
Zastosowanie AR i nowych form interakcji w środowisku Adobe Flex Air 2

Marcin Wichrowski

Polsko-Japońska
Wyższa Szkoła
Technik Komputerowych
Warszawa
mati@pjwstk.edu.pl
www.wichrowski.pl

Nakul Mishra

Polsko-Japońska
Wyższa Szkoła
Technik Komputerowych
Warszawa
nakul.mishra@pjwstk.edu.pl

Abstrakt

Poniższy artykuł jest rozwinięciem artykułu „Nowe metody interakcji w Adobe Flash” z ubiegłorocznej edycji Kansei 2009. Prezentuje trzy prototypy aplikacji stworzonych w środowisku Adobe Flex łączących role elearningu i rozrywki.

Powszechna dostępność odtwarzacza Adobe Flash Player i Adobe Air Player oraz nowo powstające biblioteki Actionscript 3 pozwalają na dostarczanie coraz bogatszych form interakcji bezpośrednio przez przeglądarkę internetową niezależnie od platformy systemowej użytkownika lub w postaci aplikacji typu desktop. Programiści pracujący w środowiskach Flash i Flex mają do dyspozycji rosnącą liczbę rozwiązań umożliwiających współpracę z zewnętrznymi kontrolerami (np. do konsoli Nintendo Wii), kamerami internetowymi oraz zastosowanie różnych form interakcji/prezentacji np. w postaci rozpoznawania gestów czy rzeczywistości wzbogaconej (augmented reality). W prezentowanych przykładach użyto bibliotek FLARManager, WiiFlash, Flexmdi, Papervision3D oraz Flex4SDK z Adobe Air 2

(beta). Nadrzędnym celem autorów jest zaprezentowanie możliwości bezpłatnych bibliotek i korzyści wynikających z połączenia różnych form interakcji.

Słowa kluczowe

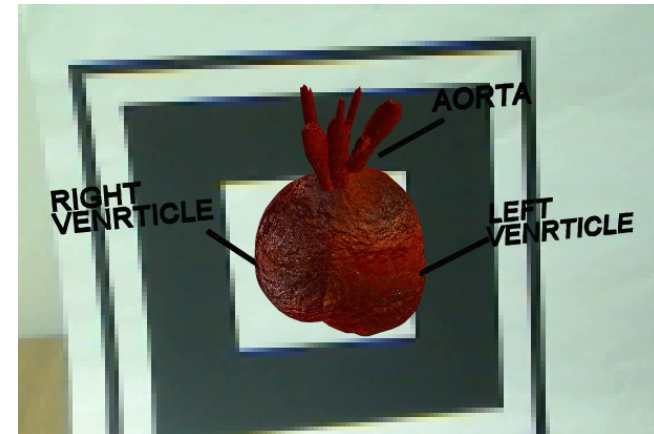
Augmented reality, rzeczywistość wzbogacoma, GUI, interfejs użytkownika, elearning, gry, rozpoznawanie gestów, Flash, Flex, Nintendo Wii, WiiFlash, FLARToolKit, FLARManager, Flexmdi, Papervision3D

Projekt „Body parts”

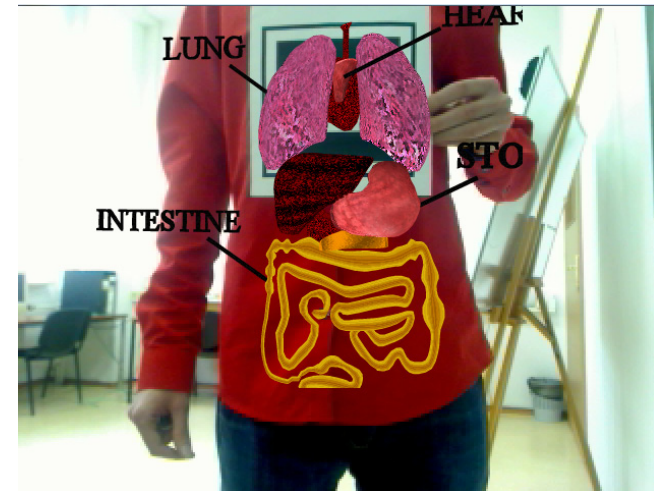
Projekt w przystępny sposób prezentuje wybrane części ciała/narządy człowieka (serce, narządy wewnętrzne, kości przedramienia i ramienia) łącznie z ich opisem. Dzięki wykorzystaniu rzeczywistości wzbogaconej możliwe stało się umieszczenie na markerach modeli 3D odpowiadających danym organom. Markery można umieścić bezpośrednio w dedykowanych im miejscach np. na ramieniu dla lepszego zobrazowania jak wygląda jego szkielet. Dodatkowo zaimplementowano rozpoznawanie gestów pozwalających powiększać/zmniejszać, skalować oraz obracać obiekty w przestrzeni 3D.

