

Vývoj mobilní aplikace s využitím JavaFX Mobile
Developing JavaFX Mobile applications

Bakalářská práce

Jan Trestr

Vedoucí bakalářské práce: RNDr. Jaroslav Icha

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

2010

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 13.12.2010

Jan Trestr

Anotace

Bakalářská práce se zabývá možnostmi skriptovacího jazyka JavaFX pro vývoj mobilní aplikace a následnými možnostmi pro zveřejnění této aplikace .

Abstract

Bachelor work deals with use of JavaFX script language in a development of a mobile application and consequently possibilities for publishing these application.

Poděkování

Rád bych poděkoval vedoucímu mé závěrečné práce RNDr. Jaroslavu Ichovi za to, že mi umožnil pracovat s touto tematikou, za konzultace a rady, které souvisely s vytvořením práce.

Dále bych chtěl poděkovat rodině za podporu v čase, který jsem této práci věnoval.

Obsah

Prohlášení.....	2
Anotace	3
Poděkování.....	4
Obsah	5
1. Úvod.....	7
1.1 Úvod	7
1.2. Metodika práce	8
2. Popis platformy JavaFX.....	10
2.1. JavaFX.....	10
2.2. Historie a budoucnost JavaFX	10
2.3. Architektura platformy JavaFX.....	12
2.4. JavaFX Script.....	13
2.5. RIA Aplikace.....	15
2.6. Popis konkurenčních produktů pro JavaFX.....	16
2.6.1. Adobe Flash	16
2.6.2 Microsoft Silverlight	17
2.6.3 AJAX	17
2.7. JavaFX Mobile.....	18
2.8. Architektura JavaFX Mobile	19
2.9. Instalace JavaFX Mobile na mobilní zařízení.....	20
2.10. Distribuce programů v JavaFX Mobile	20
3. Technologický a didaktický návrh aplikace	21
3.1. Technologický návrh	21
3.2. Požadavky na aplikaci z hlediska didaktického	22

3.3 Zhodnocení výsledků dotazníků.....	23
4. Popis vývoje a implementace aplikace	26
4.1. Vývojové prostředí NetBeans IDE	26
4.2. JavaFX Scene Graph	28
4.5. Popis prvků použitých při tvorbě aplikace	29
4.5.1. Stage	29
4.5.2. Scene	31
4.5.3. Group.....	34
4.5.4. Objekty scény	35
4.5.6. Animace	52
4.5.7. Media.....	53
4.5.8. Layouts.....	55
4.5.9. Charts.....	56
4.5.10. Alert	58
4.5.11. Actions	58
4.6. Schéma aplikace	60
4.7. Finální verze aplikace	61
5. Shrnutí.....	63
Reference:	655

1. Úvod

1.1 Úvod

V dnešní době technických zázraků, kdy se vlastnictví mobilního telefonu stalo téměř stejnou samozřejmostí jako například vlastnictví deštníku, by bylo škoda omezit využití právě zmíněného mobilního telefonu jen na komunikaci, což je jeho původní účel. Současné mobilní telefony umožňují spuštění mnoha různých aplikací, jako jsou různé hry nebo přehraavače. Zajímavou možností proto představuje využití mobilních telefonů z pohledu pedagoga. Pro dnešní mládež je určitě zajímavější variantou výuky použití mobilní aplikace než „suchý“ výklad u tabule.

Hlavním cílem této mé práce bylo vytvoření mobilní aplikace s použitím nového skriptovacího jazyka JavaFx Script od firmy Sun Microsystems. Aplikace by měla fungovat jako pomůcka při výuce anglického jazyka pro začátečníky. Zadáním práce není zaměření se na didaktickou stránku problému, ale na technické provedení aplikace z pohledu inženýrského. Didaktická část byla konzultována s vyučujícími cizích jazyků.

Skriptovací jazyk JavaFX Script umožňuje vytváření bohatých aplikačních rozhraní, která integrují video, zvuk, grafiku i animace, čehož bylo při vývoji aplikace využito.

Pro vývoj aplikace bylo použito integrované prostředí NetBeans 6.9.1 od firmy Sun Microsystems, funkčnost aplikace byla ozkoušena na importovaném mobilním emulátoru přímo ve vývojovém prostředí NetBeans.

Dalšími cíli práce je také stručný popis nově vzniklé platformy JavaFX, její historie a architektury. Představeny budou stručně i hlavní konkurenti JavaFX Adobe Flash, Microsoft Silverlight a Flex. Konkrétně se práce dále zaměřuje na představení JavaFX Mobile a její architektury.

Další kapitola je pak věnována návrhu aplikace po stránce technologické a krátce také didaktickému pohledu na aplikaci. Tato část obsahuje i stručné zhodnocení dotazníku, který měl přinést informace o využitelnosti a funkčnosti finální aplikace. Ta byla představena několika uživatelům, kteří pak odpověděli v krátkém dotazníku, zda se jim aplikace líbí a zda se jim jeví jako přínosná. Mezi dotázanými byli studenti i dospělí lidé, a to i z řad pedagogů, takže svým názorem přispěli obě strany, které by měly na vývoji aplikace mít nějaký zájem.

Předposlední část této práce je pak věnována popisu implementace celé výukové aplikace.

V závěrečné kapitole jsou shrnuty vlastnosti, výhody i nevýhody JavaFx jakož i shrnutí celé bakalářské práce.

1.2. Metodika práce

Při vypracovávání bakalářské práce bylo postupováno následujícím způsobem. Nejprve bylo nutné dostatečně nastudovat dostupné informace o JavaFX. Jelikož se jedná o relativně novou platformu, není zrovna snadné informace získat. Knižně vyšlo jen několik málo publikací a kvalitních zdrojů na internetu také není mnoho. Proto k vypracování bakalářské práce pomohli nejvíc knihy *Pro JavaFX Platform* a *JavaFX Developers guide*. První pomohla s pochopením základů JavaFX a JavaFX Script a druhá s možnostmi designování a programování aplikací v JavaFX. Z internetových zdrojů byly nejplatnější oficiální stránky www.javafx.com, kde je možné najít mnoho tutoriálů a příkladů pro pochopení fungování platformy, a stránky www.oracle.com, které obsahují kompletní dokumentaci pro JavaFX a mnoho diskuzních fór, kde bylo možné hledat inspiraci pro řešení vzniklých problémů.

Pro vlastní programování byla rozhodně přínosná znalost programování na platformě Java.

Při návrhu aplikace bylo využito informací získaných z dotazníku, který byl zadán k vyplnění několika studentům ve stáří přibližně kolem 15 let a několika vyučujícím. Následně jim byla aplikace představena a ve druhé části dotazníku odpovídali, jak jsou s aplikací spokojeni.

2. Popis platformy JavaFX

2.1. JavaFX

Platforma JavaFX je softwarová platforma, která vychází z programovacího jazyka Java od firmy Sun Microsystems. JavaFX byla prvotně vytvořena jako vývojové prostředí pro tvorbu takzvaných RIA aplikací, tedy Rich Internet Applications, což znamená v překladu bohaté internetové aplikace. Prostředí JavaFX navíc umožňuje celou aplikaci stáhnout, a spouštět z plochy jako klasický desktopový program.

Firma Sun Microsystems přichází s platformou JavaFX ve snaze konkurovat dnes již zaběhlým produktům, zvláště takovým jako Adobe Flash, který ovládá v současnosti většinu trhu s internetovými aplikacemi, nebo například Microsoft Silverlight či Flex.

Hlavní myšlenkou JavaFX je tedy vytvoření vývojového prostředí, které by rozšiřovalo možnosti programovacího jazyka Java, vyvarovalo by se jeho dosavadních chyb a zároveň by přineslo mnohem snazší a elegantnější možnost tvorby UI (User interface), tedy uživatelských rozhraní. K tomu využívá JavaFX svůj vlastní skriptovací jazyk JavaFXScript, kterému bude věnována jedna z dalších kapitol.

Dalším cílem JavaFX je rozšíření na zařízení každodenní potřeby, jako jsou právě počítače, mobilní telefony nebo televize. Jak uvádí samy tvůrci, chtějí obsáhnout všechna zařízení od plochy počítače až po obývací pokoj.

2.2. Historie a budoucnost JavaFX

Původní myšlenka na vytvoření nového jazyka pro tvorbu RIA pochází z hlavy Chrise Olivera, toho času zaměstnance firmy SeeBeyond. Ten přichází s jazykem F3 (Form Follow Functions), který měl přinést usnadnění tvorby právě výše zmíněných internetových aplikací. Firma SeeBeyoun následně

přechází pod společnost Sun Microsystems a ta mění název z F3 na JavaFX Script a v roce 2007 oznamuje projekt JavaFX na konferenci JavaOne. Již v březnu roku 2008 představuje firma Sun Microsystems první zkompileované pluginy pro NetBeans, které jsou následně v červenci téhož roku uvolněny společně s prvními náhledy pro Windows a Macintosh. Na konci roku 2008 pak dochází k představení první verze platformy JavaFX, a to JavaFX 1.0 SDK. [5]

Verze JavaFX 1.1 SDK, která je zveřejněna v únoru 2009, pak přináší podporu tvorby mobilních aplikací.

Na konferenci JavaOne v roce 2009 firma Sun Microsystems přichází s verzí JavaFX 1.2 SDK. To sebou nese lepší podporu Linuxu a Solaris, vestavěná ovládací zařízení a layouty nebo například grafy.

22.dubna 2010 je zveřejněna verze JavaFX 1.3 SDK. Tato verze vylepšuje běh programu, přináší další nové ovladače, nový font pro JavaFX zvaný Amble a díky propojení s novou verzí NetBeans 6.9 umožňuje tvorbu uživatelských rozhraní v editoru JavaFX composer.

Na druhou polovinu roku 2011 je pak plánováno vydání verze JavaFX 2.0 SDK. Asi nejvýznamnější změnou by mělo být ukončení vývoje skriptovacího jazyka JavaFX Script.

2.3. Architektura platformy JavaFX

Architektura platformy JavaFX je vystavěna jako cross – platform a platform – specific platforma. To znamená, že aplikace naprogramované v JavaFX fungují na různých platformách.[1]

Celá architektura JavaFX je vystavěna na virtuálním stroji Java Virtual Machine. Ten pomocí svých programů a datových struktur umožňuje spouštění programů napsaných v jazyce Java. Tento virtuální stroj pak převádí kód psaný v jazyce Java na mezikód Java bytecode, který je pak dále zpracováván.

