Projekt i implementacja języka do generowania slajdów

Wojciech Bajurny, Boris Komarov, Adrian Sośniak

1) Założenia i zakres projektu

W ramach zadania należy zaprojektować język (idea, skaner, parser) służący do generowania slajdów. Na podstawie pliku z tekstem w tym języku wygenerowane zostaną slajdy, na których można umieścić tytuł, bloki tekstu, a także obrazki. Każdy element slajdu może być konfigurowany. W przypadku tekstu można określać jego pozycję, rozmiar bloku i czcionkę. W przypadku obrazka należy określić jego ścieżkę, a także pożądany rozmiar i pozycję.
W celu zaimplementowania projektu wykorzystano język Java wraz z narzędziem ANTL4. W plikach *PresentationLexer.g4* i *PresentationParser.g4* zdefiniowano odpowiednio tokeny oraz gramatykę języka zgodnie z definicjami zamieszczonymi w punktach 2 i 3.

Na podstawie pliku tekstowego ze zdefiniowanym językiem aplikacja będzie kompilowana do pliku *.java*, w którym to wykorzystując bibliotekę *Graphics2D* opisane slajdy z ich konfiguracją są przetłumaczone na obiekty *BufferedImage* o odpowiedniej wielkości z zamieszczonymi blokami tekstu/obrazkami. Każdy obrazek będzie generowany do pliku o nazwie według schematu *nazwa_prezentacji*/*nr_slajdu.png*.

2) Lista tokenów

SLIDE :'SLIDE';
TEXT : 'TEXT';
IMAGE : 'IMAGE';
COLON : ':' ;
NEW_LINE : '\n' ;
COMMENT : '//' [a-zA-Z_0-9]* '\n' -> skip;

INT : [0-9]+ ;
DOUBLE : [0-9]+'.'[0-9]+ ;
ID : [a-zA-Z_][a-zA-Z_0-9]* ;
TEXT_BLOCK :'{\n'[a-zA-Z_0-9 \n]*'}' ;
PATH : '"'[a-zA-Z0-9.-/]+'"';
WS : [\t\r\f]+ -> skip ;

COKOLWIEK:.;

Słowa klucze definiują nowy slajd, blok tekstu i wstawiany obrazek. Oprócz tego mamy tokeny pomocnicze (dwukropek, nowa linia, przerwy w tekście), typowe int, double i id, a także leksem TEXT_BLOCK, który definiuje blok tekstu oraz PATH dla ścieżki do obrazka.

3) Gramatyka języka

```
program : (slide)+ EOF ;

slide : SLIDE (NEW_LINE)+ (func)* #someSlide
   | SLIDE #lastSlide
   ;

func: TEXT (NEW_LINE)+ (parameter)* #textFunction
   | IMAGE (NEW_LINE)+ (parameter)* #imageFunction
   | (TEXT|IMAGE) #lastFunction
   ;

parameter : ID COLON (NEW_LINE)* expr (NEW_LINE)+ #someParameter
   | ID COLON (NEW_LINE)* expr #lastParameter
   ;

vector2 : L_PAR INT COMMA INT R_PAR
   ;

expr: INT
   | DOUBLE
   | vector2
   | TEXT_BLOCK
   | PATH
   ;
```

Domyślnym wyprowadzeniem jest program, który składa się z conajmniej jednego slajdu. Slajd zaczyna się od wcześniej zdefiniowanego słowa klucz, a od nowej linii można wstawiać blok tekstu lub obrazek (opisane w wyprowadzeniu dla *func*). Te z kolei mają swoje określone parametry (przy braku używane są domyślne), które mogą przyjmować typy zdefiniowane w wyprowadzeniu dla *expr* -> INT, DOUBLE, TEXT_BLOCK lub PATH. Wyprowadzenie expr -> vector2 służy do przedstawienia wektora (2 wartości w nawiasach), za pomocą których można określić pozycję, czy rozmiar bloku tekstu/obrazka.

Dodać instrukcje użytkowania

W pierwszym kroku należy przygotowywać plik *.txt napisany zgodnie ze zdefiniowaną gramatyką. Oto przykład:

```
SLIDE
TEXT
position:(40,160)
font_size:72
text:{
It handles
}
TEXT
position:(80,240)
font_size:72
text:{
Images
}
IMAGE
path:"res/Slime.png"
position:(250,250)
size:0.4

SLIDE
TEXT
position:(180,300)
font_size:64
text:{
Thank you for your attention
}
```

Tworząc bloki tekstu możemy zdefiniować parametry:
position (vector2),
font_name (z dostępu i środowiska, np. “Serif”),
font_size,
font_style (`liczba całkowita będąca maską bitową, która może mieć wartości PLAIN albo być kombinacją BOLD/ITALIC`),
font_color (podawany jako hex string, np. “#FF0000”),
oraz sam *text* - ten musi być umieszczony pomiędzy nawiasami klamrowymi {}.
W przypadku nieokreślenia jakiegoś parametru przyjmowane są domyślne - “Courier”, czarny, PLAIN, pozycja (0,0) i tekst “example text”.
Tworząc bloki dla obrazków możemy zdefiniować również *position*,
a także *path*,
size (double).
Domyślnie mamy pozycję (0,0), path “res/meme.jpg” i size = 1.0.
W obu przypadkach zdefiniowanie nieujętego słowa klucz dla danego typu (obrazek/tekst), to zostanie to pominięte.
Następnie należy uruchomić program (*klasa ParserTestsMain.java*), a ścieżkę do utworzonego pliku podać jako argument linii komend. Po uruchomieniu programu w obecnym folderze utworzony zostanie podfolder */output*, a w nim kolejne slajdy o nazwach *slide_0.png, slide_1.png* i kolejno aż do wygenerowania wszystkich obrazków.

Instrukcja instalacji

Do poprawnego działania projektu należy mieć zainstalowane openjdk (testowano na wersji 18), a także ANTLR4. Następnie można pobrać projekt z następującego repozytorium https://github.com/Axyu01/MIASI-Presentation-Generator i uruchomić go - najprościej pod ścieżką out/artifacts/miasi_jar/miasi.jar znajduje się gotowy .jar do uruchomienia. Wystarczy wywołać go z linii komend poleceniem java -jar miasi.jar *ścieżka_do_pliku_z_tekstem_prezentacji*.