

Použití jazyka CQML

1 Syntaxe jazyka *CQML*

V této kapitole je na příkadech vysvětlena syntaxe jazyka *CQML* pro tvorbu *GUI*.

1.1 Tvorba elementů

Kód 1 vytvoří hierarchii elementů, kde se na vrcholu nachází element základního typu *Element* a v něm se nachází dva elementy typu *Rectangle*, z nichž jeden obsahuje element typu *MouseArea*.

Listing 1: Syntaxe pro tvorbu hierarchie elementů.

```
Element
{
    Rectangle
    {
    };
    Rectangle
    {
        MouseArea
        {
        };
    };
}
```

1.2 Přiřazení hodnoty atributu

Kód 2 ukazuje nastavení hodnoty atributu *width* pomocí výrazu a hodnotu atributu *height* pomocí funkce vracející požadovanou hodnotu.

1.3 Provázání atributů

Kód 3 demonstruje možnost provázat mezi sebou jednotlivé atributy. Vrchní element má nastavenou hodnotu atributu *width* na 500, s tímto atributem je následně provázán atribut *height*, který se nastaví na hodnotu o 100 větší. Parametry elementu *Rectangle* jsou pak provázány s hodnotami atributů nadřazeného elementu pomocí klíčového slova *parent*. Tudíž oba elementy mají schodné hodnoty atributů *width* a *height*.

Listing 2: Syntaxe přiřazení hodnoty elementu pomocí výrazu nebo funkce.

```
Element
{
    width : 500;
    height : {
        return 500;
    };
}
```

Listing 3: Ukázka provázání atributů mezi sebou.

```
Element
{
    width : 500;
    height : width+100;
    Rectangle
    {
        width : parent.width;
        height : parent.height;
    };
}
```

1.4 Použití identifikátoru pro přístup k elementu

Prostřednictvím atributu *id* je možné označit element identifikátorem, pomocí něhož je možné k elementu přistupovat v rámci daného *CQML* souboru. V kódu 4 je jednomu z elementů přiřazen identifikátor *firstRectangle*. Prostřednictvím tohoto identifikátoru je pak hodnota atributu *width* označeného elementu přiřazena jinému.

1.5 Rozšíření elementu o nový atribut

Kód 5 ukazuje rozšíření elementu o nový atribut jménem *alpha* typu *int* pomocí klíčového slova *property*, přičemž je mu přiřazena hodnota 100.

1.6 Reference na element

V kódu 6 je kořenový element rozšířen o atribut *refElement*, v němž se uchovává reference na atribut označený identifikátorem *firstRectangle*. Následně je přistupováno k atributu *width* referencovaného elementu prostřednictvím dané reference.

Listing 4: Ukázka přístupu k elementu a jeho atributům pomocí identifikátoru.

```
Element
{
    Rectangle
    {
        id : firstRectangle;
        width : 500;
    };
    Rectangle
    {
        width : firstRectangle.width;
    };
}
```

Listing 5: Syntaxe rozšíření elementu o nový atribut.

```
Element
{
    property int alpha : 100;
}
```

Listing 6: Ukázka přístupu k atributům elementu pomocí reference.

```
Element
{
    property Element refElement : firstRectangle;
    width : refElement.width;
    Rectangle
    {
        id : firstRectangle;
        width: 500;
    };
}
```

Listing 7: Syntaxe přiřazení funkce pro obsluhu události.

```
Element
{
    MouseArea
    {
        width : 100;
        height : 100;
        MouseClicked : {
            width = 500;
            height = 500;
        };
    };
}
```

1.7 Zpracování událostí

Kód 7 ilustruje přiřazení funkce pro obsluhu události kliknutí myši. V tomto případě se po kliknutí myši na element *MouseArea* změní hodnoty jeho atributů *width* a *height* ze 100 na 500.

1.8 Použití externí logiky

Pro použití externí logiky definované v rámci uživatelské aplikace lze do kódu vygenerovaného překladačem zahrnout hlavičkový soubor pomocí příkazu *include*. Kód 8 ukazuje použití funkce *GetWidth*, která je deklarovaná uvnitř importovaného souboru „logic.h“.

Listing 8: Ukázka kódu používajícího externí funkci deklarovanou v hlavičkovém souboru.

```
include "logic.h"
Element
{
    Rectangle
    {
        width : {
            return GetWidth();
        };
    };
}
```

1.9 Použití komponenty

Samostatné soubory v jazyce *CQML* lze používat jako samostatné komponenty v rámci jiných *CQML* souborů, s nimiž je možno manipulovat jako s jakýmkoli jiným elementem.

Listing 9: Zdrojový soubor komponenty.

```
Rectangle
{
    property int status : 0;
    width: 500;
    height: 500;
}
```

Toto je demonstrováno v ukázce kódu 10, kde je pomocí příkazu *import* načtena komponenta definovaná v souboru „component1.h“ (viz. kód 9), která je pro použití označena klíčovým slovem *Square*. Pomocí daného klíčového slova jsou pak vytvořeny dvě instance dané komponenty a to stejným způsobem, jako by tomu bylo u běžných elementů.

Listing 10: Ukázka použití komponenty z jiného souboru.

```
import "component1.h" as Square
Element
{
    width : 500;
    height : 500;
    Square
    {
        width : 100;
        height : 100;
    };
    Square
    {
        width : 200;
        height : 200;
    };
}
```

2 Datové typy poskytované jazykem *CQML*

Knihovna *CQML* poskytuje několik hodnotových a referenčních datových typů pro použití při tvorbě *GUI*. Pro reprezentaci grafických elementů slouží referenční typy, aby se například při přiřazení elementu do nějaké proměnné element nekopíroval a bylo možné s ním manipulovat i prostřednictvím dané proměnné.

