Základní informace

Jméno projektu	Implementace hry Go a související funkcionality
Zkratka	Omega Go
Vedoucí	Jakub Gemrot <jakub.gemrot@gmail.com>
Konzultanti	-
Anotace	Projekt si klade za cíl vytvořit aplikaci, pomocí které může uživatel hrát deskovou hru Go. Aplikace umožní uživateli hrát proti umělé inteligenci, proti ostatním hráčům (lokálně nebo přes internet), řešit hádanky (problémy života a smrti) a učit se vhodné tahy. Další funkce hry jsou popsané v těle dokumentu.

Motivace

Go je abstraktní strategická desková hra, která se dosud těší oblibě po celém světě, s důrazem na země východu. První server pro hraní Go přes internet vznikl v roce 1992 a od té doby stále více hráčů hraje tuto hru elektronicky.

Existuje mnoho implementací hry. Různé implementace podporují různé funkce a fungují na různých platformách. Provedli jsme předběžný průzkum devatenácti z těchto aplikací a jejich podrobná analýza bude součástí naší práce.

Hra, kterou vytvoříme, bude mít především následující výhody oproti ostatním aplikacím:

- Bude fungovat nativně na platformě Universal Windows Platform (dále UWP). Doposud všechny nedesktopové aplikace pro Windows podporují jen velmi málo funkcí a nejsou stabilní a vyladěné.
- Neexistuje žádná aplikace – na žádné platformě – která by nabízela přívětivý úvod ke hře pro začátečníky (webová aplikace OGS má k tomuto blízko, ale funguje pouze s připojením k internetu a vyžaduje registraci). Naše aplikace bude vypadat jako *hra*, a tedy bude smysluplná pro nováčky, a přitom bude obsahovat i funkce pro pokročilé.
- Neexistuje žádná aplikace, která by umožnila uživateli se připojit k několika různým serverům – každá aplikace je spojená nejvýše jedním serverem.

Popis projektu

Go je tahová desková hra, ve které se hráči střídají v pokládání kamenů na hrací plochu. Když kameny jednoho hráče zcela obklíčí skupinu soupeřových kamenů, soupeřova skupina je odstraněna ze hry. Zjednodušeně, na konci hry se stane vítězem ten hráč, který obsadí větší plochu herního pole. Plná pravidla jsou lehce složitější a existuje několik různých pravidlových systémů, které se liší v některých detailech.

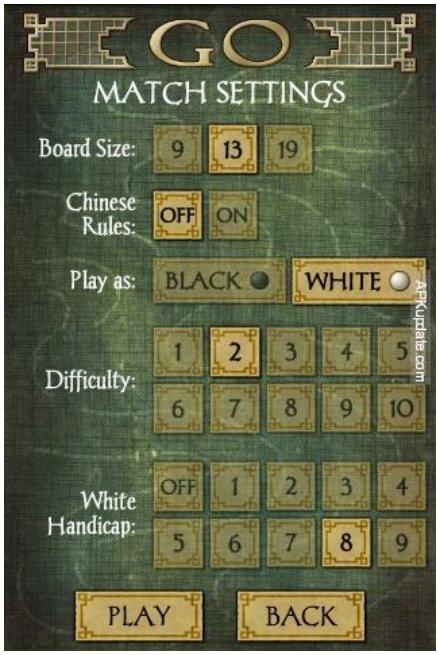
Obrázek 1: Deska Go v průběhu hry

Náš projekt, zatím nazvaný *Omega Go*, je počítačový program, který uživateli umožní hrát tuto hru,

s vynucováním pravidel, proti umělé inteligenci nebo proti ostatním hráčům lokálně nebo přes internet. Bude také nabízet několik dalších doplňkových funkcí, které popíšeme ve zbytku této sekce. Ještě podrobnější specifikaci, včetně návrhu obrazovek uživatelského rozhraní, máme v odděleném dokumentu.

Obrázky v této sekci jsou ilustrativní a pocházejí z různých programů, pomocí nichž se dá hrát Go.

Základní hra *hotseat*. Základní a nejpřístupnější část této hry bude lokální hra *hotseat*, při které se dva hráči budou střídat v pokládání kamene na jednom zařízení. Tohoto módu budou hráči využívat nejčastěji, když jsou fyzicky u sebe a chtějí si spolu zahrát. Hra jim umožní nastavit základní vlastnosti hry (tj. zvolit systém pravidel, nastavit handicap, velikost plochy a režim časovače) a poté je nechá spolu hrát, a bude vynucovat pravidla (tj. nepovolí zakázané tahy). Na konci hry pak hra pomůže hráčům s počítáním výsledného skóre.



