

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA INFORMATIKY A VÝPOČETNÍ TECHNIKY

Bakalářská práce

FAVagotchi
Real-timeová hra pro Windows Phone

Plzeň, 2014

František Kohout

Obsah

1	Úvod	6
1.1	Historie herního průmyslu	7
2	Cíle hry	9
3	Návrh aplikace	10
3.1	Scénář hry	10
3.1.1	Vstup do hry	10
3.1.2	Pravidla hry (studia)	11
3.2	Simulace času ve hře	13
3.2.1	Výpočet herního času	13
3.3	Konec hry	14
3.3.1	Řádný konec	14
3.3.2	Náhly konec hry	15
4	Použité technologie	16
4.1	Microsoft XNA Framework	16
4.2	.NET XAML	17
4.3	Výběr technologie	18
4.4	LINQ a manipulace s daty	18
4.4.1	LINQ to SQL	20
4.4.2	LINQ to XML	22
5	Programátorská dokumentace	24
5.1	Struktura aplikace	24
5.2	Seznam tříd a funkcí	25
5.2.1	třída App : Application	25
5.2.2	třída core	26
5.2.3	třída SQLQuery	27
5.2.4	třída SQLTables	28
5.2.5	třída XMLQuery	28
5.2.6	třída gameTime	28
5.2.7	třída life	29
5.2.8	třída study	30
5.2.9	třída move:PhoneApplicationPage	30
5.2.10	třída mainPage:PhoneApplicationPage	31

5.2.11	třída about:PhoneApplicationPage	32
5.2.12	třída positionChecker	32
5.2.13	třída Schedule:PhoneApplicationPage	33
5.2.14	třída exam:PhoneApplicationPage	34
5.3	Informační obrazovky	35
5.4	Mapové podklady	35
5.5	Zdrojová data SQL	36
5.6	Zdrojová data XML	37
6	Testování aplikace	38
6.1	Výběr zařízení	38
6.1.1	Komunikace s reálným zařízením	39
6.2	Testování běhu aplikace	40
6.2.1	Životospráva	40
6.2.2	Testování studijních podmínek	41
6.3	Dosavadní úspěchy	41
7	Uživatelská dokumentace	42
7.1	Instalace	42
7.2	Hardwarové a softwarové požadavky	42
7.3	Ovládání hry	43
8	Možná rozšíření	45
8.1	Datová rozšíření	45
8.2	Grafika hry	46
8.2.1	Změna mapových podkladů	46
8.3	Umělá inteligence a životospráva	47
8.4	Rozšíření scénáře a příběhu hry	47
8.5	Více platforem	47
9	Závěr	48
	Použitá literatura a zdroje	49
	Seznam obrázků	51
	Seznam tabulek	52
	Seznam algoritmů	53

Přílohy	54
1. Instalační příručka	55
2. Náhledy hry	59

Poděkování

Rád bych poděkoval především vedoucímu své bakalářské práce Ing. Ladislavu Pešíčkovi za nemalé množství poskytnutých rad, postřehů, nápadů a času, který mi při psaní této práce věnoval. Dále děkuji své rodině a přátelům, kteří mi během celého mého studia a následného psaní bakalářské práce poskytli podporu. Zejména pak těm, kteří mi pomáhali mojí hru FAVagotchi testovat. Můj velký dík, patří samozřejmě i všem členům Katedry informatiky za znalosti a konzultace, jež mi v průběhu mého bakalářského studia poskytovali.

Čestné prohlášení

Prohlašuji, že jsem svou bakalářskou práci zpracoval samostatně pod vedením vedoucího bakalářské práce a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal. Jako autor práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Plzni dne 7.5.2013

Abstrakt

English

The main goal of this bachelor thesis is to create and document development of a real-time desk game for platform Windows Phone 7 and higher which will simulate and describe real progress of a study on The Faculty of Applied Sciences in Pilsen. The name of the game is FAVagotchi.

Application is developed in C# language alongside with XAML language and LINQ technology in development environment of Microsoft Visual Studio 2010 Professional and Microsoft .NET framework 4.0. Strictly for the development of mobile applications serve Windows Phone 7.1 SDK library.

This whole bachelor thesis contains a description of all parts of development starting from draft, continuing with choosing technologies and finishing with programming and testing. testování. In addition, this bachelor thesis serve as user and program documentation for the whole project of FAVagotchi.

Česky

Hlavním cílem této bakalářské práce je vytvořit a zdokumentovat vývoj real-timeové deskové hry pro platformu Windows Phone 7 a vyšší, která bude simulovat a popisovat reálný průběh studia na Fakultě Aplikovaných věd v Plzni. Hra se jmenuje FAVagotchi.

Aplikace je vyvíjena v jazyce C# společně s jazyky XAML a technologií LINQ ve vývojovém prostředí Microsoft Visual Studio 2010 Professional a Microsoft .NET framework 4.0. Pro vývoj ryze mobilních aplikací pak slouží knihovna Windows Phone 7.1 SDK.

Celá práce obsahuje popis všech částí vývoje od návrhu, přes výběr technologií až po samotné programování a testování. Tato bakalářská práce navíc slouží jako uživatelská a programátorská dokumentace k celému projektu FAVagotchi.

Kapitola 1

Úvod

Předkládaná bakalářská práce si klade za cíl seznámit čtenáře s tvůrčím postupem vývoje hry pro platformu Windows Phone 7 a vyšší, která bude simulovat a popisovat reálný průběh studia na Fakultě Aplikovaných věd v Plzni. Navíc má za cíl napomoci budoucím studentům k lepší orientaci právě na této fakultě a zároveň ji propagovat mezi širokou veřejností. Hru jsem pojmenoval Favagotchi.

Celá práce je rozčleněna do pěti hlavních kapitol:

V první kapitole bych rád uvedl, proč jsem si vybral dané téma bakalářské práce a dále bych rád zmínil požadované cíle a propagační charakter celé aplikace.

Druhá z kapitol se věnuje samotnému návrhu hry, jejímu scénáři a popisu pravidel, jak časových (semestrálních) událostí tak i těm obecným v průběhu hry. Poté vysvětlím z jaké míry bude scénář příběhu shodný vzhledem k reálnému průběhu studia na fakultě. Aby byla zaručena alespoň přibližná hrátelnost, celý příběh se samozřejmě musí vměstnat do několika málo reálných dní, po které bude hra potenciálnímu hráči příjemňovat život. Proto se tato kapitola bude týkat hlavně časových souvislostí ve hře. Tuto a následující dvě kapitoly bakalářské práce lze považovat za programátorskou dokumentaci.

Třetí část se zabývá převážně funkcionalitou aplikace, použitými technologiemi a jejich popisu. Řeč bude o herním XNA Frameworku od společnosti Microsoft a technologiích XAML, LINQ to SQL a LINQ to XML. Následovat bude ryze programátorská dokumentace, tedy grafické znázornění projektu a výčet tříd a funkcí.

Předposlední kapitola bude dokumentovat postup při testování a ladění celého projektu a jeho součástí. Protože se jedná o aplikaci pro mobilní telefon, nebude v této kapitole chybět ani popis reálného zařízení na kterém byla hra testována, respektive i hrána. Budou rozebrány testovací prostředky, kterými jsem aplikaci obohatil, abych mohl otestovat určité úseky a funkční záležitosti hry, které jsou od sebe časově více vzdálené. Rád bych se v této kapitole také zmínil o vyvíjející se politice Microsoftu v oblasti uveřejnění aplikace v AppStore a samotné registraci zařízení pro testování aplikace.

Nebylo by správné opomenout nedostatky aplikace a možná rozšíření. Tuto kapitolu jsem zařadil právě na konec bakalářské práce. Aplikaci je možné rozšiřovat více směry, zejména pak obsáhlostí dat, protože většina textů, předmětů a zkuškových otázek je reprezentována pomocí značkovacího jazyka XML.

1.1 Historie herního průmyslu

Počátky vzniku prvních „počítačových“ her jsou datovány od roku 1947, kdy Thomas Goldsmith s Estlem Mannem vytvořili hru na osmi katodových trubicích, která simulovala ránu vystřelenou na cíl. Hovoří se o domněnce, že se autoři zřejmě inspirovali radarovými displeji za druhé světové války.

Mezi další předchůdce dnešních her lze považovat klasické piškvorky vyvinuté A.S.Douglasem, které v roce 1952 umožňoval hrát dnes již archaický počítač EDSAC. Avšak za vynálezce prvních videoher je považován William Higinbotham, který vytvořil hru simulující tenisový zápas z pohledu z boku. V této hře tenisový míček podléhá gravitaci a musí být odehrán přes síť. Hru ovládali dva hráči dvěma mohutnými ovladači. Vývoj her šel nezastavitelně dál přes šedesátá léta až do přítomnosti. Hry nejprve vyvíjeli jednotlivci v jejich volném čase a jako běhové prostředí většinou využívali univerzitní servery, kde tak mohly své výtvořky testovat a ladit. Programování šlo skrze komerční sféry, univerzity a různé spolky, také i přes garáže různých nadšenců.

Mezi největší korporace, které samy produkovaly nebo odkupovaly hry od dalších tvůrců, je podle mě společnost Atari, kterou v roce 1972 založil Nolan Bushnell. O Bushnelovi se říká, že Atari založil kvůli své nespokojenosti s tím, jak malé finanční ohodnocení mu ostatní výrobci her nabízeli za jím dodávané polotovary, již dnes legendárních arkádovek. V éře her pro normální smrtelníky, kteří je mohli hrát na domácích počítačích, se nejčastěji používali přístroje od firem, jako jsou DEC (dnešní HP), Apple, Commodore, Tandy a další.

O vývoji počítačových či konzolových her se jednoznačně nedá říci, že už je vše vyvinuto nebo že už potenciálnímu hráči nebude nabídnuto nic nového. Naopak, každý rok v této branži výrobci svět omráčí něčím nevídaným, co si lidé na příštích 365 dní v roce dozajista oblíbí a odvezou domů z regálů počítačových prodejen.

V 90. letech, konkrétně v roce 1997 se po celém světě začal rozvíjet nový herní trend. Odstartoval ho japonský tvůrce her Aki Maita společně s hračkářskou firmou Bandai. Na trh totiž vstoupilo tzv. Tamagoči, ve kterém se uživatel musí starat o určité zvířátko či bytost. Obecně se tomuto hernímu stylu přezdívá „Virtual Pet Care“. Úkolem hráče Tamagoči bylo tuto bytost

krmit, hrát si s ní, plnit určité úkoly a celkově jít s vývojem této postavy na herním displeji. Odměnou byla spokojenost právě této postavy, její fyzický růst či bonusy, které mohl hráč získat při průběhu hraní hry. Tvar samotného Tamagoči připomíná vajíčko a ovládá se pouze třemi tlačítky, čtvrté miniaturní tlačítko slouží pro restart celého herního procesu a „znovunarození“ herní postavy. Postupem času bylo samozřejmě možné hrát Tamagoči i na různých počítačových emulátorech, nebo přímo jako hru vydanou pro určitá zařízení jakými jsou XBOX, stolní PC, GameBoy, chytré telefony a další.

Dnes, v době chytrých telefonů, které pracují na nepřehledném množství verzí operačních systémů Android, Windows Phone nebo iOS je jasné, že jen velmi malý okruh lidí si opravdu koupí Tamagoči a bude ho aktivně hrát. Jako alternativa naproti tomuto problému přicházejí hry přímo do těchto telefonů. Jako první bych rád uvedl hru Tamagochi L.i.f.e. Ta byla uvedena na trh k 16. výročí hry Tamagoči jako její verze do mobilu. Oproti původní verzi má samozřejmě hned několik vylepšení – je barevná, ve vyšším rozlišení a máte na výběr z více “mazlíčků”. Své úspěchy navíc můžete sdílet se svými přáteli na Facebooku. Dále existuje mnoho dalších variant této hry pro mobilní telefony, avšak rád bych ještě vyzdvihl jednu hru. Řeč je o mimozemském mazlíčkovi jménem Pou. Ta se stala hitem během roku 2012.

