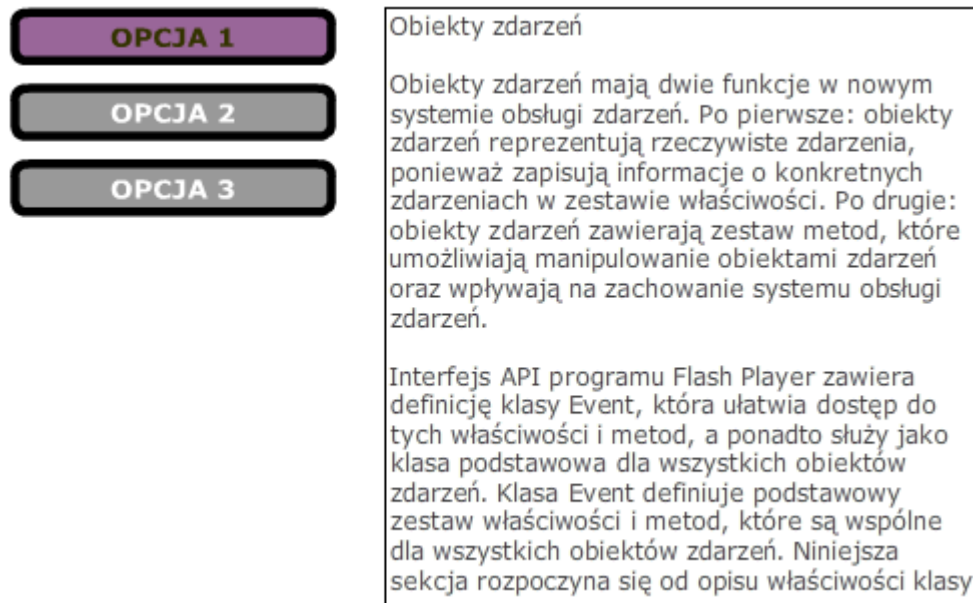


Ćwiczenie 4 – Dynamiczne wczytywanie tekstu z pliku .txt

Wykonamy dwa zadania z wczytywaniem zewnętrznych plików tekstowych. W pierwszym zadaniu wczytamy jeden plik tekstowy. W drugim podejściu za pomocą przycisków będziemy mogli decydować, który plik tekstowy zostanie wczytany. W obu przypadkach będzie możliwe przewijanie długiego tekstu w polu tekstowym, gdyż zbudujemy odpowiednią funkcję z użyciem metody scrollV.



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

PRZYKŁAD 1

Krok 1

Tworzenie pliku kontenera

Otwórz nowy plik i nie zmieniaj jego wyjściowych ustawień. Zapisz go na dysku lokalnym. W tej samej lokalizacji zapisz plik tekstowy „target.txt” (znajdziesz go w materiałach do ćwiczenia na mojej stronie www).

To koniec przygotowań. Kolejny etap ćwiczenia polega na napisaniu funkcji odpowiedzialnych za dynamiczne tworzenie pola tekstowego, ładowanie do niego odpowiedniego tekstu z zewnętrznego pliku tekstowego, dynamiczne formatowanie tekstu oraz dodanie funkcji, która powoduje, że za długi tekst będzie mógł być przewijany w polu tekstowym.

Krok 2

Pianie kodu

Na początek inicjalizujemy obiekt klasy URLLoader, która obsługuje wczytywanie zewnętrznych plików tekstowych oraz danych z plików XML. Następnie także inicjalizujemy instancję obiektu z klasy URLRequest, który pobierze nam docelowy adres wczytywanego pliku (lub po prostu jego nazwę).

Metoda URLLoader.load() przyjmuje pojedynczy parametr *request*, czyli mówiąc prościej dopiero ta metoda wczyta do naszego filmu docelowy plik tekstowy:

```
var textLoader:URLLoader = new URLLoader();  
var textURLRequest:URLRequest = new URLRequest("target.txt");  
textLoader.load(textURLRequest);
```

Brakuje nam jeszcze możliwości zobaczenia naszego tekstu. Plik musi być wczytany do określonego pola tekstowego. W tym celu wykonamy kolejne dwa kroki w pisaniu kodu.