Obrázek 2: Nastavení hry v programu AI Factory Go

Jádro hry. Hra zobrazí uživatelům herní plochu a umožní jim intuitivně pokládat kameny s minimalizací pravděpodobnosti překliku (i v prostředí bez přesného ukazatele myši – dotykové obrazovky). Hra uživateli zabrání provést neplatné pohyby podle zvoleného pravidlového systému (tj. opakovaný pohyb, pohyb mimo pořadí a sebevražedný pohyb). Jakmile se oba hráči shodnou na ukončení hra, aplikace se pokusí odhadnout skóre každého hráče a dá hráčům možnost označit kameny jako mrtvé nebo živé, aby vylepšili odhad aplikace. Hra umožní hráčům pokračovat ve hře nebo potvrdit výsledek, který se pak stane finálním výsledkem hry.

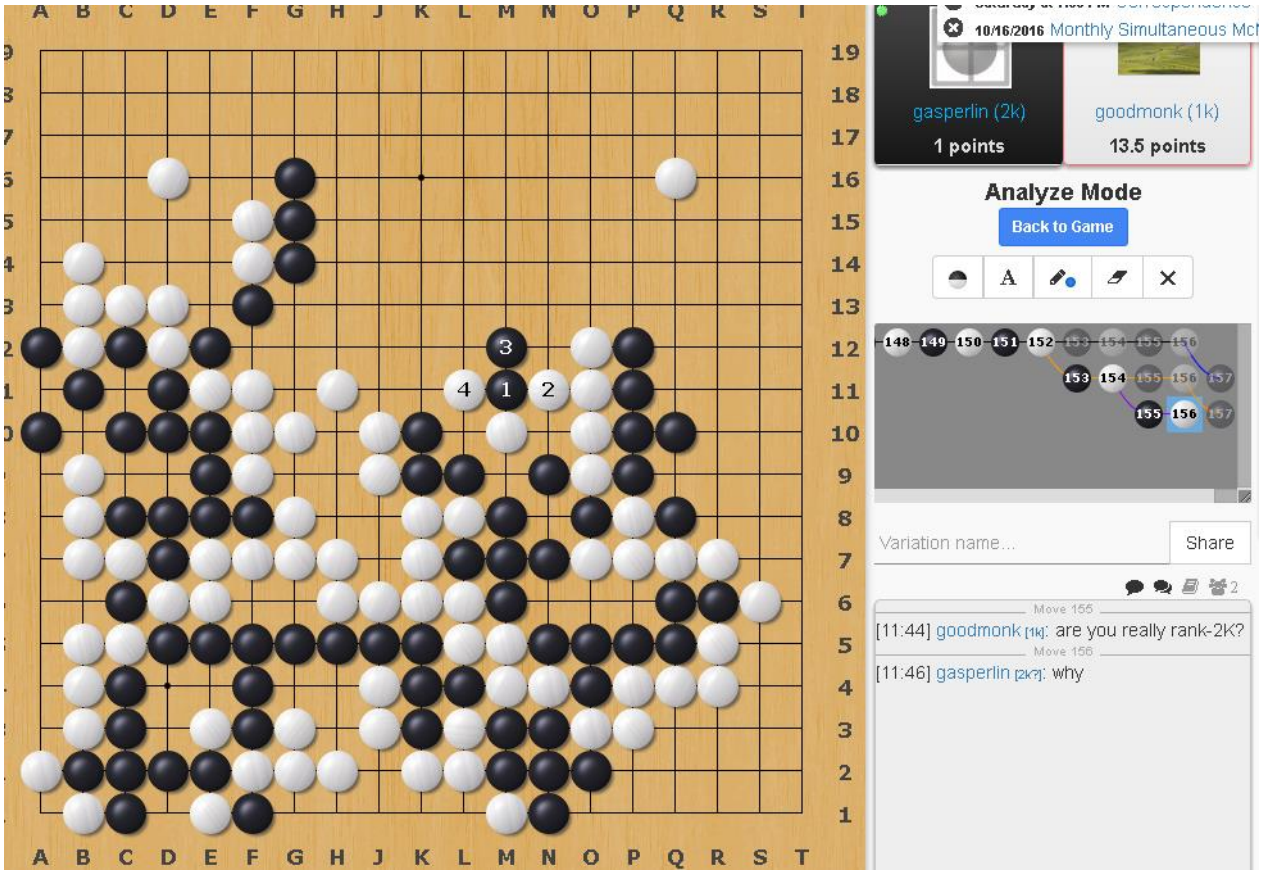
Obrázek 3: Obrazovka se statistikami ve hře AI Factory Chess