[3]

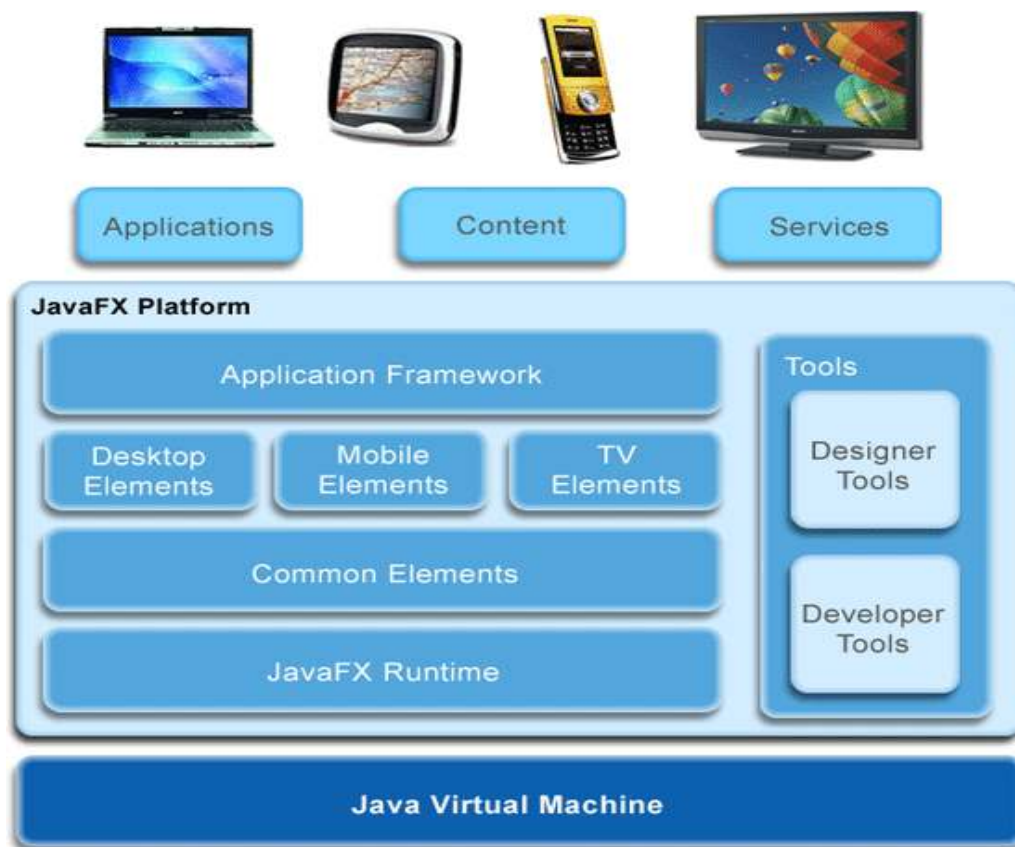
JavaFX Runtime je prostředí, na němž už funguje přímo JavaFX. Toto prostředí se skládá z nativních knihoven a souborů JAR (Java tříd a jejich zdrojů). Toto prostředí je společné všem platformám. [4]

Common elements obsahuje všechny společné prvky, které jsou podobně jako JavaFX Runtime funkční pro všechny platformy. Jejich součástí jsou veškeré knihovny a další prostředí.[1]

Dalšími součástmi architektury platformy JavaFX jsou prvky pro tvorbu desktopových, mobilních a televizních aplikací. Ty obsahují prostředí a knihovny nutné ke spuštění aplikace na dané platformě. Pro tuto práci se dále zabýváme konkrétně prvky pro mobilní aplikace.

Nad těmito prvky pak funguje Application Framework, který umožňuje tvorbu aplikací, služeb a produktů pro mobilní zařízení. [1]

Pro potřeby vývojářů poskytuje architektura JavaFX různé nástroje, stejně jako pro usnadnění práce designérů nabízí snadný převod médií do formátu JavaFX a jejich zobrazení.



Obrázek1: Architektura JavaFX

2.4. JavaFX Script

Platforma JavaFX má vlastní skriptovací jazyk, JavaFX Script. JavaFX Script používá deklarativní syntaxi, která popisuje, čeho se má v programu dosáhnout, ale ne jak. Tím je kód zjednodušen například oproti klasické Javě. [19]

Jazyk JavaScriptFX je staticky typovaný, takže hodnoty všech proměnných musí být známe již při překladu programu.

Jednou z výhod skriptovacího jazyka je takzvaný Data binding, což je vlastně technika, která umožňuje automatickou synchronizaci dat aplikace

s komponentami uživatelského rozhraní, kdy při každé změně dat dojde i ke změně komponent vázaných k těmto datům.

Propojení jazyka JavaFX Script s knihovnamí Java Swing, Java 2D a Java 3D umožňuje snadnou tvorbu uživatelských rozhraní.

Zdrojové kódy psané v jazyce JavaFX Script jsou ukládány do souboru s koncovkou fx.

Podle slov vývojářů JavaFX by měl být kód napsaný v jazyce JavaFX Script až pětkrát kratší než kód napsaný v klasickém jazyce Java.[20] Zde je uveden krátký příklad vykreslení scény s červeným kruhem.

Ukázka kódu v jazyce JavaFX Script:

```
Stage
{
  style:StageStyle.TRANSPARENT

  scene:Scene {
    content:Circle {
      centerX:50
      centerY:50
      radius:50
      fill:Color.RED
    }
    fill:null
  }
}
```

Ukázka kódu v jazyce Java:

```
JFrame f = new JFrame();
if
(
```

```

    com.awt.sun.util.AWTUtilities.isTranslucencySupported()

    {
    com.awt.sun.util.AWTUtilities.setTranslucency(true );
    }
    frame.setUndecorated(true);
    JPanelpanel = newJPanel()
    {
    protected void paintComponent(Graphics g )
    {
        Graphics2D gfx = (Graphics2D)g.create ();
        gfx.setRendering Hints (Rendering
        Hints.KEY_ANTIALIASING ,
        Rendering Hints.VALUE _ANTIALIASING_ O N );
        gfx.setColor(Color.RED);
        gfx.drawOval (0,0,100,100);
        gfx.dispose ();
    }
    }
    frame.getRootPane().setContent (panel);

    frame.setVisible(true );

```

[21]

2.5. RIA Aplikace

Hlavním zaměřením platformy JavaFX by měla být tvorba RIA aplikací. Ta je v dnešní době v oblasti zájmu mnoha firem a proto existuje mnoho konkurentů pro JavaFX. Mezi ty hlavní patří Microsoft Silverlight, AJAX a Adobe Flash. Tito konkurenti budou dále krátce představeni. Výhodou JavaFX oproti konkurenci je právě skutečnost, že není omezená jen jednou platformou, ale je možné aplikace vyvíjet i pro mobilní zařízení nebo televize.

RIA aplikace by měly hlavně zaujmout uživatele. K tomu používají množství grafických prvků, animací, efektů, zvuků nebo videí. Odtud název

RIA, tedy bohaté internetové aplikace. Po nainstalování pluginu umožní internetový prohlížeč spouštění těchto aplikací. [7]

2.6. Popis konkurenčních produktů pro JavaFX

2.6.1. Adobe Flash

Jedním z největších konkurentů na poli tvorby bohatých internetových aplikací je grafický vektorový program Flash od firmy Adobe. Jeho hlavní výhodou je právě uchování souborů ve vektorovém formátu, což umožňuje následnou velice malou velikost výsledného souboru.

Flash má v sobě implementovaný také vlastní programovací jazyk ActionScript, díky kterému lze celou vytvořenou aplikaci uvést do pohybu a interaktivně s ní operovat.

Dvěma základními výstupními formáty Flashe jsou formáty typu swf a exe. Formát swf je určen ke zveřejňování na internetu. Proto má co nejmenší možnou velikost a vyžaduje nainstalování přehrávače Adobe Flash Player ve formě pluginu například ActiveX pro Internet Explorer. Formát exe pak slouží ke spouštění přímo pod Windows a nese v sobě zabudovaný flash player, díky čemuž není nutný další přehrávač. Proto se takovému formátu říká projektor. Zabudování flash playeru s sebou nese navýšení velikosti souboru.

Vývoj Flashe se datuje od roku 1996, kdy společnost Macromedia nastupuje směr tvorby nástroje pro vývoj webových aplikací. V roce 2005 je společnost Macromedia zakoupena společností Adobe, která dále pokračuje ve vývoji. Současnou verzí programu je Adobe Flash CS4.

V porovnání s platformou JavaFX je produkt Adobe Flash zaměřen mnohem více na grafickou stránku tvorby bohatých internetových aplikací, část

programovací zůstává více v pozadí. Právě tento problém se snaží platforma JavaFX řešit a vyrovnat poměr programování a tvorby grafiky.[22]

2.6.2 Microsoft Silverlight

Microsoft Silverlight je moderní technologie pro tvorbu RIA aplikací, která umožňuje po instalaci plug-inu interaktivitu aplikace spouštění ve velkém množství současných internetových prohlížečů, například Internet Explorer nebo Opera. Takto funkční je pro platformy Windows a Mac OS X. Pro Linux je pod názvem Moonlight vyvíjen společností Novell. [14]

2.6.3 AJAX

Zkratka AJAX znamená asynchronní java script a XML. To v praxi znamená, že AJAX je souborem několika technologií umožňujících tvorbu RIA aplikací. Konkrétně se jedná o tyto technologie. HTML a XHTML, které slouží k prezentování dat, Java Script a DOM pro dynamické změny těchto dat a XMLHttpRequest, který slouží ke komunikaci s webovým serverem. Hlavní výhodou technologie AJAX je možnost právě této komunikace bez nutnosti znovu načítat stránku. [15]

AJAX má široké využití. K vývoji mnoha aplikací byl využit společností Google. Jako příklad slouží aplikace Google Suggest, tedy takzvaný našeptávač, který pomáhá při vyhledávání v databázi, kdy při napsání části výrazu nabídne našeptávač možnosti, které uživatel vyhledává. Největším problémem při používání aplikací naprogramovaných pomocí technologie AJAX je návyk používat při používání internetového prohlížeče tlačítko Zpět. Jelikož aplikace v AJAX funguje jako samostatný prvek, po zmáčknutí tlačítka

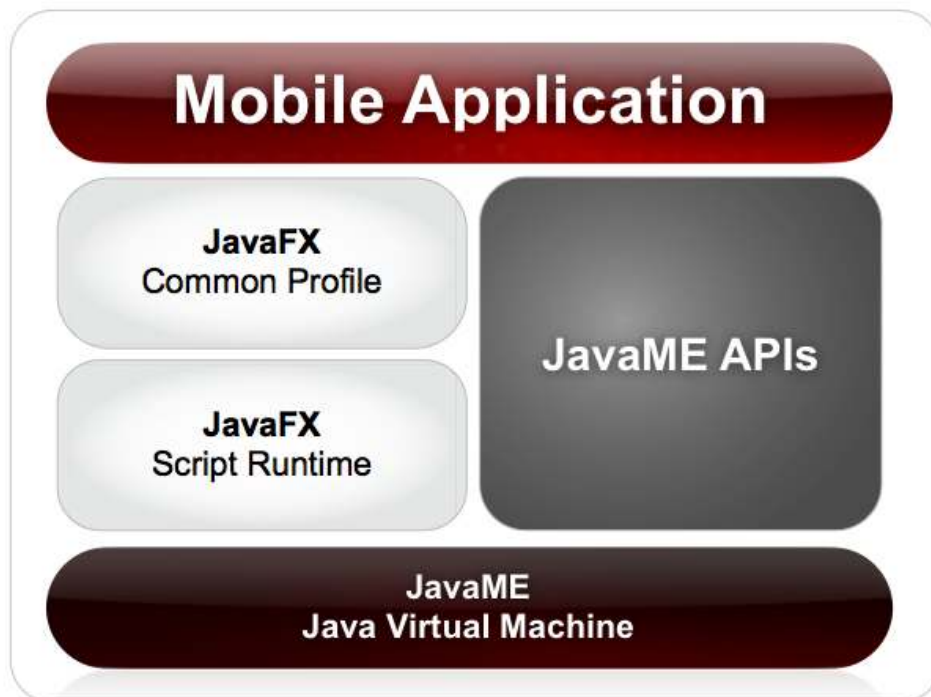
Zpět dochází k načtení předchozí stránky a ztrátě dat, které uživatel sdílel s aplikací.[16]

2.7. JavaFX Mobile

JavaFX Mobile je implementace JavaFX, která slouží k vývoji aplikací, které by měly fungovat na mobilních zařízeních. K psaní těchto aplikací by měl fungovat stejně jako v celé JavaFX skriptovací jazyk JavaFX Script. JavaFX Mobile se stal veřejně přístupnou součástí platformy JavaFX ve verzi 1.1, a od té doby počítá firma Sun s podporou operačních systémů pro mobilní zařízení Google Android a Windows Mobile. Společně s představením JavaFX Mobile byla ohlášena i spolupráce s firmami LG a Sony Ericsson.

