

Piotr Kisiel

Aplikacje interakcyjne Technologii Flash w Action Script 2.0 jako wprowadzenie do programowania obiektowego i strukturalnego

Edukacja - Technika - Informatyka 2/2, 209-214

2011

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.

Piotr KISIEL

Uniwersytet Rzeszowski, Polska

Aplikacje interaktywne Technologii Flash w Action Script 2.0 jako wprowadzenie do programowania obiektowego i strukturalnego

Analizując podstawę programową przedmiotów kształcenia informatyki oraz technologii informacyjnej (i komunikacyjnej) szkół gimnazjalnych i ponadgimnazjalnych, wyraźnie da się zauważyć rozbieżne obszary zainteresowań tych dwu przedmiotów.

Technologia informacyjna kładzie nacisk na aspekty pojmowania, wyszukiwania i posługiwania się informacją zapisaną w formie tekstowej, wizualnej czy video, obecnej dzięki procesowi konwergencji mediów w szeroko pojmowanej warstwie telematycznej.

Przedmiot informatyka w głównej mierze skupia się na zagadnieniach związanych z algorytmiką i szeroko rozumianym programowaniem. Przy takim podejściu nauczyciel przedmiotu informatyka staje przed wyzwaniem rozwijania u swoich uczniów myślenia algorytmicznego, a nawet komputacyjnego¹.

Takie ujęcie problemu determinuje zakres omawianych treści i chronologię ich realizacji w szkołach.

Śledząc proces nauczania przedmiotów informatycznych, począwszy od stopnia podstawowego, uczeń zapoznaje się z aspektami korzystania z komputera, urządzeń peryferyjnych, uruchamiania systemu operacyjnego, bezpieczeństwa i podstaw komunikowania.

Poziom gimnazjum obejmuje dodawanie i usuwanie programów, dodawanie urządzeń, korzystanie z sieci lokalnej i globalnej, zaznajamia ucznia z aspektami korzystania z systemu operacyjnego Windows, Mac OS, Linux. Wprowadza w tajniki tworzenia prostej grafiki, tabel, wyszukiwania i zapisywania informacji, zapoznaje z podstawami arkusza kalkulacyjnego, odnosi się do społecznych skutków nowoczesnej technologii i problemów własności intelektualnej.

Na poziomie podstawowym szkoły średniej wiedza wzbogacana jest o rozbudowane dokumenty, kalkulacje kosztów, porządkowanie baz danych, korespondencję elektroniczną, korzystanie z forów i komunikatorów, tworzenie prezentacji, na poziomie poszerzonym omawiane są problemy prawne i społeczne informatyzacji, pozytywne skutki informatyki, nowe formy zatrudnienia [Krawczyński, Talaga, Wilk 2007].

¹ ang. *computational thinking*

Reasumując, jest to nauka stosowana, głównie związana z aspektami, z którymi uczeń ma styczność w swoim otoczeniu, zarówno domowym, jak i szkolnym, w postaci infrastruktury sieci szkolnej czy prezentacji multimedialnych wykorzystywanych jako narzędzie edukacyjne na lekcjach innych przedmiotów niż technologia informacyjna [Piecuch 2008: 90].

Znacznie poważniejsze wyzwania stoją przed nauczycielami realizującymi zajęcia na rozszerzonym poziomie szkoły średniej, gdzie na lekcjach informatyki omawiane być powinny tematy obejmujące podstawy programowania w języku wysokiego poziomu, budowę i projektowanie relacyjnych baz danych czy przetwarzanie dźwięku i obrazu oraz publikowanie materiałów w sieci.

Przedmiot ten dedykowany jest dla wszystkich uczniów, a ukierunkowany w sposób naturalny profilem obecnie zdawanej matury z przedmiotu informatyka², predestynującym w głównej mierze przyszłych programistów.

W praktyce rodzi to wiele problemów, albowiem nietrudno się domyślić, iż z osób interesujących się poszerzoną informatyką sporo osób stanie się w przyszłości nie tylko programistami, ale również webdesignerami, administratorami systemów, testerami złożonych aplikacji komputerowych czy choćby grafikami komputerowymi.

Przytoczony powyżej zakres tematyczno-chronologiczny toku nauczania na poziomie gimnazjalnym i ponadgimnazjalnym, obszarów związanych z informatyką jasno wskazuje, iż z dydaktycznego punktu widzenia potrzebne jest płynne przejście pomiędzy informatyką stosowaną (technologią informacyjną) a programowaniem realizowanym na zajęciach z przedmiotu informatyka (rozumianą jako nauka podstawowa).

