

# Textový editor obohacený o grafické prvky



Tímový projekt 2011/12  
tp-team-10@googlegroups.com

# Úvod do problematiky

- Myšlienka vytvoriť multiplatformový grafický editor
  - Využiť grafické prvky na zvýraznenie štruktúr textu (grafické bloky ...)
  - Podporiť myšlienku „literate programming“
  - Málo podobných riešení na súčasnom trhu
- 
- Pokračujeme vo vývoji už existujúceho riešenia tímu UFOPAK
  - Základná funkcionálna aplikácia je implementovaná

# Ciel' projektu

- Plne sa sústrediť na rozšírenie problematických častí editora
- Doplnenie editora o ďalšie špecifikované funkcionality
- Moderné používateľské rozhranie
- Zefektívnenie výpočtovo náročných operácií
- Vytvorenie plnohodnotného editora a jeho nasadenia pre reálne využitie
- Používateľský prívetivé rozhranie, ktoré by uľahčovalo prácu s textom

# Ukážka editora

The image shows a screenshot of a code editor window titled "xtursky2 zle.cpp\* — TrollEdit". The editor displays C++ code for a linked list. The code includes headers for `<stdio.h>`, `<string.h>`, `<stdlib.h>`, and `<ctype.h>`. A `struct` named `zoznam` is defined with fields for `char nazov[100]`, `char miesto[50]`, `char ulica[50]`, `int rozloha`, `int cena`, `char popis[205]`, and a pointer `zoznam *dalsi`. The `main` function initializes a pointer `*pos = NULL`. A `void uvolni(zoznam **pos)` function is defined to free the list. The function uses a `while` loop to traverse the list, updating a pointer `n` to `n->dalsi` and freeing each node until `n` is `NULL`. The function then sets `*pos = NULL`.

Annotations in the image explain the code:

- An arrow points to the `<ctype.h>` include with the text: *definujem si strukturu reality, do ktorej nacitam prvky suboru*
- An arrow points to the `zoznam *dalsi` field in the struct with the text: *pointer na nasledujucy zaznam*
- An arrow points to the `**pos` parameter in the `uvolni` function with the text: *dve \*\*, lebo sa bude menit pri beh funkcie pomocne pointery, aby som sa mohol pohybovat v danom dynamickom zozname*
- An arrow points to the `n=m;` line in the `while` loop with the text: *n ako pomocny pointer na predchadzaju polozku*
- An arrow points to the `m=n->dalsi;` line in the `while` loop with the text: *uz mozem uvolnit, lebo som sa posunul s m na dalsiu polozku*

# Aké technológie sme použili?

Editor je realizovaný ako multiplatformová desktopová aplikácia

Vychádzame z už použitých technológií:

- Qt Creator                      vývojové prostredie
- Qt framework                široké možnosti práce s grafikou
- C++                                programovací jazyk editora
- Lua (LuaJit)                    realizuje veľkú časť vypočtu
- Knižnica Lpeg                 syntaktická analýza textu

# Analýza riešenia

- Rozpracovaný projekt po minulom tíme UFOPAK
- Klasická práca s textom obohatená o grafické prvky ( práca s blokmi )
- Vizualizácia zdrojového kódu je neefektívna
- Podpora jazykov: C, Lua, Xml
- Analýza zdrojového kódu a samotná práca s editorom v jednom vlákne
- Jednoduchšie a nezaujímavé používateľské rozhranie
- Chýba podpora základných operácií ako: Undo/Redo, Copy/Paste

# Funkcionálne požiadavky na editor

- Modernejšie používateľské prostredie
- Práca s editorom založená na dvoch módoch (textový a grafický)
- Paralelizovanie výpočtovo náročných operácií
- Prenesenie spracovania AST stromu na stranu Lua
- Zabudovanie pokročilej práce s textom – Undo, Redo, Copy/Paste
- Zabudovanie a vytvorenie vlastných Shortcuts

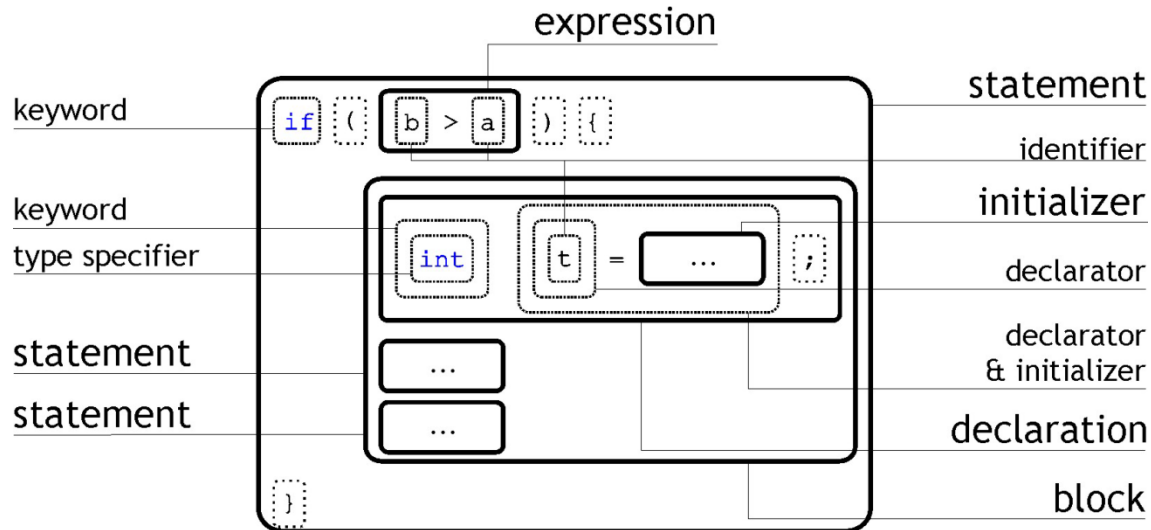
# Nefunkcionálne požiadavky na editor

- Rýchlosť a spoľahlivosť
- Modulárnosť
- Poskytnúť možnosti konfigurácie editora
- Redesign používateľského rozhrania GUI