V létě 2009 přichází firma Sun s verzí binárního zásobníku JavaFX 1.2 for Windows Mobile OS. Ten po nainstalování na mobilní zařízení umožňuje spuštění aplikací vytvořených v JavaFX. Požadavky této instalace na dané mobilní zařízení jsou operační systém Windows Mobile 6.1, možnost připojení k síti, 32-bitový RISC procesor a alespoň 64 MB paměti RAM. Počínaje verzí NetBeans 6.7.1 IDE jsou aplikace vytvořené v tomto vývojovém prostředí funkční a je možné vyzkoušet jejich funkčnost na emulátoru, který je součástí NetBeans. Firma Sun provedla zkoušky na mobilních zařízeních HTC Diamond a LG Incite. Na nich by tedy JavaFX měla být funkční. [6]

2.8. Architektura JavaFX Mobile



Obrázek2: Zobrazení architektury JavaFx mobile [8]

Stejně jako celá JavaFX funguje i JavaFX na virtuálním stroji Java Virtual Machine. Konkrétně JavaFX Mobile funguje na virtuálním stroji pro Java Micro Edition, který umožňuje spouštění aplikací na mobilních zařízeních. Nad virtuálním strojem pak pracují prostředí pro jazyk JavaFX Script a JavaME APIs, obsahující knihovny pro mobilní zařízení. Dále pak knihovny ze společného profilu JavaFX, které jsou funkční na všech platformách podporujících JavaFX. [9]

2.9. Instalace JavaFX Mobile na mobilní zařízení

Jak bylo uvedeno v kapitole 2.7., pro instalaci JavaFX Mobile je nutné, aby koncové zařízení obsahovalo operační systém Windows Mobile 6.1. Pokud toto mobilní zařízení splňuje, připojíme ho například k počítači, na který jsme si ze stránky <http://www.javafx.com/downloads/all.jsp> stáhli zásobník JavaFX Mobile 1.2 for Windows Mobile. Po připojení zkopírujeme soubor SUN_JAVAFX.CAB na paměťovou kartu mobilního zařízení. Po dvojkliku na tento soubor dojde k nainstalování JavaFX do mobilu. Následně se po otevření prohlížeče programů zobrazí nová ikona JavaFX. Po jejím poklikání se objeví menu, které obsahuje několik ukázkových aplikací v JavaFX, například kalkulačku. [17]

2.10. Distribuce programů v JavaFX Mobile

Pro spuštění aplikace na mobilním zařízení je nutné sbalit aplikaci do dvou souborů.

První souborem je soubor ve formátu JAD. Jedná se o Java Application Descriptor, který poskytne všechny potřebné informace o aplikaci cílovému zařízení.

Druhý soubor je uložen ve formátu JAR, tedy soubor typu Java Archive. Ten obsahuje všechny zkompileované soubory typu Class a všechny zdroje. [18]

Po nahrání těchto dvou souborů do mobilního zařízení jednoduše poklikáme na soubor typu JAD, a spustí se instalace aplikace.

Soubory JAR a JAD nalezneme v podsložce dist hlavní složky vaší aplikace. [18]

3. Technologický a didaktický návrh aplikace

3.1. Technologický návrh

Zadáním práce bylo vytvořit mobilní aplikaci pro výuku cizího jazyka. Z hlediska technologického tedy bylo nutné počítat s jistými omezeními, která přináší mobilní zařízení a to hlavně nefunkčnost knihoven z desktopového profilu, například knihoven Swing controls. Další omezení byla dána technickými možnostmi mobilních zařízení. Proto byla aplikace vyvíjena jako co možná nejmenší. Neobsahuje tedy žádná objemná videa či zvuky.

Cílem této práce je také ukázat možnosti JavaFX mobile. Za tím účelem je použito široké množství komponent, které mají tyto možnosti demonstrovat. Proto je někdy na řešení podobného problému použito jiných řešení, přesto že by jedno splnilo svůj účel.

Koncept aplikace je jednoduchý. Celá aplikace se po spuštění přepne do hlavního menu, kde je možnost výběru mezi možnostmi zahájení výuky, zobrazením textu o autorovi, zobrazením nápovědy a ukončením aplikace.

Po zvolení možnosti zahájení výuky se aplikace přepne na menu zobrazující výběr z několika lekcí, slovníku a závěrečné zkoušky.

Uživatel po zvolení lekce shlédne animaci, která představí téma dané lekce a dále pokračuje. Zobrazí se slovník, který obsahuje slovíčka pro tuto lekci. Po prostudování a zapamatování se po stisknutí tlačítka zobrazí postupně několik úkolů, kde má uživatel možnost procvičit naučená slovíčka. Po procvičení slovíček následuje návrat do menu s výběrem lekcí.

Pokud uživatel zvolí slovník, zobrazí se mu všechna slovíčka, která jsou v aplikaci vyučována.

Když uživatel vybere závěrečné přezkoušení, je mu postupně zobrazeno několik úkolů, jejichž výsledky jsou započítávány a na závěr je uživateli zobrazen graf jeho úspěšnosti.

Jelikož se jedná o aplikaci, která bude provozována na mobilním zařízení, bylo k jejímu ovládní použito klávesnice telefonu. Konkrétně pro přepínání mezi ovládacími zařízeními je využito joysticku a pro potvrzení vybraného ovládacího zařízení klávesa Enter.

Technologický návrh lekcí je koncipován jako několik postupně zobrazovaných skupin. První dvě skupiny obsahují animaci, uvádějící lekci, a text, kde je uvedena slovní zásoba. Další skupiny jsou věnovány úkolům.

Ty jsou řešeny formou obrázku doprovázeného otázkou a nabídkou několika možných odpovědí. Po stisknutí tlačítka zvolené odpovědi se mohou objevit dvě možnosti. Při volbě správné možnosti se objeví tlačítko s nápisem „Right“. Pokud je vybrána špatná možnost, objeví se tlačítko „False“, po jehož stisknutí má uživatel možnost vybrat novou odpověď. Pokud uživatel zvolí správnou odpověď, objeví se nová skupina s dalším úkolem.

Dalším typem úkolu je zvuková ukázka. Uživatel spustí tlačítkem ukázkou a podle ní pak vybírá mezi možnostmi správnou odpověď na otázku zobrazenou u tlačítek přehrávače.

V lekci o číslech je několik jednoduchých početních úloh, jako spočítání obrazců a nebo sčítání čísel.

Posledním typem úkolů je vybrání jednoho ze tří nabídnutých obrázků. Je nutno vybrat ten obrázek, který představuje nejvhodnější odpověď na otázku.

3.2. Požadavky na aplikaci z hlediska didaktického

Po dohodě byl přijat návrh zaměřit aplikaci jen na začátečníky a připravit několik jednoduchých lekcí, které by jim pomohly s naučením praktických začátků anglického jazyka, hlavně s rozšířením slovní zásoby.

V každé lekci byla připravena slovní zásoba čítající okolo deseti slovíček a dále několik cvičení na jejich vyzkoušení v praxi. Celá výuková aplikace se záměrně nezabývá výukou gramatiky, protože by zaprvé implementace do

aplikace byla velmi složitá a zadruhé by byla nutná hlubší znalost angličtiny ze strany autora, nebo větší spolupráce s odborníky v této oblasti.

Při vývoji aplikace se počítalo s alespoň základní znalostí angličtiny, jelikož by nebylo možné vytvořit aplikaci, umožňující výuku pro úplné začátečníky. Proto jsou v aplikaci použity základní otázky, zájmena a věty, u kterých se předpokládá, že je uživatel bude znát.

3.3 Zhodnocení výsledků dotazníků

Pro vyučující i pro studenty byly připraveny dva krátké dotazníky. V prvním byly položeny otázky týkající se směru dalšího vývoje aplikace.

Položené otázky zněly:

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?
2. Jak byste si takovou výukovou aplikaci představovali?
3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?
4. Z čeho byste měli největší strach při používání této aplikace?
5. Jaké by mělo být ovládání aplikace?

Na první otázku odpovídali vyučující vesměs kladně, ale s obavou ohledně složité realizace. Jednak je tento způsob výuky založen na individuální činnosti a ta se pro prostředí školní třídy nehodí. Je zde především problém kázně, kterou by takový způsob výuky narušil. A také vyžaduje určité materiální vybavení (pro každého žáka mobilní telefon s podporou aplikace). Takovými možnostmi zatím rozhodně školství nedisponuje. Proto jediným možným způsobem šíření aplikace je její poskytnutí ke stažení pro uživatele, kteří mají vhodné vybavení ke spuštění.

Studenti byli možnostmi výuky na mobilních telefonech nadšeni. Považovali to za celkem moderní způsob výuky a v dnešní době moderních

technologií za logický posun způsobu vyučování, se kterým se oni dokážou ztotožnit.

Největší obavy tedy představovala hlavně nutnost vlastnit mobilní zařízení, které podporuje JavaFX Mobile.

Na otázku, jak by si uživatelé takovou aplikaci představovali, byly odpovědi většinou dost nejasné a neurčité. Většina dotázaných uvedla, že by aplikace měla obsahovat možnost prověření naučených poznatků. K tomu bylo přihlédnuto při tvorbě aplikace a byl přidán i test, který zobrazí vyhodnocení úspěšnosti uživatele, a je tedy možné zjistit, jak nám aplikace pomohla.

Studentům záleželo hlavně na atraktivitě. Bylo tedy nutné vytvořit aplikaci, která bude pokud možno, interaktivní, s obrázky a zvuky, aby uživatele zaujala.

Vyučující na otázku ohledně upřednostnění výuky na mobilním zařízení před klasickou knihou většinou zohledňovali náklonnost dnešní mládeže k technickým novinkám a také možnost spustit si aplikaci kdekoliv, jelikož mobilní telefon každý nosí většinou pořád s sebou.

Studenti odpovídali podobně. Možnost nést si místo těžké učebnice pouze mobilní telefon se jim zamlouvala. V dnešní době je mládež se svými mobilními telefony tak sžitá, že pro ně představa psaní úkolů do sešitu zní mnohem hrozivěji, než klikání úkolů na mobilním telefonu.

Studenti vesměs nejevili obavy z používání tohoto typu výuky.

Požadavkem na ovládání byla co možná největší jednoduchost, a to jak u studentů, tak u vyučujících.

Ve druhém dotazníku, který byl předložen společně s aplikací k ozkoušení, byly jen tři otázky.

1. Líbila se vám aplikace?
2. Jak hodnotíte ovládání aplikace?

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Na první otázku většina dotázaných odpovídala převážně kladně. Aplikace se jevila jako přehledná a celkem vhodně graficky navržena. Její grafický styl byl hodnocen jako nerušivý, což odpovídalo zaměření na výuku.

I v odpovědích na druhou otázku vyjádřili dotazovaní spokojenost. Ovládání jim přišlo velmi intuitivní a nekomplikované.

Aplikace se zdála být přínosnou a většině zájemců o naučení angličtiny přišla tato možnost procvičování slovíček jako příjemná a moderní varianta výuky.