Po pierwsze tworzymy instancję klasy TextField, czyli nasze pole tekstowe i w kolejnej linii zdecydujemy o dodaniu go do listy wyświetlania:

```
var mytextField:TextField = new TextField();  
addChild(mytextField);
```

Drugi konieczny element decydujący o możliwości zobaczenia tekstu to dodanie danych (tekstu), które obecnie zawarte są w obiekcie textLoader, do naszego pola tekstowego. Zrobimy to za pomocą właściwości htmlText pola tekstowego, która da nam możliwość formatowania tekstu, tak jak robi się to w dokumentach HTML.

Nie wykonamy tego jednak bezpośrednio. Wykonamy pewną funkcję, która wykorzysta zdarzenie complete obiektu textLoader. Inaczej mówiąc funkcja ta zadziała (pozwoli na wyświetlenie tekstu w polu tekstowym) tylko pod warunkiem, że plik tekstowy zostanie poprawnie pobrany (np. istnieje w podanej lokalizacji).

W ciele funkcji zawrzemy nie tylko samo załadowanie danych tekstowych do pola tekstowego (linia 1 w ciele funkcji) ale także formatowanie wyglądu i ustawienia pola tekstowego (linie 2-9 w ciele funkcji), dodanie formatu dla samego tekstu (linia 10 w ciele funkcji, format za chwilę zostanie utworzony – dlatego teraz jest ujęty w komentarz), a także ustawienie wygładzania tekstu (linia 11 w ciele funkcji) – bez tego nasz tekst nie wyglądałby zbyt ciekawie.

Na początku tego fragmentu kodu określamy, że celem zdarzenia complete i „odbiornikiem” tej funkcji jest obiekt textLoader:

```
textLoader.addEventListener(Event.COMPLETE, dataOK);  
  
function dataOK(myevent:Event):void{  
    mytextField.htmlText = textLoader.data;  
    mytextField.multiline = true;  
    mytextField.embedFonts = true;  
    mytextField.wordWrap = true;  
    mytextField.width = 300;  
    mytextField.x = 130;  
    mytextField.y = 25;  
    mytextField.height = 300;  
    mytextField.border = true;  
    //mytextField.setTextFormat(myformat);  
    mytextField.antiAliasType = AntiAliasType.ADVANCED;  
}
```

Ustalimy teraz format dla tekstu. Bez niego nasz tekst byłby wyświetlany z domyślnymi ustawieniami.

```
var myformat:TextFormat = new TextFormat();  
myformat.color = 0x333333;  
myformat.size = 12;  
myformat.font = "Verdana";
```

Można usunąć już komentarz (//) z funkcji dataOK.

Dodamy jeszcze funkcję do przewijania tekstu:

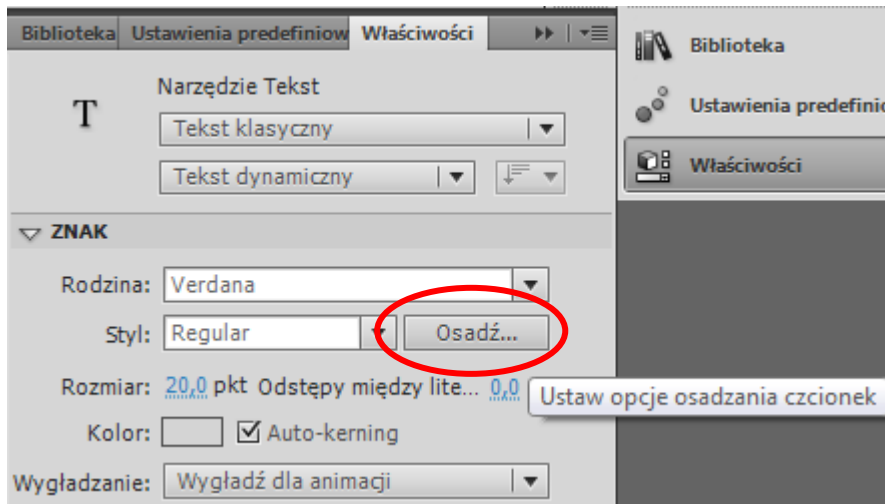
```
mytextField.addEventListener(MouseEvent.CLICK, mouseDownScroll);
```

```
function mouseDownScroll(event:MouseEvent):void{  
    mytextField.scrollV++;  
}
```

