

Wprowadzenie do ActionScript

"Action Script" to prosty język skryptowy umożliwiający dodawanie do strony wielu ciekawych elementów takich jak: linki do stron WWW, drukowania, przewijania, podmieniania kursora, sterowania obiektem za pomocą klawiatury i wiele innych. Tworząc skrypt masz w sumie niewiele do roboty, gdyż skrypty są pogrupowane i wstawienie każdego z nich polega na dwukrotnym kliknięciu na odpowiedni przycisk w panelu "Actions", przy zaznaczonej klatce lub obiekcie, do którego będzie się odnosił (np. dla linku będzie to przycisk).

Teraz wykonamy kilka prostych przykładów z użyciem "Action Script'u".

1. Nawigacja:

Trzy podstawowe polecenia odnoszące się do sterowania animacją:

- `stop()`; - zatrzymuje animację w klatce, w której umieszczony jest skrypt
- `gotoAndPlay(y)`; - przenosi animację z klatki x i puszcza ją od klatki y (umieszczamy w buttonie razem z opcją `onMouseEvent(Press){wydarzenie po wciśnięciu}`
- `gotoAndStop(y)`; - przenosi animację z klatki x i zatrzymuje ją w klatce y (również umieszczamy w akcjach przycisku)

2. Przeciąganie obiektów:

- Utwórz "Button" (może być animowany).
- Dzięki "Action Script" będzie możliwe przenoszenie go za pomocą myszki w obrębie animacji.
- Teraz wstaw go do animacji i nadaj mu nazwę poprzez: zaznaczenie go i wprowadzenie w panelu "Properties" nazwy ("Instance Name")
/Mój ma nazwę po prostu obiekt./
- Przy cały czas zaznaczonym przycisku, otwórz panel "Actions" i wpisz taki kod:

```
on (press) {  
startDrag("obiekt");  
}  
on(release) {  
stopDrag();  
}
```

W tym momencie trzymając klawisz myszy można go przenosić, a gdy klawisz zwolnisz to obiekt pozostanie tam gdzie go zostawisz.

3. Detektory – przemieszczanie Movie Clipa

- Narysuj dowolny kształt i przekonwertuj go w movie clipa.
- Umieść jeden klon tego movie clipa na głównej scenie (w prawej części sceny!).
- W oknie Properties w polu Instance Name wpisz ball i naciśnij Enter.
- Otwórz okienko Actions i wpisz:

```
onClipEvent (enterFrame) {  
_root.ball._x--;  
}
```

- Uruchom film (Ctrl+Enter)

Co widać? Coś, co narysowałeś przemieszcza się ze stałą prędkością w lewo. Czy to Ci nie przypomina liniowej animacji, którą do tej pory robiłeś przy pomocy *tweeningu*? Może prędkość nie jest ta sama, ale daje to te same efekty.

Ważna sprawa jest jeszcze detektor onClipEvent (mouseDown). Działa on jak detektor (release) przycisku, ale (UWAGA!) nie jest ważne w który obszar sceny klikniesz.

Znając już polecenie speed zmodyfikujmy kod:

```
onClipEvent (mouseDown) {  
    speed = 5  
    _root.ball._x-=speed;  
}
```

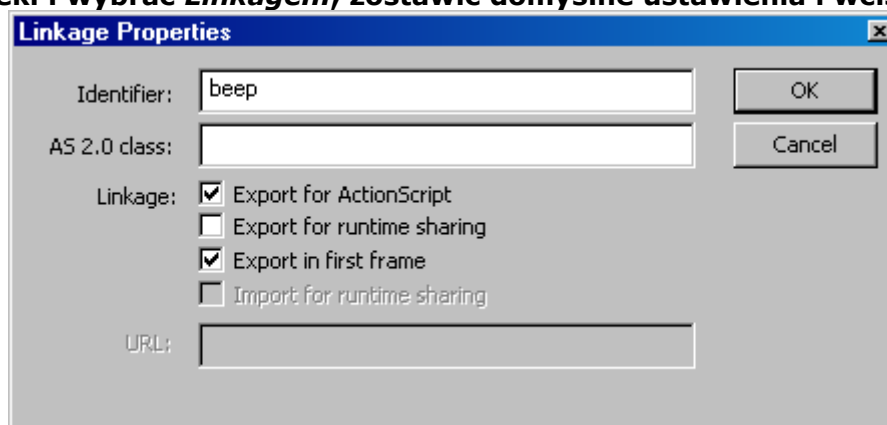
ActionScript jest językiem zorientowanym obiektowo. Oznacza to, że występują tu obiekty, które mają np. swoje właściwości.