W ślad za jedną z fundamentalnych zasad pedagogiki, przechodzenia od zagadnień prostych do coraz bardziej złożonych, także narzędzia, którymi w trakcie nauki posługuje się młody człowiek, powinny być w miarę opanowania stosownych umiejętności wymieniane, poczynając od uproszczonych do bardziej skomplikowanych i wydajnych. Tymczasem mamy do czynienia z gwałtownym przejściem od zagadnień związanych z arkuszami biurowymi, grafiką komputerową czy projektowaniem stron internetowych wprost w świat abstrakcyjnego myślenia związanego z programowaniem [Sadowski 1996] i relacyjnymi bazami danych [Bujnowski, Talaga 2003b: 47–85].

Uczniowie uprzednio prowadzeni powyższym tokiem nauczania w zetknięciu z zadaniami z algorytmiki oraz podstawami programowania strukturalnego i obiektowego [Bujnowski, Talaga 2003a; Walnum 1994] tracą niejednokrotnie zainteresowanie przedmiotem, pozostając w biernej opozycji dla prezentowanych treści.

² Analiza własna zadań maturalnych z lat 2008–2011, źródło: www.cke.edu.pl

2 lub 3 zadania programowania i algorytmiki

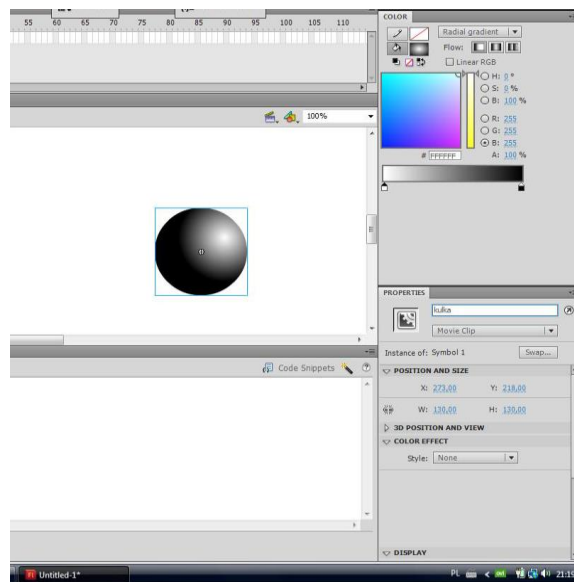
1 zadanie z relacyjnych baz danych

1 zadanie z arkusza kalkulacyjnego

1 zdanie wiedza ogólna: systemy liczbowe, sieci itp.

By uniknąć takiego stanu rzeczy, warto zastanowić się nad celowością wprowadzenia w tok nauczania aplikacji wiążącej w sobie elementy zarówno grafiki, jak i animacji (łatwo przyswajalnej i akceptowalnej wśród młodzieży) z elementami prostego, aczkolwiek posiadającego ogromne możliwości języka ActionScript zorientowanego obiektowo [Mapes 2002: 302].

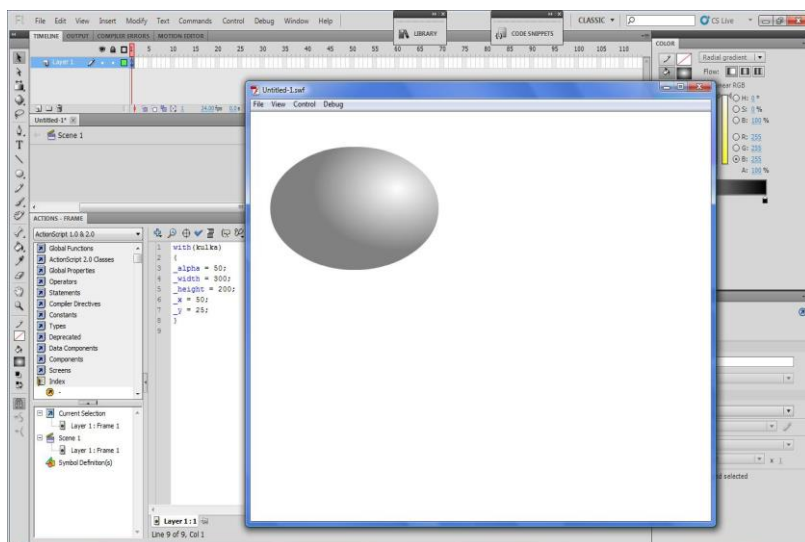
Przykładowo, rysując figurę w programie Flash z łatwością, tj. poprzez właściwości (znajdujące się na pasku properties) [Iskierka 2008: 162] możemy zmienić jego wygląd, a logicznie przekonwertować go na obiekt MovieClip, nadając mu atrybut nazwy, powiedzmy *kulka*.



Tak zaimplementowany obiekt poprzez język skryptowy ActionScript możemy modyfikować, zmieniając położenie na ekranie, wielkość czy chociażby stopień przezroczystości poprzez następujący kod:

```
with(kulka)
{
    _alpha = 50; //stopień przezroczystości obiektu
    _width = 300; //szerokość obiektu w pikselach
    _height = 200; //wysokość obiektu w pikselach
    _x = 50; //współrzędne położenia obiektu względem osi x
    _y = 25; //współrzędne położenia obiektu względem osi y
}
```

Uruchomienie powyższego kodu skutkować będzie następującą zmianą wyglądu i położenia obiektu *kulka*.