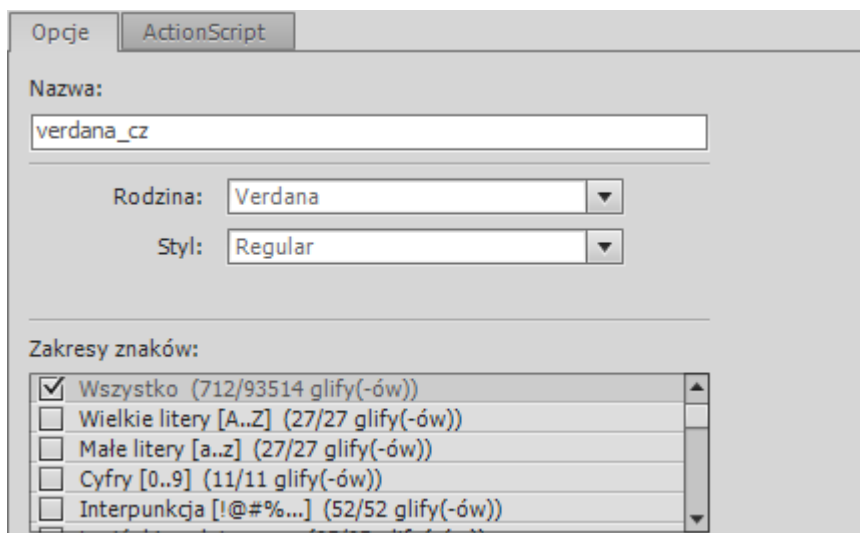
Oto cały kod do tego ćwiczenia:

```
1  import flash.net.URLLoader;  
2  import flash.net.URLRequest;  
3  import flash.text.TextField;  
4  import flash.text.TextFieldAutoSize;  
5  import flash.events.Event;  
6  import flash.net.URLVariables;  
7  
8  var textLoader:URLLoader = new URLLoader();  
9  var textURLRequest:URLRequest = new URLRequest("target.txt");  
10 textLoader.load(textURLRequest);  
11  
12 var mytextField:TextField=new TextField();  
13 addChild(mytextField);  
14  
15 textLoader.addEventListener(Event.COMPLETE, dataOK);  
16  
17 function dataOK(myevent:Event):void{  
18     mytextField.htmlText = textLoader.data;  
19     mytextField.multiline = true;  
20     mytextField.embedFonts = true;  
21     mytextField.wordWrap = true;  
22     mytextField.width = 300;  
23     mytextField.x = 130;  
24     mytextField.y = 25;  
25     mytextField.height = 300;  
26     mytextField.border = true;  
27     mytextField.setTextFormat(myformat);  
28     mytextField.antiAliasType = AntiAliasType.ADVANCED;  
29 }  
30  
31 mytextField.addEventListener(MouseEvent.CLICK, mouseDownScroll);  
32  
33 function mouseDownScroll(event:MouseEvent):void{  
34     mytextField.scrollV++;  
35 }  
36  
37 // ustalenie formatu tekstu  
38 var myformat:TextFormat = new TextFormat();  
39 myformat.color = 0x333333;  
40 myformat.size = 12;  
41 myformat.font = "Verdana";  
42
```