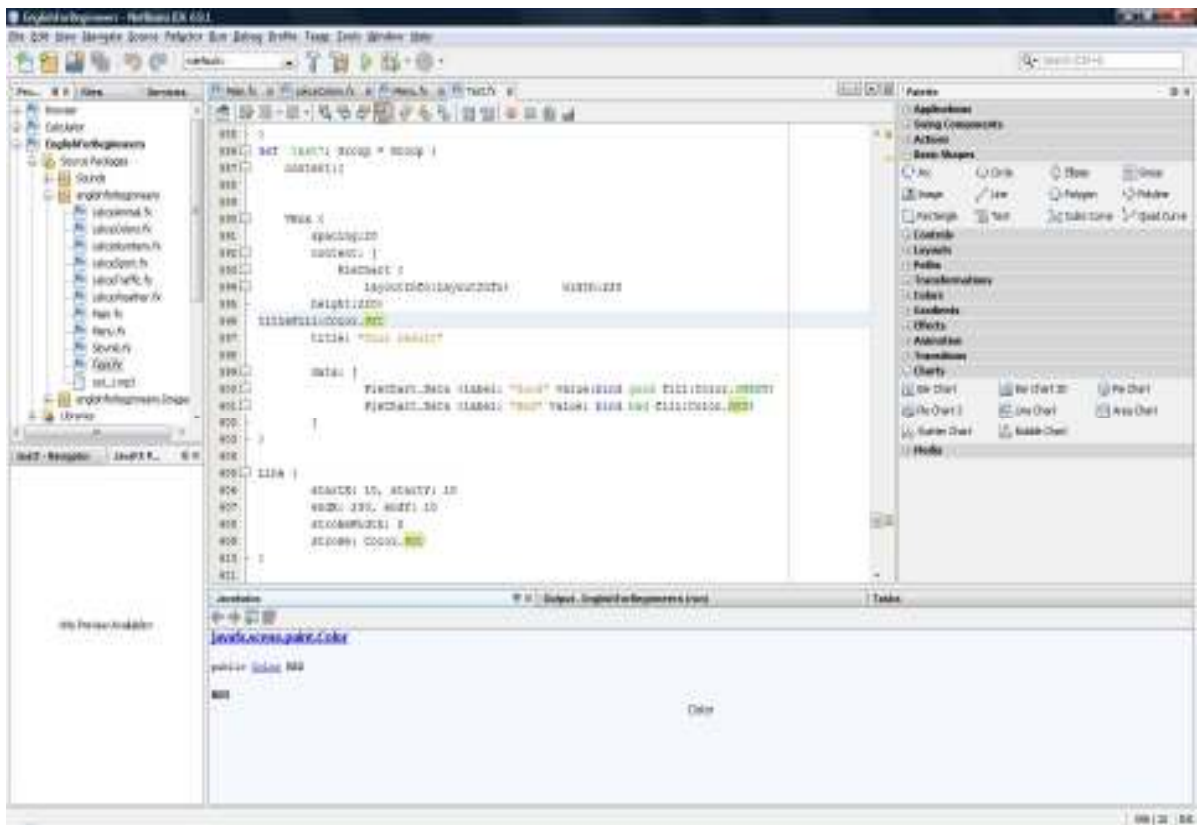
4. Popis vývoje a implementace aplikace

4.1. Vývojové prostředí NetBeans IDE

Vývojové prostředí NetBeans IDE (Integrated Development Environment) je nástroj pro programátory, který jim slouží k psaní kódu, jeho překladu a ladění. Celé vývojové prostředí je napsané v jazyce Java a je založeno na platformě NetBeans. [10]

Celá platforma NetBeans původně vznikla na MFF Univerzity Karlovy, posléze byla odkoupena firmou Sun Microsystems a dále je šířena jako takzvaný OpenSource software. To ve zkratce znamená, že je program šířen s otevřeným zdrojovým kódem, tedy je umožněno uživatelům tento kód v souladu s licenčními podmínkami využívat a dále upravovat.

V létě 2010 vychází verze NetBeans IDE 6.9.1, která plně podporuje JavaFX 1.3.1 a přináší několik novinek. Jednou z nich je JavaFX Composer, což je vizuální editor uživatelských rozhraní pro JavaFX, rozšiřuje palety pro tvary, grafy, barvy a efekty, nebo usnadňuje formátování kódu. Součástí této verze je samozřejmě i emulátor mobilního zařízení, který umožňuje otestování funkčnosti mobilní aplikace. Pro vývoj finální aplikace bylo použito právě verze NetBeans IDE 6.9.1.



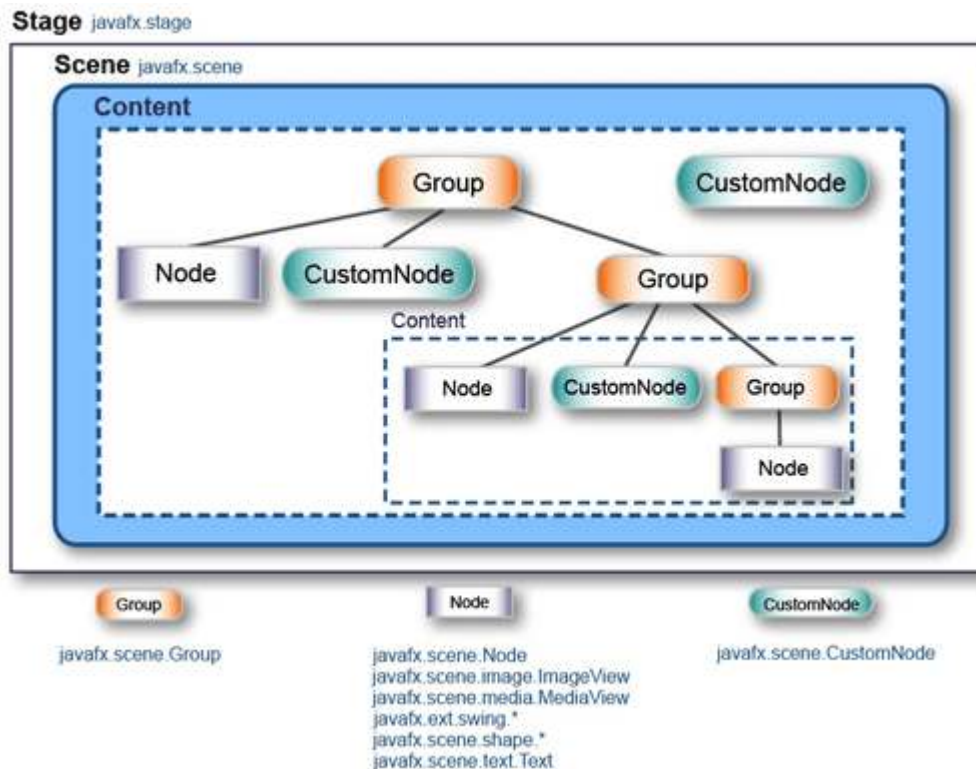
Obrázek 3: Pracovní plocha Netbeans 6.9.1



Obrázek 4: Ukázka emulátoru mobilního zařízení vývojového prostředí NetBeans

4.2. JavaFX Scene Graph

Pro tvorbu aplikace v JavaFX je důležité pochopit, jak je vlastně celá scéna aplikace koncipována. Tedy pochopení JavaFX Scene Graph. Hlavní prvek, který obsahuje veškeré další prvky, se nazývá `Stage`. Jeho součástí je objekt `Scene`. Do něho jsou dále vykreslovány veškeré elementy aplikace, jako například tvary, obrázky nebo ovladače, a to konkrétně do obsahové části zvané `Content []`. Každý prvek, který je objektem scény, se nazývá `Node`. Základní `Node` je pouze jeden prvek, který nemůže mít žádného dědice. Druhou možností je zobrazení skupiny `Group`, která obsahuje několik prvků, které dědí její vlastnosti. Poslední možností je tvorba vlastního prvku pomocí přepsání funkce `create () :Node`. [11]



Obrázek 5: JavaFX Scene Graph [12]

4.5. Popis prvků použitých při tvorbě aplikace

4.5.1. Stage

Základním prvkem aplikace programované v JavaFX je takzvaná `Stage`. Jedná se o plochu, na které se zobrazují veškeré naprogramované prvky. Pro většinu aplikací vystačí jedna `Stage`, i když je možné jich vytvořit i více. Pro použití objektu `Stage` je nutno nainportovat do programu knihovni třídu tímto způsobem `import javafx.stage.Stage;`. V samotné kostře programu pak použijeme k její implementaci definici proměnné `var stg = Stage {}`. A právě mezi závorky jsou dopisovány prvky, které má aplikace obsahovat (například obrázky, animace, text a podobně, vše ve scéně, jak je vysvětleno dále).

Samotná Stage nabízí několik možností úpravy. Jedním z nich je možnost nastavení titulku nastavením proměnné `title`: "Titulek".

Další důležitou možností je určení velikosti `Stage` {}, pomocí proměnných `width` a `height`, kdy je velikost reprezentována počtem pixelů. Pro použití v mobilních telefonech platí, že je `Stage` {} nastavena přes celou obrazovku použitého zařízení. Pro ozkoušení na mobilním emulátoru, který je součástí vývojového prostředí NetBeans je velikost obrazovky nastavena na `width:240` a `height:320`.

Určení výchozí pozice `Stage` {} je dáno nastavením proměnných `x` a `y`. Tedy například `x:15` `y:10` znamená, že `Stage` {} se objeví 10 pixelů od horního okraje a 15 pixelů od pravé strany obrazovky.

Dalšími možnými nastaveními jsou pak `style` a `opacity`, kde nastavíme vzhled okna aplikace `style: StageStyle.DECORATED` zobrazuje aplikaci s rámečkem, kde vidíme titulek, možnost maximalizace, ikonu, nebo křížek pro zavření aplikace. Dále nabízenými možnostmi stylu `Stage` {} jsou `UNDECORATED` a `TRANSPARENT`, kdy první ukazuje aplikaci bez rámečku, a druhý s rámečkem průhledným. Proměnná `Opacity` určuje průhlednost `Stage` {}, kdy 1.0 značí neprůhlednost a 0.0 absolutní průhlednost. Nastavením těchto proměnných ovšem nezměníte vzhled `Stage` {} při spuštění v mobilním telefonu.

Pro vypnutí `Stage` {} je možno použít buď klasického kříže v pravém horním rohu rámečku a nebo, pokud není rámeček zobrazen nebo používáme aplikaci na mobilním telefonu, je připravena funkce `close()`. Proto přidání funkce `Stage.close()` například tlačítka a následné poklikání na tlačítko zapříčiní vypnutí `Stage`. [19]

4.5.2. Scene

Pro vkládání prvků do `Stage{}` je v JavaFX určen objekt `Scene{}`. Pro jeho využití je nutné naimportovat knihovni třídu `import javafx.scene.Scene ;`. `Scene{}` má několik proměnných, z nichž nejdůležitější je `Content: []`. Jedná se vlastně o obsah scény, do těchto hranatých závorek jsou vkládány všechny prvky, které mají být zobrazeny v aplikaci. Dalšími proměnnými jsou podobně jako u `Stage` `x`, `y`, `width` a `height`, fungující obdobně. Krom těchto proměnných patří mezi důležité ještě proměnná `fill`. Ta slouží k vyplnění dané scény buďto barvou nebo přechodem barev. Pro překrytí barvou je nutné naimportovat knihovni třídu `import javafx.scene.paint.Color ;` a definovat proměnnou například jako `fill: Color.BLACK`. Takto nastavená proměnná vyplní scénu černou barvou. Pokud chceme použít překrytí scény přechodem barev, máme možnost výběru mezi lineárním a kruhovým přechodem. Pro přechod lineární musí být použita a naimportována knihovni třída `import javafx.scene.paint.LinearGradient ;` a `import javafx.scene.paint.Stop ;`. Proměnná `fill` je pak definována jako `fill: LinearGradient{}`.

Zde jsou definovány proměnné `startX` a `startY`, které označují startovní pozici pro přechod. Dalšími proměnnými jsou `endX` a `endY`. Ty označují konečnou pozici překrytí. Takto je definován směr přechodu. Poslední proměnnou je `stops`, která ve hranatých závorkách definuje objekty `Stop{}`, podle počtu použitých barev při překrývání přechodem, z nichž každý má dvě proměnné, `offset`, která určuje pozice na ose, kde začíná daná barva a `color`, určující barvu.

Pro překrytí kruhovým přechodem musí být naimportovány knihovní třídy `import javafx.scene.paint.RadialGradient; a import javafx.scene.paint.Stop ;`. V programu je pak použito zápisu `fill:RadialGradient{}`. Závorky obsahují proměnné `centerX` a `centerY` které určují střed kruhového přechodu. Dále je v závorkách proměnná `radius`, která určuje rádius přechodu a podobně jako při lineárním přechodu proměnnou `stops` s objekty `Stop{}`. [19]

Pro potřeby aplikace byla nastavena velikost `Scene{}` na 240 pixelů krát 320 pixelů. Scéna byla vyplněna lineárním přechodem.