Jeśli natomiast obiektowi *kulka* nadamy funkcjonalność przycisku, a następnie opiszemy następującym kodem:

```
on (rollOver) {
    with(kulka)
    {
        _alpha = 50;
        _width = 300;
        _height = 200;
        _x = 50;
        _y = 25;
    }
}
```

otrzymamy w pełni efektywną aplikację interaktywną.

Jak widać język ActionScript posiada w pełni cechy nowoczesnego języka zorientowanego obiektowo [Underdahl 2002: 37–39], sama zaś aplikacja Flash nie odrywa ucznia od namacalnego wręcz postrzegania obiektów, jako elementów graficznych. Jest to szczególnie ważne w początkowym okresie nauki programowania i pozwala stopniowo wprowadzać pojęcia związane z kodem źródłowym, pojęciem zmiennych czy też rekurencji i iteracji.

Literatura

- Bujnowski I., Talaga Z. (2003a), *Informatyka. Podręcznik*, t. 1, Warszawa.
- Bujnowski I., Talaga Z. (2003b), *Informatyka. Podręcznik*, t. 2, Warszawa.
- Iskierka I. (2008), *Aktywne prezentacje multimedialnew e-edukacji [w:] Multimedia w teorii i praktyce szkolnej*, red. W. Furmanek, A. Piecuch, Rzeszów.

- Krawczyński E., Talaga Z., Wilk M. (2007), *Technologia Informacyjna nie tylko dla uczniów Podręcznik*, Warszawa.
- Mapes R. (2002), *Flash Techniki zaawansowane*, Gliwice.
- Piecuch A. (2008), *System multimedialny w teorii i praktyce szkolnej* [w:] *Multimedia w teorii i praktyce szkolnej*, red. W. Furmanek, A. Piecuch, Rzeszów.
- Sadowski T.M. (1996), *Praktyczny kurs Turbo Pascala*, Gliwice.
- Underdahl B. (2002), *Flash MX Programowanie w języku ActionScript*, Gliwice.
- Walnum C. (1994), *Pierwsze kroki w Turbo C++*, Warszawa.

Streszczenie

Niniejsze opracowanie jest próbą zwrócenia uwagi na trudności występujące w procesie nauczania informatyki. Młodzież w newralgicznym momencie, kiedy potrzebne jest porzucenie prostych aplikacji i rozpatrywanie problemów na poziomie wymagającym abstrakcyjnego, twórczego myślenia, traci zainteresowanie przedmiotem i łatwo się zniechęca. Problem taki jest zauważalny, zwłaszcza na lekcjach informatyki w szkołach średnich, kiedy rozpoczynamy zajęcia z podstaw algorytmiki i programowania. Wprowadzenie aplikacji technologii Flash w procesie nauczania ułatwiłoby płynne przejście od grafiki i aplikacji webowych poprzez język ActionScript, posiadający ogromne możliwości, do programowania w językach C++ czy Turbo Pascal. Możliwość prostego sterowania obiektami w języku ActionScript, połączona z efektywnymi możliwościami ich wizualizacji, daje szansę inspirowania wiedzą podczas omawiania podstawowych algorytmów.

Finalnie przyszli programiści stając przed rzeczywistymi wymaganiami klientów, możliwościami takiej czy innej technologii, a w końcu kodem źródłowym aplikacji, dysponować będą elastycznym podejściem w rozwiązywaniu problemów i ich wizualizacji, co ważne jest zwłaszcza w pracy zespołowej.

Słowa kluczowe: nauczanie informatyki, myślenie twórcze, aplikacje interakcyjne.

Interactive applications of Flash Technology in Action Script 2.0 as an introduction to object and structural designing

Abstract

This case study attempts to highlight the difficulties experienced in teaching information technology. In the key moment, when it is necessary to employ abstract and creative thinking, pupils, who are used to using simple applications, become disinterested in the subject. This problem is particularly prominent in the secondary schools when the concepts of programming and algorithm design are first introduced. Introducing Adobe Flash applications in the schools' curric-

ulum would ease transition from graphical and web applications through ActionScript, and ultimately lead to programming in C++, or Turbo Pascal. The ActionScript's capability to manipulate objects, together with the ease of their visualisation, gives the ability to enthuse and inspire whilst demonstrating basic algorithms. Ultimately, future programmers, facing customer requirements, new technologies and application source code, will have at their disposal a flexible approach to problem visualisation and its potential solutions, both of these abilities are extremely important particularly whilst working within a team.

Key words: education computer science, creative thinking, interactive applications.