteki obraz „background”, który będzie tłem animacji. Za pomocą panelu Wyrównaj precyzyjnie umieść zdjęcie w centrum stołu montażowego.

Wstaw nową warstwę, nazwij ją „klip”. Będąc w tej warstwie wstaw na stół montażowy obraz „woman” z biblioteki. Można postać nieznacznie zmniejszyć i dopasować jej położenie do tła.

Zaznacz obraz z postacią i konwertuj go do klipu (F8). Nazwij klip „teleportacja”.

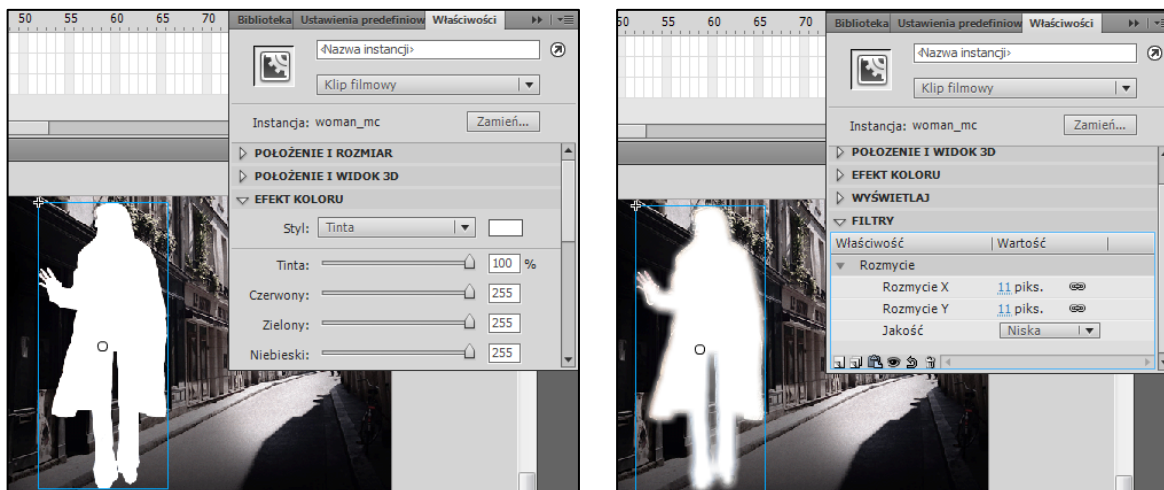


Krok 3 – Praca wewnątrz klipu

Edytuj klip w miejscu (poprzez dwukrotne kliknięcie postaci). Wewnątrz klipu zaznacz postać i ponownie konwertuj ją do klipu (nazwa klipu to np. „woman_mc”). Wciśnij Ctrl+C (czyli skopiuj klip).

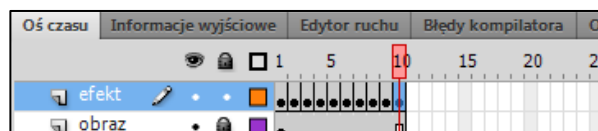
Wstaw nową warstwę i nazwij ją „efekt”. Warstwę pierwszą możesz nazwać „obraz”. Zablokuj warstwę „obraz”.

Zaznacz nową warstwę („efekt”) i będąc na scenie wstaw do niej skopiowany klip poleceniem: **Edytuj > Wklej w miejscu**. Wklejonej postaci dodaj efekt w panelu **Właściwości**: w sekcji **EFEKT KOLORU** wybierz styl **Tinta** i ustaw kolor biały i jego wartość na „100%”:

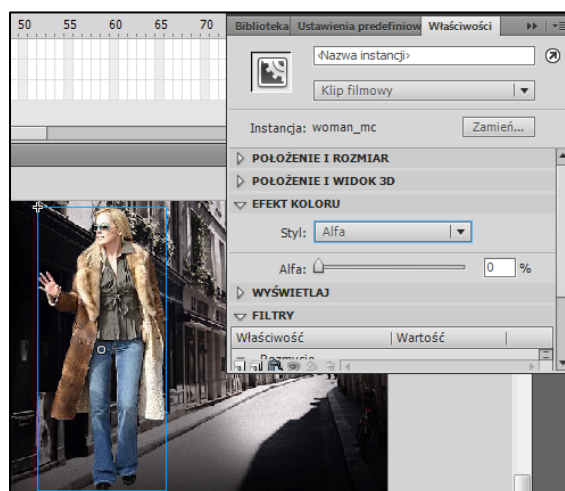


W tym samym panelu **Właściwości** przejdź do sekcji **FILTRY**. Dodaj filtr **Rozmycie** z ustawieniami RozmycieX = RozmycieY = 11 piks.