Ukázka zdrojového kódu `Stage{}` a `Scene{}` z aplikace:

```
var stg = Stage {
    title: "English for beginners"
    scene: Scene {
        fill: LinearGradient {
            startX : 1.0
            startY  : 0.1
            endX    : 1.1
            endY    : 1.1
            stops: [
                Stop {
                    color
: Color.BLACK
```



```

offset: 0.0
                                },
                                Stop {
                                    color
: Color.GRAY
offset: 1.0
                                },
                                ]
                                }
width: 240
height: 320
content: [
    mainMenu
]
}

```

4.5.3. Group

Pro snadnější orientaci v programu a lehčí manipulaci s objekty je v JavaFX implementována třída `Group`. Ta umožňuje seskupovat kolekce objektů do skupiny, se kterou je pak nakládáno jako s jednolitým celkem. To umožňuje například změny pozice, různé transformace nebo rotace celé skupiny, aniž by bylo nutné provádět tyto operace pro každý objekt zvlášť.

Pro použití těchto skupin je nutné importovat knihovni třídu `import javafx.scene.Group;`. V programu pak definujeme objekt typu `Group{}`. Například `def group1: Group = Group {}`. Takto definovaný objekt v sobě nese proměnnou `Content[]`, do které jsou vepsány veškeré objekty, které skupina obsahuje.

V programu, který je zadán jako téma bakalářské práce, bylo využito možnosti smazání skupiny a její znovu vyvolání. Jelikož program nutně potřebuje časté měnění objektů na scéně, bylo toto řešení použito jako ideální. Byla vytvořena funkce:

```
function showGroup(group : Group) : Void {  
  
    delete stg.scene.content;  
  
    insert group into stg.scene.content;  
  
}
```

Tato funkce vždy vymazala skupinu, kterou obsahovala stávající scéna a nahrála do ní skupinu novou. Proto přiřazením této funkce tlačítku, které je definováno v obsahu skupiny, bylo dosaženo plynulého přechodu mezi různými skupinami bez nutnosti tvořit nové scény. Bohužel rozsáhlost programu s sebou nese velké množství těchto skupin. [19]

4.5.4. Objekty scény

Dostáváme se k použitým objektům scény, které vlastně určují, co najdeme na scéně, zda obrázky, tvary, text, animace nebo něco jiného.

4.5.4.1. Image a ImageView

Pro použití obrázků ve scéně musí být nainportovány knihovny

```
import javafx.scene.image.ImageView; a import
javafx.scene.image.Image; . Pro nahrání obrázku slouží objekt
Image{ }. Ten ve své proměnné url nese cestu k obrázku uloženou
v datovém typu String. Pro potřeby aplikace je využito uložení obrázku přímo
v adresáři programu. Cesta je pak udávána tímto způsobem :
```

```
url: "{__DIR__}britishflag.png" .
```

Mezi proměnné objektu `Image{ }` patří klasické `width` a `height`, udávající šířku a výšku v pixelech. Dalšími proměnnými u obrázku jsou `preserveRatio`, typu `boolean`, které umožňuje nebo zakazuje zachování poměru stran při změně velikosti obrázku. Proměnná `smooth` nastavená na `true` umožňuje použití algoritmu, které při změně velikosti zachová hladkost obrazu.

Pro potřeby mobilní aplikace byli použity formáty obrázků `png` a `jpg`. Formát `gif` byl ozkoušen jako nefunkční, i když při spuštění jako nemobilní aplikace byl v JavaFX podporován.

Pokud chceme obrázek na scéně zobrazit, je v JavaFx použit objekt `ImageView{ }`, kdy tento objekt v sobě obsahuje objekt typu `Image` . Aplikuje se následující cestou :

```
ImageView {
```

```
image: Image { url: "{__DIR__}britishflag.png"
}

}
```

Základními proměnnými jsou `x` a `y`, které určují vzdálenost od levého horního rohu danou v pixelech. Další proměnné `fitWidth` a `fitHeight` jsou vlastně ekvivalentní k k proměnným `width` a `height` u objektu `Image{}`, tedy určují velikost obrázku v pixelech. Podobně i `preserveRatio` rozhoduje o zachování poměru stran a `smooth` o vyhlazení obrázku. Poslední proměnnou u objektu `ImageView{}` je `viewport`, která je typu obdélník a jedná se vlastně o zobrazení jen obdélníkového výřezu z obrázku, kdy je v proměnných `minX` a `minY` určena vzdálenost levého horního rohu obrázku k levému hornímu rohu výřezu na osách `x` a `y`. V proměnných `width` a `height` je pak určena velikost výřezu. [19]

4.5.4.2. Tvary

JavaFX umožňuje vkládání různých tvarů, jako jsou čtverec, kruh nebo elipsa. Také umožňuje přizpůsobování jejich vlastností jako je velikost nebo barva. Všechny tyto tvary obsahuje balíček knihovnických tříd `javafx.scene.shape`. Ten obsahuje proměnné společné pro všechny tvary v kolekci `shapes`. Jsou to `fill`, která určuje barvu nebo přechod výplně, a `stroke`, která určuje barvu ohraničení. Z dále použitých v aplikaci je uvedena `strokeWidth`, která udává velikost ohraničení v pixelech.

4.5.4.2.1. Čáry

Základním objektem třídy `Shapes` je čára `Line{}`. Importuje se `import javafx.scene.shape.Line;`. Má jen čtyři základní proměnné, `startX` a `startY`, které určují pozici začátku čáry, a dále `endX` a `endY`, určující souřadnice konce. Barva čáry je určena proměnnou `stroke`. Další možnou proměnnou je `smooth`, která při nastavení na `true` vyhladí čáru, což se projeví u čar, které nejsou vedeny svisle nebo vodorovně. [19]

Příklad použití čáry v aplikaci:

```
Line {  
  
    startX: 10, startY: 10  
  
    endX: 230, endY: 10  
  
    strokeWidth: 2  
  
    stroke: Color.RED  
  
}
```

4.5.4.2.2. Obdélníky

Obdélníky se dají používat po importaci knihovny třídy `javafx.scene.shape.Rectangle;`. Objekt obdélník je v JavaFX označen jako `Rectangle{}`. Jeho pozice je určena proměnnými `x` a `y`, jeho velikost pak proměnnými `width` a `height`. Speciálními proměnnými jsou pak `arcwidth` a `archeight`, které určují průměr zakřivení rohových oblouků shora a ze strany. Tím dosáhneme obdélníku se zaoblenými rohy. [19]

Příklad použití obdélníku:

```
Rectangle {  
    x: 10, y: 40  
  
    arcWidth: 10  
    arcHeight: 5  
  
    width: 220, height: 40  
  
    fill: Color.WHITESMOKE  
}
```

4.5.4.2.3. Kruhy

Kružnice jsou v JavaFX tvořeny objektem `Circle{}`. Do aplikace je nutné nainportovat třídu z knihovny `Shape` `javafx.scene.shape.Circle;`. Proměnné, které jsou u objektu typu `Circle{}` jsou `centerX`, určující v pixelech umístění středu na ose `x`, `centerY`, určující pozici na ose `Y`, a `radius`, který určuje poloměr zadaného kruhu. [19]

Příklad použití kruhu:

```
Circle {  
    centerX: 170, centerY: 70  
  
    radius: 30  
  
    fill: Color.RED  
}
```

4.5.4.2.4. Elipsa

Objekt `Ellipse{}` slouží k vykreslení elipsy do scény. Po naimportování třídy `javafx.scene.shape.Ellipse`; můžeme nastavit podobně jako u kruhu pozici středu v proměnných `centerX` a `centerY`. V proměnné `radiusX` pak nastavíme poloměr horizontální a v proměnné `radiusY` poloměr vertikální. [19]

Příklad použití v programu:

```
ellipseColor =Ellipse {  
  
    centerX: 140, centerY: 70  
  
    radiusX: 90, radiusY: 70  
  
    fill: Color.WHITESMOKE  
  
}
```

4.5.4.2.5. Mnohoúhelník

Posledním objektem z knihovny třídy `Shape`, který je použit v aplikaci je mnohoúhelník `Polygon{}`. Ten je importován třídou `javafx.scene.shape.Polygon`; Mnohoúhelník má jen jednu proměnnou `Points{}`, která je typu `Number[]` a obsahuje sdružené dvojice čísel, reprezentující souřadnice vrcholů na ose `x` a `y`. [19]

Příklad použití v programu:

```
Polygon {  
  
    points: [ 70,60, 100,30, 130,60,115,90,85,90 ]  
  
}
```

```
fill: Color.YELLOW

    strokeWidth:1

    stroke:Color.GREEN

}
```

4.5.4.2.6. Text

Text je v JavaFX chápán jako tvar. Proto ho najdeme v sekci BasicShapes. Je importován knihovně třídou `javafx.scene.text.Text`. Jelikož se jedná o tvar, je možné Textu nastavit stejné proměnné jako pro další tvary, tedy `fill` a `stroke`, a s nimi související proměnné.

Proměnnými souvisejícími přímo s textem jsou `content`, `x`, `y`, `textOrigin`, `textAlignment` a `wrappingWidth`.

Proměnná `content[]` obsahuje text, který má být zobrazen.

Proměnné `x` a `y` znázorňují souřadnice začátku textového pole. Pro umístění textu na řádek, nad nebo pod něj, slouží proměnná `textOrigin`, kterou je nutno naimportovat, `javafx.scene.text.TextOrigin`. Existují tři možná nastavení. `BASELINE`, pro zobrazení na řádek, `TOP`, pro zobrazení pod řádkem a `BOTTOM`, které text zobrazí nad řádkem. Proměnná se nastavuje takto `textOrigin: TextOrigin.BASELINE`.

Pokud je potřeba text zarovnat, poskytuje JavaFX tuto možnost pomocí naimportování knihovně třídy `javafx.scene.text.TextAlignment` a její proměnné `textAlignment`. Ta je defaultně nastavená na `LEFT`, ale

lze ji zaměnit za `CENTER` nebo `RIGHT`. Proměnná se inicializuje takto
`textAlignment: TextAlignment.CENTER` .

Pro text víceřádkový existují dvě možnosti. Je možno doplnit do textu znak pro odřádkování `\n` . Druhou možností je nastavení proměnné `wrappingWidth`, která určí vzdálenost v pixelech od levého okraje textu, po které dojde k odřádkování. Tato možnost se zapisuje takto
`wrappingWidth: 100` , kdy 100 je vzdálenost pro odřádkování v pixelech. [19]

Dekorace textu

Pro potřeby dekorace textu obsahuje třída `Text` proměnné `underline` a `strikethrough`. Obě jsou nastavená na `false`, jejich nastavením na `true` dostaneme text podtržený, respektive přeškrtnutý.

Fonty

JavaFX poskytuje možnost použití různých fontů. Pro jejich použití musí být nainportována knihovná třída `javafx.scene.font.Font`; Pro použití fontu nastavím proměnnou `font: Font{}` . Objekt `Font{}` v sobě nese informace o fontu. Základní proměnnou je `name`, tedy název fontu. Pro určení stylu fontu je možné nastavit hodnotu `FontWeight` na různé hodnoty (tučné, kurzíva,...). Dalším možným nastavením je `size`, které udává velikost fontu. Ta je defaultně nastavená na velikost 12. Pro potřeby aplikace bylo využito speciálního fontu `Amble`, který umožňuje zobrazování diakritiky na mobilních zařízeních. [19]

Ukázka použití textu v programu:

```
Text {  
  
    textAlignment: TextAlignment.RIGHT
```

```
font: Font {  
    name: "Amble "  
    size: 16  
}  
x: 15, y: 175  
content:  
    "What color is it?"  
}
```

4.5.5.3. Ovládací prvky

Celá aplikace naprogramovaná v JavaFX Mobile je ovládaná pomocí ovládacích prvků z balíčku knihovných tříd `javafx.scene.control`. Tento balíček je variantou pro Swing ovladače, které se používají v programovacím jazyce Java a jsou i součástí JavaFX, ale nejsou kompatibilní s mobilními zařízeními a proto JavaFX poskytuje možnost využití balíčku `Control`. Při vývoji aplikace bylo použito největší možné množství druhů ovládacích prvků, aby bylo demonstrováno jejich použití v JavaFX.

