

Iwona Iskierka, Sławomir Iskierka

Symulacja naturalnego ruchu w programie Adobe Flash

Dydaktyka Informatyki 5, 136-142

2010

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej bazhum.muzhp.pl, gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

Iwona Iskierka, Sławomir Iskierka

**SYMULACJA NATURALNEGO RUCHU
W PROGRAMIE ADOBE FLASH**

Wstęp

Program Flash jest programem przeznaczonym do tworzenia animacji dwuwymiarowych. Możliwości tego programu są olbrzymie. Aby animacje Flasha dawały wrażenie realizmu, często projektanci tworzący projekty w tym programie odwołują się do praw fizyki, naśladując naturalne zachowanie przedmiotów. Przykładami klasycznej mechaniki zaczerpniętymi z życia są: rzut piłką, spadający obiekt, hamujący samochód, zderzające się kule bilardowe czy drgająca sprężyna. W wielu animacjach symulowane są kolizje [Dolecki 2002]. Podstawą tworzenia animacji w programie Flash w oparciu o prawa fizyki jest znajomość zagadnień matematycznych związanych z układem współrzędnych, funkcjami trygonometrycznymi, rzutami geometrycznymi oraz wektorami.

1. Tworzenie i modyfikowanie obiektów w programie Flash

Flash na początku funkcjonował pod nazwą Future Splash Animator, jako niewielki program do tworzenia i animowania obrazów wektorowych. Była to aplikacja amerykańskiej firmy FutureWave. W roku 1997 firma Macromedia przejęła program Future Splash Animator i zmieniła jego nazwę na Macromedia Flash [Bhargal 2000]. Flash jest technologią tworzenia grafiki, animacji i interaktywnych filmów w formacie wektorowym. Zawiera narzędzia do tworzenia i modyfikacji obrazów wektorowych. Obrazy te układa się w sekwencje, dodaje dźwięk. Prezentacje i filmy tworzone w programie Flash mogą być wyświetlane w dowolnej przeglądarce internetowej. Flash, co jest ogromną zaletą tego programu, nie odtwarza tylko biernie animacji, lecz stwarza również możliwości interaktywnej współpracy z użytkownikiem [Świdarska 2002]. Program pozwala również zaawansowanym użytkownikom na wykorzystanie umiejętności programowania, bowiem wyposażony został w język programowania ActionScript. Użytkownik może tworzyć filmy i interaktywne animacje dla WWW. Nie musi wykazywać się bardzo dobrą znajomością skryptów CGI, składni języka HTML

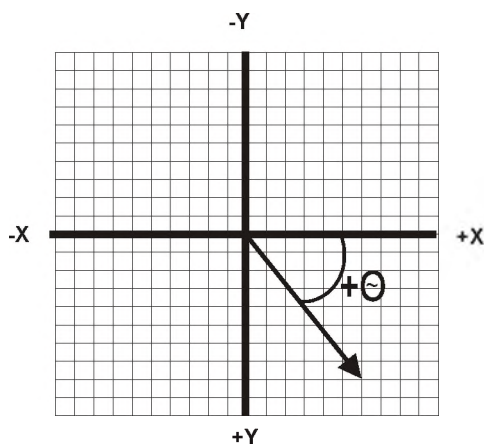
czy JAVA. Nawet bez umiejętności programowania można stworzyć interaktywną stronę WWW, która będzie spełniała wymagania użytkownika.