Udržování statistik: Hra udržuje pro hráče statistiky jeho her proti umělé inteligenci, popř. jeho úspěchy ve hře jednoho hráče (singleplayer) a některé další údaje, které mohou být zajímavé. Hra také umožňuje hráči, aby se podíval na svoje statistiky udržované na multiplayer serverech, kam je přihlášen.

Hra proti lokální umělé inteligenci: Ve hře budeme mít zabudovanou umělou inteligenci, proti které bude moci uživatel hrát. Zde bude záležet na tom, jak snadné bude portovat existující umělou inteligenci, jako například *Gnu Go*, na platformu UWP. Pokud to bude realistické, provedeme port a tato umělá inteligence bude součástí programu. Pokud se to ukáže být příliš složitým, naprogramujeme vlastní umělou inteligenci, která ovšem nebude tak silná jako například zmíněné *Gnu Go*. V každém případě tato umělá inteligence bude mít nastavitelnou úroveň obtížnosti. Sílu a rychlost umělé inteligence změříme a uvedeme v dokumentaci. Měření provedeme hraním na některém online serveru.

Doporučování tahů: Umělá inteligence bude schopná hráči radit, jaký tah by měl provést. Hráč bude moci nastavit, jakou sílu má umělá inteligence vykazovat při doporučování tahů. Pokud umělou inteligenci budeme programovat sami, je ovšem možné, že tahy, které bude doporučovat, nebudou dostatečně silné.

Režim analýzy hry: Po hře, ale i během hry, může hráč vstoupit do *režimu analýzy*, ve kterém může plánovat svoje i soupeřovy tahy dopředu. Hra pak nepředstavuje časovou osu, ve které se hráči střídají s pokládáním kamenů, ale spíše „*herní strom*“, kde hráč může v režimu analýzy vytvářet další větve a pohybovat se mezi nimi. V režimu analýzy lze také pokládat nebo odstraňovat značky na herní plochu a pro každý tah vkládat a číst komentáře. Analyzovat lze jak lokální, tak internetové hry.



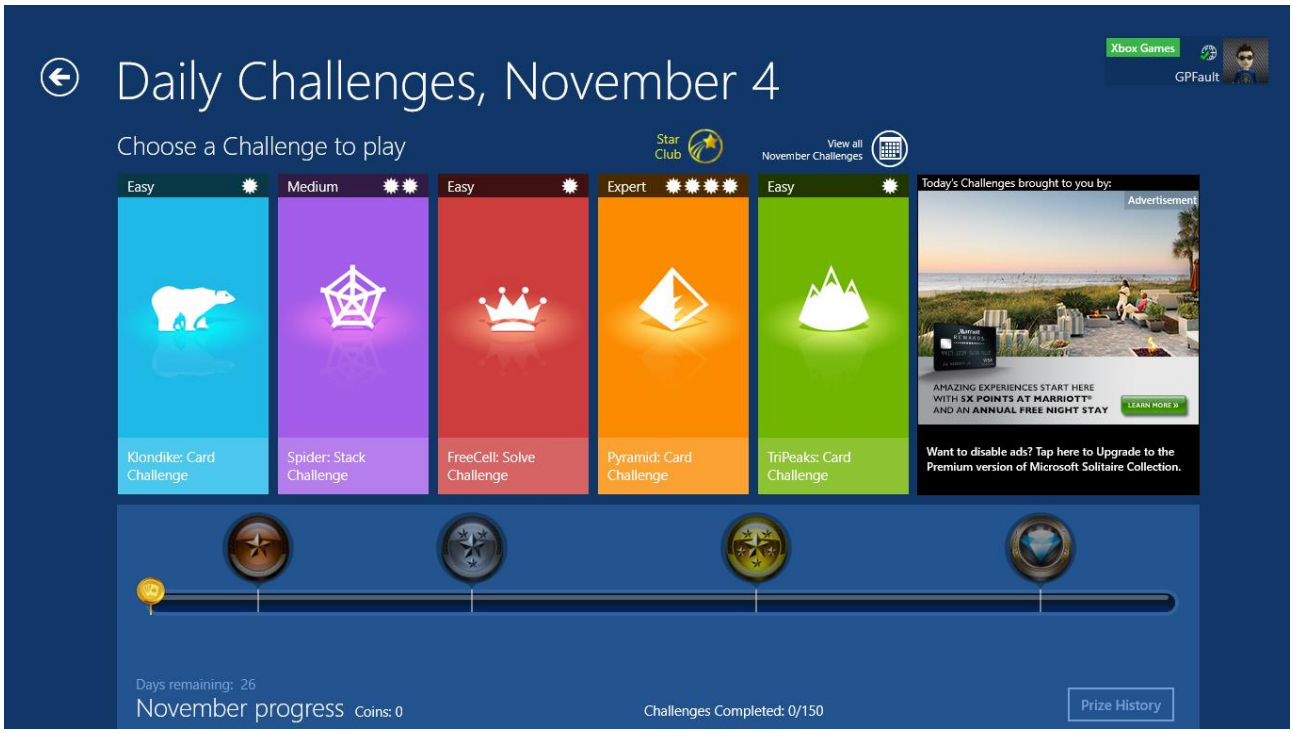
Obrázek 4: Režim analýzy ve hře online-go.com (OGS)