4.5.5.3.1. Button

Základním prvkem, který slouží k ovládní aplikace je objekt `Button{}`, nainportovatelný knihovná třída `javafx.scene.control.Button`. Jedná se o klasické tlačítko, jež spouští určitou akci, vyvolanou uživatelem například stisknutím. Vzhled tlačítka je určen pomocí proměnných třídy `Button`. Jsou to proměnná `text`, která zobrazí na tlačítku nápis a `visible`,

kteřá určuje viditelnost tlačítka. Pro umístění tlačítka dle osy x a y se používá proměnných `translateX` a `translateY`, které posunou tlačítko na danou pozici. Proměnná `action` udává funkci, která se provede po provedení dané akce uživatelem. Jedná se tedy o hlavní funkci tlačítka. Dalšími proměnnými, které je možno při tvorbě tlačítka použít, jsou `strong` a `disable`. Obě jsou typu `Boolean`. Proměnná `strong` tlačítko zvýrazní a proměnná `disable` znemožní akci s tlačítkem. [19]

V aplikaci jsou přidány tlačítku akce, které spouští určité funkce po stisku klávesy.

Ukázka zdrojového kódu pro tlačítko, použitého v aplikaci:

```
Button {  
  
    translateX:10  
  
    translateY:270  
  
    font:Font {name : "Amble"}  
  
    text: "Start animace"  
  
    onKeyPressed: function(e: KeyEvent): Void {  
  
        if (e.code == KeyCode.VK_ENTER)  
  
            {  
  
                timelineColor.play();  
  
            }  
  
    }  
  
}
```

```
}
```

4.5.5.3.2. Toggle Button

Pro potřeby ovládání aplikace nabízí JavaFX možnost ovládání pomocí takzvaného Toggle Button. Jedná se o přepínací tlačítko se dvěma stavy, buď vybráno nebo nevybráno. Využití v aplikaci je při výběru správné odpovědi z několika možností.

Pro jeho použití se importuje knihovna `javafx.controls.ToggleButton`. V aplikaci se pak `ToggleButton` použije jako objekt `ToggleButton{}`. Jeho dvěma základními proměnnými jsou `selected`, která označuje, zda je `ToggleButton` vybrán a která je implicitně nastavená na `false`, a `toggleGroup`, ve které je název skupiny, do které může `ToggleButton` patřit. [19]

Příklad použití v programu:

```
ToggleButton  
  
{  
  
    translateX:20  
  
    translateY:220  
  
    text: "Cat"
```

```

        onKeyPressed: function(e: KeyEvent): Void
    {
        if (e.code == KeyCode.VK_ENTER)
            {
                buttonAnimal2.visible=true;
                buttonAnimal2.text =
"Right"
            }
        }
    }

```

4.5.5.3.3. Check Box

Dalším ovládacím prvkem, použitým v aplikaci je takzvaný CheckBox. Využívá se v momentě, kdy je uživateli nabídnut výběr z několika možností. Uživatel pak zaškrtně zvolený CheckBox. V aplikaci je použit pro výběr správné odpovědi na otázku.

Pro použití CheckBoxu je nutné naimportovat knihovnu `javafx.controls.CheckBox;`. Objekt `CheckBox{ }` má několik proměnných. Ta hlavní se nazývá `selected`. Je nastavena defaultně na `false`. Tato proměnná značí, zda je tento prvek vybrán nebo ne. Dalšími proměnnými jsou `allowTriState` a `defined`, které umožňují více stavů u

objektu `CheckBox {}`. Funkcí těchto proměnných v aplikaci nebylo využito.
[19]

V ukázce je zobrazen `CheckBox`, který po zaškrtnutí ukáže další skupinu scény.

Ukázka zdrojového kódu s využitím `CheckBoxu` v aplikaci:

```
CheckBox {  
  
    translateX:100 translateY: 220  
  
    text: "Hammster?"  
  
    allowTriState: false  
  
    selected: bind box1Selected  
  
    var box1Selected = false  
onKeyPressed:function(e: KeyEvent): Void {  
  
    if (e.code == KeyCode.VK_ENTER)  
  
        {  
  
            showGroup(vyukaAnimal2);  
  
        }  
  
    }  
  
}
```



Obrázek 6: Ukázka ovládacího prvku `CheckBox { }`

4.5.5.3.4. ChoiceBox

Pro zobrazení menu pro výběr lekcí bylo využito ovládacího prvku `ChoiceBox { }`. Pro jeho použití se importuje knihovná třída `javafx.scene.control.ChoiceBox`. Ovládací prvek `ChoiceBox { }` má několik proměnných. Pro zobrazení položek menu je tu proměnná typu pole `items : []`, do které jsou zapsány všechny položky `ChoiceBoxu`, které mají být zobrazeny. Prvky jsou postupně indexovány čísly od nuly vzestupně.

Další proměnnou je `selectedIndex`, která obsahuje index vybrané položky. Proměnná `selectedItem` pak obsahuje vybranou položku.

Pro realizaci funkčního menu je použito funkce `bindingu`, díky které je při zvolení položky z `ChoiceBoxu` zobrazena vždy správná nabídka lekce.

Ukázka zdrojového kódu s použitím ovládacího prvku `ChoiceBox{}`:

```
cb = ChoiceBox {  
  
    translateX:30  
  
    translateY:180  
  
    items: ['Lekce Animal','Lekce Colors','Lekce  
Sport','Lekce Numbers','Lekce Traffic','Lekce  
Weather']  
  
}  
  
ImageView {  
  
    y:35  
  
    x: 50  
  
    image: bind images[cb.selectedIndex]  
  
}
```




Obrázek 7: Ukázka využití prvku `ChoiceBox{ }`

4.5.5.3.5. `ListView`

Pro možnost výběru z několika možností je v JavaFX implementován i objekt `ListView{ }`. Ten umožňuje výběr z daného seznamu prvků. V aplikaci je využito objektu `ListView{ }` k výběru ze seznamu obrázků.

Použití v aplikaci je možné po importování knihovny třídy `javafx.scene.control.ListView`. Použitými proměnnými jsou `item[]`, což je seznam prvků v aplikaci. Další možností volby je proměnná `vertical`, která udává orientaci seznamu. Ta je implicitně nastavena na vertikální. Pro další použití prvků v seznamu slouží proměnné `selectedIndex`, která udává index prvku v seznamu, respektive

selectedItem, obsahující přímo prvek ze seznamu. Indexování prvků v seznamu se počítá od nuly. [19]

Ukázka zdrojového kódu aplikace s prvkem `ListView{}:`

```
var lv:ListView;

        lv = ListView {

                layoutInfo:LayoutInfo{
width:100

                height:80}

        items: ['Image 1','Image 2','Image 3']

        onKeyPressed:function(e: KeyEvent): Void {

                if (e.code == KeyCode.VK_ENTER)

                        {

if (lv.selectedIndex==1 )

{

                bad = bad +1; showGroup(test6);} else if

(lv.selectedIndex==2) {

                bad = bad +1; showGroup(test6); }

else if (lv.selectedIndex==0){

                good=good+1;
```

```
showGroup (test6);  
  
}  
  
}  
  
}
```



Obrázek8: Ukázka ovládacího prvku `listView{ }`

4.5.6. Animace

Pro prezentaci možností JavaFX Mobile bylo v aplikaci využito i animací. Každá lekce je uvedena krátkým animovaným klipem.

Pokud má aplikace obsahovat animace, je nutné nainportovat knihovni třídu `javafx.scene.animation.Timeline`. Ta nám pak umožní použít časovou osu `Timeline{}`. Její hlavní proměnnou je `keyFrames[]`. Ta obsahuje objekty typu `keyFrame`, tedy klíčový snímek. Tento objekt obsahuje informace o tom, jaká akce se za jaké časové období má vykonat.

Ke spuštění animace slouží příkaz `timeline.play()`; a k vypnutí `timeline.stop()`; . Pro potřeby aplikace, která vyžaduje znovupřehrávání animací, pak bylo využito příkazu `timeline.playFromStart()`; .[19]

Animace je použita v aplikaci ve formě ubíhajícího textu, který mění svoji pozici vůči ose y, po uplynutí určité doby mizí, objevuje se text nový, několik objektů scény mizí a zobrazí se tlačítko umožňující postup dále do lekce.

Ukázka zdrojového kódu animace:

```
var timelineAnimal = Timeline {  
  
    keyFrames: [  
  
        at (10s){ellipseAnimal.visible  
=> false}
```

```

        at (10s){polylineAnimal.visible
=> false}

        at (5s) {textAnimal.translateY=>-100}

        at (10s)
{textAnimal2.translateY=>-200}

        at (5s){textAnimal.visible =>
false}

        at (4s){textAnimal2.visible =>
true}

        at (10s){textAnimal2.visible =>
false}

        at (10s){buttonAnimal.visible =>
true}

        ];

    }

    timelineAnimal.playFromStart();

```

4.5.7. Media

Další ukázkou možností platformy JavaFX Mobile je možnost přehrávání médií. Pro již výše zmíněné zohlednění velikosti aplikace nebylo použito přehrávání videí, ale jen přehrávání zvuku. Toho je využito ve výukových lekcích pro ukázky, ke kterým jsou položeny otázky.

Pro používání objektu `Media {}` je nutný import knihovny `javafx.scene.media.Media;`

Hlavní proměnnou, která je inicializována u objektu `Media {}`, je proměnná `source`. V té je zadána adresa přehrávaného souboru. Na mobilních zařízeních je zatím podporováno přehrávání zvuků ve formátu MP3. [19]

Ukázka použití objektu `Media {}` v programu:

```
media: Media { source:
"{__DIR__}Sounds/cat_1.mp3" }
```

4.5.7.1. MediaPlayer

K přehrávání médií slouží v JavaFX objekt `MediaPlayer {}`. Do programu se importuje jako knihovní třída `javafx.scene.media.MediaPlayer;`

Jeho součástí je objekt typu `Media {}`, který udává adresu přehrávaného souboru, jak bylo uvedeno výše.

Objekt typu `MediaPlayer {}` má několik funkcí, z nichž hlavní jsou `start()`, `pause()` a `stop()`. Ty jsou v aplikaci přiřazeny tlačítkům, s jejichž pomocí je řízeno přehrávání souborů. [19]

Ukázka zdrojového kódu s použitím objektu `MediaPlayer {}`:

```
var mediaPlayerAnimal = MediaPlayer {

media: Media { source:
"{__DIR__}Sounds/cat_1.mp3" }
```

```
}  
  
mediaPlayerAnimal.play();  
  
mediaPlayerAnimal.stop();
```

4.5.8. Layouts

Pro rozdělení prvků scény je možné využít balíčku knihovnic tříd `javafx.scene.layout.*`. S jejich pomocí je určena pozice a velikost daných prvků. Možnými zobrazeními jsou `HBox`, `VBox`, `Flow`, `Panel`, `Stack` a `Tile`.

`Hbox` organizuje prvky tak, že jsou zobrazovány horizontálně vedle sebe. `VBox` zobrazí prvky scény pod sebe. `Layout Flow` umožňuje nastavení počtu prvků v řádku a sloupci, kdy jsou prvky řazeny postupně řádku po řádce. S pomocí `layout Panel` je možné nastavit pozice prvků individuálně pomocí proměnné `onLayout`. `Stack` zobrazí prvky scény na sebe. A posledním `layoutem` je `Tile`, který poskládá prvky do stejně velkých dlaždic.