W programie Flash grafikę tworzy się, rysując kształty geometryczne w obszarze roboczym. Narzędzia służące do tworzenia i modyfikowania obiektów znajdują się w panelu narzędziowym [Ulrich 1999]. Początkującym użytkownikom Flash zapewnia odpowiednie narzędzia do tworzenia prostych grafik i animacji, natomiast użytkownicy zaawansowani mogą posłużyć się bardziej precyzyjnymi narzędziami. Panel Właściwości wyświetla wszystkie właściwości wyselekcjonowanego obiektu, które można określać bezpośrednio w tym panelu. Narzędzie przyciągania obiektów pomaga w precyzyjnym rysowaniu. W programie Flash wszystkie kształty można zamykać w grupach kształtów. Można również tworzyć symbole, które są używane w animacji. Większość grafiki jest konwertowana do symboli, dlatego że symbole są obiektami przygotowanymi do wielokrotnego zastosowania. Istnieje możliwość tworzenia statycznych symboli graficznych, symboli animowanych oraz symboli-przycisków [Bargiel 2002]. Symbole takie są tworzone albo z obiektów już utworzonych, albo są rysowane w trybie edycji symbolu. Po przekonwertowaniu dowolnego obiektu do symbolu, w obszarze roboczym umieszczany jest odnośnik tego symbolu. Utworzony symbol znajduje się w bibliotece dokumentu, w którym został stworzony. Jeśli symbol zostanie skopiowany z dokumentu do innego dokumentu, symbol ten będzie występował oddzielnie w bibliotece każdego z tych dokumentów. Biblioteka każdego dokumentu Flasha zawiera wszystkie symbole zastosowane w tym dokumencie. Może również zawierać symbole, które nie są w nim wykorzystywane. Umieszczanie odnośnika do symbolu w obszarze roboczym jest bardzo proste. Należy mianowicie odnośnik do symbolu przeciągnąć z panelu biblioteki do obszaru roboczego. Tryb edycji symbolu pozwala zmodyfikować grafikę, która znajduje się w symbolu. Symbole i odnośniki stanowią podstawę pracy w programie Flash. Ponieważ można utworzyć wiele odnośników do jednego symbolu, oszczędza się wielkość animacji oraz czas potrzebny na tworzenie i modyfikację projektu. Flash jest programem służącym do tworzenia interaktywnych animacji. Oferuje w związku z tym możliwość tworzenia filmów metodą tradycyjną, czyli klatka po klatce, co oznacza umieszczenie różnych obrazów w różnych ujęciach. Ujęcia, w których zawarty jest nowy materiał, nazywa się ujęciami kluczowymi. Jednak animacje typu klatka po klatce wymagają dużego nakładu pracy oraz tworzą duże pliki. Dlatego Flash udostępnia użytkownikowi jeszcze jeden rodzaj animacji, zwanej animacją uzupełnianą. Animacja ta polega na tym, że użytkownik w ujęciach kluczowych wprowadza zmiany istotne dla animacji, natomiast program przelicza obraz w klatkach tzw. pośrednich, czyli znajdujących się między klatkami kluczowymi [Ulrich 1999]. Flash udostępnia dwa rodzaje automatycznej animacji: automatyczną animację ruchu (*Motion Tweening*) oraz automatyczną animację kształtu (*Shape Tweening*).

2. Matematyczny opis zagadnień fizycznych w programie Flash

W programie Flash funkcjonuje układ współrzędnych, w którym oś y jest odwrócona – ujemna półoś znajduje się na górze, a dodatnia na dole. Kąt obrotu w układzie współrzędnych Flasha jest mierzony od dodatniej półosi x. Wartość tego kąta jest dodatnia, gdy obrót zachodzi zgodnie z kierunkiem wskazówek zegara. W programie Flash można oczywiście posługiwać się kątami obrotu od 0° do 360° . Gdy użytkownik nada właściwości `_rotation` wartość 270° , Flash potraktuje ją tak samo jak wartość kąta -90° [Dolecki 2002].

Funkcje trygonometryczne są wykorzystywane w programie Flash do rozwiązywania różnorodnych problemów geometrycznych. Na przykład za pomocą funkcji trygonometrycznych użytkownik może obliczyć odległość między dwoma dowolnymi punktami, obliczyć amplitudę fali lub też wyznaczyć nową trajektorię obiektu w chwili kolizji.