Kapitola 2

Cíle hry

Samotná podstata hry by nemusela být až tak v jejím technickém a herním provedení ve snaze přilákat svojí hratelností širokou veřejnost, ale spíše v samotné myšlence prezentace školy a jejího pravidelného chodu.

Každého studenta střední školy ve výběru z nepřeberného množství univerzit v České republice ovlivňuje celá řada faktorů. Prvním z nich je jméno školy a fakt, jak si stojí v porovnání s ostatní konkurencí. Dále pak její zaměření, náplň studia a samozřejmě co o ní říkají přátelé, kteří na ní již studují, nebo ji právě z nějakého důvodu ke studiu nezvolili. Do jisté míry ho ovlivní i marketingová činnost a síla propagace dané univerzity, o kterou se starají právě univerzitní pracovníci a studenti. Dle mého názoru středoškoláka v tomto výběru také velmi ovlivní samotné pochopení fungování univerzity, interních pravidel a nejnmutnější byrokracie.

Hra obsahuje poměrně veliké množství informačních textů o fakultě, rozmístění různých budov a také autobusových zastávek. Proto bych jako student fakulty aplikovaných věd velmi rád vývojem této hry napomohl budoucím studentům k lepší orientaci právě na této fakultě a zároveň tímto způsobem svou fakultu, respektive celou Západočeskou univerzitu propagoval v tom nejlepším světle.

Kapitola 3

Návrh aplikace

Při návrhu aplikace bylo nutné navrhnout scénář hry, logiku časového prostředí a hlavně obsáhnout komplexnost dat v podobě seznamu a rozvrhu předmětů, zkušových otázek a především textů které se ve hře vyskytují.

3.1 Scénář hry

3.1.1 Vstup do hry

Po spuštění a načtení hry FAVagotchi se hráč dostane do hlavního menu kde bude mít na výběr ze dvou tlačítek. Viz. obrázek 3.1. Prvním tlačítkem započne FAVákův příběh. Pokud se hráč rozhodne něco si o hře přečíst, tento úkon provede druhým tlačítkem. Pokud by hra nebyla spuštěna poprvé, vsune se mezi tyto dvě možnosti ještě třetí a tou je načtení hry z rozehraného stavu, tedy v určité dějové části hry.

Při již zmiňovaném spuštění příběhu, viz. obrázky 3.2 a 3.3, se hráč ocitne přímo na univerzitním kampusu a je vtažen do děje, který započíná na počátku tříletého bakalářského studia 1.ledna 2000. V tu chvíli se spustí herní čas a postava ve hře započíná svůj virtuální život, který je potřeba udržovat, přesně jako je tomu v klasické hře Tamagotchi.

Herní strategie je prostá, pohybovat postavíčkou po virtuálním studijním kampusu pomocí dotekové obrazovky, plnit pravidla hry a úspěšně hru dovést do jejího zdárného konce.



Obrázek 3.1: Hlavní nabídka



Obrázek 3.2: Uvítací obrazovka



Obrázek 3.3: Herní prostředí - simulace vysokoškolského kampusu

3.1.2 Pravidla hry (studia)

Hned po započetí hry začne herní postavička hladovět, žíznivět a pociťuje nedostatek spánku. Mimo jiné jsou toto hlavní tři faktory, které je potřeba zásahem hráče ovlivňovat a stimulovat. Ve hře jsou různě po areálu, stejně jako je tomu v reálném prostředí, umístěny automaty, kavárny a menza pro doplnění jídla i pití. Pro zdravý studijní život je potřeba FAVáka udržet bdělého a odpočatého, proto musí hráč odcházet pravidelně z kampusu na kolej, kde může jeho virtuální svěřenec v klidu odpočívat do nového studijního dne.

Pro zdárný průchod hrou až do jejího samotného konce musí být splněno ještě několik nezbytných pravidel. Posláním hry je samozřejmě propagovat fakultu v tom nejlepší světě, ale přes to má hlavně výukový charakter, proto je hráč povinen splňovat pravidla školní docházky, a splnit všechny zkoušky z předmětů, které spadají do aktuálního semestru. Virtuální stu-

dent by se měl účastnit všech přednášek z vybraných předmětů v jeho rozvrhu a mít i aktivní účast na cvičeních. Rozvrh předmětů je znázorněn na obrázku 3.4. Studijní aktivita je zaznamenávána v herních minutách pro přesnější výpočet času stráveného na těchto přednáškách. Pokud tomu tak nebude a docházka klesne pod 50%, stane se mu následně u zkoušky to, co se pravděpodobně stane každému studentovi v této situaci - obdrží od zkoušejícího nedostatečnou. Ve hře je tato situace simulována sadou velmi těžkých otázek, na které bude déle vzpomínat i osoba znalá v dané problematice. Pokud nastane situace opačná, FAVák řádně studuje a plní více jak padesáti procentní docházku, dostane u zkoušky takové otázky, na které by měl být schopen odpovědět i průměrný středoškolák.

The screenshot shows a weekly study plan for 'LS 2001'. The interface includes navigation arrows and a 'Zpět' button. The schedule is as follows:

	1 7:30 8:15	2 8:25 9:10	3 9:20 10:05	4 10:15 11:00	5 11:10 11:55	6 12:05 12:50	7 13:00 13:45	8 13:55 14:40	9 14:50 15:35	10 15:45 16:30	11 16:40 17:25	12 17:35 18:20	13 18:30 19:15
Po				FYA1	FYA1	FYA1		PPA2	PPA2				
Út		M2	M2			DMA	DMA	DMA					
St			M2	M2	M2			FYA1	FYA1				
Čt		DMA	DMA			WEB	WEB						
Pá			WEB	WEB				PPA2	PPA2	PPA2			

Obrázek 3.4: Studijní plán

Jediné, o co bude hráč této aplikace "ochuzen", je zápočet z kteréhokoliv předmětu. Ať už formou semestrální práce, aktivity či testu. Tento klasický studijní milník do hry nebyl zakomponován z pochopitelného důvodu. Nejen, že by pro zápočtové testy musel být vytvořen minimálně dvojnásobek otázek, které by musely být obsahově i datově stejně obsáhlé jako zkouškové testy, ale také by aplikace naprosto přesáhla mé časové možnosti. S tímto omezením je také spjata problematika rozvrhu předmětů. Na počátku analýzy této aplikace byla myšlenka provedení reálné simulace vysokoškolského zápisu a libovolného výběru z reálných předmětů na fakultě. Posléze bylo od této myšlenky upuštěno ze stejného důvodů jako je tomu u zápočtu. Proto má hráč studijní plán pevně určen dle reálného rozložení hlavních předmětů vycházejících ze studia oboru Inženýrská informatika při fakultě aplikovaných věd. V číslech to znamená 5 předmětů na semestr, s tím, že u posledního roku studia předmětů poměrně ubyde.

Vzhledem k poměrně velké obsáhlosti dat a logiky v této aplikaci byl v předchozích odstavcích scénář hry kompletně uživatelsky popsán. Nyní bude popsáno, jak je to vlastně s časem ve hře.

3.2 Simulace času ve hře

K tomu aby byla hra reálnová, tj. aby simulovala reálný průběh studia na vysoké škole, je zapotřebí vytvořit simulovanou dějovou linku, aby mohla postava ve hře splňovat veškeré aspekty reálného studenta.

Vzhledem k hratelnosti celé hry FAVagotchi nelze simulovat čas reálně, tzn. že časový souběh událostí musí být několikanásobně zrychlen, aby se celkový průběh šesti bakalářských semestrů dal vměstnat do několika málo reálných dní, po které bude uživatel hru hrát. Navíc se ve hře neuvažují ani víkendové dny, proto hra na přelomu pátku a soboty skočí rovnou na pondělí.

3.2.1 Výpočet herního času

Hra nabízí dva herní časové režimy.

Normální:

10 reálných vteřin, odpovídá jedné minutě ve hře

Do tohoto režimu se herní postavička dostává při každé návštěvě přednášky, cvičení, menzy, bufetu nebo automatu na pití či občerstvení. Z těchto dvou herních módů se jedná o pomalejší časovou variantu. Viz. tabulka 3.1

Herní čas	Reálný čas
1 s	160,666 s
1 min	10 s
1 hod	10 min / 600 s
1 den	4 hod / 240 min
1 týden	28 hod / 1680 min
1 měsíc	120 hodin / 7200 min
1 rok	1460 hod

Tabulka 3.1: Tabulka časů pro normální herní režim

Zrychlený:**1 reálná vteřina, odpovídá jedné minutě ve hře**

V tomto zrychleném režimu se hra nachází jakmile je hráč mimo výše zmiňovaná místa, pohybuje se po kampusu, nebo se nachází na koleji, kde nabírá síly do nového dne. Viz. tabulka 3.2.

Herní čas	Reálný čas
1 s	16,666 s
1 min	1 s
1 hod	1 min / 60 s
1 den	24 min / 1440 s
1 týden	2,8 hod / 168 min
1 měsíc	12 hodin / 720 min
1 rok	146 hod / 8760 min

Tabulka 3.2: Tabulka časů pro zrychlený herní režim

3.3 Konec hry

Každá hra má svůj začátek a také konec, proto nesmím zapomenout uvést kdy hra končí a zmínit zde několik případů, kdy může k této situaci dojít předčasně.

3.3.1 Řádný konec

Na toto místo se hra dostane v případě, že hráč splnil veškerá pravidla hry a řádně ukončil studium, tj. splnil všechny zkoušky, odevzdal závěrečnou bakalářskou práci a splnil státní zkoušku. Nejedná se o žádný zvrat či nečekanou událost, spíše se jedná o vtipné podchycení konce, na kterém je hráč povinen hledat místnost, ve které se odehrává obhajoba bakalářské práce a státní zkouška. Jakmile ji v daném časovém rozmezí najde, hra informuje o řádném ukončení a přejde zpět do hlavního menu.

3.3.2 Náhlý konec hry

K náhlému ukončení hry může dojít i při nesplnění základních pravidel.

Životospráva:

V případě, že se uživatel hře dostatečně nevěnuje, může postavička zemřít na dehydrataci či vyhladovění, přesně jako je tomu u klasické hry Tamagotchi. Hra v tento moment končí informační obrazovkou s odůvodněním. Jakmile virtuální student neustále bdí a nedostane se mu pravidelného odpočinku na koleji, hra končí též. V tomto případě dojde ukončení hry z nedostatku spánku.

Studijní podmínky:

Hra bude na přelomu každého virtuálního semestru kontrolovat stav studia. Pokud nejsou splněny všechny podmínky pro postup do dalšího semestru, v případě aplikace FAVagotchi, splněny zkoušky ze všech předmětů vypsanych na daný půl rok, hra opět končí pro nesplnění základních pravidel.

Kapitola 4

Použité technologie

Celá aplikace je vyvíjena pro platformu Windows Phone 7 v jazyce C#. Jako IDE bylo pro vývoj hry použito Microsoft Visual Studio 2010 Professional a Microsoft .NET framework 4.0. Pro vývoj ryze mobilních aplikací pak Windows Phone 7.1 SDK jako doplněk Visual Studia.

Prvotním úkolem bylo rozhodnout, zda se hra bude odvíjet čistě na herní platformě Microsoft XNA (XNA Game Studio) nebo bude vyvíjena za pomoci frameworku .NET a jeho součástí, jazykem XAML v kombinaci s klasickými C# funkcionalitami. Informace o těchto technologiích jsem čerpal z publikací Programming Windows Phone 7 ¹, Building Windows 8 Apps with C# and XAML ², Pro LINQ: Language Integrated Query in C# 2008 ³ a veřejně dostupných informací přímo na serveru www.microsoft.com ⁴

4.1 Microsoft XNA Framework

Dle dostupných informací z knihy Programming Windows Phone 7 byla tato platforma vyvinuta společností Microsoft převážně pro vývoj 2D a 3D her nejen pro herní platformu Windows Phone, ale také pro platformy Xbox 360 či Zune. Obsahuje tři hlavní knihovny - Microsoft.Xna.Game, Microsoft.Xna.Pipeline, Microsoft.Xna.Framework. Hra FAVagotchi byla již od počátku zamýšlena ve 2D grafice proto se nebudeme 3D problematikou vůbec zabývat.