V programu je nejvíce využito objektu `VBox{}`. Ten je možné použít po nainportování knihovnic třídy `javafx.scene.layout.VBox`. Proměnné využití u tohoto objektu jsou `spacing`, která označuje vzdálenost mezi prvky, které má objekt `VBox{}`, a `padding`, který označuje volný prostor okolo zobrazovaného prvku. Pozice `VBoxu` jsou určeny proměnnými `layoutX` a `layoutY`. Pro zarovnání prvků slouží proměnné `hPos` a `vPos`.
[19]

Ukázka zdrojového kódu:

```
VBox {  
  
    layoutX:10
```

```
spacing:25

content: [

    lv = ListView {

        ...

    }

    Button {

        ...

    }

    ImageView { ...

    }

]

}
```

4.5.9. Charts

Počínaje verzí 1.2 umožňuje JavaFX využití grafů. V aplikaci bylo grafu využito k zobrazení výsledků závěrečného textu. JavaFX nabízí několik typů grafů. Jedná se o dvě základní skupiny. Jedna, takzvaná PieCharts obsahuje grafy typu PieChart a PieChart3D, tedy takzvané koláčové grafy, které graficky znázorňují poměr mezi několika hodnotami. Právě grafu PieChart bylo v aplikaci využito. Druhou skupinou jsou takzvané XYChart, v níž jsou grafy,

keré zobrazují výsledky na osách. Mezi ně patří BarChart, BubbleChart, AreaChart, LineChart a ScatterChart.

Grafu PieChart bylo využito pro zobrazení poměru správných a špatných odpovědí v části aplikace, která má otestovat znalosti uživatele. Správné odpovědi jsou vyznačeny zelenou, špatné červenou barvou.

Pro implementaci grafů je nutné naimportovat knihovní třídy z balíčku `javafx.scene.chart.*`; . Přímou v aplikaci se pak volá objekt `PieChart{}`. Hlavní proměnnou grafu je `data[]`. Ta obsahuje objekt `data` s proměnnými `value`, která udává hodnotu, kterou bude tato část grafu zobrazovat, `fill`, která označuje barvu výplně grafu, `stroke`, rozhodující o okraji grafu a `label`, která přidá text ke grafu. [19]

Ukázka použití grafu z aplikace:

```
PieChart {  
  
        layoutInfo:LayoutInfo{  
width:220  
  
        height:220}  
  
        titleFill:Color.RED  
  
        title: "Your result"  
  
        data: [  
  
                PieChart.Data {label: "Good" value:bind  
good fill:Color.GREEN}  
  
                PieChart.Data {label: "Bad" value: bind bad  
fill:Color.RED}
```

```
]
}
```

4.5.10. Alert

V aplikaci bylo nutné upozorňovat uživatele na volbu špatné odpovědi při různých výukových úkolech.

Na řešení tohoto problému bylo využito knihovná třída `Alert`. Ta při chybné akci zobrazí varování, které pak uživatel odklikne a vrátí se do původní scény. [19]

Pro použití je nutné naimportovat knihovná třída `javafx.scene.Alert`;

Ukázka použití v aplikaci:

```
Alert.inform("False",
    "Click OK to close.");
```

4.5.11. Actions

Pro interakci s ovládacími prvky využívá JavaFX balíček knihovných tříd `javafx.scene.input.*`; . Jelikož aplikace má být využívána na mobilním zařízení, používají se k interakci klávesy a joystick telefonu. Proto byly naimportovány knihovná třída `javafx.scene.input.KeyEvent`; a `javafx.scene.input.KeyCode`; . Ty umožní volání funkce `onKeyPressed()`, která po stisknutí určené klávesy vykoná zvolenou akci.

Zde ukázaná akce vykoná po stisknutí klávesy Enter přičtení k proměnné good a zavolá funkci showGroup() .[19]

Příklad použití v programu:

```
onKeyPressed:function(e: KeyEvent): Void {  
    if (e.code == KeyCode.VK_ENTER)  
        {  
            good=good+1;  
            showGroup(test3);  
        }  
}
```

4.5.12. Binding

Pro snadné přiřazování hodnot proměnným umožňuje JavaFX takzvaný binding dat. To znamená, že jakkoli jedna proměnná změní hodnotu, je tato hodnota aktualizována pro proměnné, na které je vázána bindingem. Toho bylo využito například při výměně scén v hlavní Stage, nebo při zobrazování hodnoty u grafu. [19]

Ukázka využití bindingu:

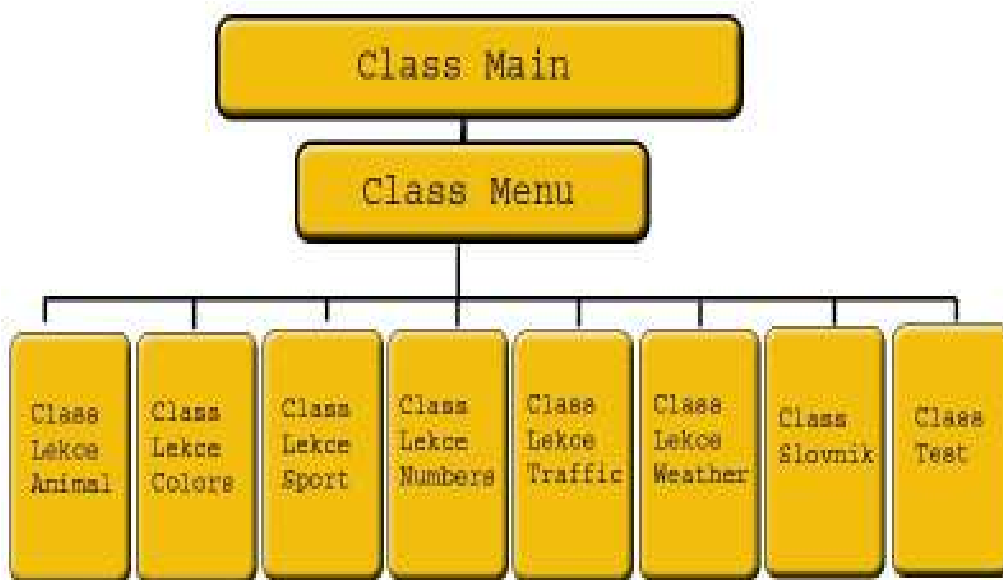
```
PieChart.Data {label: "Good" value:bind good  
fill:Color.GREEN}
```

4.6. Schéma aplikace

Celá aplikace je koncipována do několika tříd. Hlavní třída, která se spustí při zapnutí aplikace je třída Main. Ta nainicializuje hlavní menu a obsahuje nadefinované funkce ke spouštění dalších tříd. Dále obsahuje hlavní Stage, do které jsou zobrazovány veškeré scény aplikace.

Třída menu obsahuje skupinu mainMenu, ve které je tlačítko pro spuštění výuky, dále odkazy na nápovědu, na text o autorovi a tlačítko pro ukončení aplikace. Po spuštění výuky je nabídnuto menu. Zde je pomocí objektu `ChoiceBox{ }` předložen výběr z lekcí, slovníku a testu. Po vybrání je další výuka spuštěna klepnutím tlačítka „Pokračuj“. To volá funkci z třídy main, pro spouštění dalších tříd.

Třídy s výukou, slovník a test obsahují vlastní scénu a několik skupin, které jsou postupně zobrazovány do hlavní Stage pomocí změny scény. Po projití lekce je pak znovu volána funkce ze třídy Main, která nás vrátí do třídy Menu.



Obrázek 9: Schéma tříd aplikace

4.7. Finální verze aplikace

Výsledná aplikace byla vyzkoušena v mobilním emulátoru a měla by být plně funkční na mobilních telefonech s operačním systémem Windows Mobile 6.1.

Finální verze obsahuje šest lekcí zaměřených na rozšíření slovní zásoby. Dále je součástí aplikace slovník a test, který prověří naučené znalosti.

Celá aplikace má přibližně velikost 6 MB, soubor JAR má velikost přibližně 1 MB a soubor JAD 0.3 MB.

Screenshots finální verze:



Obrázek 10: Úvodní nabídka



Obrázek 11: Ukázka úlohy

5. Shrnutí

V této práci bylo hlavním úkolem vytvořit aplikaci pro výuku angličtiny s využitím JavaFX Mobile. To bylo splněno. Aplikace byla vyzkoušena v mobilním emulátoru vývojového prostředí NetBeans 6.9.1, jelikož vyzkoušení na reálném zařízení nebylo možné vzhledem k nedostatečnému materiálnímu vybavení.

Dále byly vypracovány dva krátké dotazníky. V prvním sdělilo několik studentů a vyučujících, jak by si představovali mobilní výukovou aplikaci. Na to pak byl brán ohled při vývoji finální aplikace. Ve druhém dotazníku odpovídali dotázaní na otázky o svojí spokojenosti s programem poté, co jim byl ukázán a bylo jim umožněno ho vyzkoušet. Výsledky obou dotazníků byly zpracovány a uvedeny v bakalářské práci.

Co se týče té části práce, ve které je popsána platforma JavaFX a konkrétně pak JavaFX Mobile, je nutné přihlídnout k tomu, že se jedná o velice mladou platformu, která se zatím rozvíjí. Proto o ní nelze nalézt nijak velké množství informací. Přesto v této práci byla platforma Java FX představena, nastíněny její hlavní cíle a představeni její hlavní konkurenti. Dále byla pak popsána architektura JavaFX Mobile a její současný stav.

Poslední, nejobsáhlejší část práce byla věnována popisu vývoje a implementaci výukové aplikace. Zde byly popsány veškeré komponenty, které byli v aplikaci použity, spolu s ukázkami zdrojových kódů daných komponent.

Při hodnocení JavaFX Mobile jako platformy pro tvorbu mobilních aplikací je tedy nutno uvést, že zásadním problémem zatím zůstává omezená působnost JavaFX Mobile. Ta je v současnosti funkční jen na mobilních

zařizování vybavených operačním systémem Windows Mobile 6.1. Dalším problémem je ne zcela jasná budoucnost platformy JavaFX Mobile. Zatím se zdá, že její vývoj má být upozaděn. [13] Jinak se na této platformě programuje velice příjemně, skriptovací jazyk JavaFX Script je snadné se naučit. Stejně tak se povedlo ideálně spojit práci designéra s prací vývojáře. Proto, pokud dojde k většímu rozšíření a větší podpoře JavaFX Mobile, je možné ji označit za kvalitní platformu pro tvorbu mobilních aplikací.

Reference:

- [1] *JavaFX Overview* [online]. 2010 [cit. 2010-12-01]. JavaFX Develop Expressive Content with the JavaFX Platform Dostupné z WWW: <<http://javafx.com/about/overview/>>.
- [2] *Oracle for Java Developers* [online]. Únor 2010 [cit. 2010-12-01]. Three Reasons Why Your Next Java ME Mobile Application Should Include JavaFX Mobile. Dostupné z WWW:<<http://java.sun.com/developer/technicalArticles/javame/javafxmobile-javame/figure1.gif>>.
- [3] *Java Virtual Machine* [online]. Únor 2010 [cit. 2010-12-01]. Java Virtual Machine from Sun. Dostupné z WWW: <<http://java-virtual-machine.net/sun-java-virtual-machine.html>>.
- [4] *Oracle Technology Network* [online]. 2.6.2009[cit. 2010-11-27]. Deploying Java™ Technology to The Masses: How Sun Deploys The JavaFX™ Runtime. Dostupné z WWW: <<http://www.oracle.com/technetwork/server-storage/ts-4381-159141.pdf>>.
- [5] WEAWER, James, et al. *Pro JavaFX™ Platform : Script, Desktop and Mobile RIA with Java™ Technology*. USA : Apress, 2009. 543 s. ISBN 978-1-4302-1875-3.
- [6] *JavaFX Downloads* [online]. Listopad 2009 [cit. 2010-11-28]. JavaFX™ Mobile 1.2 for Windows Mobile OS Release Notes . Dostupné z WWW: <<http://javafx.com/downloads/mobile/ReleaseNotes.pdf>>.
- [7] *Linux Software* [online]. 30.7.2010 [cit. 2010-11-28]. JavaFX - prostředí pro tvorbu RIA aplikací. Dostupné z WWW: <http://www.linuxsoft.cz/article.php?id_article=1744>
- [8] *Rich Internet Applications (RIA)*[online]. 2009 [cit. 2010-11-25]. Jazoon 2009: JavaFX versus Android. Dostupné z WWW: <<http://www.canoo.com/blog/2009/06/24/jazoon-2009-javafx-vs-android/>>.