Rys 1. Miara kąta w programie Flash [źródło: Dolecki 2002]

Bardzo często korzysta się z metod obiektu *Math*: *Math.sin()*, *Math.cos()*, *Math.sqrt()*. Funkcje trygonometryczne Flasha posługują się kątami wyrażanymi w radianach, zaś właściwość `_rotation` posługuje się kątami wyrażanymi w stopniach. Wektory są wykorzystywane do przedstawienia oddziaływania na obiekty różnymi siłami. Operacje na wektorach są wykonywane na przykład w celu znalezienia wypadkowej siły działającej na obiekt. W filmie Flasha można zatem zaimplementować grawitację, wiatr i tarcie i po dodaniu do siebie odpowiednich wektorów wyznaczyć trajektorię ruchu obiektu.

Dużo interesujących efektów realizowanych w programie Flash z uwzględnieniem praw fizyki wiąże się z zastosowaniem przyspieszeń. Przyspieszenie może ulegać zmianie w kolejnych klatkach. Mówi się wtedy o przyspieszeniu drugiego stopnia.

Równania ruchu w odniesieniu do czasu i do liczby klatek.

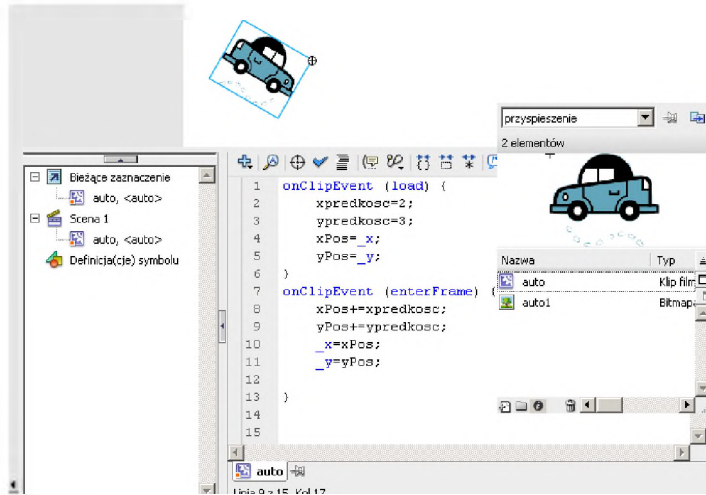
w fizyce :

$$x = x_{pocz} + xpredkosc * czas$$

we Flashu :

$$x = x_{pocz} + xpredkosc * klatki$$

We Flashu oblicza się nowe położenie w każdej klatce. Na rysunku 2 umieszczono przykład skryptu dotyczącego klipu poruszającego się ze stałą prędkością.



Rys 2. Przykład klipu poruszającego się ze stałą prędkością

Poniżej umieszczono zmodyfikowany skrypt symulujący przyspieszenie, ze zdefiniowaną zmienną *przys* określającą przyspieszenie.

```
onClipEvent (load) {  
    xpredkosc=2;  
    ypredkosc=3;  
    przys=1;  
    xPos=_x;  
    yPos=_y;  
}
```

```

onClipEvent (enterFrame) {
    xpredkosc+=przys;
    ypredkosc+=przys;
    xPos+=xpredkosc;
    yPos+=ypredkosc;
    _x=xPos;
    _y=yPos;
}

```

Dobierając odpowiednią wartość zmiennej *przys* można symulować działanie oporu powietrza oraz przyspieszenie grawitacyjne. Tarcie można symulować na dwa sposoby: zgodnie z prawami fizyki oraz w sposób uproszczony (w każdej klatce mnoży się prędkość przez współczynnik mniejszy od jedności). Interesujące efekty daje także symulacja ruchu ciężarka zawieszonoego na sprężynie.