V XNA Frameworku 2D hra obvykle sestává z grafického pozadí a prvků zvaných Sprites. Sprites jsou 2D grafické objekty manipulované nezávisle na zbytku scény. Rovněž lze využít i animované Sprites, kde se jednotlivé obrázky mění v závislosti na předdefinovaném časovém intervalu, tyto změny

¹Programming Windows Phone 7 - Charles Petzold - ISBN: 978-0-7356-4335

²Building Windows 8 Apps with C# and XAML (Microsoft Windows Development Series - Jeremy Likness - ISBN-13: 978-0321822161

³Pro LINQ: Language Integrated Query in C# 2008 - Joseph C. Rattz, Jr. - ISBN: 978-1-59059-789-7

⁴XAML for Windows Phone 8 - <http://msdn.microsoft.com/en-us/library/windowsphone/develop/cc189036>

v čase poté vytvářejí iluzi pohybu. Souřadný systém při vytváření 2D scény má standardně počátek (0,0) v levém horním rohu obrazovky. Pro vykreslování textu, tedy vytvoření 2D obrázku na základě požadovaného textu, slouží objekt `SpriteFont`, reprezentující font definovaný prostřednictvím XML souboru s příponou `.spritefont`.

Celý vývoj probíhá tedy spíše v grafickém režimu. Je zapotřebí vytvořit mnoho grafických komponent a mapových podkladů, které zajistí hratelnost hry.

4.2 .NET XAML

.NET XAML je značkovací jazyk podobný klasickému HTML využívaný k popisu grafického rozhraní v aplikacích společnosti Microsoft nové generace. Je vyvíjen výhradně Microsoftem a založený na XML. Pomocí tohoto jazyka lze definovat grafickou strukturu jak webových, tak i desktopových či mobilních aplikací. Jedná se o aplikace které převážně obsahují standardní ovládací prvky, jakými jsou například tlačítka, seznamy, textová a výběrová pole a mnoho dalších. Již z popisu tohoto jazyka je zřejmé, že oproti XNA Frameworku se dá XAML využít nejen pro hry ale i pro širší spektrum aplikací. Viz. kód 4.1.

Kód 4.1: Využití XAML prvku `Storyboard` pro animaci pohybu herní postavičky ve třídě "move"

```
<phone:PhoneApplicationPage.Resources>
  <Storyboard x:Name="sb">
    <DoubleAnimationUsingKeyFrames
      Storyboard.TargetName="translate"
      Storyboard.TargetProperty="Y">
      <SplineDoubleKeyFrame x:Name="whereY" />
    </DoubleAnimationUsingKeyFrames>

    <DoubleAnimationUsingKeyFrames
      Storyboard.TargetName="translate"
      Storyboard.TargetProperty="X">
      <SplineDoubleKeyFrame x:Name="whereX" />
    </DoubleAnimationUsingKeyFrames>
  </Storyboard>
</phone:PhoneApplicationPage.Resources>
```

4.3 Výběr technologie

Porovnáním .NET XAML a XNA Framework nelze jednoznačně určit, která z těchto technologií je obecně lepší pro vývoj FAVagotchi. Lze je mezi sebou i libovolně kombinovat, tzn. že se nejedná o dvě oddělené technologie. Je ale třeba rozhodnout, která z nich bude v celé aplikaci převažovat. Celá hra je koncipována spíše v naučném smyslu. Obsahuje větší množství textových polí a snaží se spíše simulovat a popsat průběh studia na Západočeské univerzitě. Graficky vypracované animace a 2D grafika by jistě byly nespornou výhodou této hry ale z hlediska velké obsáhlosti dat ve hře, úspory času a nedostatku grafických podkladů pro hru, jsem vybral druhou variantu, tedy technologii .NET XAML.

Pokud zvážím různá pro a proti, určitě by se XNA Framework dal využít v grafických animacích, jakými jsou například pohyb postavičky či přepínání mapových podkladů ve hře, ale vzhledem k poměrně složité implementaci této kombinace dvou technologií, jsem tuto možnost zavrhl.

4.4 LINQ a manipulace s daty

S příchodem .NET frameworku 3.5 byla zveřejněna finální verze projektu, který se na základě informací na serveru „<http://msdn.microsoft.com>“⁵ dá označit za revoluci v přístupu k datům na platformě .NET. Jedná se o projekt LINQ. V .NET aplikacích lze pracovat s relačními či hierarchickými daty a to hned několika způsoby. V případě dat, která jsou uložena v relační databázi bylo možné přistupovat skrze známé rozhraní ADO.NET, a to jak v připojeném tak odpojeném režimu. Pokud jsou data reprezentována hierarchicky v populárním formátu XML, lze využívat typů z assembly System.Xml a pracovat s těmito daty buď pomocí DOM nebo je zpracovávat sekvenčně.

Tyto přístupy by se daly považovat za standardizované do té chvíle, než přišla technologie LINQ, což znamená Language Integrated Query. LINQ přináší pomocí nové palety klíčových slov podporu dotazování přímo do .NET programovacího jazyku a to s sebou nese mnoho výhod oproti výše zmíněným postupům.

Pomocí této technologie lze totiž pracovat takřka s libovolnými daty pomocí již zmíněných klíčových slov podobně jako je tomu u klasického dotazování v SQL aj. Zmíněná klíčová slova jsou uvedeny v tabulce 4.1.

⁵msdn.microsoft.com/cs-cz/library/bb397926.aspx - Oficiální dokumentace k technologii LINQ

Select	výběr hodnoty kterou chceme použít
SelectMany	výběr více hodnot najednou (např. pole)
Join	spojení více poskytovatelů dat
GroupBy	rozdělení dat do více skupin podle určitého klíče
Where	omezení výběru prvků podle specifikované podmínky
OrderBy, OrderByDescending	specifikace třídění, umožňuje výběr elementu podle kterého se má třídit
First, Last	výběr prvního nebo posledního prvku z kolekce
ElementAt	výběr prvku podle udaného indexu
Count	počet prvků v kolekci
Union, Intersect, Except	definice množinových operací sjednocení, průnik a rozdíl
Sum, Min, Max, Average	vrací součet, minimální, maximální či průměrnou hodnotu z dané kolekce
Reverse	otočí pořadí prvků v kolekci
Concat	spojí dvě kolekce dohromady
OfType	výběr pouze těch prvků, které jsou specifikovaného typu

Tabulka 4.1: Tabulka klíčových slov LINQ

LINQ je navržena tak, že je možné vytvářet její implementace pro jednotlivé datové zdroje. Je to podobné jako v ADO.NET, kde je možné implementací rozhraní vytvořit .NET provider pro specifický typ databáze, avšak v LINQ tato rozšiřitelnost nezůstává jenom u relačních databází a je mnohem abstraktnější. Spolu s .NET frameworkem 3.5 je dodáno několik oficiálních implementací LINQ od Microsoftu, které by měly pokrýt běžné potřeby pro tvorbu aplikací pracujících s daty. Jedná se o implementace uvedené v tabulce 4.2.

LINQ to Objects	Implementace LINQ pro standardní kolekce nacházející se v paměti
LINQ to SQL	Implementace LINQ pro Microsoft SQL Server 2000 a vyšší
LINQ to XML	Implementace LINQ pro práci s XML daty
LINQ to DataSet	Implementace LINQ pro práci s ADO .NET datasety

Tabulka 4.2: Tabulka implementací LINQ

Na internetu jsou dostupné i další implementace LINQ, jako je například LINQ to Amazon, který slouží pro vyhledávání knih v tomto internetovém obchodě. Ve hře FAVagotchi jsou však využity implementace **LINQ to SQL** a **LINQ to XML**.

4.4.1 LINQ to SQL

Při vývoji jakékoliv mobilní aplikace nebo hry je důležité myslet na data, která je potřeba uchovat po minimalizaci, deaktivaci ale i po ukončení procesu aplikace. Klasicky pro takovou situaci slouží databáze, do kterých se data mohou po dobu nečinnosti aplikace ukládat. Po důkladném pátrání v literatuře Pro LINQ: Language Integrated Query in C# 2008 ⁶ a ve světě internetu se dá najít hned několik realizací databáze na platformě Windows Phone. V porovnání s LINQ to SQL se tyto další realizace implementačně ani výkonostně dle mého názoru nedají ani srovnávat. Moji volbu navíc ovlivnil také fakt, že technologie LINQ se dá aplikovat i na jazyk XML, proto jsem neváhal a začal o implementaci LINQ to SQL čerpat dostupné informace.

V aplikaci je ukládání dat vyřešeno pomocí dvou hlavních tříd `SQLTables`, reprezentující databázi a tabulky v ní obsažené a `SQLQuery`, obsahující jednotlivé operace právě pro tyto tabulky. Viz. ukázky kódů 4.2, 4.3. a 4.4.

Kód 4.2: Ukázka třídy `SQLTables` představující databázi a třídy `gameData` sloužící jako tabulka v této databázi

```
using System.Data.Linq;  
using System.Data.Linq.Mapping;  
public class SQLTables : DataContext  
{  
    public SQLTables(string connectionString)
```

⁶Pro LINQ: Language Integrated Query in C# 2008 - Joseph C. Rattz, Jr. - ISBN: 978-1-59059-789-7

```

        : base(connectionString) { }
        public Table<gameData> gameDatas;
    }
    [Table]
    public class gameData
    {
        public gameData() { }
        [Column(IsPrimaryKey = true, IsDbGenerated = true)]
        public int ID { get; set; }

        [Column(CanBeNull = false)]
        public string name { get; set; }

        [Column(CanBeNull = false)]
        public string data { get; set; }
    }

```

Kód 4.3: Ukázka z třídy SQLQuery reprezentující dotazy na databázi. Zde konkrétně zápis a výpis herních zdrojových dat

```

public class SQLQuery
{
    public SQLTables DB;
    public SQLQuery(string connectionString)
    {
        DB = new SQLTables(connectionString);
        if (DB.DatabaseExists() == false) DB.CreateDatabase();
    }
    public void insertTo(string name, string data)
    {
        gameData ud = new gameData();
        ud.name = name;
        ud.data = data;
        DB.gameDatas.InsertOnSubmit(ud);
        DB.SubmitChanges();
    }
    public IList<gameData> selectAll()
    {
        IQueryable<gameData> query;
        query = from c in DB.gameDatas select c;
        IList<gameData> countryList = query.ToList();
    }
}

```

```
        return countryList;
    }
}
```

Kód 4.4: Dotaz vkládající do tabulky gameData informaci o časovém módu při načítání hry může vypadat například takto.

```
App.SQL.insertTo("gameTimeMode", "speed");
```

4.4.2 LINQ to XML

Jak bylo zmíněno výše, LINQ se dá využívat i pro manipulaci s daty ve formátu XML. Abych mohl zpracovávat tento formát dat, byl jsem doposud nucen v mých aplikacích používat klasický DOM nebo sekvenční přístup. LINQ to XML však umožňuje tato data zpracovávat mnohem více efektivněji a rychleji. Ve výsledku tato manipulace s XML skrze implementaci LINQ vypadá velmi podobně jako přístup k SQL.

Ve hře FAVagotchi se XML oproti SQL využívá zejména pro neměnná zdrojová data, jakými jsou zobrazované texty ve hře (viz. kódy 4.5, 4.6), názvy předmětů, rozvrhové akce či testové otázky. Jedná se tedy spíše o režim čtení dat, než o režim zápisu a mazání jako je tomu u dynamických dat v SQL.