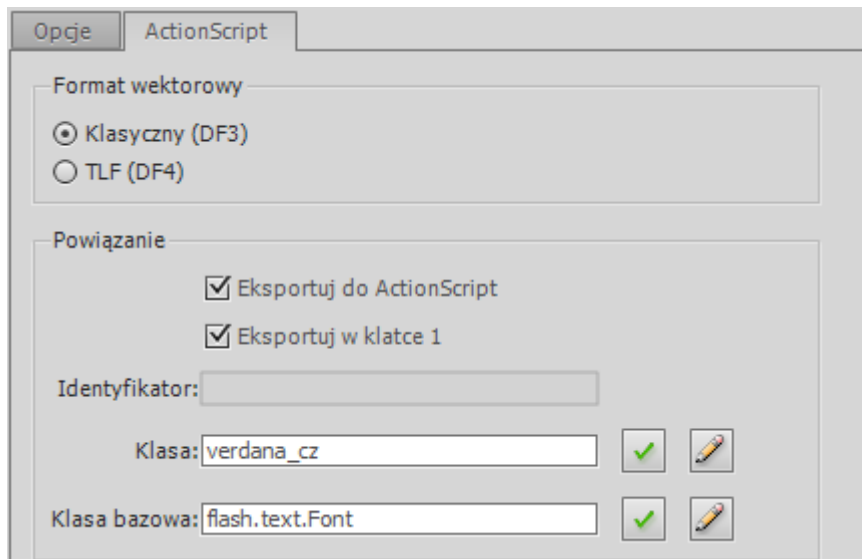
Pozostał jeszcze jeden ważny element, a mianowicie konieczność osadzenia czcionki w dokumencie. Jest to konieczne w przypadku każdego tekstu dynamicznego. Jeśli chcemy, aby tekst wyświetlany był czcionką Verdana wybieramy narzędzie **Tekst** i w panelu **Właściwości** zaznaczamy tę czcionkę. Wybieramy przycisk **Osadź**:



Na karcie **Opcje** wprowadź nazwę czcionki i wybierz glyfy do osadzenia:



Na karcie **ActionScript** zaznacz opcję **Eksportuj do ActionScript**. Okno **Klasa** i **Klasa bazowa** muszą być wypełnione według poniższego schematu:



Po zaakceptowaniu ustawień może pojawić się ostrzeżenie o konieczności utworzenia nowej klasy dla osadzonej czcionki.

PRZYKŁAD 2

Krok 1

Tworzenie pliku kontenera

Otwórz nowy plik i nie zmieniaj jego wyjściowych ustawień. Zapisz go na dysku lokalnym. W tej samej lokalizacji zapisz trzy pliki tekstowe: „obiekty.txt”, „target.txt” i „currenttarget.txt”.

Umieść w projekcie trzy przyciski (lokalizacja tych przycisków w projekcie pokazana jest na przykładowym obrazku na pierwszej stronie tego konspektu). Aby nie tworzyć przycisków od początku możesz wykorzystać te zrobione w ćwiczeniu nr 21. Otwórz tamten plik .fla i z jego głównej sceny skopiuj przyciski. Wklej je od razu na stół montażowy nowego projektu. W ten sposób zachowasz istniejące nazwy instancji (b1_btn, b2_btn, b3_btn).

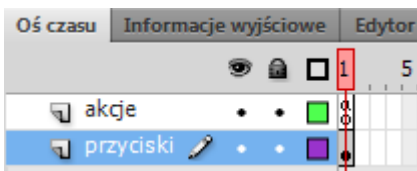
Nazwij warstwę z przyciskami „przyciski”.

Tak jak w przykładzie 1 z tego ćwiczenia osadź w projekcie czcionkę Verdana.

Krok 2

Listwa czasowa

Dodaj warstwę i nazwij ją „akcje”.



Krok 3

Pianie kodu

Skopiuj kod z poprzedniego przykładu w tym ćwiczeniu. Potrzebne będą tylko niewielkie poprawki, aby współdziałał z przyciskami.

Początek kodu będzie wyglądał następująco:

```
var textLoader:URLLoader = new URLLoader();
var mytextField:TextField = new TextField();
addChild(mytextField);
```

Jak widać tylko deklarujemy obiekt URLRequester bez określenia pliku do wczytania (pominięta jest na razie deklaracja obiektu URLRequest). Niezwołując tworzymy teraz pole tekstowe i dodajemy je do listy wyświetlania.

Obiekt URLRequest zostanie utworzony dopiero w parametrycznej funkcji addText. W tej samej funkcji nastąpi także załadowanie tekstu do obiektu URLRequester i nasłuchiwanie zdarzenia COMPLETE dla tego obiektu.

Dzięki temu, że funkcja addText jest parametryczna (parametrem przekazywanym do funkcji jest nazwa pliku tekstowego), będzie można decydować, który tekst zostanie wczytany. Funkcja addText będzie wywoływana w funkcji chooseBtn, która obsłuży zdarzenie CLICK myszy. Celem („odbiornikiem”) tej funkcji będą nasze przyciski.