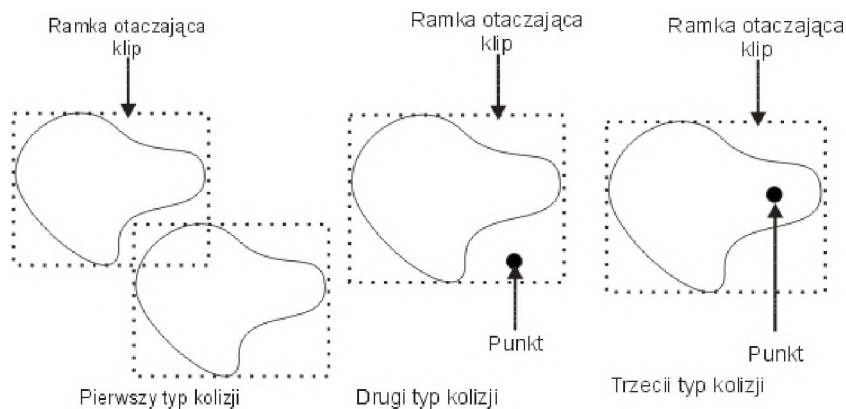
3. Implementacja realistycznych zachowań w języku ActionScript

Język ActionScript 2.0 jest zorientowanym obiektowo, skryptowym językiem pozwalającym na kontrolę wszystkich parametrów aplikacji Flash. Daje możliwość kontroli przebiegu odtwarzania animacji i reagowania na zdarzenia, np. generowane przez użytkownika. Do dyspozycji użytkownik ma szereg narzędzi i wbudowanych klas. Gdy zachodzi potrzeba użycia klasy, tworzy się jej instancję (klon) nazwaną obiektem. Obiekt zawiera wszystkie instrukcje opisane w klasie. Obiekty posiadają metody, właściwości i zdarzenia. ActionScript zawiera wiele klas wbudowanych – złożonych typów danych, które mogą być wykorzystywane do określonych celów. Przykładami takich klas mogą być: *Math*, *Button* lub *Date*. Często używaną klasą jest *MovieClip* oraz jej metody i właściwości, np. *stop* i *gotoAndPlay*. Klasa *MovieClip* zawiera także właściwości, które można dla klonu zmieniać lub definiować – np.: *_visible*, *_width*, *_height*. Obiekty bazujące na klasach *MovieClip*, *Button*, *TextField* i *Component* są we Flashu reprezentowane w sposób wizualny. Instancje wielu innych klas, na przykład *LoadVars* i *Sound*, tworzy się wyłącznie za pomocą kodu ActionScript. Gdy tworzy się nową instancję klasy, wywołuje się funkcję konstruktora: specjalną funkcję, która najczęściej odpowiada za inicjalizację obiektu [English 2006].

Bardzo często w aplikacjach programu Flash tworzy się funkcje, które są blokami kodu wykorzystywanymi wielokrotnie w programie. Zwracają one określoną wartość (po wykonaniu określonych zadań) i są do nich przypisane parametry. Parametry nazywane czasem argumentami, umożliwiają wysłanie do funkcji wartości stałej lub zmiennej, która może być następnie używana lub modyfikowana w kodzie funkcji. Parametry pozwalają modyfikować zachowanie

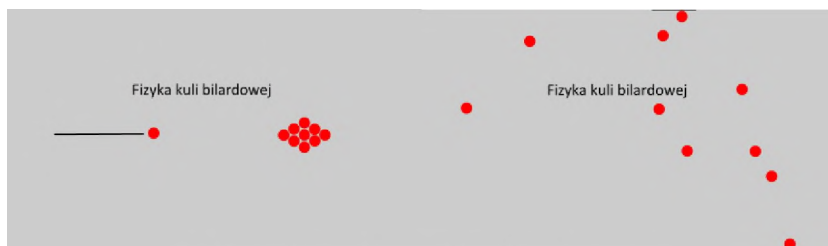
lub efekt działania funkcji. Język ActionScript zawiera również instrukcje warunkowe i trzy typy pętli, z których wszystkie wykonują akcję (lub ich zbiór) wówczas, gdy dany warunek jest prawdziwy.