Kód 4.5: Ukázka z třídy XMLQuery reprezentující přístup k textovému generátoru

```
using System.Xml;
using System.Xml.Linq;

public class XMLQuery
{
    public XElement sen;
    public XMLQuery(string sePath)
    {
        sens = XElement.Load(sePath);
    }
    public string textGenerator(string key
                                , string toReplace)
    {
        Random r = new Random();
```

```
var query = from c in sens.Elements("sen")
            where c.Element("key").Value == key select c;

return
    query.Elements("text")
        .ToList()[
            r.Next(0, query.Elements("text")
                .Count() - 1)]
        .Value.Replace("{toReplace}", toReplace);
}
}
```

Kód 4.6: Soubor sentences.xml pro generování vět ve hře

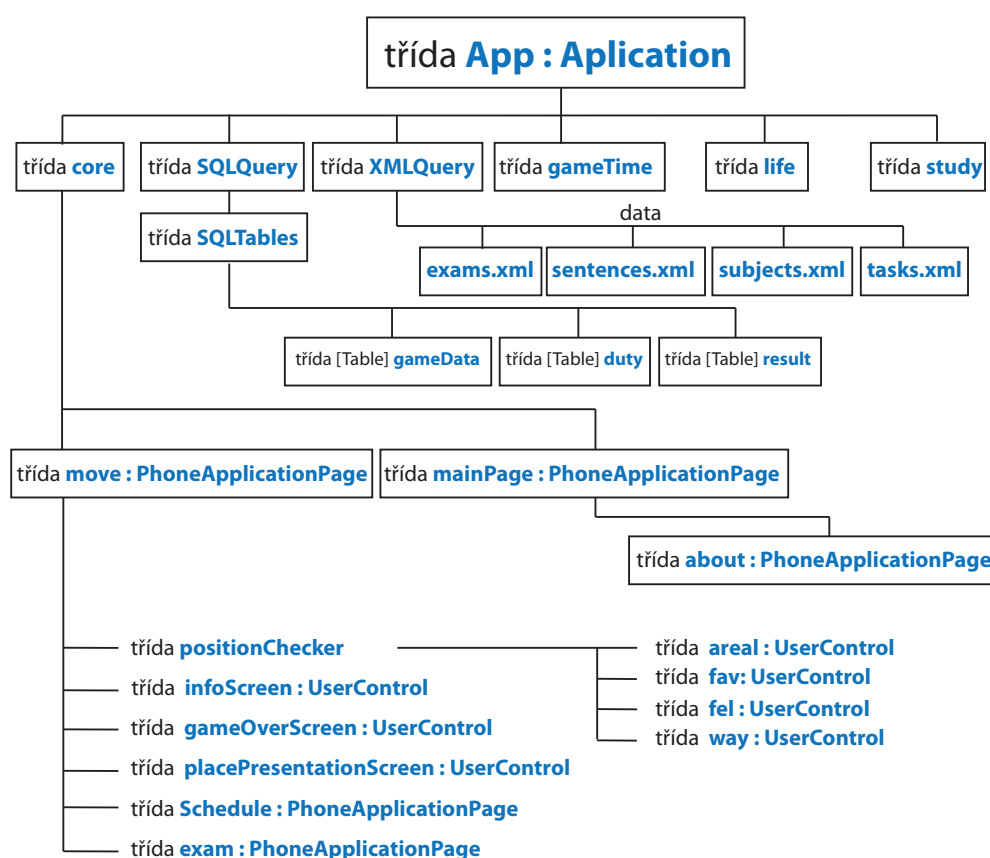
```
<?xml version="1.0" encoding="utf-8"?>
<sentences>
  <sen>
    <key>s1</key>
    <text>...</text>
    <text>...</text>
  </sen>
</sentences>
```


Kapitola 5

Programátorská dokumentace

5.1 Struktura aplikace

Graficky popisuje seznam tříd a funkcí celé aplikace. Jediné, co se v tomto diagramu nenachází jsou funkce a třídy zajišťující zobrazení komponent a jsou automaticky generovány prostřednictvím Visual Studio 2010. Zjednodušeně lze říci, že se v grafickém znázornění FAVagotchi vyskytuje pouze to, co bylo naprogramováno. Viz. obrázek 5.1



Obrázek 5.1: Grafické znázornění seznamu tříd

Jak je vidět na tomto grafickém znázornění struktury hry FAVagotchi, celou aplikaci zastřešuje třída `App : Application`, která se stará o chování aplikace v závislosti na akcích uživatele a dále inicializuje třídy: `gameTime` - simulující čas ve hře, `life` - zajišťující životosprávu a `study` - starající se o studijní profil.

Poslední třídou, která je inicializována skrze `App : Application` pomocí `mainPage : PhoneApplicationPage` je třída `core`. Zde se na pozadí aplikace inicializují globální proměnné uvnitř výše vyjmenovaných doprovodných tříd a prostřednictvím LINQ se postupně inicializují SQL a XML zdrojová data. Jako struktura SQL databáze slouží třída `SQLTables` a v ní obsažené třídy: `gameData` - základní aplikační data, `duty` - studijní aktivita a `result` - výsledky zkoušek. Na tuto databázi je možno se dotazovat pomocí třídy `SQLQuery`. XML zdrojové soubory jsou pak načítány a hře zprostředkovány skrze třídu `XMLQuery`. Jedná se o tyto soubory: `exams.xml` - rozpis zkoušek, `sentences.xml` - zdroj textových podkladů pro hru, `subjects.xml` - seznam studijních předmětů společně s instrukcemi pro zařazení předmětů do studijního plánu, `tasks.xml` - zdroj testových otázek.

Na základě těchto zinicizovaných dat se postupně vyhodnocují události v souvislosti s pohybem herní postavičky a časem, které nastávají ve třídách využívajících právě "jádro" `core` a běžící pod instancí třídy `move : ApplicationPhonePage`. Pod touto třídou se na základě pozice postavičky asynchroně na pozadí volá třída `positionChecker`, která má na starosti rozpoznat kde se hráč právě nachází. V této souvislosti skrze třídu `move` mění mapové podklady `areal`, `fav`, `fel` a `way` reprezentované pomocí `UserControls` a vyvolává informační obrazovky `infoScreen`, `gameOverScreen`, `placePresentationScreen`.

Studijní profil hráče je dostupný skrze horní uživatelské menu. Reprezentuje ho třída `Schedule : PhoneApplicationPage`. To nejdůležitější co však třída `positionChecker` a `move` zajišťují, je obsluha zkuškového testu reprezentovaná třídou `exam : PhoneApplicationPage`.

5.2 Seznam tříd a funkcí

5.2.1 třída `App : Application`

Tato třída definuje celé chování aplikace v závislosti na akcích uživatele. Dále inicializuje tři hlavní třídy: `gameTime`, `life` a `study`. Inicializační funkce které v této třídě automaticky vytváří Visual Studio a nebylo v nich proveden žádný zásah zde nebudou uvedeny.

void Application_Launching() Inicializace XML a LINQ.

void Application_Activated() Změna globální statické proměnné `gameWasDeactivated` na hodnotu `true`

void Application_Deactivated() Minimalizace aplikace na telefonu. Volání funkce `core.saveGame()`

void Application_Closing() Ukončení procesu aplikace v telefonu. Volání funkce `core.saveGame()`

void RootFrame_NavigationFailed() Selhání navigačního prostředí (odkaz na neexistující soubor či obsah, neznámá chyba). Volání funkce `core.saveGame()` navíc zobrazení chybového hlášení.

void Application_UnhandledException() Neznámá chyba. Volání funkce `core.saveGame()`

void BackgroundMusic() Přehrávání MIDI souborů na pozadí hry pomocí `GlobalMediaElement`

5.2.2 třída `core`

Hlavní statická třída programu obsahující globální statické proměnné.

void startNewGame() Spouštění nové hry, inicializace proměnných, zavedení nových dat do LINQ.

void loadGame() Načtení rozehrané hry, inicializace proměnných pomocí dat načtených z LINQ, dopočítání herního času (odečet času zavření aplikace od aktuálního času), dopočet času v závislosti na poslední pozici hráče, před vypnutím hry (v případě že se hráč nacházel na místě ve kterém čas ubíhal pomaleji např. v menze, nebo na místě kde záleželo na čase, ku příkladu na přednášce nebo zkoušce).

void gameOver(string why) Ukončení hry, smazání dat z LINQ, zobrazení info obrazovky.

saveGame Uložení statických proměnných dat do LINQ.

5.2.3 třída `SQLQuery`

`SQLQuery(string connectionString)` Kontruktor třídy vytvářející nebo inicializující databázi na základě cesty k datovému souboru `.sdf`.

`void insertTo(string name, string data)` Vkládání dat do tabulky `gameData`

`IList selectAll()` Vrací celou tabulku `gameData`

`string Select(string what)` Vrací konkrétní prvek tabulky `gameData` na základě hledaného klíčového slova.

`void Update(string what, string to)` Editace záznamu v tabulce `gameData` dle klíčového slova a nahrazujícího prvku.

`void TruncateGameData` Vyprázdnění tabulky `gameData`

`void insertToSubjectsDuty(string subject, string dateTimeIn)`
Vkládání počátečního času přístupu na přednášku nebo cvičení dle zadaného předmětu.

`void UpdateSubjectsDateTimeOut(string subject)` Dokončení záznamu o studijní aktivitě v tabulce `duty` na základě aktuálního času a odečtení času přístupu na přednášku nebo cvičení, který je vložen do tabulky skrze funkci `void insertToSubjectsDuty` .

`int getSumMinutes(string subject)` Vrací počet minut strávených na přednáškách a cvičeních dle zadaného předmětu.

`IList selectAllSubjectsDuty()` Vrací celou tabulku `duty`

`void TruncateDutys()` Vyprázdnění tabulky `query`

`void insertToResults` Vkládání záznamu o výsledku zkoušky do tabule `result`.

`void countExamTry(string subject)` Vrací počet pokusů o zkoušku dle zadaného předmětu.

int countExamEnter(string subject, string timeTo) Vrací počet přístupů na zkoušku dle zadaného předmětu a času zkoušky.

void TruncateResults() Vyprázdnění tabulky `result`.

void TruncateDatabase() Vyprázdnění všech tabulek v databázi.

void DeleteDB() Smazání celé databáze.

5.2.4 třída `SQLTables`

Obsahuje inicializaci tříd `gameData`, `dutys`, `results`

5.2.5 třída `XMLQuery`

XMLQuery(string sPath, string ePath, string tPath, string sePath)
Kontruktor třídy inicializující globální proměnné `XElement` `subjects`, `exams`, `tasks`, `sentences` na základě cesty k datovému souboru `.xml`.

string array getSubjectsBySchedule(string room) Vrací předmět dle zadané místnosti ve které se odehrává.

string array getSubjectsBySortcut(string sc) Vrací předmět dle jeho zadané zkratky.

string array getExam(string room) Vrací termín zkoušky dle zadané místnosti ve které se odehrává.

string textGenerator(string key, string toReplace) Vrací náhodně vybraná textový podklad na základě zadaného klíče. Sekvence `toReplace` slouží pro případ, že bychom v daném XML chtěli dynamicky měnit text. Například číslo místnosti pro zkoušku či název předmětu.