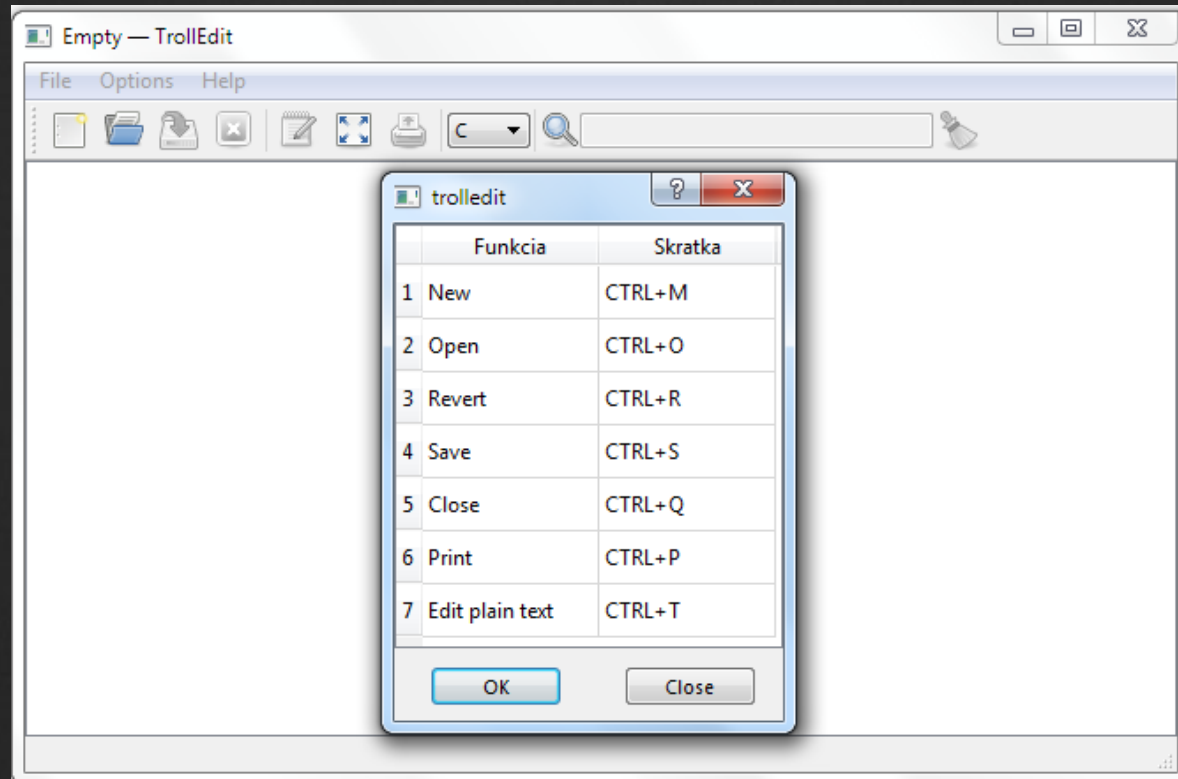
# Spracovanie AST stromu

- Abstract Syntax Tree – stromová reprezentácia štruktúry zdrojového kódu
- Abstraktný = neviaže sa na konkrétny programovací jazyk



# Shortcuts

- Konfigurovatelné klávesové skratky



# Dva módy

Blok s grafickými prvky



Textový blok

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h> // dve **

void uvolni(zoznam **pos) {
    zoznam *m, *n;
    m= *pos;

    while (m!=NULL) {
        n=m;
        m=m->dalsi;
        free(n);
    }
    *pos=NULL;
}

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
//dve ** [!223,74,1!]
void uvolni(zoznam **pos) {
    zoznam *m, *n;
    m= *pos;

    while (m!=NULL) {
        //n ako pomocny pointer
        n=m;
        m=m->dalsi;
        free(n);
    }
    *pos=NULL;
}
```

# Implementácia prototypu

- Vykonaný refactoring podľa zadefinovaných štýlov
- Implementácia funkcií pre prácu s AST stromom na strane Lua
- Zabudovanie dvoch módov – Undo Redo v rámci práce s textom
- Zabudovanie paralelizmu – spracovanie syntaktickej analýzy na pozadí
- Zabudovanie Shortcuts – rozšírenie existujúcich skratiek
- Úprava editora za použitia CSS štýlov

# Demo

*Ukážka nástroja Trolledit*

# Plán do LS

- Držanie AST stromu na strane Lua – dopytovanie na strom z Qt
- Rozšíriť paralelizmus aj na stranu Lua
- Vlastné konfigurovateľné shortcuts
- Undo Redo rozšíriť aj pre prácu s blokmi
- Zabudovanie pokročilého dokumentačného bloku podporúceho RTF
- Využiť QML a CSS štýly