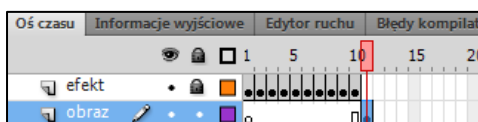
W warstwie „efekt” kliknij na klatkę 1 i wciśnij klawisz F6 dziewięć razy.



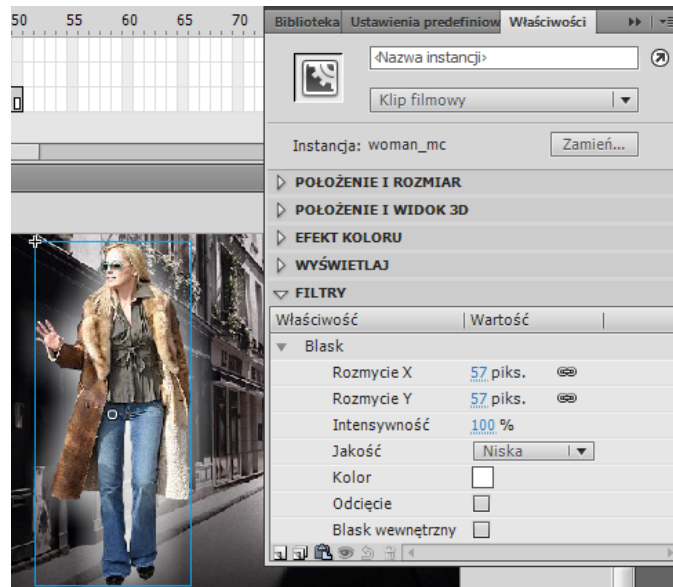
Klikamy na co drugą klatkę, z wyjątkiem dwóch ostatnich, zaznaczamy obrazek i w panelu **Właściwości** w sekcji **EFEKT KOLORU** zmieniamy właściwość **Alfa** na 0%.



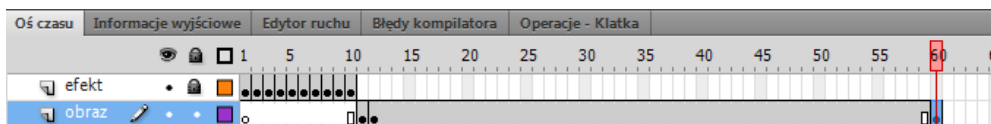
Blokujemy warstwę „efekt”, a odblokowujemy warstwę „obraz”. Zaznaczamy w niej klatki 2-10 i usuwamy je poleceniem **Usuń klatki** z menu podręcznego. Klikamy na pierwszą klatkę i metodą przeciągnij i upuść przenosimy ją do klatki 11:



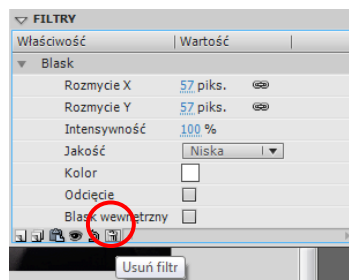
Będąc w klatce 11 w warstwie „obraz” zaznaczamy postać, w panelu **Właściwości** wybieramy znów sekcję **FILTRY** i dodajemy filtr **Blask**. Ustawiamy opcje jak na rysunku poniżej:



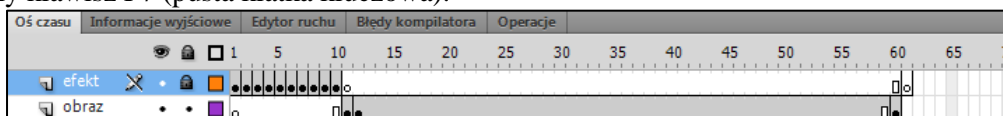
Klikamy kolejno na klatkę 12 i 60 i wciskamy klawisz F6.



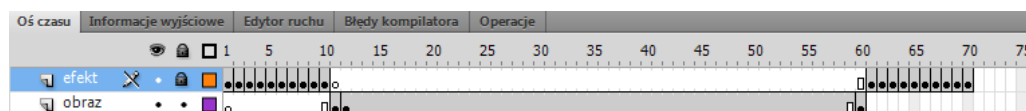
Zaznaczamy klatkę 12, a następnie postać i w sekcji **FILTRY** usuwamy efekt **Blask** (używając ikony kosza).