5.2.6 třída `gameTime`

Obsahuje proměnné nesoucí informace o herním čase a režimu hry.

void saveTimes() Uložení herního času, režimu a času deaktivace aplikace do LINQ.

int getMyTime() Vrací speciální konverzi z časového formátu HH:MM na číselný formát typu integer, pouhým spojením HH+MM, v aplikace je tento údaj využít pro sestavování rozvrhových akcí.

string getSemester() Vrací aktuální označení semestru ZS nebo LS.

int getDay() Vrací číslo dne v týdnu, pondělí se bere jako 0.

string getNameDay() Vrací název dne v českém jazyce.

void setMode(string m) Nastavuje herní režim na **SPEED** nebo **NORMAL**.

void stopTime() Zastaví čas ve hře a nastaví hodnotu globální statické proměnné **isRunnable** na **false**.

void startTime() Spustí čas ve hře a nastaví hodnotu globální statické proměnné **isRunnable** na **true**.

void incrTime() Inkrementace herního času o minutu. Dekrementace životosprávy. V případě přetečení času na Sobotu nebo Neděli, inkrementuje o jeden nebo dva dny.

5.2.7 třída **life**

Uchovává a manipuluje s informacemi o životosprávě herní postavičky.

void decrLifes() Odečítá žízeň, hlad a vyvolává konec hry v případě že dojde k nulovému stavu jednoho z těchto prvků životosprávy.

void decrSleep() Odečítá spánek a vyvolává konec hry v případě jeho nulového stavu.

void setLifes(double w, double f, double s) Nastavení životosprávy na danou úroveň.

double getWater() Vrací stav úrovně vody.

double getFood() Vrací stav úrovně jídla.

double getSleep() Vrací úroveň spánku.

bool plusWater(double w) Přičítá úroveň vody.

bool plusFood(double f) Přičítá úroveň jídla.

bool plusSleep(double s) Přičítá úroveň spánku.

void saveLives() Uložení proměnných spojených s životy do LINQ.

5.2.8 třída `study`

void checkStudy() Testuje zda byly splněny podmínky studia při přechodu do nového semestru. Pokud ne, vyvolá `stac game over` pro nesplnění podmínek studia.

5.2.9 třída `move:PhoneApplicationPage`

move() Konstruktor třídy inicializující aplikační komponenty, globální proměnné a asynchronní vlákna v podobě `DispatcherTime` komponent. Svými funkcemi spíše simuluje `globalEvent listener`, naslouchající všem akcím uživatele a herní postavy na základě jednotlivých herních úkonů.

void PhoneApplicationPageLoaded(object sender, RoutedEventArgs e) Odstraňuje zpětné navigační vazby při stisknutí tlačítka zpět na telefonu. Při spuštění hry dále nastavuje výchozí pozici postavy.

void stopAllTimers() Zastaví všechna aktivní vlákna.

void changeMap(string map, bool move) Nastavuje a mění mapový podklad hry na základě zadané proměnné `map`.

void checkPositionFunction() Asynchronně pracující funkce zjišťující zda postava dorazila na zadané souřadnice X a Y. Jakmile tato situace nastane, pomocí globální proměnné `core.position` vyhodnotí akci a provede.

void positionTimerToGetOut() Jakmile se postava dostane do jakékoli místnosti, menzy, zkoušky nebo automatu na jídlo či pití, spouští se tato asynchroně pracující funkce, která neustále zjišťuje zda postava není z tohoto místa pryč aby opět mohla spustit buďto jiný mapový podklad, nebo vyhodnotit jinou akci.

void eventCheckerFunc() Asynchroně obsluhuje zobrazování herního času, jednotek životosprávy a zároveň slouží jako eventListener při vyvolání info obrazovky nebo gameOver stavu.

void animateFunction() Animační funkce obstarávající správné vykreslení, natočení postavy a střídání nohou (levá - pravá)

void moveObject(double kX, double kY, string target, string kom) Zajišťuje samotný výpočet pro pohyb postavy na obrazovce. Pomocí klasického Pythagorova vzorce dopočítá směr pohybu a spustí animační funkci `animateFunction`.

sekvence funkcí void Tap Funkce zajišťující vykonání pohybu, změnu obrazovky či konec hry na základě dotyku na jeden z ovládacích prvků rozmístěných na herní ploše.

void arrowAnimationFunc() Animace ukazatele po dotyku na určitý ovládací prvek.

void markTarget(int x, int y) Nastavení souřadnic ukazatele a spuštění samotné animace.

5.2.10 třída `mainPage:PhoneApplicationPage`

mainPage() Konstruktor třídy inicializující ovládací komponenty hlavního menu.

void loadGameClick() Voláním funkce `core.loadGame()` načte potřebná data ke spuštění rozehrané hry, spustí herní čas a přesměruje hráče do hlavní herní obrazovky zobrazené pomocí třídy `move`.

void newGameClick() Voláním funkce `core.newGame()` načte potřebná data ke spuštění nové hry, spustí herní čas a přesměruje hráče do hladní herní obrazovky zobrazené pomocí třídy `move`.

void PhoneApplicationPageLoaded() Odstraňuje zpětné navigační vazby při stisknutí tlačítka zpět na telefonu. Navíc zastaví herní čas.

void aboutClick() Přesměrování na obrazovku informující hráče o hře FAVagotchi.

5.2.11 třída **about:PhoneApplicationPage**

about() Konstruktor třídy inicializující ovládací komponenty stránky o hře FAVagotchi.

void PhoneApplicationPageLoaded(object sender, RoutedEventArgs e) Odstraňuje zpětné navigační vazby při stisknutí tlačítka zpět na telefonu

void backClick(object sender, RoutedEventArgs e) Návrat ze stránky o hře do hry.

5.2.12 třída **positionChecker**

void startTimer(int interval) Spouští asynchronní vlákno (timer) **positionTimerFunc** se zadaným časovým intervalem v milisekundách.

void check() Kontrola proměnné `core.targetPosition`. Po vyhodnocení proměnné vyvolá příslušnou funkci ve třídě `positionChecker`

void menza() Postava se nachází v menze. Jakmile funkce proběhne prvotní podmínkou na čas přístupu do menzy vyvolá funkce informační obrazovku o vstupu do menzy. Dále naplní globální proměnnou `core.position`, vynuluje `core.targetPosition` a nastaví herní čas na normální režim.

void favBufet() Postava se nachází v bufetu budovy FAV. Naplní globální proměnnou `core.position`, vynuluje `core.targetPosition`, nastaví herní čas na normální režim a spustí `positionTimerFunc` s intervalem 200ms.

void felBufet() Postava se nachází v bufetu budovy FEL. Naplní globální proměnnou `core.position`, vynuluje `core.targetPosition`, nastaví herní čas na normální režim a spustí `positionTimerFunc` s intervalem 200ms.

void foodMat() Postava se nachází u jednoho z automatu na jídlo. Naplní globální proměnnou `core.position`, vynuluje `core.targetPosition`, nastaví herní čas na normální režim a spustí `positionTimerFunc` s 200ms intervalem.

void waterMat() Postava se nachází u jednoho z automatu na pití. Naplní globální proměnnou `core.position`, vynuluje `core.targetPosition`, nastaví herní čas na normální režim a spustí `positionTimerFunc` s 200ms intervalem.

void sleep() Postava se nachází na koleji. Naplní globální proměnnou `core.position`, vynuluje `core.targetPosition`, nastaví herní čas na normální režim a spustí `positionTimerFunc` s 200ms intervalem.

void learningProgres() Postava se nachází v jednom ze tří stavů (přednáška, cvičení, zkouška). Probíhá kontrola těchto stavů a dále vyhodnocení. Pak už nastává podobný proces jako u předchozích pozičních funkcí. Naplní globální proměnnou `core.position`, vynuluje `core.targetPosition`, nastaví herní čas na normální režim a spustí `positionTimerFunc` s 200ms intervalem.

void positionTimerFunc() Kontrola globální proměnné `core.position`. Na základě její hodnoty v nastaveném intervalu vykonává časově závislé úlohy, kterými jsou přičítání prvků životosprávy (pití, jídlo, spánek), jestliže se postava nachází na místě kde tyto prvky lze doplnit a nebo přičítá studijní aktivitu pokud je postava na přednášce nebo cvičení z nějakého předmětu.

void positionInterrupt() Přerušuje asynchroně pracující funkci `positionTimerFunc` a vyvolá informační obrazovku s příslušným textem. Při přerušení procesu stravování v menze si například postava postěžuje, že se ani nestihla najíst.

void savePosition() Ukládá do SQL tabulky `gameData` informace o poloze postavy a aktuální mapový podklad.

5.2.13 třída `Schedule:PhoneApplicationPage`

Schedule() Konstruktor třídy inicializující ovládací komponenty studijního profilu.

void createTable() Vytvoření podkladu rozvrhu pro rozmístění předmětů.

void insertSubject(int day, string blocks, string type, string shortcut) Rozmístění předmětů do předpřipraveného podkladu.

void subjectTap() Zobrazení detailu předmětu.

void backToHome() Návrat z detailu do rozvrhové akce studenta.

void button5Click() Zobrazení rozpisu zkoušek.

void button6Click() Přepínání semestrálního období v rozvrhu vpřed.

void button7Click() Přepínání semestrálního období v rozvrhu vzad.

void button8Click() Zobrazení rozvrhu.

void back() Návrat ze studijního profilu do hry.

5.2.14 třída exam:PhoneApplicationPage

exam() Konstruktor třídy inicializující ovládací komponenty zkouškového procesu. Výpočet složitosti zkoušky na základě studijní aktivity. Inicializace pole otázek vypočtené úrovně obtížnosti.

void makeQuestion(int dir) Inicializace ovládacích komponent, vypsání zkouškové otázky, možností a výpočet stránkování pro přechod na další otázku.

void checkEndFunction() Asynchroně pracující funkce, hlídající konec zkoušky dle stanoveného časového limitu. Na konci limitu končí zkoušku a vyzve k odevzdání testu.

void button1Click() Po stisku tlačítka vyvolá opuštění zkoušky a info obrazovku s potvrzením odchodu ze zkoušky hráčem.

void zacitClick() Po stisku tlačítka vyvolá začátek zkoušky, spustí kontrolu konce zkoušky a načte první otázku skrze funkci `makeQuestion`

void odevzdatClick() Po stisku tlačítka vyhodnotí a uloží výsledky zkoušky do SQL tabulky `result`.

void zustatClick() Zruší volbu odchodu ze zkoušky.

void nextClick() Přejede na další testovou otázku.

void prevClick() Přejede na předchozí testovou otázku.

void checkAnswers() Zkontroluje vybranou otázku a zanesou výsledek do pole s výsledky otázek.

void opt1Tap() Vyvolá funkci `checkAnswers()`

void opt2Tap() Vyvolá funkci `checkAnswers()`

void opt3Tap() Vyvolá funkci `checkAnswers()`

void opt4Tap() Vyvolá funkci `checkAnswers()`

void PhoneApplicationPageLoaded() Odstraňuje zpětné navigační vazby při stisknutí tlačítka zpět na telefonu

5.3 Informační obrazovky

třída `infoScreen:UserControl`

třída `gameOverScreen:UserControl`

třída `placePresentationScreen:UserControl`

třída `about:PhoneApplicationPage`

5.4 Mapové podklady

třída `areal:UserControl` Mapový podklad areálu Bory.

třída `fav:UserControl` Mapový podklad budovy FAV.

třída `fel:UserControl` Mapový podklad budovy FEL.

třída `way:UserControl` Mapový podklad cesty mezi areálem Bory a kolejemi.

5.5 Zdrojová data SQL

třída [Table] `gameData`

Obsahuje herní data, která jsou průběžně ukládána do SQL tabulek ze kterých se pak provádějí výpočty životosprávy a studijní aktivity při načítání a pokračování ve hře. Jedná se o tyto data:

startGameTime Čas spuštění hry.

gameTime Herní čas.

closeTime Čas ukončení hry.

gameTimeMode Časový režim.

lastPosition Poslední pozice postavy.

lastPositionOutside Poslední pozice postavy mimo vnitřní prostory objektů.

map Aktuální mapový podklad.

life_Water Stav pití.

life_Food Stav potravy.

life_Sleep Stav spánku.

subjectDutyTimeOut Čas odchodu z přednášky nebo cvičení.

třída [Table] `duty`

Obsahuje záznamy studijní aktivity tj. čas strávený na přednáškách a cvičeních.

třída [Table] `result`

Obsahuje výsledky zkouškových testů.