Soubory SGF: Soubory ve formátu SGF („smart game format“) obsahují jeden nebo více herních stromů. Mohou tak obsahovat rozehranou hru, jeden nebo více problémů života a smrti nebo třeba profesionální komentovanou hru. Naše aplikace nebude podporovat všechny SGF soubory (nemá například smysl podporovat soubory, které obsahují hry šachu), ovšem bude přinejmenším schopná ukládat herní stromy z režimu analýzy jako SGF soubory a načítat tyto soubory, anebo soubory odjinud, do svého režimu analýzy. Soubor SGF je textový soubor ve speciálním formátu. Aplikace bude také zaregistrována jako handler formátu SGF, takže bude možné kdykoliv otevřít aplikaci a načíst hru přímo spuštěním souboru tohoto typu.

Online režim: Hráč se může připojit k alespoň dvěma různým online serverům. Tato část programu bude rozšiřitelná a nebude obtížné později přidat přístup k dalšímu serveru. Každý server má

odlišný způsob komunikace (většinou jde o nějaký druh textového protokolu přes TCP spojení) a nabízí lehce odlišnou funkcionalitu. Hráč se pomocí této aplikace přihlásí k vybraným serverům a může na nich zahájit nebo přijmout hru. Hra pak bude komunikovat se serverem a zobrazovat na ploše tahy soupeře a vynucovat pravidla tak, jak je vynucuje server. Nebudeme podporovat všechny funkce určitých pokročilých serverů jako je OGS, ale minimálně bude hráč schopen navazovat a přijímat hry, chatovat se soupeřem, pokládat kameny a provádět jiné herní akce, účastnit se fáze počítání skóre, a zobrazit si svoje statistiky.

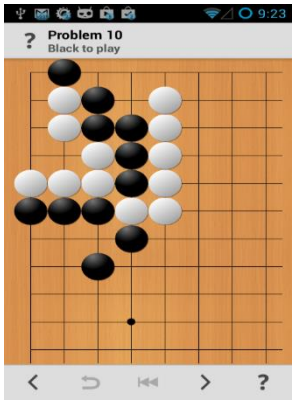
Režim jednoho hráče: Budeme poskytovat nějaké další *gamifikované* funkce pro sólo hru, aby pro hráče bylo zábavnější hrát jen samostatně, i pokud nemá zájem se stát velkým expertem v Go. Tato podpora hra jednoho hráče bude přinejmenším ve formě *denních výzev*, kdy hra bude nabízet hráči různé úkoly (stylu „Vyhraj 3 hry za bílého.“), které hráč může splnit, aby získal *body*.



Obrázek 5: Denní výzvy ve hře Microsoft Solitaire Collection

Je možné, že se v průběhu projektu ukáže, že zvládneme do hry dodat mnohem více funkcí pro hru jednoho hráče, včetně například příběhové kampaně. V takovém případě bychom se mohli rozhodnout denní výzvy neimplementovat. V každém případě ovšem tato komponenta bude alespoň tak velká a složitá, jako jsou denní výzvy v hrách od firmy Microsoft na Windows Store.

