

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

# ROČNÍKOVÝ PROJEKT

Projektový seminář

Latrunculi



Červen 2016

Ondřej Grätz  
Aplikovaná informatika, III. ročník

## **Abstrakt**

Implementace deskové hry Latrunculi s použitím *.NET Framework* pro operační systém *Windows*.

# Obsah

<b>1. Zadání projektu</b>	<b>5</b>
<b>2. Volba technologie</b>	<b>5</b>
2.1. Výběr operačního systému . . . . .	5
2.2. Výběr vývojového prostředí . . . . .	5
<b>3. Organizace kódu</b>	<b>6</b>
3.1. Zvyklosti pro .NET aplikace . . . . .	6
3.2. Soubory projektu . . . . .	6

## Seznam obrázků

1.	Závislosti souborů . . . . .	6
2.	Vrstvy aplikace . . . . .	6
3.	Případy užití . . . . .	8

# 1. Zadání projektu

Cílem projektu je vytvoření hry pro desktopový operační systém s grafickým uživatelským rozhraním (GUI) dle standardů.

Požadavek na přenositelnost spustitelných souborů nebyl stanoven. Uživatelské rozhraní se předpokládá objektově orientované s použitím oken a standardních prvků (hlavní nabídka, nástrojová lišta, tlačítka).

## 2. Volba technologie

### 2.1. Výběr operačního systému

Já jsem pro vývoj i běh hry (aplikace) zvolil operační systém *Windows*, jelikož je mi dobře známý a také proto, že je to s podílem 52 % (viz [4]) mezi uživateli i programátory nej-používanější operační systém pro osobní počítače.

### 2.2. Výběr vývojového prostředí

Pro vývoj byl zvolen nástroj *Visual Studio* od firmy *Microsoft*, jazyky *C#* a *F#* a pro vývoj uživatelského rozhraní *Windows Presentation Framework* (WPF).

Pro zvolený operační systém (*Windows*) by s ohledem na zadání bylo výhodné použít jeden z následujících typů aplikací.

- Win32 (Visual C++ s použitím Windows API nebo MFC)
- .NET Framework + WinForms
- .NET Framework + WPF
- Windows Runtime (C++/CX)

*Win32* a *Windows Runtime* poskytují nejmenší úroveň abstrakce. Vývoj tohoto typu aplikací vyžaduje větší znalosti a zkušenosti programátora. *Windows Runtime* aplikace je navíc možné spustit pouze pod operačním systémem *Windows 8* (nebo novějším). Výhodou je ale standardní vzhled výsledných aplikací a nejlepší výpočetní výkon.

Použití *WinForms* by bylo výhodné, jelikož vývoj je jednoduchý (údálostmi řízený, objektově orientovaný). Výsledná aplikace má vzhled v souladu se standardy operačního systému a očekáváním uživatele. Nevýhodou je strohý design, který se příliš nepřizpůsobuje rozlišení obrazovky a jehož vzhled se může jevit zastaralý. Design je navíc velmi svázan s kódem, který má na starosti výpočty a samotnou logiku aplikace.

Naproti tomu u *WPF* je již při vývoji myšleno na responzivní design. Prvky uživatelského rozhraní mají velmi bohaté a pro programátora snadno použitelné vlastnosti, které umožňují přizpůsobit aplikaci různým velikostem obrazovky a různým způsobům vstupu od uživatele (dotykové obrazovky, přizpůsobení pro zrakově nebo tělesně hendikepované uživatele). Návrh uživatelského rozhraní je navíc velmi dobře oddělen od samotného kódu a umožňuje s použitím nástroje *Microsoft Blend* výrazně zasáhnout do designu aplikace i grafikovi (bez nutnosti znalosti programování a bez zásahu do modelu aplikace).

## 3. Organizace kódu

### 3.1. Zvyklosti pro .NET aplikace

Při vývoji pro *.NET Framework* je nutno veškerý kód umisťovat do metod tříd. Třídy musejí být povinně organizovány do jmenných prostorů.

Pro *.NET* aplikace (s výjimkou aplikací v jazyku F#) je také typické rozdělení kódu do několika binárních souborů. Rozdělení se provádí na základě jmenných prostorů (*namespaces*).

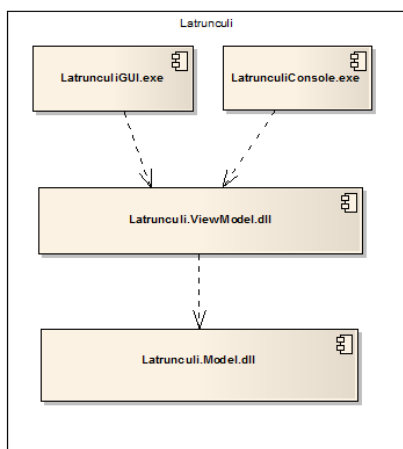
Spustitelný (EXE) soubor obsahuje jádro programu a definice hlavních částí uživatelského rozhraní, ale komponenty uživatelského rozhraní i třídy obsahující model domény, rozhodovací logiku a třídy pro práci se soubory, databázemi či síťovou komunikací jsou umístěny v tzv. knihovnách tříd (DLL soubory).

DLL i EXE soubory poté obsahují překladačem vytvořený *bytecode*, který je poté při spuštění aplikace částečně kompilován do strojového kódu (just-in-time kompilace) a částečně také interpretován běhovým prostředím *.NET*.

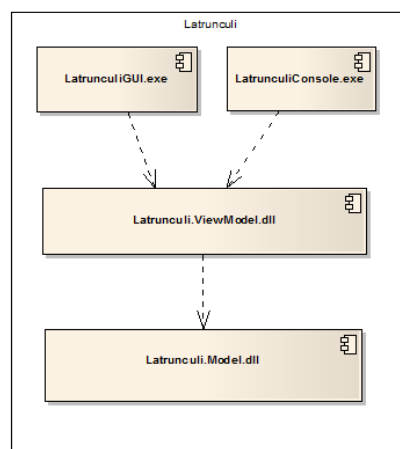
### 3.2. Soubory projektu

Třídy projektu jsou organizovány v podprostorech jmenného prostoru *Latrunculi*.

Závislosti přeložených souborů (*assemblies*) jsou zobrazeny na obr.1.. Tomu odpovídající logické rozvrstvení aplikace na obr.2.. Názvy souborů jsou shodné s názvy jmenných prostorů, které jsou v jednotlivých souborech obsaženy.



Obrázek 1. Závislosti souborů



Obrázek 2. Vrstvy aplikace

## Reference

- [1] Mgr. Jan Outrata, Ph.D.: *Projekt - implementace*, <http://outrata.inf.upol.cz/courses/ps/navrh.pdf>, listopad 2008.
- [2] Mgr. Jan Outrata, Ph.D.: *Projekt - analýza a návrh*, <http://outrata.inf.upol.cz/courses/ps/implementace.txt>, listopad 2010.
- [3] Mgr. Tomáš Kühn, Ph.D.: *Algoritmy realizující počítačového hráče*, <http://www.inf.upol.cz/downloads/studium/PS/algoritmy.pdf>, říjen 2011.
- [4] Wikipedia: *Usage share of operating systems*, [https://en.wikipedia.org/wiki/Usage\\_share\\_of\\_operating\\_systems#Desktop\\_and\\_laptop\\_computers](https://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Desktop_and_laptop_computers), květen 2016.