5.6 Zdrojová data XML

exams.xml rozpis zkoušek pro daný předmět.

sentences.xml reprezentace textů ve hře pomocí klíče a následného textu dané věty. Jedná se prakticky o slovní zásobu postavy.

subjects.xml seznam a struktura studijních předmětů.

tasks.xml seznam testových otázek.

Kapitola 6

Testování aplikace

Nedílnou součástí vývoje aplikací i her je bez pochyby testování. V případě hry FAVagotchi tomu není jinak. Hned po výběru technologií, naprogramování základní logiky a spuštění aplikace na emulátoru Windows Phone SDK, bylo potřeba vybrat minimálně jedno reálné zařízení, na kterém by se hra dala otestovat. Hra také obsahuje mnoho časových úseků a událostí, ve kterých se postupně kontrolují jednotlivá pravidla pro zdárný průchod hrou, tedy životospráva a studijní výsledky. Pro účely tohoto testování jsem do aplikace postupně umístil několik ovládacích prvků, které manipulují s časovými závislostmi.

6.1 Výběr zařízení

Vzhledem k velké nabídce dnešních chytrých telefonů není zrvona jednoduché vybrat zařízení, na kterém by se celý projekt FAVagotchi dal otestovat. Bylo testování na katedrou zapůjčeném zařízení HTC Mazaa, tento telefon je dodáván s operačním systémem Windows Phone 7.5 Mango. Dle oficiálního prohlášení výrobce HTC není určen k volnému prodeji, je k dostání pouze pro developerské účely a to v následující konfiguraci v tabulce 6.1.

Procesor	1 GHz
RAM	576 MB RAM
Interní paměť	8 GB
Rozlišení	TFT, 800 px x 480 px, 3.8 palců
Operační systém	Windows Phone 7.5 Mango

Tabulka 6.1: Technické parametry HTC Mazaa

Další zařízení, na kterém jsem měl možnost testovat hru FAVagotchi, posloužil chytrý telefon Nokia Lumia 800 16GB Cyan, který mi poskytl kamarád téměř na dva měsíce vývoje. Technické parametry telefonu viz. následující tabulka 6.2

Procesor	1,4 GHz
RAM	576 MB RAM
Interní paměť	16 GB
Displej	AMOLED, 800 x 480 px, 3.8 palců
Operační systém	Windows Phone 7.5 Mango

Tabulka 6.2: Technické parametry Nokia Lumia 800 16GB Cyan



Obrázek 6.1: HTC Maza a Nokia Lumia 800 16GB Cyan

6.1.1 Komunikace s reálným zařízením

Aby bylo možné testovat aplikaci na reálném zařízení, je nutné provést řadu kroků, které tuto kompilaci z Visual Studio na Windows Phone zařízení umožní. Tyto kroky budou podrobněji popsány v příloze č. 1 kapitole Instalační příručka. Oba dva testované telefony na obrázku 6.1 komunikují s počítačem pomocí micro USB kabelu.

Po načtení a instalaci hry do obou dvou telefonů se hned na první pokus zdálo vše v pořádku. Od emulátoru Windows Phone se hra na těchto telefonech nelišila prakticky v ničem. Mapové podklady, animace pohybu

postavičky ani výpočetní rychlost aplikace se nijak výrazně od emulátoru nezměnily. Jediným viditelným rozdílem je mnohem snadnější dotykové ovládání hry a mnohem větší rozlišení než na monitoru PC. Ve výsledku hra na telefonu vypadala mnohem lépe než jsem očekával.

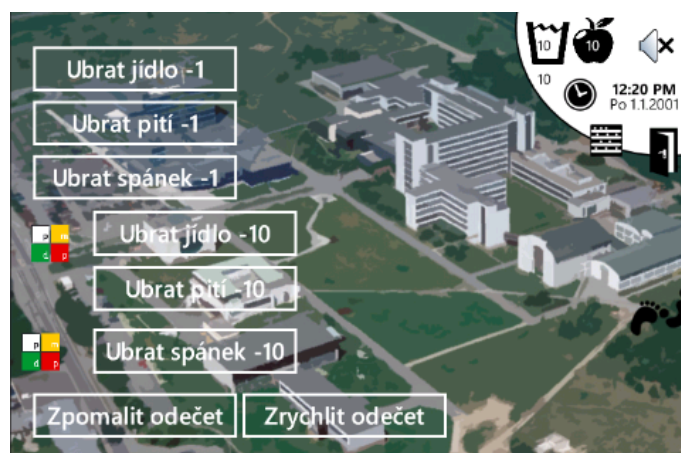
U obou zařízení je instalován operační systém Windows Phone 7.5 Mango. Pro Windows Phone 8 aplikace nebyla podrobena dlouhodobějšímu testování z důvodu dostupnosti této platformy a její chování tedy nemusí vyhovovat požadavkům scénáře a vývoje hry. Samozřejmě lze předpokládat kompatibilitu s nižšími verzemi OS.

6.2 Testování běhu aplikace

Osobně bych zde rád uvedl, že testování aplikace tohoto typu je poměrně složité. Hra není ve stylu sbírání bodů či snaze postoupit do další úrovně. Proto je nutné testovat události hlavně simulovanými časovými posuny. Toto testování bylo v rámci vývoje však nad moje síly i časové možnosti, proto mi s ním skrze emulátor pomáhal úzký okruh přátel. Rád bych jim tímto velmi poděkoval.

6.2.1 Životospráva

Po uplynutí každé časové jednotky (v normálním režimu každých 10 reálných vteřin, ve zrychleném jednu vteřinu), se odečítá stav jídla, pití i spánku. Aby bylo možné rychle a efektivně testovat události závislé na tomto odpočtu, je třeba s ním manipulovat, tuto manipulaci jsem zajistil umístěním ovládacích prvků, které jsou ilustrovány níže na obrázku 6.2.



Obrázek 6.2: Manipulace odpočtu životosprávy

6.2.2 Testování studijních podmínek

Opět jako je tomu u testování odpočtu životosprávy jsem na herní obrazovku umístil ovládací prvky, které mohou posunout herní čas o určité úseky různé délky. V závislosti na těchto časových posunech jsem poté mohl testovat střídání semestrů, přeskoky víkendových dní, začátek i konec hry, počátky přednášek a zkoušek a například i studijní aktivitu. Viz. obrázek 6.3



Obrázek 6.3: Posouvání herního času

6.3 Dosavadní úspěchy

Hra byla představena široké veřejnosti na dni otevřených dveří Fakulty aplikovaných věd dne 29. 1. 2014, kde se mohla pyšnit velikou oblibou jak u studentů zájímavých se o studium v jednom z oborů, které fakulta nabízí, tak i u studentů, kteří již na fakultě působí. Mnoho dotazů a připomínek si také neodpustila i řada univerzitních pracovníků, zájímavých se zejména o jazyk, ve kterém je hra programována a kde si mohou hru stáhnout a zahrát. Součástí prezentace bylo i představení oficiálního loga hry FAVagotchi 6.4.



Obrázek 6.4: Logo hry FAVagotchi

Kapitola 7

Uživatelská dokumentace

7.1 Instalace

V tuto chvíli je hru možné nainstalovat skrze vývojové prostředí Microsoft Visual Studio 2010. Hra se dá instalovat na emulátor Windows Phone v počítači, i na reálné zařízení. Celý tento instalační proces je popsán v příloze č.1.

V budoucnu by hra měla být publikována v oficiálním AppStoru společnosti Microsoft, kde bude veřejně ke stažení a instalaci pro Windows Phone zařízení.

7.2 Hardwarové a softwarové požadavky

Pro úspěšnou instalaci a spuštění hry, není nutné vlastnit chytrý telefon s vysokými technickými parametry. Hra FAVagotchi je vytvořena primárně v jazyce C# a XAML a není nijak výrazně náročná na zařízení na kterém je instalována.

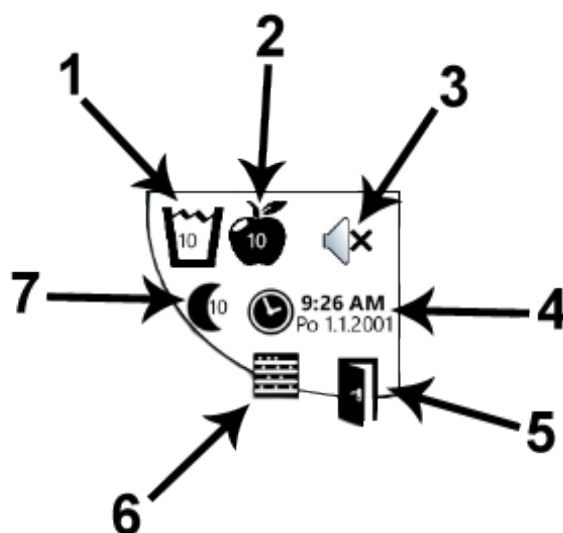
Minimální hardwarové a softwarové požadavky v tabulce 7.1, jsou uvedeny v souvislosti s telefony, na kterých byla aplikace testována.

Procesor	0,25 GHz
RAM	128 MB RAM
Displej	alespoň 800 x 480 px, 3.8 palců
Operační systém	Windows Phone 7.5 Mango a vyšší

Tabulka 7.1: Minimální hardwarové a softwarové požadavky

7.3 Ovládání hry

Všechny ovládací a informační prvky hry jsou obsaženy v info panelu v pravém horním rohu obrazovky. Viz. obrázek 7.1.



Obrázek 7.1: Herní info panel

- **1.** Indikátor pití - informuje o stavu tekutin, pokud postava tento stav průběžně nedoplňuje v menze, bufetech, nebo některém z automatů, může zemřít na dehydrataci - hra končí.
- **2.** Indikátor potravy - má naprosto stejný význam jako stav tekutin. Vždy když klesne na nulu, FAVák umírá na vyhladovění - hra končí.
- **3.** Obsluha doprovodné hudby - spuštění / vypnutí MIDI hudebního podkladu hry.
- **4.** Herní čas - informuje hráče o čase ve hře, tento informační prvek je velmi důležitý při hlídání začátku zkoušek, přednášek a cvičení.
- **5.** Hlavní menu - po kliknutí na tento ovládací prvek se hra uloží a hráč je přesměrován do hlavního menu ze kterého lze pak snadno hru opustit.

- **6.** Studijní profil - Hráč si zde může prohlédnout svůj rozvrhový plán, kde se u každého předmětu může dozvědět svou aktuální docházku, sledovat vypsání termínů zkoušek pro daný semestr a kontrolovat si výsledky těchto zkoušek.
- **7.** Indikátor spánku - stejně jako je tomu u pití a potravy, jakmile klesne úroveň spánku na nulovou hodnotu, hra končí z důvodu nedostatku spánku.

Kapitola 8

Možná rozšíření

Po návrhu i při samotném vývoji celé aplikace jsem přemýšlel, jak by se hra dala v budoucnu rozšířit a vylepšit. V tuto chvíli je to možné hned několika způsoby.

8.1 Datová rozšíření

Ta spočívají především v obohacení XML souborů o další data. Ty lze rozšířit dle následující tabulky 8.1.

XML / <code>exams.xml</code>	rozpis zkoušek pro daný předmět. Stačí přidat jen další termín zkoušky v DOM struktuře a hra se automaticky přizpůsobí a načte termín zkoušky do studijního profilu.
XML / <code>sentences.xml</code>	v tomto souboru je reprezentována každá věta ve hře pomocí klíče a následného textu dané věty. Slovní zásoba FAVáka se dá rozšířit pouhým přidáním dalšího elementu <code>text</code> .
XML / <code>tasks.xml</code>	testové otázky. Rozšířit tyto otázky je možné pouhým přidáním zápisu do struktury XML souboru. Pro každou otázku jsou k dispozici čtyři odpovědi.
XML / <code>subjects.xml</code>	seznam a struktura studijních předmětů. Lze rozšířit o libovolný počet v daném tří letém rozsahu studia. Při vkládání dalších předmětů je však nutné hlídat si kolidující rozvrhové časy jednotlivých předmětů. Dále musí být dodržena datová souvislost s tímto předmětem. Je proto nutné mu dodat termíny zkoušek a zkouškové otázky ve výše vyjmenovaných souborech.

