

GIML - Podstawy

Jacek Matulewski, wersja z 2 lipca 2016

1. Platforma GCAF

GCAF jest platformą, w której korzystając z nowego języka znaczników GIML można tworzyć aplikacje sterowane wzrokiem. Platforma ta obsługuje w tej chwili eyetrackery firmy SMI, ale zaawansowane są już prace nad wdrożeniem otwartego protokołu, który wykorzystywany jest m.in. w tańszych urządzeniach firm Mirametrix i Tobii.

Odpowiednio przygotowane pliki XML z kodem GIML są parsowane przez platformę i na ich podstawie budowany jest zestaw scen pokazywanych użytkownikowi. Scena może zawierać wiele obszarów pokazujących tekst, rysunki lub filmy. Użytkownik poprzez spojrzenia na poszczególne obszary może sterować działaniem aplikacji, w tym: włączanie, wyłączanie i aktywacje poszczególnych obszarów, uruchamianie animacji i nawigację pomiędzy scenami.

W kontekście projektowania badań psychologicznych, GIML i GCAF wyróżnia pełna interaktywność tworzonych z ich pomocą eksperymentów. Możliwa jest dowolna ścieżka przebiegu eksperymentu – to badany wybiera kolejność aktywowanych bodźców. Jednak istnieje również możliwość narzucenia kolejności aktywowanych obszarów lub ograniczenia ich dostępności w pewnych warunkach, co pozwala w szczególności na przeprowadzenie ściśle kontrolowanych eksperymentów.

2. Gramatyka języka GIML

Język GIML jest językiem deklaratywnym, opisującym efekt, który chcemy uzyskać (tj. wygląd i zachowanie aplikacji), a nie algorytm, który do takiego efektu prowadzi. To czyni go łatwym w użyciu przez osoby, które nie są profesjonalnymi programistami. Wygląd aplikacji opisywany jest znacznikami. Tworzenie aplikacji polega zatem na edycji plików XML, które zawierają odpowiednią strukturę znaczników określonych standardem języka GIML. Ponadto programowanie w GIML ułatwia walidator wbudowany w GCAF, który wskazuje błędne znaczniki w pliku.

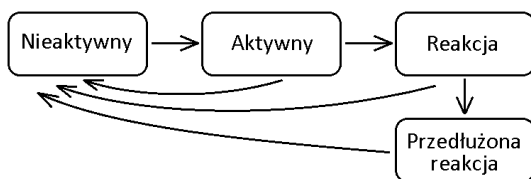
Łatwość używania języka GIML bierze się również z tego, że jest on stosunkowo prosty – składa się z zaledwie kilku znaczników. Ogromna jest natomiast liczba atrybutów, jakimi możemy kontrolować wygląd i zachowanie kontrolek. Jednak początkujący użytkownik, aby móc pisać pierwsze aplikacje musi znać tylko kilka z nich.

2.1. Sceny

Listing 1. Szablon interfejsu opisany w GIML. Z lewej w języku polskim, z prawej - w angielskim

<pre><sceny nazwaScenyDomyślnej="0" ...> <scena nazwa="0" ...> <obszar ...> <aktywacja /> <reakcja /> </obszar> ... </scena> ... </sceny></pre>	<pre><scenes nameOfDefaultScene="0" ...> <scene name="0" ...> <region ...> <activation /> <reaction /> </region> ... </scene> ... </scenes></pre>
---	---

Struktura interfejsu aplikacji GIML odzwierciedlona jest w hierarchii znaczników (listing 1). Interfejs aplikacji GIML zorganizowany jest w sceny; każda scena zawiera obszary, które mogą znajdować się w różnych stanach przełączanych spojrzeniem (por. rysunek 1). Podstawowym stanem obszaru jest *nieaktywny*. Spojrzenie użytkownika, ale nie dłuższe niż jedna sekunda (ten parametr można modyfikować), przełącza obszar w stan *aktywny*. Natomiast spojrzenie dłuższe niż sekunda przełącza go w stan *reakcji*. Przeniesienie spojrzenia na inny fragment sceny powoduje powrót obszaru do stanu nieaktywnego. Możliwe jest niezależne ustalenie wyglądu i zachowania obszaru w każdym z tych trzech stanów, włączając w to zmianę tekstu, rysunku, dźwięku, filmu, animacji czy akcji jakie może powodować.



Rysunek 1. Diagram stanów obszaru. Stan „przedłużona reakcja” dotyczy sytuacji, w której obszar wykonuje dłuższą czynność aktywowaną przez zmianę stanu

Jedynym obowiązkowym atrybutem znacznika `scena` jest jej nazwa. Więcej obowiązkowych atrybutów ma znacznik `obszar`, w którym poza nazwą musimy określić jego kształt, rozmiar i położenie na ekranie. Poza tym dla każdego stanu osobno można określić wyświetlany tekst, czcionkę, kolor i grubość ramki, rysunek, dźwięk, film, obszary włączane i wyłączane przy przejściu lub wyjściu ze stanu oraz scenę, do której nawigujemy. Można również określić animację obszaru lub samej jego zawartości. Ważnym atrybutem jest komunikat zapisywany w danych rejestrowanych przez eyetracker przy zmianie stanu obszaru – jego użycie znakomicie ułatwia późniejsze analizy danych okولوجraficznych. Opisy stanów aktywnego i

reakcji umieszczone są w osobnych pod-znacznikach `aktywacja` i `reakcja` zagnieżdżonych w znaczniku `obszar`.