- [9] WHITE, Jim. *DevX* [online]. 3.4.2009 [cit. 2010-11-25]. Using JavaFX Mobile to Combat Device UI Fragmentation. Dostupné z WWW: <<http://www.devx.com/Java/Article/41395/1763/page/2>>.
- [10] *ABC Linuxu* [online]. 7.8.2006 [cit. 2010-11-25]. NetBeans. Dostupné z WWW: <<http://www.abclinuxu.cz/software/programovani/ide/netbeans>>.
- [11] CHURCH, Cindy. *JavaFX Highlights* [online]. 2009 [cit. 2010-11-25]. Understanding the JavaFX™ Scene Graph. Dostupné z WWW: <<http://javafx.com/launch/archive/video9.jsp>>.
- [12] Media browser tutorial [online]. 2009 [cit. 2010-11-25]. *JavaFX Tutorials and Articles*. Dostupné z WWW: <http://download.oracle.com/javafx/1.2/tutorials/mediabrowser/module1_task1.html>.
- [13] KRILL, Paul. *Developer world* [online]. 20.9.2010 [cit. 2010-11-25]. JavaFX moves forward while mobile variant on hold. Dostupné z WWW: <<http://www.infoworld.com/d/developer-world/javafx-moves-forward-while-mobile-variant-hold-037>>.
- [14] *Microsoft web* [online]. 2008 [cit. 2010-11-25]. Microsoft Silverlight . Dostupné z WWW: <<http://www.microsoft.com/cze/web/silverlight/>>.
- [15] LÁSLÓ, Petr. *Programujte.com* [online]. 26.6.2008 [cit. 2010-11-25]. Ajax – Úvod. Dostupné z WWW: <<http://programujte.com/?akce=clanek&cl=2008062101-ajax-%96-uvod>>.
- [16] AŠENBRYL, Ondřej. *Petr Pexa - pepe* [online]. 2008 [cit. 2010-11-25]. Tvorba internetových aplikací pomocí Rich Internet Application AJAX Dostupné z WWW: <<http://home.pf.jcu.cz/~pepe/Diplomky/asenbryl.pdf>>.
- [17] MENON, Rakesh . *Java, JavaFX and beyond...* [online]. 27.6.2009 [cit. 2010-12-10]. JavaFX for Windows Mobile (EA). Dostupné z WWW: <http://blogs.sun.com/rakeshmenonp/entry/javafx_for_windows_mobile_ea>.
- [18] *JavaFX Downloads* [online]. Listopad 2009 [cit. 2010-12-10]. Installing

and Testing JavaFX™ Mobile 1.2 for Windows Mobile OS. Dostupné z
WWW: <<http://javafx.com/downloads/mobile/InstallGuide.pdf>>.

[19] TOPLEY, Kim. *JavaFX™ Developer's Guide : Developer's Library*. Ann Arbor (Michigan) : Addison - Wesley, Říjen 2010. 1110 s. ISBN 978-0-321-60165-0.

[20] *Wikipedie* [online]. 2008, 4.11.2010 [cit. 2010-11-25]. JavaFX. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/JavaFX>>.

[21] SCHULZ, Clemens. *Open Source University Meetup* [online]. 2008 [cit. 2010-12-13]. JavaFX.: <http://osum.fontysvenlo.org/javafx_intro.pdf>.

[22] *Adobe* [online]. 2010 [cit. 2010-12-10]. Adobe Flash Player. Dostupné z WWW: <<http://www.adobe.com/products/flashplayer/>>.

Přílohy na CD:

[1] aplikace EnglishForBeginners

[2] dotazníky k aplikaci

Přílohy v elektronické podobě práce

1. Dotazníky přiložené k tomuto dokumentu
2. Aplikace předána k archivaci na katedře informatiky

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Irena Trestrová

Datum narození: 4.4.1959

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

V dnešní době se výuka dosti modernizuje, ale představa, že ve třídě řknu žákům, ať se učí na mobilních telefonech se mi zdá dosti nereálná. Rozhodně by bylo dost těžké uhlídat, jestli všichni pracují.

2. Jak byste si takovou výukovou aplikaci představovali?

Taková aplikace by pravděpodobně byla ideální pro lidi, kteří opravdu mají zájem se učit. Mohli by si například opakovat slovní zásobu.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Knihu bych mobilem nenahrazovala. Jen bych přidala výuku na telefonu jako doplněk. Přeci jen má pro studenty mobilní telefon větší atraktivitu než kniha.

4. Z čeho byste měli největší strach při používání této aplikace?

Pokud by se měla použít ve škole, tak právě z kázně žáků. Jinak bych se využití nebála.

5. Jaké by mělo být ovládání aplikace?

Jednoduché.

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Aplikace je povedená, i když celkem jednoduchá. Pro začátečníky se hodí.

2. Jak hodnotíte ovládání aplikace?

I pro netechnické typy jako jsem já bylo snadné ho zvládnout.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Ano.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Kateřina Bečvářová

Datum narození: 16.3.1994

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

Učení mne moc nebaví. Muselo by to být nějakou zábavnou formou.

2. Jak byste si takovou výukovou aplikaci představovali?

Měla by být komplexní. Něco naučit a pak si vyzkoušet, jestli jsem se učila dobře.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Radši si zapnu telefon a pustím aplikaci, než někde shánět knihu. Telefon mívám většinou u sebe.

4. Z čeho byste měli největší strach při používání této aplikace?

Že bych ji nainstalovala, ale stejně nikdy nepustila.

5. Jaké by mělo být ovládání aplikace?

Na pár kláves, aby to nebylo náročné.

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Ano. Vypadá dobře. Ale stejně nevím jestli přehodnotím svůj postoj ohledně zapínání takových aplikací.

2. Jak hodnotíte ovládání aplikace?

Nijak nevybočilo, ovládá se podobně jako většina aplikací na mobilech.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Ti kdo mají zájem, by to určitě využili.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Lucie Lüftnerová

Datum narození: 28.3.1994

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

Určitě, telefon mám neustále s sebou a ráda ho používám i jinak, než jen k telefonování.

2. Jak byste si takovou výukovou aplikaci představovali?

Hlavně aby byla zábavnější než výuka ve škole.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Mobilní telefon je mnohem skladnější a můžu se učit kdekoli, třeba při cestě autobusem.

4. Z čeho byste měli největší strach při používání této aplikace?

Z toho že jí můj telefon nebude zvládat.

5. Jaké by mělo být ovládání aplikace?

Hlavně ne nijak složité!

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Docela ano, líbily se mi obrázky a celkově grafický vzhled.

2. Jak hodnotíte ovládání aplikace?

Ovládání bylo celkem snadné. Nebylo nutné používat zbytečně moc kláves.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Myslím si že ano, ráda bych se takhle učila slovíčka.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Lenka Perníková

Datum narození: 24.5.1988

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

Přijde mi to jako logická cesta pro vývoj vyučovacích metod.

2. Jak byste si takovou výukovou aplikaci představovali?

Určitě by měla vypadat nějak atraktivně, aby dokázala zaujmout, což je u výuky asi nejtěžší.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Větší praktičnost mobilního telefonu.

4. Z čeho byste měli největší strach při používání této aplikace?

Neumím si představit, jak bych takovou aplikaci používala například při výuce ve škole. Určitě by se všechny děti neodhodlali k tomu, aby propůjčili svůj telefon pro výuku a škola to rozhodně platit nikomu nebude.

5. Jaké by mělo být ovládání aplikace?

V těchto technických věcech se nevyznám, takže co nejméně komplikované je ideální.

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Myslím, že taková aplikace by se dětem mohla líbit.

2. Jak hodnotíte ovládání aplikace?

Neměla jsem s ním větší problémy.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Myslím že ano, umožňuje to učení slovíček prakticky kdykoli a kdekoli.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Lucie Maryšková

Datum narození: 9.5.1995

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

To by bylo skvělé. Mobilní telefon využívám k různým aplikacím, proč se při tom i něco nenaučit.

2. Jak byste si takovou výukovou aplikaci představovali?

Musí mne zaujmout. Takže by to neměl být jen suchý text, ale měla by být barevná, s obrázky a podobně.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Knihy jsou těžké a nepraktické.

4. Z čeho byste měli největší strach při používání této aplikace?

Z toho, jaký mobilní telefon bude potřeba, aby aplikaci podporoval.

5. Jaké by mělo být ovládání aplikace?

Dnes už se všechny aplikace ovládají celkem příjemně, takže bych nic neměnila a držela se klasiky.

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Líbila. Nebyla nijak přeplácáná a přitom bylo dost obrázku, i zvuková ukázka.

2. Jak hodnotíte ovládání aplikace?

Klasické, tedy v pořádku.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Já bych si takle představovala výuku slovíček.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Michal Herman

Datum narození: 28.9.1994

1. část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

To by bylo super, normální výuka ve škole je nuda.

2. Jak byste si takovou výukovou aplikaci představovali?

Měla by dobře vypadat a něco naučit.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Radši půjdu do školy s telefonem než s knihou.

4. Z čeho byste měli největší strach při používání této aplikace?

Aby nebylo nutné kupovat si kvůli tomu nový drahý telefon.

5. Jaké by mělo být ovládání aplikace?

Co nejjednodušší.

2. část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Líbila. Jen je škoda že nebyla prezentována na mobilním telefonu.

2. Jak hodnotíte ovládání aplikace?

Nebylo těžké se naučit aplikaci ovládat.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Mohla by nahradit školní výuku, tohle by mne bavilo víc.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Pavel Hrdý

Datum narození: 14.3.1995

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

Zní to zajímavě.

2. Jak byste si takovou výukovou aplikaci představovali?

Těžko říct, dokážu si představit různé plnění úloh, podobně jako v učebnici, ale bez nutnosti dlouhého a nudného vypisování.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Jak už jsem uvedl v minulé odpovědi, plnění úkolů z učebnic je zdlouhavé a nudné.

4. Z čeho byste měli největší strach při používání této aplikace?

Možná z ceny a z podpory mobilního telefonu.

5. Jaké by mělo být ovládání aplikace?

Ideální by bylo, kdyby se dala aplikace ovládat například pomocí joysticku.

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Aplikace vypadala dobře, žádné divoké barvy, které by rušili soustředění při učení.

2. Jak hodnotíte ovládání aplikace?

Celkem se přiblížilo k mé představě.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Asi ano, pokud by se její vývoj nezastavil a přinášel nové lekce a úkoly.

Dotazník k bakalářské práci Vývoj mobilních aplikací s využitím JavaFX Mobile

Jméno: Karel Falta

Datum narození: 5.9.1976

1.část, před zahájením vývoje aplikace

1. Zaujala vás možnost výuky cizího jazyka na mobilním telefonu?

Ve dnešní době je to aktuální. Všichni používají mobilní telefony, tak proč ne.

2. Jak byste si takovou výukovou aplikaci představovali?

Pro studenty je dobré, když mají možnost si zkontrolovat svoje výsledky ihned. Takže by je aplikace mohla upozorňovat na chyby nebo něco podobného.

3. Co by pro vás bylo důvodem dát přednost tomuto způsobu výuky před jiným, například před knihou?

Právě větší atraktivita pro dnešní mladou generaci. Já už asi zůstanu u knih.

4. Z čeho byste měli největší strach při používání této aplikace?

Z její ceny a z realizace výuky například ve škole. Bylo by to dost komplikované.

5. Jaké by mělo být ovládání aplikace?

V jednoduchosti je síla.

2.část, po ozkoušení aplikace

1. Líbila se vám aplikace?

Nebyla moc obsáhlá, ale pokud by se obsah dal měnit, nebo by byly nové díly, tak by byla aplikace určitě dobrou alternativou. Oceňuji že byla vhodně graficky navržena.

2. Jak hodnotíte ovládání aplikace?

I já jsem to zvládl, tedy ovládání je v pořádku.

3. Byla by tato aplikace přínosná pro výuku angličtiny ?

Pro ty kdo mají zájem určitě, ve škole bych se bál toho, že studenti radši budou hrát hry na mobilu a přitom se tvářit, jako že studují.