Blokujemy warstwę „obraz”, odblokowujemy warstwę „efekt”. Na tej warstwie klikamy na klatkę 11 i 61 i wciskamy klawisz F7 (pusta klatka kluczowa).



Zaznaczamy równocześnie pierwsze dziesięć klatek warstwy „efekt” i kopiujemy je (polecenie **Kopiuj klatki** z menu podręcznego). Idziemy do klatki 61 i wklejamy tam skopiowane klatki (poleceniem **Wklej klatki**).



Krok 4 – Dodanie skryptu wewnątrz klipu

Pozostało tylko w klatce powiedzmy 90-tej klipu „teleportacja” umieścić skrypt, który będzie odwoływał się do kolejnej klatki na głównej listwie czasowej. Skrypt możemy umieścić na dowolnej warstwie, np. warstwie „efekt” lub specjalnie utworzonej do tego celu. Jeśli wstawiasz skrypt na warstwie „efekt” to nie zapomnij o wstawieniu pustej klatki kluczowej (F7) do klatki 71 i następnie klatki kluczowej (F6 lub F7) do klatki 90.

Otwórz panel **Operacje** i wpisz następujący kod:

The image shows a code editor window titled "Operacje - Klatka" with a menu bar containing "Modyfikuj", "Tekst", "Polecenia", "Sterowanie", "Debuguj", "Okno", and "Pomoc". The code is as follows:

```
1 var n:uint;
2 if(n<2){
3     MovieClip(root).nextFrame();
4     n++;
5 }
6 else{
7     MovieClip(root).gotoAndPlay(1);
8     n=0;
9 }
```

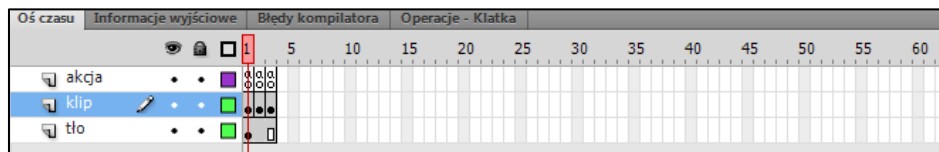
Krok 5 – Końcowe ustawienie zawartości warstw na scenie głównej

Powróć do głównej sceny filmu.

Na warstwie „klip” wstaw dwie kolejne klatki kluczowe (F6; w ten sposób powieli się zawartość klatki 1). W dodanych klatkach zmień położenie i wielkość klipu.

Na warstwie „tło” wstaw dwie zwykłe klatki kluczowe (F5).

Dodaj nową warstwę z trzema klatkami kluczowymi – nazwij ją „akcja”. W każdej klatce oddzielnie umieść akcję **stop()**;



Przetestuj i opublikuj film.

2.4 Lupa

Celem ćwiczenia jest pokazanie zastosowań warstwy maskującej oraz efektów dodawania skryptów ActionScript 3.0.

ETAPY PRACY:

- Krok 1: Import zdjęcia i przekształcenie go w klip
- Krok 2: Tworzenie nowego klipa z efektem maski
- Krok 3: Praca w scenie głównej – dodanie akcji do obsługi lupy
- Krok 4: Tworzenie przycisków
- Krok 5: Dodanie akcji w scenie głównej



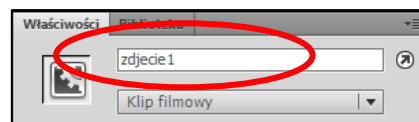
Krok 1 – Import zdjęcia i przekształcenie go w klip

Na początek importujemy dowolne ale niezbyt duże zdjęcie np. z [internetu](#). Jeśli zdjęcia mamy już na dysku stacji roboczej to we Flashu za pomocą polecenia z menu **Plik** → **Importuj** → **Importuj do biblioteki** umieszczamy zdjęcie w **Bibliotece** projektu.

W panelu **Biblioteka** pojawia się nowy wpis, który przenosimy na scenę główną.

Leżące na scenie głównej zdjęcie zaznacz i przekształć w *Klip filmowy* (poleceniem z menu kontekstowego **Konwertuj na symbol** lub skrótom **F8** dostępnym tylko w Windowsie).