Wyróżnikiem GIML jest to, że atrybuty mogą określać nie tylko wygląd, ale również zachowanie obiektów na ekranie, a także ich reakcję na działania użytkownika, w tym przede wszystkim na spojrzenie. Odpowiadają za to atrybuty znacznika `obszar`. Przykład takiego znacznika widoczny jest w listingu 2 (na listingu wytluszczone są atrybuty obowiązkowe). W tym przykładowym znaczniku w obszarze wyświetlany jest jedynie tekst, inny w każdym stanie. Zmienia się również kolor czcionki. Jak widać z listingu oznacza to, że atrybuty `tekst` i `kolorCzcionki` powinny być nadpisywane w pod-znacznikach `aktywacja` i `reakcja`. Najważniejsze jest jednak to, że w stanie `reakcja` określona jest akcja polegająca na zmianie bieżącej sceny na inną. Inne możliwe akcje to narysowanie ramki ściągającej uwagę, czy przesunięcie obszaru w inne miejsce.

Listing 2. Prosty przykład znacznika `Obszar` – najbardziej rozbudowanego znacznika w języku GIML

```
<obszar nazwa="PrzykładowyObszar"
  kształt="prostokąt" rozmiarX="400" rozmiarY="50"
  położenieŚrodkaX="300" położenieŚrodkaY="450"

  tekst="Dłuższe spojrzenie na obszar zmienia scenę"
  czcionka="Times" rozmiarCzcionki="20"
  kolorCzcionki="Black" >

  <aktywacja tekst="Nastąpiła aktywacja"
    kolorCzcionki="Navy" />

  <reakcja typAkcji="PrzejscieDoSceny"
    nazwaDocelowejSceny="innaScena"
    tekst="Nastąpiła reakcja"
    kolorCzcionki="Blue" />

</obszar>
```

Obszary są domyślnie włączone, ale można także zadeklarować obszar, który jest początkowo wyłączony, a włączany jest dopiero po zajściu jakiegoś zdarzenia. To daje możliwość dowolnego zmieniania zawartości sceny w wyniku interakcji programu z użytkownikiem.

2.2. Zasoby i szablon pliku

Jak wspomniano każdy z obszarów może mieć przypisany obraz, dźwięk lub film. W praktyce zasoby te często są wykorzystywane wielokrotnie w różnych obszarach i na różnych scenach. Aby uniknąć redundantnego alokowania pamięci, wprowadziliśmy abstrakcyjną warstwę zasobów. Dla przykładu każdy wykorzystywany rysunek musi być najpierw zadeklarowany w znaczniku `rysunek`, gdzie nadawana jest mu unikalna nazwa (por. listing 3). Obszary odwołują się do zasobów korzystając właśnie z tych nazw. To pozwala nie tylko uniknąć nadmiarowego obciążenia pamięci (ważne szczególnie w przypadku filmów), ale także umożliwia centralne zarządzanie zasobami oraz dodatkową kontrolę nad nimi (np. ustalenie liczby powtórzeń lub poziomu głośności).

Listing 3. Szablon pliku GIML pokazujący sposób wykorzystywania zasobów

```
<?xml version="1.0" encoding="UTF-8"?>
<ustawienia katalog="D:\GIML\Zasoby" kodJęzyka="pl">

  <rysunki>
    <rysunek nazwa="elmo" ścieżka="rysunki\elmo.png" />
    ...
  </rysunki>
  <dźwięki>
    <dźwięk ... />
  </dźwięki>
  <filmy>
    <film nazwa="ciastko" ścieżka="filmy\ciastko.avi"
      liczbaPowtórzeń="2" głośność="0.3" />
  </filmy>

  <sceny nazwaScenyDomyślnej="ulicaSezamkowa">
    <scena nazwa="ulicaSezamkowa">
      <obszar nazwa="Obszar wyświetlający rysunek" ...
        nazwaRysunku="elmo" />
      <obszar nazwa="Obszar wyświetlający film" ...
        nazwaRysunku="ciastko" nazwaDźwięku="ciastko" />
    </scena>
  </sceny>
</ustawienia>
```

Listing 3 pokazuje ogólny szablon pliku GIML. Tzw. korzeniem drzewa XML jest w nim znacznik `ustawienia`. Nieobowiązkowymi atrybutami tego znacznika są: ścieżka katalogu, w którym znajdują się zasoby oraz kod języka używanego w bieżącym pliku (na razie może to być język polski lub angielski). Wewnątrz tego znacznika mogą znajdować się cztery znaczniki: `rysunki`, `dźwięki`, `filmy` i `sceny`. W każdym z nich może znajdować się tylko jeden rodzaj znacznika, odpowiednio: `rysunek`, `dźwięk`, `film` i `scena`. Ten ostatni został już pobieżnie omówiony. Pozostałe są od niego znacznie prostsze. W każdym z nich obowiązkowe są tylko dwa atrybuty: nazwa zasobu oraz ścieżka do pliku, w jakim ów zasób się znajduje.