Planowany rozwój projektu

Dzięki obsłudze wielu markerów możliwe staje się rozbudowywanie projektu o nowe obiekty 3D. Wzbogacenie interfejsu o przyciski znajdujące się na markerze pozwoli na dodatkową interakcję. Prezentację można rozbudować o animację danego organu i dotyczące go filmy. Możliwe również jest wykorzystanie „naturalnych” markerów, które mogą być np. zdjęciem danego narządu (Natural Feature Tracking).



Rysunek 1. Model serca na markerze



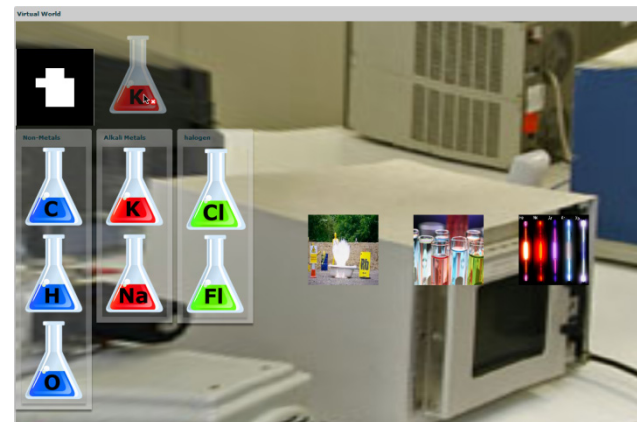
Rysunek 2. Model organów wewnętrznych na postaci

Projekt „Chemical Lab”

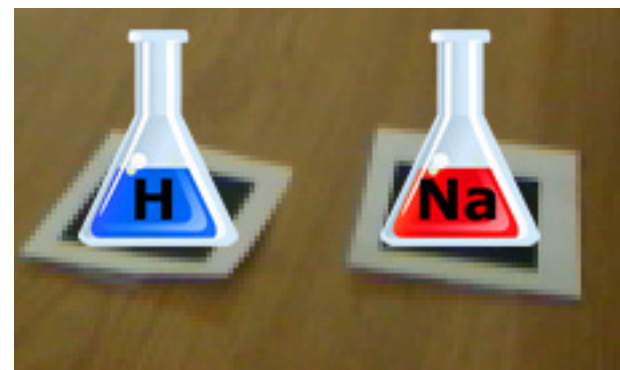
Kolejny projekt jest również prototypem aplikacji elearningowej, której zadaniem jest przekazanie wiedzy o budowie podstawowych związków chemicznych. Głównym założeniem projektu było otrzymanie modelu wirtualnego laboratorium chemicznego. Użytkownik wybierając poszczególne pierwiastki z okna zasobnika i przesuwać je na ikonę markera powoduje pojawienie się ich na faktycznym markerze znajdującym się na np. na biurku. Gdy budowa związku dobiegnie końca można sprawdzić prawidłowość swoich działań – osobny marker wizualizuje to w postaci ikon.