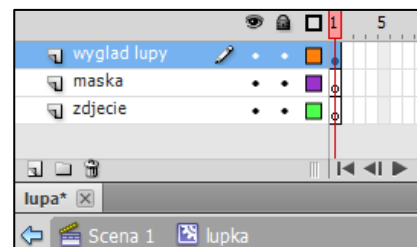
Zdjęciu-klipowi nadaj nazwę instancji *zdjecie1* (w okienku '<Nazwa instancji>' panelu **Właściwości**). **UWAGA!! – pisane bez polskich ogonków.**



Krok 2 – Tworzenie nowego klipa z efektem maski

Wstaw nowy *Klip filmowy* (**Wstaw** → **Nowy symbol** <typ: *Klip filmowy*>, nazwa np. „lupka”), w którym skonstruuj lupę powiększającą.

Kategorycznie musimy mieć do tego trzy warstwy – dodaj je, nazwij i przestrzegaj ich kolejności jak na rysunku obok.



W pierwszej warstwie od góry („wygląd lupy”) rysujemy kształt naszej lupy: okrąg i rączka lupy. Okrąg może posiadać wypełnienie – ale tylko po to, aby je za chwilę wyciąć (Windows: Ctrl+x, Mac: Cmd+x) i wkleić (Windows: Ctrl+v, Mac: Cmd+v) na warstwę „maska”.

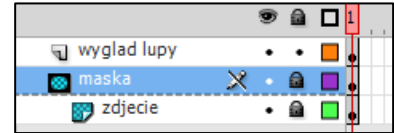


Na warstwie „maska” umieszczamy okrąg (wklejamy wypełnienie wycięte wcześniej). Okrąg ten będzie stanowić szkło, przez które będziemy oglądać zdjęcie. Okrąg ten musi leżeć dokładnie w tym samym położeniu jako wypełnienie lupy. Jeśli mamy więc wycięty wcześniej wypełnienie to stosujemy polecenie **Edycja > Wklej w miejscu**.



Na warstwie „zdjęcie” wstaw utworzony w kroku 1 symbol *Klipu filmowego*. Instancji tego symbolu nadajemy nazwę <Nazwa instancji>: *zdjecie2*.

Warstwie „maska” nadajemy tryb maski (patrz rysunek po prawej). Opcję „maska” znajdziecie w menu podręcznym po zaznaczeniu warstwy.

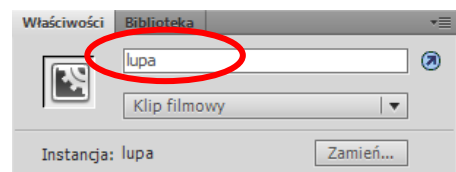


I to koniec pracy z tym klipem. Całość powinna wyglądać jak na rysunku:



Krok 3 – Praca w scenie głównej – dodanie akcji do obsługi lupy

Przechodzimy do sceny głównej, gdzie jest już nasze *zdjecie1*. Wstaw tu klip filmowy „lupka” z panelu **Biblioteka** i nadaj mu nazwę instancji: *lupa*. Nie trzeba dodawać nowej warstwy.



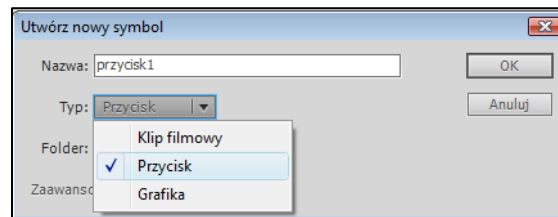
W głównej scenie tworzymy teraz warstwę „akcje” (rysunek obok). W klatce tej warstwy wpisujemy akcję:

```
1 var skala:Number = 2;
2
3 lupa.addEventListener(Event.ENTER_FRAME, moveLupa)
4
5 function moveLupa(e:Event):void{
6 // położenie lupy
7     lupa.x = mouseX-40;
8     lupa.y = mouseY-40;
9
10 // definiujemy zmienne
11
12     var xpz1:Number = zdjecie1.x;
13     var xp1:Number = lupa.x;
14     var ypz1:Number = zdjecie1.y;
15     var yp1:Number = lupa.y;
16
17 //skala zdjęcia 2
18     lupa.zdjecie2.scaleX=this.scaleX*skala
19     lupa.zdjecie2.scaleY=this.scaleY*skala
20
21 //położenie zdjęcia 2
22     lupa.zdjecie2.x=(xpz1-xp1+(10*skala))*skala
23     lupa.zdjecie2.y=(ypz1-yp1+(10*skala))*skala
24 }
```

Wartość 10 wynika z przesunięcia środka "szkiełka" lupy do rzeczywistego środka klipu filmowego.