Tsumego: Hráč bude moci ze souboru načíst *problémy života a smrti* („tsumego“), hádanky, ve kterých má hráč za úkol pokládat kameny tak, aby vyřešil nejlepším možným způsobem zadanou situaci na ploše. Tato hra bude disponovat nějakými takovými problémy a další bude možné načíst z uživatelem zadaného souboru. U každého problému je přednastaveno jeho správné řešení, vůči kterému se hráčovy tahy porovnávají.



Obrázek 6: Řešení problému ve hře Tsumego Pro

Hráčovy výsledky z řešení problémů tsumego budou započítávány do jeho

statistik.

Tutoriál: Hráč by měl být schopen nevědět o hře *vůbec nic,* interagovat s touto hrou, a po chvíli umět hrát Go relativně dobře. Za tímto účelem bude hra obsahovat režim tutoriálu, který hráče v sérii interaktivních cvičení naučí pravidla a základní principy. Naší inspirací bude tutoriál OGS, ovšem náš půjde ještě trochu více do hloubky.

Vícejazyčnost: Aplikace bude fungovat ve více jazycích. Začne v jazyce operačního systému (pokud ten není podporován, výchozím jazykem bude angličtina), ale uživatel bude mít možnost jazyk aplikace přepnout. Abychom dokázali, že aplikace je vícejazyčná, budeme podporovat alespoň angličtinu a češtinu. Vícejazyčnost je důležitá pro budoucnost obzvláště vzhledem k tomu, že je hra Go populární spíše ve východních zemích a mnoho tamních hráčů nerozumí angličtině.

Režim diváka: Hráč se může přihlásit k multiplayerovému serveru (např. IGS), zobrazit si seznam probíhajících her a přidat se k některé z nich jako divák. Divák vidí hráče pokládat kameny, vidí jejich chat a může s ostatními diváky komunikovat v odděleném chatu pro diváky, který je skryt hráčům, ale je sdílen s ostatními diváky, kteří se k serveru připojili přes jiné klienty. Tuto komunikaci („kibitz chat“) nesmí vidět hráč, protože může obsahovat strategické rady.

Platforma, technologie

Hra bude vyvíjena v C#/.NET, primárně pro platformu UWP. Aplikace UWP se dají spustit na zařízeních se systémem Windows 10 – převážně počítačích, ale také tabletech a mobilech – to jsou zařízení, která chceme hlavně podporovat. Volitelně můžeme v průběhu přidat ještě další platformy (zajímá nás hlavně Xbox One).

V každém případě je naším cílem co nejlépe oddělit frontend od jádra aplikace, aby přidání dalšího frontendu nebylo složité. Toho docílíme částečně oddělením projektu jádra od projektu UWP a návrhem aplikace dle vzoru Model-View-ViewModel (MVVM). Ten poskytuje dobré oddělení na platformě závislého UI od na platformě nezávislého jádra, které lze díky formátu knihovny Portable Class Library (PCL) použít na různých platformách bez nutnosti zásadních změn. Pokud nám zbyde čas, další frontendy, o kterých uvažujeme, jsou Android a desktop (pro Windows a Linux, pravděpodobně pomocí frameworku *Monogame*).

Pro správu týmu a zdrojového kódu využijeme služeb Visual Studio Team Services (VSTS). Kód projektu bude uložen v Gitu, který je službou nabízen. Pokusíme se ve VSTS vést jednotlivé úlohy při práci a udržovat pravidelné sprinty. Na projekt se budeme také snažit aplikovat *continuous integration*, která je součásti VSTS pro automatizované buildy a testování.

Odhad náročnosti

Počet řešitelů: 4 (všichni členové mají magisterský obor *Softwarové a datové inženýrství*)

Termín dokončení: květen 2017

- Započali jsme první práce v září 2016.
- Obhajovat budeme v květnu 2017.
- Počítáme s tím, že každý bude věnovat 300-350 hodin za dobu projektu.

Plán prací:

- Září: Specifikace, skládání týmu, analýza konkurenčních aplikací
- Říjen: návrh architektury, začátek implementace, kódování jádra a jádra UI
- Listopad-březen: Implementace a testování
- Březen: Uzavírání textu, žádost o překladatele, překlad
- Duben: dokončování implementace a testování, beta test
- Květen: vyladění chyb nalezených v beta testu, prezentace

V následujícím prozatímním časovém odhadu „1 týden“ značí zhruba 10 hodin práce jednoho člena týmu. Pokud každý člen bude pracovat 320 hodin, tak bude pracovat 32 „normálních týdnů“, vytvořit aplikaci pak potrvá 128 normálních člověkotýdnů.