Piszemy więc:

```
b1_btn.addEventListener(MouseEvent.CLICK, chooseBtn);
b2_btn.addEventListener(MouseEvent.CLICK, chooseBtn);
b3_btn.addEventListener(MouseEvent.CLICK, chooseBtn);
```

```
function chooseBtn(event:MouseEvent):void{
    if(event.target==b1_btn){
        addText("obiekty.txt");
    }
    else if(event.target==b2_btn){
        addText("target.txt");
    }
    else{
        addText("currenttarget.txt");
    }
}
```

```
function addText(plikText:String){
    var textURLRequest:URLRequest = new URLRequest(plikText);
    textLoader.load(textURLRequest);
    textLoader.addEventListener(Event.COMPLETE, dataOK);
}
```

Pozostałe elementy skryptu z poprzedniego ćwiczenia pozostają prawie bez zmian. W funkcji dataOK zmienione zostały właściwości x i y, po to aby pole tekstowe leżało obok przycisków – w razie konieczności wypróbuj tu własne ustawienia:

```
function dataOK(myevent:Event):void{
    mytextField.htmlText = textLoader.data;
    mytextField.multiline = true;
    mytextField.embedFonts = true;
    mytextField.wordWrap = true;
```

```
mytextField.width = 300;
mytextField.x = 200;
mytextField.y = 15;
mytextField.height = 300;
mytextField.border = true;
//mytextField.setTextFormat(myformat);
mytextField.antiAliasType = AntiAliasType.ADVANCED;
}
```

Tak jak poprzednio ustalamy format dla tekstu i dodajemy funkcję do przewijania tekstu:

```
var myformat:TextFormat = new TextFormat();
myformat.color = 0x333333;
myformat.size = 12;
myformat.font = "Verdana";

mytextField.addEventListener(MouseEvent.MOUSE_DOWN, mouseDownScroll);

function mouseDownScroll(event:MouseEvent):void{
    mytextField.scrollV++;
}
```

Oto cały kod do tego ćwiczenia:

```
1 import flash.net.URLLoader;
2 import flash.net.URLRequest;
3 import flash.text.TextField;
4 import flash.text.TextFieldAutoSize;
5 import flash.events.Event;
6 import flash.net.URLVariables;
7
8 var textLoader:URLLoader = new URLLoader();
9 var mytextField:TextField=new TextField();
10 addChild(mytextField);
11
12 b1_btn.addEventListener(MouseEvent.CLICK, chooseBtn);
13 b2_btn.addEventListener(MouseEvent.CLICK, chooseBtn);
14 b3_btn.addEventListener(MouseEvent.CLICK, chooseBtn);
15
16 function chooseBtn(event:MouseEvent):void{
17     if(event.target==b1_btn){
18         addText("obiekty.txt");
19     }
20     else if(event.target==b2_btn){
21         addText("target.txt");
22     }
23     else{
24         addText("currenttarget.txt");
25     }
26 }
```

```
27
28 function addText(plikText:String){
29     var textURLRequest:URLRequest = new URLRequest(plikText);
30     textLoader.load(textURLRequest);
31     textLoader.addEventListener(Event.COMPLETE, dataOK);
32 }
33
34 function dataOK(myevent:Event):void{
35     mytextField.htmlText = textLoader.data;
36     mytextField.multiline = true;
37     mytextField.embedFonts = true;
38     mytextField.wordWrap = true;
39     mytextField.width = 300;
40     mytextField.x = 200;
41     mytextField.y = 15;
42     mytextField.height = 300;
43     mytextField.border = true;
44     mytextField.setTextFormat(myformat);
45     mytextField.antiAliasType = AntiAliasType.ADVANCED;
46 }
47 //scrollowanie tekstu
48 mytextField.addEventListener(MouseEvent.CLICK, mouseDownScroll);
49
50 function mouseDownScroll(event:MouseEvent):void{
51     mytextField.scrollV++;
52 }
53 // określenie formatu tekstu
54 var myformat:TextFormat = new TextFormat();
55 myformat.color = 0x333333;
56 myformat.size = 12;
57 myformat.font = "Verdana";
58
```