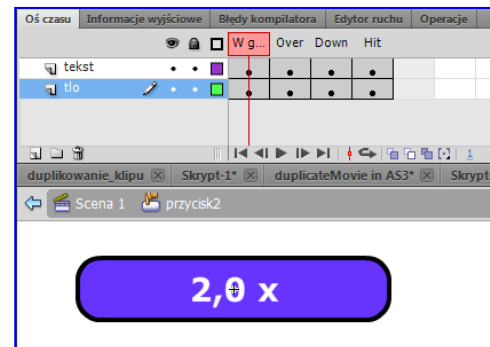
Krok 4 – Tworzenie przycisków

Tworzymy teraz dwa przyciski, dzięki którym będziemy mogli zmieniać skalę powiększenia. Przycisk to symbol, który wstawiamy poprzez polecenie **Wstaw > Nowy symbol**:



Przycisk ma charakterystyczną listwę czasową. Uzupełniamy tu z reguły wszystkie klatki (**Błąd! Nie można odnaleźć źródła odwołania.**). Wypełnijmy klatki grafiką i tekstem (mogą być na osobnych warstwach). Stany (klatki) mogą różnić się między sobą kolorystyką znajdujących się tam elementów. Kolory nie mają znaczenia tylko w klatce HIT oznaczającej pole reakcji przycisku.

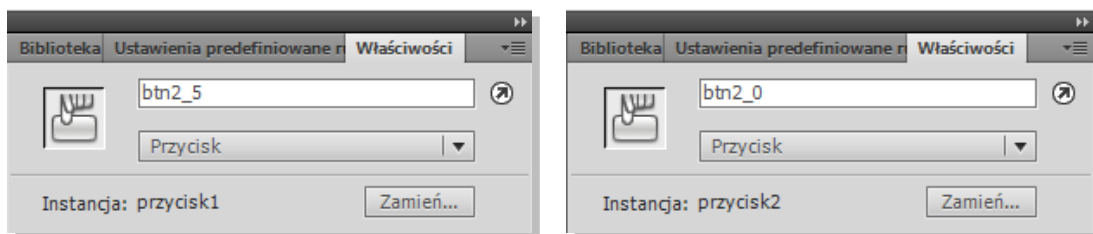
Z boku przedstawiono ostateczny wygląd listwy czasowej naszego przycisku.



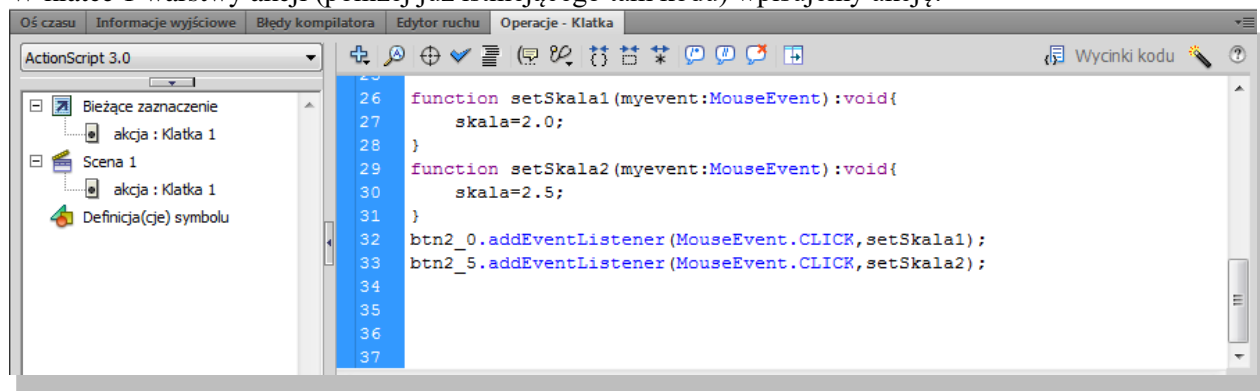
Potrzebujemy dwa przyciski. Gotowy przycisk w bibliotece możemy powielić (polecenie **Powiel...**) i w duplikacie zmienić np. treść tekstu na przycisku.

Krok 5 – Dodanie akcji w scenie głównej

Przyciski wstawiamy z biblioteki do sceny głównej i pozostanie nam jeszcze dodanie akcji. Zanim to jednak nastąpi musimy nadać nazwy instancji naszym przyciskom:



W klatce 1 warstwy akcji (poniżej już istniejącego tam kodu) wpisujemy akcję:



W ramach eksperymentowania wartość przypisaną do zmiennej *skala* można dowolnie zmieniać (dozwolone są nawet wartości ujemne).