We Flashu tworzy się również aplikacje, w których obiekty oddziałują ze sobą. W sytuacji, gdy poruszający obiekt wykrywa obecność innych obiektów, może reagować na nie w określony sposób. Przyczyną takich reakcji mogą być różne zdarzenia, najczęstsze to kolizja i zbliżenie. Istnieje wiele odmian kolizji, zależnych od kształtów kolidujących przedmiotów.



Rys 3. Typy kolizji w programie Flash [źródło: Dolecki 2002]

Wykorzystuje się dwie popularne metody detekcji kolizji: za pomocą metody *MovieClip.HitTest()* oraz za pomocą obliczeń matematycznych. W aplikacjach wykorzystuje się możliwości symulacji kolizji obiektu ze ścianą, kolizji prostokątów, kolizji kół. Aby uzyskać realistyczne efekty, należy również spowodować, aby kolidujące obiekty odpowiednio reagowały na tę kolizję. W programie Flash istnieją dwa typy reakcji na kolizję: reakcje niefizyczne oraz reakcje fizyczne. Reakcje fizyczne – bazują na równaniach fizycznych lub w jakiś sposób naśladują naturalne zachowanie. Bardzo często rozważa się reakcje fizyczne bazujące na zasadzie zachowania pędu i zasadzie zachowania energii kinetycznej. Efekty można przeanalizować w mechanizmie odbić kul bilardowych.



Rys 4. Mechanizm odbić kul bilardowych

```
74 // -----równania zasad zachowania pędu i energii kinetycznej---oblicz nowe p
75 P=(mass[n]*xVel1prime+mass[m]*xVel2prime);
76 V=(xVel1prime-xVel2prime);
77 var v1f=(P-mass[m]*V)/(mass[n]+mass[m]);
78 var v2f=(V+v1f);
79 var xVel1prime=v1f;
80 var xVel2prime=v2f;
81 // -----przenieś nowe prędkości z powrotem do układu współrzędnych Flasha
82 var xVel1=xVel1prime*cosTheta-yVel1prime*sinTheta;
83 var xVel2=xVel2prime*cosTheta-yVel2prime*sinTheta;
84 var yVel1=yVel1prime*cosTheta+xVel1prime*sinTheta;
85 var yVel2=yVel2prime*cosTheta+xVel2prime*sinTheta;
86 //-----gdy znamy nowe prędkości, przepisujemy je z powrotem do tablicy prędkości
87 xVelocity[n]=xVel1;
88 xVelocity[m]=xVel2;
89 yVelocity[n]=yVel1;
90 yVelocity[m]=yVel2;
```

Rys 5. Fragmenty skryptów dotyczących fizyki kuli bilardowej

W programie Flash poprawna implementacja praw fizyki pozwala użytkownikowi uzyskiwać fizycznie realistyczne rezultaty. Przedstawione przykłady wskazują na duże możliwości ActionScript umożliwiające implementację wielu zjawisk fizycznych w tym języku. Za pomocą skryptów ActionScript można implementować prawa fizyki, tak by odtwarzać realistyczny ruch z uwzględnieniem grawitacji, lepkości, tarcia i oporów powietrza oraz kolizje i reakcje obiektów na kolizje.

Bibliografia

- Bargiel D. (2002), *Flash MX. Ćwiczenia praktyczne*, Gliwice.
Beard D. (2002), *Flash5 ActionScript Techniki zaawansowane*, Gliwice.
Bhargal S., Farr A., Rey P. (2000), *Flash 5. Podstawy*, Gliwice.
Dolecki E. (2002), *Flash Super Samuraj*, Gliwice.
English J. (2006), *Macromedia Flash8. Oficjalny podręcznik*, Gliwice.
Świdarska M. (2002), *Animacje i WWW. Ćwiczenia z Macromedia Flash 5*, Warszawa.
Ulrich K. (1999), *Flash 4*, Wydawnictwo Helion, Gliwice.