Zacznijmy od obiektu typu **sound**. Dzięki AS możesz tworzyć obiekty sound i ustawiać ich dwie właściwości (głośność i panoramę dźwięku). Wykonamy proste ćwiczenie, wykorzystując wiedzę zdobytą w poprzednim paragrafie i trochę nowych rzeczy.

1. Utwórz nowy film Flash
2. Narysuj dowolny kształt i przekonwertuj go w przycisk
3. Wstaw do biblioteki swojego filmu jakiś plik dźwiękowy i nazwij go beep
3. Mając zaznaczony ten przycisk otwórz okno Actions i wpisz:

```
on (release) {  
    MySound = new Sound ();  
    MySound.attachSound ("beep");  
    MySound.setVolume (100);  
    MySound.setPan (-100);  
    MySound.start ();  
}
```

Wiem, że wygląda to skomplikowanie, ale tak nie jest. Po kliknięciu w przycisk tworzony jest nowy obiekt **sound** o nazwie MySound (ważny jest operator new). Najgorsza jest druga linia tego kodu. Dołącza ona z biblioteki symbol o nazwie beep. I tu pojawia się problem... Film nie będzie działał poprawnie, jeśli nie każemy Flashowi dołączyć tego symbolu do pliku swf (ponieważ on sam tego nie zrobi). Dlaczego? Flash jest "inteligentny" i nie dołącza do końcowego pliku symboli, które nie znajdują się na scenie filmu. A ten symbol się nie znajduje. Można sobie z tym jednak łatwo poradzić. **Musisz otworzyć okno *Library*. Następnie zaznaczyć symbol beep, otworzyć okno *Options biblioteki* i wybrać *Linkage...*, zostawić domyślne ustawienia i wcisnąć OK.**



To już działa! Tak trzeba dołączać do pliku swf, każdy symbol z biblioteki, który nie znajduje się na scenie, ale występuje w kodzie. Dwie następne linie ustawiają głośność i panoramę dźwięku. Jego

głośność to 100 %, a dźwięk usłyszymy w 100 % z lewego głośnika. Następna linia nakazuje, by Flash zagrał ten dźwięk.

Teraz obiekt **key**, który reaguje na naciskane klawisze klawiatury (a głównie na strzałki). Jeśli będziesz chciał zrobić grę we Flashu (np. łapanie czegoś, biegającym stworkiem), to wtedy docenisz obiekt key. Obiekt ten reaguje na naciśnięcie jakiegoś przycisku i wykonuje określoną akcję.

1. Utwórz nowy film Flasha. Ustaw 60 fps (to ważne!).
2. Narysuj dowolny kształt i przekonwertuj go w *movie clipa*
3. Nadaj instancji (klonowi) tego movie clipa nazwę circle
4. Otwórz okienko Actions tego klipu i wpisz:

```
onClipEvent(enterFrame) {  
    if (Key.isDown(Key.RIGHT)) {  
        _root.circle._x++;  
    }  
    if (Key.isDown(Key.LEFT)) {  
        _root.circle._x--;  
    }  
}
```

5. Ustaw klon w na środku dolnej części sceny.
6. I znów Ctrl+Enter.

Działa to dość prosto. Obiekt key sprawdza, czy przycisk został naciśnięty. Jeśli tak się stało zostaje wykonana akcja w nawiasach klamrowych.

ActionScript jest językiem zorientowanym obiektowo. Może zauważyłeś pewną rzecz... Często w kodzie AS używana jest kropka. Jeśli programowałeś już w innych językach, to zapewne już się z tym spotkałeś (np. w C++ do korzystania z akcesoriów używa się kropki). We Flashu kropka jest bardzo ważna. Często to przez złe zaadresowanie cały skrypt nie działa. Ale wróćmy do tematu. Wiesz, że we Flashu są trzy typy symboli, My zajmiemy się tymi najbardziej interaktywnymi. Są to klipy filmowe. Klipy filmowe mają swoje właściwości. Jak z nich korzystać? Musisz użyć kropki. Wykonamy teraz proste ćwiczenie, które wyjaśni o co chodzi w składni kropki. Będziemy zmniejszać właściwość `_alpha` klipu filmowego.