Planowany rozwój projektu

Wykorzystanie okularów z wbudowaną kamerą i ekranami oraz kontrolerów Wiimote wzbogaciłoby wrażenie przebywania wewnątrz laboratorium i wchodzenia w interakcję z otaczającym światem. Dodanie funkcjonalności wykrywania odległości między markerami pozwoliłoby na symulowanie wchodzenia w reakcję poszczególnych związków chemicznych



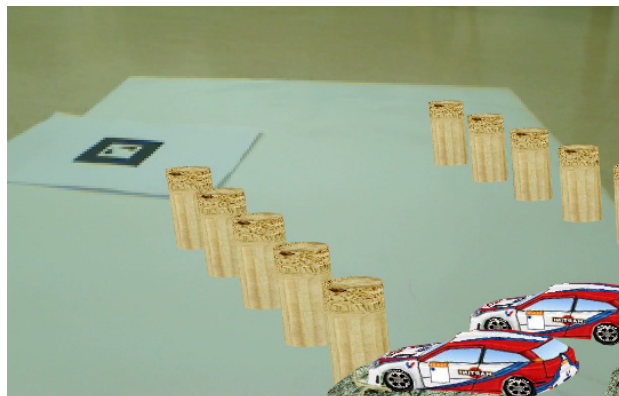
Rysunek 3. Ekran „Laboratorium” z zasobnikami pierwiastków i ikoną markera, która przenosi je na rzeczywisty marker



Rysunek 4. Pierwiastki na markerach

Projekt „AR Car Game”

Kolejny projekt pełni rolę rozrywkową – zaprojektowana gra pozwala graczom poruszać pojazdy ulokowane na markerach za pomocą kontrolerów Wiimote. Sterowanie pojazdem odbywa się przez przechylenie kontrolera i używanie wbudowanych przycisków. Jeden z graczy wciela się w rolę „uciekiniera”, natomiast zadaniem drugiego jest pościg. Gracz „złapany” przegrywa.



Rysunek 5. Modele pojazdów i przeszkód na planszy do gry (wykorzystano modele dostarczane z WiiFlash)

Dodatkowym utrudnieniem są przeszkody w postaci obiektów (cylindry lub kamienie), które w wypadku uderzenia przez pojazd powodują utratę jednego punktu (początkowo gracz otrzymuje 20 punktów). Utrata wszystkich punktów jest równoznaczna z przegraną. W projekcie zawarto również markery odpowiedzialne za reprezentację drzew i podłoża.

Planowany rozwój projektu

Dołączenie edytora, którego zadaniem byłoby dodawanie markerów i przypisanie im nowych obiektów oraz znaczeń pozwoli na rozbudowanie zasad gry. Użycie piko projektora pozwoliłoby na wyświetlanie różnych markerów w różnej konfiguracji i zmieniałoby dynamicznie pole gry. W celu ukrycia markerów na polu gry istnieje możliwość dynamicznego nałożenia na niego filtrów np. efektu blur.

Realizacja projektów

W projektach wykorzystano framework FLARManager (v0.7) [1] do rozpoznawania markerów i osadzania na nich obiektów 3D. Bazuje on na bibliotekach FLARToolkit i jest kompatybilny z wieloma silnikami 3D dla środowiska Flash - Alternativa3D, Away3D, Papervision3D, and Sandy3D. W przypadku opisywanych aplikacji posłużono się Papervision3D (v. 2.0. 883) [2]. Modele do projektu „Body Parts” wykonano w programie Maya (autor Radosław Kosierkiewicz). Obiekty low-polygon zostały przekonwertowane do formatu collada (*.dae) dzięki dodatkowi COLLADA for Maya. Tworząc modele z myślą o ich zastosowaniu w środowiskach 3D dla aplikacji Flash należy pamiętać ich o optymalizacji – zbyt skomplikowane struktury i duża objętość tekstur zdecydowanie osłabia szybkość działania aplikacji.

Osadzenie modelu umożliwia kod:

```
private var heartModel:DAE;
```

```
heartModel= new DAE(true, "model", true);
```

```
heartModel.load("heart.dae");
```

```
heartModel.addFileSearchPath("assets/tex")
```

Jeśli projekt „Body parts” zostanie uruchomiony na urządzeniu wspierającym funkcję multitouch (np. HP TouchSmart lub MacPro) możliwe staje się sterowanie aplikacją za pomocą gestów. W tym celu wykorzystano biblioteki wbudowane w Flex4SDK z Adobe Air 2 (wersja beta) do rozpoznawania gestów [3]. Użyto rozpoznawanie gestów powiększania/oddalania, skalowania i rotacji.