2.1 Grafické elementy

Grafické elementy jsou základními stavebními součástmi *GUI* a každý z nich může být umístěn v hierarchii. Ke všem z nich je v rámci jazyka *CQML* přistupováno jako k referenčním typům.

2.1.1 *Element*

Jedná se o základní element pro použití uvnitř hierarchie *GUI*. Sám nemá žádnou grafickou podobu a využívá se zejména jako kontejner pro umístění dalších grafických elementů.

Seznam atributů:

- x** - horizontální souřadnice elementu. (typ *int*)
- y** - vertikální souřadnice elementu. (typ *int*)
- width** - šířka elementu. (typ *int*)
- height** - výška elementu. (typ *int*)
- visible** - hodnota 1 znamená, že se element vykreslí. Hodnota 0 zamezí jeho vykreslování. (typ *int*)
- enabled** - hodnota 1 udává, že je element zapnutý a bude reagovat. Hodnota 0 element vypne. (typ *int*)

Seznam funkcí pro obsluhu událostí:

- KeyPressed** - funkce pro obsluhu stisknutí klávesy.
- KeyReleased** - funkce pro obsluhu puštění klávesy.

2.1.2 *Rectangle*

Jednoduchý grafický element sloužící pro vykreslení jednoduchého obdélníku o specifikovaných rozměrech. Lze mu nastavit barvu a šířku okraje a také barvu výplně.

Seznam atributů:

- border** - šířka okraje. (typ *float*)
- borderColor** - barva okraje. (typ *Color*)
- color** - barva výplně. (typ *Color*)
- všechny atributy z typu *Element*

Seznam funkcí pro obsluhu událostí:

- všechny funkce z typu *Element*

2.1.3 *MouseArea*

Podobně jako základní typ *Element*, nemá tento element žádnou grafickou podobu. Slouží zejména k obsluze vstupních událostí vyvolaných pomocí myši.

Seznam atributů:

všechny atributy z typu *Element*

Seznam funkcí pro obsluhu událostí:

MousePressed - funkce pro obsluhu stisknutí tlačítka myši.

MouseReleased - funkce pro obsluhu puštění tlačítka myši.

MouseClicked - funkce pro obsluhu kliknutí tlačítkem myši.

MouseEntered - funkce pro obsluhu najetí myši na plochu vymezenou elementem.

MouseExited - funkce pro obsluhu vyjetí myši z plochy vymezené elementem.

MouseMoved - funkce pro obsluhu pohybu myši.

MouseScrolled - funkce pro obsluhu otočení kolečkem myši.

všechny funkce z typu *Element*

2.1.4 *Image*

Grafický element sloužící pro zobrazení obrázku načteného ze souboru. V případě, že rozměry obrázku v souboru jsou odlišné od rozměrů elementu, pak se obrázek roztáhne či smrští do definované plochy.

Seznam atributů:

img - data obrázku. (typ *Img*)

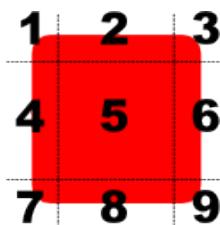
všechny atributy z typu *Element*

Seznam funkcí pro obsluhu událostí:

všechny funkce z typu *Element*

2.1.5 *ScaledImage*

Tento element je inspirován elementem *BorderImage* poskytovaný knihovnou *QML*. Načte obrázek, který následně rozřeže na devět částí (viz. obrázek 1), přičemž rohy (1, 3, 7 a 9) zůstávají stejné velikosti jako v původním obrázku, ostatní okrajové části (2, 4, 6 a 8) mohou změnit velikost jen v jednom rozměru a prostřední část (5) může velikost měnit v obou rozměrech.



Obrázek 1: Diagram ilustrující rozřezání obrázku pomocí elementu *ScaledImage* [?].

Seznam atributů:

leftBorder - hodnota udávající vzdálenost pravého řezu od pravého okraje v souřadnicích obrázku. (typ *int*)

rightBorder - hodnota udávající vzdálenost levého řezu od levého okraje v souřadnicích obrázku. (typ *int*)

topBorder - hodnota udávající vzdálenost horního řezu od horního okraje v souřadnicích obrázku. (typ *int*)

bottomBorder - hodnota udávající vzdálenost spodního řezu od spodního okraje v souřadnicích obrázku. (typ *int*)

všechny atributy z typu *Image*

Seznam funkcí pro obsluhu událostí:

všechny funkce z typu *Image*

2.1.6 *Text*

Grafický element sloužící pro vykreslení textu o definovaných parametrech.

Seznam atributů:

text - vykreslovaný řetězec. (typ *string*)

textColor - barva písma. (typ *Color*)

font - styl písma. (typ *Font*)

všechny atributy z typu *Element*

Seznam funkcí pro obsluhu událostí:

všechny funkce z typu *Element*

2.2 *Hodnotové typy*

Některá data používaná elementy jsou sloučena do struktur, aby bylo možné s nimi manipulovat najednou. K tomuto účelu je vytvořeno několik hodnotových typů.

2.2.1 *Font*

Datový typ, jenž slouží pro uchování údajů o stylu písma.

Seznam atributů:

family - název písma. (typ *string*)

size - velikost. (typ *int*)

2.2.2 *Img*

Tato datová struktura slouží k uchování dat cesty pro načtení obrázku ze souboru.

Seznam atributů:

src - cesta k obrázku. (typ *string*)

2.2.3 *Color*

Tato struktura slouží pro uchování údajů o barvě prostřednictvím hodnot pro červený, zelený a modrý kanál.

Seznam atributů:

red - hodnota červeného barevného kanálu. (typ *float*)

green - hodnota zeleného barevného kanálu. (typ *float*)

blue - hodnota modrého barevného kanálu. (typ *float*)