Časový odhad hlavních částí projektu:

- Základní architektura, MVVM: 1 týden (Martin, Víťa)
- Jádro hry: 4 týdny (Anikó), 3 týdny (Petr)
- Jádro uživatelského rozhraní: 6 týdnů (Martin), 4 týdny (Víťa)
- Obrazovky uživatelského rozhraní: 4 týdny (Martin) 2 týdny (Víťa)
- Hra proti lokální umělé inteligenci: 5 týdnů (Víťa)
- Udržování statistik: 2 týdny (Anikó)
- Komunikace hráče s hrou: 4 týdny (Martin), 3 týdny (Víťa)
- Režim analýzy hry: 7 týdnů (Anikó), 3 týdny (Martin)
- Online režim: 2 týdny (Víťa), 1 týden (Petr)
- Singleplayer: 2 týdny (Víťa), 1 týden (Petr)
- Tsumego: 1 týden (Petr), 2 týdny (Anikó)
- Tutoriál: 2 týdny (Petr)
- Vícejazyčnost: 1 týden (Petr)
- Režim diváka: 2 týdny (Anikó)
- Doporučování dobrých tahů: 1 týden (Anikó)
- Knihovna SGF: 1 týden (Víťa, Petr)
- Prototypování: 1 týden (Petr, Anikó)
- Integrační testování a finišování: 2 týdny (každý)
- Komunikace v rámci týmu: 7 týdnů (každý)
- Interní dokumentace: 2 týdny (každý) + 1 týden (Petr, Martin)
- Uživatelská dokumentace: 2 týdny (Petr)
- Vývojová dokumentace: 3 týdny (hlavně Petr)
- Beta test: průběžně – 2 týdny (hlavně Petr)
- Rezerva: 10% času, tj. 3 týdny (každý)

V závorce jsou uvedeni řešitelé, kteří se touto části budou nejpravděpodobněji zabývat, v případě více řešitelů v závorce platí časový odhad před ní pro každého z nich.

Zvláštní zaměření jednotlivých řešitelů:

- Víťa, Martin: Uživatelské rozhraní, rendering, portování AI

- Petr: Tutoriál, pravidla, design funkcí hry, multiplayer
- Anikó: Režim analýzy, statistiky, jádro

Zaměření i odhady jsou prozatímní, a můžou se změnit.

Vymezení projektu

Projekt je zaměřen na následující oblasti (zaškrtněte vyhovující):

Diskrétní modely a algoritmy	
	diskrétní matematika a algoritmy
	geometrie a matematické struktury v informatice
	optimalizace
Teoretická informatika	
	Teoretická informatika
Softwarové a datové inženýrství	
X	softwarové inženýrství
X	vývoj software
	webové inženýrství
	databázové systémy
	analýza a zpracování rozsáhlých dat
Softwarové systémy	
	systémové programování
	spolehlivé systémy
	výkonné systémy
Matematická lingvistika	
	počítačová a formální lingvistika
	statistické metody a strojové učení v počítačové lingvistice
Umělá inteligence	
	inteligentní agenti
	strojové učení
	robotika
Počítačová grafika a vývoj počítačových her	
	počítačová grafika
X	vývoj počítačových her

Poznámky

Je možné, že se v průběhu vývoje ukáže, že jsme neodhadli časovou náročnost některých funkcí. V takovém případě po dohodě s vedoucím některé funkce implementujeme v jednodušší podobě nebo

naopak další funkce přidáme. V dokumentu, který má vedoucí k dispozici, jsme připravili desítky dalších funkcí, o které by náš program mohl být obohacen.

Na začátku budeme aplikaci plánovat tak, jako by měla být volně dostupná zadarmo všem uživatelům, bez obchodního modelu. Pokud se v průběhu projektu rozhodneme, že bychom chtěli, aby aplikace byla komerční, obchodní model vymyslíme a zdůvodnění přiložíme k dokumentaci.

Tento specifikační dokument obsahuje hlavní funkce programu, které plánujeme v programu mít, a které má vedoucí vyžadovat na konci vývoje. Počítáme s tím, že naimplementujeme do aplikace i další funkcionalitu, kterou zde nemáme, protože si nejsme jisti jejím časovým odhadem nebo proto, že nesouvisí přímo s programováním, a tedy se nepočítá nutně jako součást práce na NRPG023.

Omega Go není v oficiálním názvu projektu, protože se jedná jen o kódové jméno