1. Utwórz nowy film Flasha z domyślnymi parametrami
2. Narysuj dowolny kształt i przekonwertuj go w movie clipa
3. Otwórz okienko Actions dla klipu symbol i wpisz poniższy skrypt:

```
onClipEvent (enterFrame) {  
    this._alpha -= 5;  
}
```

4. Przetestuj animacje (CTRL + ENTER).

Jest to dość prosta animacja, która po każdym przejściu klipu do następnej klatki zmniejsza o 5 jego parametr `_alpha`. Użyliśmy tu słowa **this**, które wskazuje, że skrypt odnosi się do tego klipu (bo wcale tak być nie musi! Możemy np. zmniejszać parametr `_alpha` innego klipu). Animacja bez tego słowa nie zadziała. Często w animacjach używa się także słowa `_root`, które oznacza główną linię czasu. Tu pojawia się dość duży problem... Adresowanie obiektów w skryptach. Musisz wiedzieć, że czymś innym jest klon klipu filmowego na scenie, a czymś innym symbol znajdujący się w bibliotece. Dlatego, aby wykorzystywać klipy filmowe w AS musisz nadawać ich klonom nazwę. Symbol z biblioteki możesz użyć kilka razy. Co z tego, że w bibliotece nazywa się on np. koło, jak na scenie jest 5 klonów tego symbolu z biblioteki. Jeśli napiszesz np. `_root.kolo...`, to Flash nie będzie wiedział

o co chodzi. Musisz nadać instancji klipu nazwę! Zapamiętaj to zdanie! To jest bardzo ważne. Pojawia się tu jeszcze jedna sprawa. Właściwości klipów. Wymieniono już parametr `_alpha`, ale do tego dochodzą choćby wykorzystane już przez nas współrzędne `_x` i `_y`, `_xscale` i `_yscale` (dzięki nim możesz manipulować wielkością klipu), `_xmouse` i `_ymouse`, które zwracają odpowiednie współrzędne położenia myszki. Na razie te właściwości wystarczą - teraz wykonamy jeszcze jedno ćwiczenie, które przybliży Ci adresowanie i wyjaśni, dlaczego tak ważne są nazwy instancji klonów klipów filmowych.

1. Utworz nowy...
2. Narysuj dowolny kształt i przekonwertuj go w Movie Clipa o nazwie ball.
3. Przeciągnij z biblioteki na scenę jeszcze jeden symbol.
4. Pierwszemu symbolowi nadaj nazwę symbol1, a drugiemu symbol2.
5. Dodaj następujący skrypt do symbol1:

```
onClipEvent (enterFrame) {
    _root.symbol2._x +=1;
}
```

Przetestuj film i zobacz jak to działa. Mimo, że akcję dodałeś do klonu o nazwie symbol1, to porusza się klon o nazwie symbol2. To wszystko dzięki adresowaniu. Możesz za pomocą jednego klipu zmieniać właściwości innego znajdującego się nawet na innym poziomie (o poziomach za chwilę).

6. Do klonu o nazwie symbol2 dodaj następującą akcję:

```
onClipEvent (enterFrame) {
    _root.symbol1._alpha -=2;
}
```

Klon o nazwie symbol2 zmienia natomiast właściwość `_alpha` klonu i nazwie symbol1.

4. Sterowanie animacją za pomocą myszy lub klawiatury

- a. Rzeczą ciekawą jest sterowanie animacją za pomocą myszy, lub klawiatury.
- b. Utwórz animację, którą chcesz poruszać, w formie Movie klipu
- c. I wpisz w jej panel akcji:

```
onClipEvent (load) {
    speed = 10;
}

onClipEvent (enterFrame) {
    // move up, down, left, or right

    if (Key.isDown(Key.LEFT) && !Key.isDown(Key.RIGHT)) {
        _x -= speed;
        _rotation = 270;
    }

    if (Key.isDown(Key.RIGHT) && !Key.isDown(Key.LEFT)) {
        _x += speed;
        _rotation = 90;
    }

    if (Key.isDown(Key.UP) && !Key.isDown(Key.DOWN)) {
        _y -= speed;
    }
}
```

```
    _rotation = 0;
}

if (Key.isDown(Key.DOWN) && !Key.isDown(Key.UP)) {
    _y += speed;
    _rotation = 180;
}

// move diagonally
if (Key.isDown(Key.LEFT) && Key.isDown(Key.UP) && !Key.isDown(Key.RIGHT)
&& !Key.isDown(Key.DOWN)) {
    _rotation = 315;
}

if (Key.isDown(Key.RIGHT) && Key.isDown(Key.UP) && !Key.isDown(Key.LEFT)
&& !Key.isDown(Key.DOWN)) {
    _rotation = 45;
}

if (Key.isDown(Key.LEFT) && Key.isDown(Key.DOWN) && !Key.isDown(Key.RIGHT)
&& !Key.isDown(Key.UP)) {
    _rotation = 225;
}

if (Key.isDown(Key.RIGHT) && Key.isDown(Key.DOWN) && !Key.isDown(Key.LEFT)
&& !Key.isDown(Key.UP)) {
    _rotation = 135;
}

// loop to opposite side of the area when the clip travels off-screen
if (_y<0) {
    _y = 400;
}

    if (_y>400) {
        _y = 0;
    }

if (_x<0) {
    _x = 550;
}

if (_x>550) {
    _x = 0;
}
}
```