Tabulka 8.1: Tabulka rozšiřitelných zdrojových XML souborů

8.2 Grafika hry

Grafické zpracování hry je v tuto chvíli na základní úrovni. Mapové podklady budov jsou zpracovány velmi stručně a jistě by se daly rozšířit o reálné grafické 2D prvky, které by hru dozajista velmi obohatily. Podklady pro venkovní prostory kampusu na Borech jsou zpracovány o něco lépe. Na tyto prostory jsem použil leteckou fotku areálu a převedl ji v grafickém programu do barevného náčrtu. Jednotlivé budovy, do kterých se pomocí dotykového displeje dá herní postavíčka poslat jsou prakticky výřezy z této letecké fotky použité jako pozadí jednotlivých XAML elementů.

Pro úpravu těchto mapových podkladů a grafických prvků hry, je nutné provést několik kroků, které to umožní.

8.2.1 Změna mapových podkladů

- Splnit podmínky nutné ke kompilaci a načtení projektu do Visual Studia 2010. Tyto podmínky a návod jsou popsány v příloze 1. kapitole Instalační příručka.
- V hlavním kořenovém adresáři `PhoneApp1` se ve složce `maps` nachází jednotlivé mapové podklady.
- Pokud chceme například změnit mapový podklad pro budovu FAV, vybereme tedy soubor `fav.xaml`.
- Dále do složky `images` importujeme obrázek mapového podkladu o minimálních rozměrech 800px x 481px.
- Pokud nechceme v souboru `fav.xaml` nic měnit a pouze vyměnit tento mapový podklad. Je třeba tento obrázek nahrát pod stejným názvem jako má jeho předchůdce, tedy `fav.png` a nahradit ho.
- V případě, že chceme předchozí mapový podklad zachovat, a hru pouze rozšířit o nový, je třeba změnit cestu ke zdrojovému souboru. To provedeme zásahem do zdrojového kódu XAML v souboru `fav.xaml`, u elementu `mainGrid` a jeho vlastnosti `ImageBrush`. Zde změníme `ImageSource`, tak, že přepíšeme cestu k našemu novému mapovému podkladu. Například `ImageSource=/PhoneApp1;component/images/novy-podklad.png`

Úprava grafiky a většiny prvků aplikace je velmi jednoduchá. Stačí pouze manipulovat s objekty ve zdrojových souborech s koncovkou `.xaml`. To znamená, že můžeme libovolně manipulovat s pozicemi různých budov, přednáškových

místností a ovládacích prvků aniž by to ohrozilo chování a logiku aplikace. Samozřejmě musí být tyto objekty zachovány, v případě odstranění některého z nich, nepůjde aplikace ani zkompileovat a Visual Studio bude hlásit chybu hned po smazání tohoto objektu i při samotné kompilaci.

8.3 Umělá inteligence a životospráva

Chování studenta by se určitě dalo obohatit o více simulací lidského chování, dodat mu nálady a únavu. Dalším prvkem hry, který se spíše nevešel do samotného návrhu, je generátor front a zdržení. Ten by generoval různé časové prodlevy mezi akcemi hráče. Při poledním vstupu do menzy se například může stát, že vznikne fronta na jídlo a na jejím základě FAVák nemusí stihnout následující přednášku, zkoušku apod.

Co se týká životosprávy, tu bych v budoucnu chtěl rozšířit například o náhled studentovy postavy a aktuálního zdravotního stavu. Pokud se FAVák bude neustále stravovat v bufetu, bude se to muset odrazit na jeho zdraví, postavě a poté i rychlejším odpočtu prvků životosprávy.

8.4 Rozšíření scénáře a příběhu hry

Celý příběh je navržen v rámci tří let bakalářského studia, určitě by mohl být rozšířen dál včetně reálného zápisu na počátku každého studijního roku. Hra například přeskakuje víkendy, to by se stávat nemuselo a o víkendy by hra mohla být doplněna o různé úkoly, sbírání bodů, hraní miniher a další, za které by si hráč vysloužil například možnost absence na zkoušce či přednášce. V roce 2015 se Fakulta aplikovaných věd přestěhuje do nové budovy NTIS/CTPVV, ta by se mohla ve hře určitě objevit také.

8.5 Více platforem

FAVagotchi je vyvíjeno primárně pro Windows Phone. Z hlediska porovnání funkčnosti na všech platformách bych si ji určitě chtěl zkusit naprogramovat například i na operačním systému Android nebo iOS.

Kapitola 9

Závěr

Cílem práce bylo navrhnout scénář hry pro mobilní zařízení odehrávající se v simulovaném čase. Tato hra navíc měla představovat vybrané činnosti studenta fakulty aplikovaných věd a mimo i aktivní účast na výuce a kompletní starost o životosprávu postavy. Navržená hra byla realizována a otestována na platformě Windows Phone, tedy v jazce C# a dále byla zmíněna její možná rozšíření v různých podobách.

Samotný vývoj hry byl velmi komplexní. Svou komplexností si celá práce získala zejména rozsáhlostí scénáře a XML dat, která byla do hry zanesena. Pro každý předmět bylo nutné vytvořit alespoň 6 otázek a vypsát alespoň 4 termíny na zkoušku z tohoto předmětu.

Pro vývoj celého projektu bylo zvoleno vývojové prostředí Microsoft Visual Studio 2010. Součástí testování hry byl použit emulátor Windows Phone dodávaný s knihovnou Windows Phone SDK a dva reálné mobilní telefony. V rámci tohoto testování byla navíc prozkoumána instalace hry na reálné zařízení skrze Zune software a registrace telefonu pomocí Windows Phone Developer Registration.

Na závěr bych rád uvedl, že cíle deklarované v zadání a na úvodu bakalářské práce byly splněny. Hra byla představena na dni otevřených dveří Fakulty aplikovaných věd 2014, kde zaujala potenciální studenty i studenty kteří již na fakultě působí.

Byl bych velmi rád, kdyby pokračováním této práce mohla být například realizace některých navržených rozšíření či vývoj celé hry pro jinou platformu.

Literatura

- [1] *Charles Petzold*
Programming Windows Phone 7
ISBN: 978-0-7356-4335
<http://fav.galosoft.cz/favagotchi.html>

- [2] *Joseph C. Rattz, Jr.*
Pro LINQ: Language Integrated Query in C# 2008
ISBN: 978-1-59059-789-7
<http://fav.galosoft.cz/favagotchi.html>

- [3] *Jeremy Likness*
Building Windows 8 Apps with C# and XAML (Microsoft Windows Development Series)
ISBN-13: 978-0321822161

- [4] *Adam Freeman, Matthew MacDonald, Mario Szpuszta*
ASP .NET 4 a C# 2010 - KNIHA 1 - tvorba dynamických stránek profesionálně
ISBN: 978-80-7413-131-8

- [5] *Adam Freeman, Matthew MacDonald, Mario Szpuszta*
ASP .NET 4 a C# 2010 - KNIHA 2 - tvorba dynamických stránek profesionálně
ISBN: 978-80-7413-131-5

- [6] *Andrew Troelsen*
Pro C# 5.0 and the .NET 4.5 Framework (Expert's Voice in .NET)
ISBN-13: 978-1430242338

- [7] *Windows Phone Dev Center*
DEV.WindowsPhone.com [on-line]
<https://dev.windowsphone.com>

- [8] *Microsoft Developer Network*
MSDN.Microsoft.com [on-line]
<https://msdn.microsoft.com>

- [9] *Microsoft Developer Network*
Introduction to LINQ [on-line]
<http://msdn.microsoft.com/en-us/library/bb397897.aspx>
- [10] *Microsoft Developer Network*
XAML for Windows Phone 8 [on-line]
<http://msdn.microsoft.com/en-us/library/windowsphone/develop/cc189036>
- [11] *Czech MSDN Blog* **Blogs.MSDN.com** [on-line]
<http://blogs.msdn.com/b/vyvojari>
- [12] *Komunitní server pro vývojaře* **Vyvojar.cz** [on-line]
<http://www.vyvojar.cz>
- [13] *Komunitní server pro vývojaře* **StackOverflow.com** [on-line]
[http://stackoverflow.com/questions/tagged/windows-phone-7 /](http://stackoverflow.com/questions/tagged/windows-phone-7/)

Seznam obrázků

3.1	Hlavní nabídka	10
3.2	Uvítací obrazovka	11
3.3	Herní prostředí - simulace vysokoškolského kampusu	11
3.4	Studijní plán	12
5.1	Grafické znázornění seznamu tříd	24
6.1	HTC Mazaa a Nokia Lumia 800 16GB Cyan	39
6.2	Manipulace odpočtu životospřávy	40
6.3	Posouvání herního času	41
6.4	Logo hry FAVagotchi	41
7.1	Herní info panel	43

Seznam tabulek

3.1	Tabulka časů pro normální herní režim	13
3.2	Tabulka časů pro zrychlený herní režim	14
4.1	Tabulka klíčových slov LINQ	19
4.2	Tabulka implementací LINQ	20
6.1	Technické parametry HTC Mazaa	38
6.2	Technické parametry Nokia Lumia 800 16GB Cyan	39
7.1	Minimální hardwarové a softwarové požadavky	42
8.1	Tabulka rozšiřitelných zdrojových XML souborů	45

Seznam algoritmů

4.1	Využití XAML prvku Storyboard pro animaci pohybu herní postavičky ve třídě “move”	17
4.2	Ukázka třídy SQLTables představující databázi a třídy gameData sloužící jako tabulka v této databázi	20
4.3	Ukázka z třídy SQLQuery reprezentující dotazy na databázi. Zde konkrétně zápis a výpis herních zdrojových dat	21
4.4	Dotaz vkládající do tabulky gameData informaci o časovém módu při načítání hry může vypadat například takto.	22
4.5	Ukázka z třídy XMLQuery reprezentující přístup k textovému generátoru	22
4.6	Soubor sentences.xml pro generování vět ve hře	23

Přílohy

Příloha 1. Instalační příručka

V této příloze se čtenář dozví, jak hru FAVagotchi otevřít a spustit na emulátoru Windows Phone, i na reálném zařízení.

Příloha 2. Náhledy hry

Příloha obsahuje jednotlivé náhledy na hru v různých dějových úsecích.

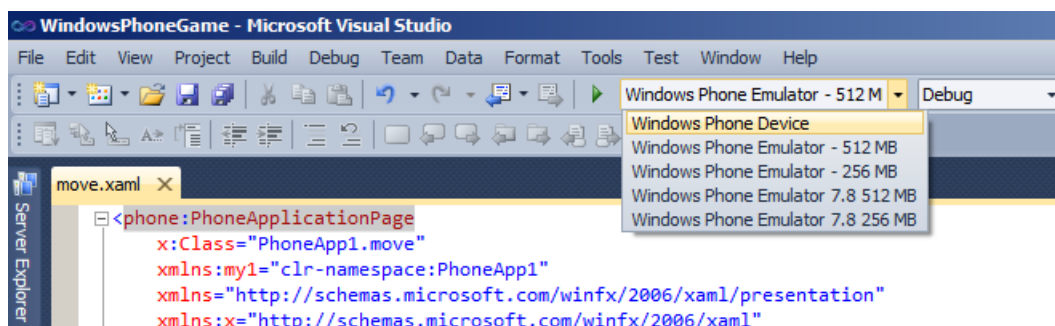
1. Instalační příručka

Prvním krokem k samotné instalaci hry FAVagotchi je stažení celého projektu. Ten je volně dostupný na adrese <http://fav.galosoft.cz/favagotchi.html>, kde se mimo jiné nachází i tato kompletní dokumentace včetně příloh a veškeré dostupné literatury týkající se použitých programovacích jazyků a technologií v tomto projektu.