Przykładowo implementacja rozpoznawania gestu powiększania wygląda następująco:

```
this.mainStage.addEventListener(TransformGestureEvent.GESTURE_ZOOM, onZoom);
private function
onZoom(e:TransformGestureEvent):void{

    if(inner_modelContainer.visible){

        innerModel.scaleX*=e.scaleX;

        innerModel.scaleY*=e.scaleY;

    }
}
```

W podobny sposób oprogramowano pozostałe gesty. W przypadku urządzeń pozbawionych tej technologii obsługa aplikacji jest możliwa z użyciem skrótów klawiaturowych.

W projekcie „AR Car Game” wykorzystano dodatkowo jeszcze bibliotekę WiiFlash, której autorami są Joa Ebert i Thibault Imbert. Pozwala ona na komunikację z kontrolerami do konsoli Nintendo Wii.

Części składowe WiiFlash to:

- WiiFlash Server (serwer odpowiadający za komunikację komputera z kontrolerami napisany w C++ i .NET)
- WiiFlash ActionScript API (interfejs programowania aplikacji w postaci komponentu SWC)

Komunikacja z kontrolerem Wiimote odbywa się poprzez połączenie Bluetooth. W prezentowanym przykładzie do obsługi Bluetooth wykorzystano program BlueSoleil.

Kod odpowiedzialny za podłączenie Wiimote i nasłuchiwanie zdarzenia wciśnięcia przycisku Home:

```
private var wiimote1:Wiimote; //stworzenie
obiektu typu wiimote

wiimote1.connect();//podlaczenie
kontrolera

wiimote1.addEventListener(ButtonEvent.HOME
_PRESS,onHomePress); //dodanie obsługi
zdarzenia wcisniecia przycisku Home
```

Obiekty znajdujące się na markerach mogą wchodzić we wzajemne kolizje. Sprawdzana jest również odległość pomiędzy nimi. Kod realizujący kolizję pojazdu ze skałami oraz samych pojazdów:

```
if(car1.hitTestObject(stoneVect[ind])){
    updateScore("car1");
}
```

```

if(car2.distanceTo(car1)<=100 &&
car1.distanceTo(car2)<=100) {

    finishGame("car1",-1);

}

```

Przedstawione projekty są w fazie rozwojowej, dodatkowe informacje dostępne będą na stronie <http://ar.pjwstk.edu.pl>

Inne realizacje

Innym coraz częstszym przykładem wykorzystania powyższych technik są zastosowania reklamowe lub artystyczne. Ilustracją takiego projektu może być interaktywna książka przygotowana na tegoroczną Noc Muzeów w PJWSTK.



Rysunek 6. Interaktywna książka (autorzy: Bogna Kowalska i Marcin Wichrowski).

Instalacja składała się z książki, której odpowiednio lewe strony były markerami, natomiast prawe zawierały tekst. Dodatkowo na statywie znajdowała się kamera internetowa i piko projektor. Dzięki rozpoznawaniu markerów przez kamerę i odpowiednio dobrane projekcje, widzowie oglądając kolejne strony mogli zaobserwować pojawiające się animacje wideo będące uzupełnieniem już istniejącego tekstu. Podczas Nocy Muzeów prezentowane były również inne projekty wykorzystujące AR (np. aplikacja Solar System, autor Nakul Mishra), które testowane były przez uczestników i spotkały się z dużym zainteresowaniem.

Podsumowanie

Prezentowane prototypy projektów są próbą przeniesienia koncepcji i pomysłów autorów do środowiska Flex i sprawdzenia możliwości oraz ograniczeń bezpłatnych bibliotek. Dostępność tych rozwiązań i łatwość ich użycia oraz łączenia otwiera drogę do realizacji ciekawych interaktywnie i wizualnie programów nie tylko z zakresu rozrywki, ale również elearningu. Zastosowane biblioteki wymagają jeszcze optymalizacji i ulepszeń jednak należy się spodziewać, że kolejne wersje przyniosą lepsze rezultaty. Zaproponowane środowisko dodatkowo pozwala na przeniesienie aplikacji do przeglądarki internetowej i tym samym zwiększenie liczby ich odbiorców.

Bibliografia

- [1] <http://words.transmote.com/wp/flarmanager/>
- [2] <http://www.papervision3d.org/http://www.mikkoh.com/blog/?p=182>
- [3] <http://labs.adobe.com/technologies/air2/>