Na adrese <http://dev.windowsphone.com/en-us/downloadsdk> je k dostání celá řada programových knihoven pro Windows Phone. Na konci tohoto seznamu je SDK 7.1, tato knihovna je podmínkou kompilace celého projektu. Po stažení a instalaci se do Visual Studia 2010 automaticky natáhne vývojové prostředí pro Windows Phone.

Projekt FAVagotchi lze otevřít pomocí spouštěcího souboru `FAVagotchi.sln` v hlavní složce projektu. Ten spustí Visual Studio 2010 a načte do něj veškeré součásti projektu, které jsou dostupné po spuštění funkce Solution Explorer.

Následuje samotná kompilace a spuštění hry. Na výběr je ze dvou možností. Emulátor nebo reálné zařízení. Tento výběr lze provést v horním panelu hned vedle zelené ikony spouštějící kompilaci. Viz. obrázek 1.



Obrázek 1: Výběr typu kompilace

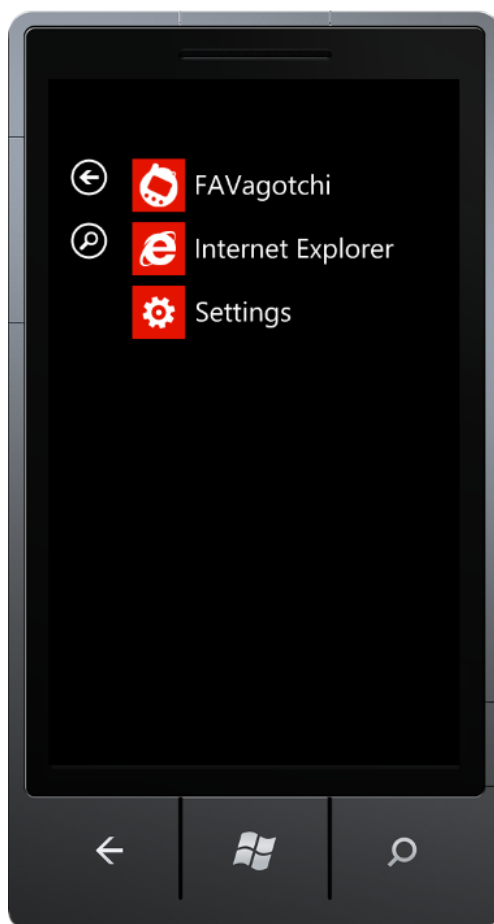
Pro každý typ kompilace je nutné provést odlišné kroky, proto už jsou následné pokyny oddělené.

Windows Phone Emulátor

Pro spuštění hry na emulátoru stačí ve výše zmiňovaném horním panelu pouze vybrat jednu z možností Windows Phone emulátoru. Tyto možnosti se liší pouze ve virtualizaci velikosti RAM paměti a verzi operačního systému.

Po výběru varianty lze kompilaci projektu na emulátor spustit klávesovou zkratkou F5 nebo kliknutím na zelené tlačítko hned vedle tohoto výběru. Celá instalace probíhá na tomto emulátoru naprosto shodně s reálným zařízením a netrvá dlouho.

S tímto emulátorem navíc lze manipulovat jako s reálným zařízením, tj. instalovat do něj aplikace, procházet operační systém apod. Tyto úkony lze provádět pouze do vypnutí emulátoru nebo celého PC. Po znovuspuštění se toto virtuální zařízení chová totiž jako po návratu do továrního nastavení.



Obrázek 2: Windows Phone Emulator

Reálné zařízení Windows Phone

Tento postup instalace hry je mírně komplikovanější. Nejprve je nutné mít účet na serveru Live.com, respektive na portálu zabývající se samotným vývojem aplikací pro platformu Windows Phone. Portál Windows Phone Dev Center se nachází na adrese <http://dev.windowsphone.com/en-us> .



Obrázek 3: Windows Phone Dev Center

Po registraci na tomto portálu je nutné mít telefon vyhovující minimálním hardwarovým a softwarovým požadavkům, které jsou uvedeny v **Kapitole 6. - Uživatelská dokumentace** .

Dále je nutné mít instalovaný Zune Software, komunikující se zařízením a sloužící k synchronizaci dat a souborů mezi počítačem a telefonem. Tento program lze volně stáhnout na adrese <https://www.windowsphone.com/cs-cz/how-to/wp7/zune-software> . Jak vypadá program Zune lze nahlédnout v následujícím obrázku 4.



Obrázek 4: Zune Software

Po těchto základních krocích je nutné provést samotnou registraci zařízení k developerskému účtu Windows Phone Dev Center. Společně s knihovnou Windows Phone SDK 7.1 a emulátorem Windows Phone se do počítače nainstaluje služba Windows Phone Developer Registration. Pro rychlejší přístup

k této službě je určitě lepší si ji v počítači najít pomocí funkce *Prohledat programy a soubory*.

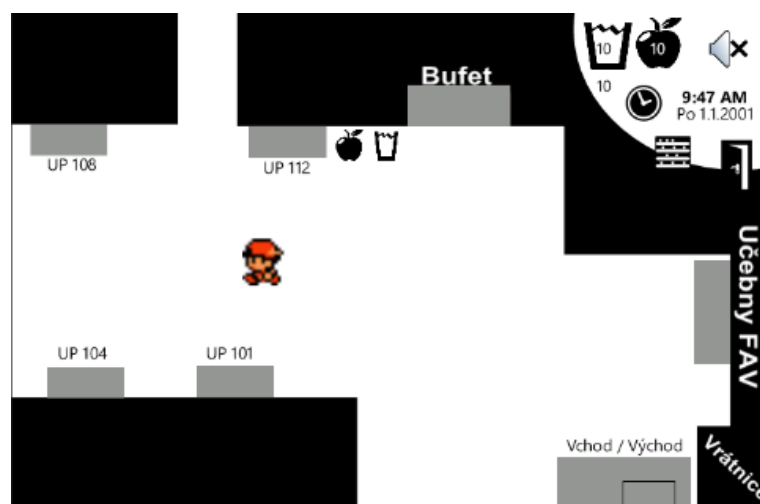
Společně s touto službou musí být spuštěn Zune Software, prostřednictvím kterého telefon s počítačem komunikuje. Celý proces registrace zařízení proběhne hned poté, co uživatel vyplní své přihlašovací údaje, které zadal při registraci do Windows Phone Dev Center.



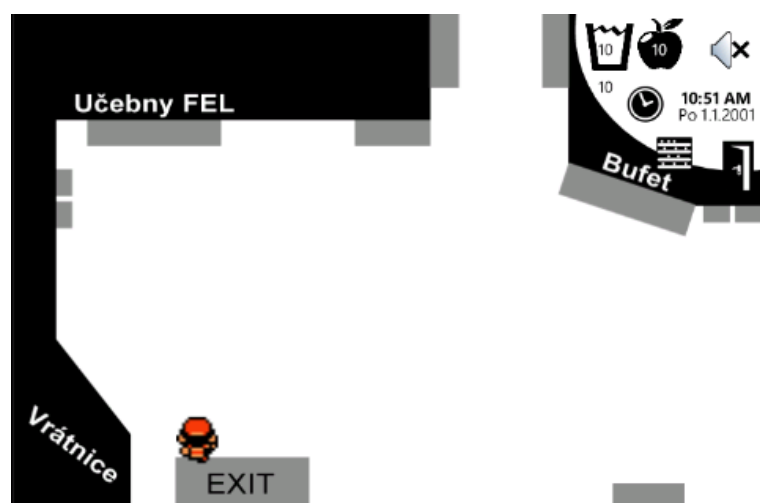
Obrázek 5: Windows Phone Developer Registration

Po těchto krocích by měl být chytrý telefon připraven k instalaci FAVa-gotchi.

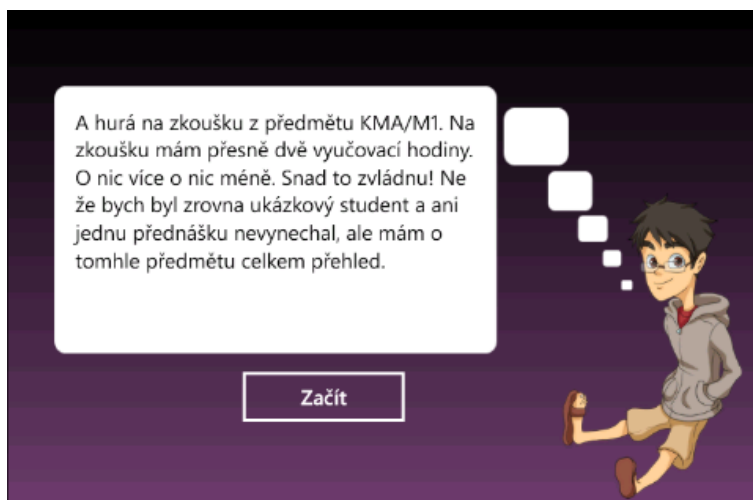
2. Náhledy hry



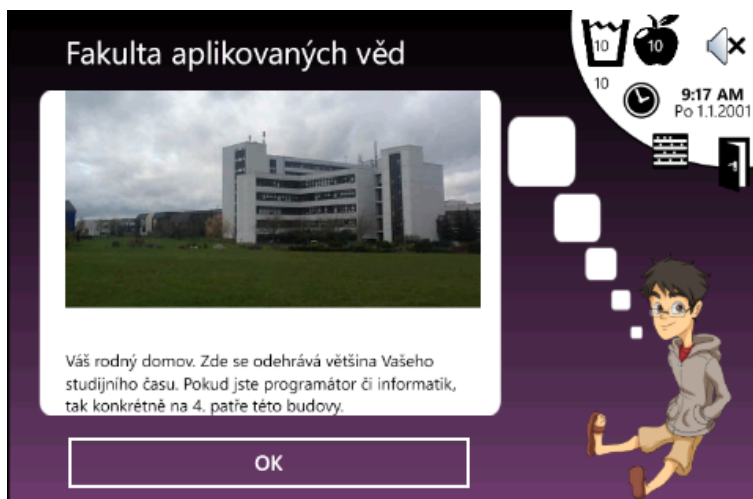
Obrázek 1: Budova FAV



Obrázek 2: Budova FEL



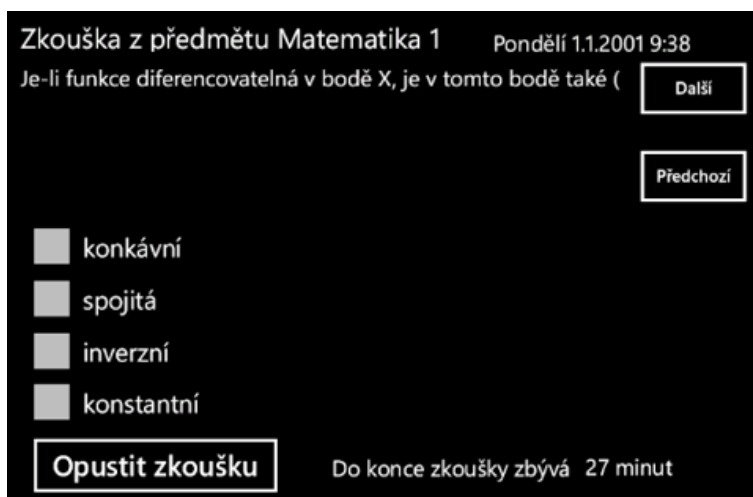
Obrázek 3: Příchod na zkoušku



Obrázek 4: Uvítání po příchodu do budovy



Obrázek 5: Úvodní obrazovka



Obrázek 6: Náhled zkouškové otázky