Česká zemědělská univerzita v Praze

Technická fakulta

Katedra technologických zařízení staveb

Bakalářská práce

Návrh univerzální programové logiky pro vývoj her

Martin Novák

2022 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Technická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Martin Novák

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Návrh univerzální programové logiky pro vývoj her

Název anglicky

Design of universal program logic for game development

Cíle práce

Cílem práce je popsat aktuální dostupné herní enginy, uživatelská rozhraní a programovací jazyky vhodné pro návrh vzorového řešení. Nejprve na návrhu aplikačního modelu popsat objektový model aplikace. Následně vytvořit návrh vzorového řešení, které bude univerzální v oblasti vývoje her typu RPG.

Metodika

1. Úvod
2. Cíl práce a metodika
3. Výběr vhodných programovacích jazyků pro vývoj her
4. Výběr herních žánrů vhodných pro implementaci
5. Grafické výstupy aplikací
6. Návrh aplikačního modelu
7. Návrh vzorového řešení
8. Zhodnocení realizace aplikace
9. Závěr

Doporučený rozsah práce

30 – 40

Klíčová slova

herní enginy, uživatelská rozhraní, programovací jazyky, objektový model, vývoj her

---

Doporučené zdroje informací

EGGES, A., Learning C# by Programming Games, Springer, 2013, ISBN 3642365795
GREGORY J., Game Engine Architecture, Third Edition, Taylor & Francis Ltd, 2018, ISBN 9781138035454
HARDMAN, C., Game Programming with Unity and C# : A Complete Beginner's Guide, aPress, 2020, ISBN 1484256557
MURRAY, J., C# Game Programming Cookbook for Unity 3D, Taylor & Francis Inc, 2014, ISBN 9781466581401

Předběžný termín obhajoby

2021/2022 LS – TF

Vedoucí práce

Ing. Marek Pačes

Garantující pracoviště

Katedra technologických zařízení staveb

Elektronicky schváleno dne 3. 2. 2021

doc. Ing. Jan Malaťák, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 2. 2021

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 12. 06. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Návrh univerzální programové logiky pro vývoj her" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne datum odevzdání _____

Poděkování

Rád(a) bych touto cestou poděkoval(a) jméno vedoucího, případně dalších osob, a informace, za co děkujete.

Návrh univerzální programové logiky pro vývoj her

Abstrakt

Souhrn práce (cca 15 řádek textu).

Klíčová slova: herní enginy, uživatelská rozhraní, programovací jazyky, objektový model, vývoj her

Design of universal program logic for game development

Abstract

Anglický překlad českého souhrnu

Keywords: game engines, user interface, programing languages, object model, game development

Obsah

Seznam obrázků

Odkazovaný seznam obrázků

Seznam tabulek

Odkazovaný seznam tabulek

Seznam použitých zkratek

Soupis a definování zkratek (vyskytuje-li se jich v textu velké množství)

1. Úvod

Text text text text text text text text text text text text text text text text text text text text text text text.

2. Cíl práce a metodika

2.1 Cíl práce

Cílem práce je popsat aktuální dostupné herní enginy, uživatelská rozhraní a programovací jazyky vhodné pro návrh vzorového řešení. Nejprve na návrhu aplikačního modelu popsat objektový model aplikace. Následně vytvořit návrh vzorového řešení, které bude univerzální v oblasti vývoje her typu RPG.

2.2 Metodika

V první části budou zhodnoceny programovací jazyky, které je možné k realizaci využít a následně vybrán nejvhodnější. Dále bude následovat stručné seznámení s herními žánry a typy aplikací. U žánrů bude posouzeno, do jaké míry je možné pro jednotlivé žánry knihovnu využít a tím pádem, zda je třeba brát charakteristické prvky žánru v potaz při návrhu logiky.

Text text text

3. Výběr vhodných programovacích jazyků pro vývoj her

Programovací jazyky dělíme na dva základní skupiny. První jsou imperativní (např. C++), kam patří většina jazyků a jejich rysem je, že kód je sekvence instrukcí a je z něj čitelné co se v jaký okamžik bude provádět. Druhá skupina jsou deklarativní (např. HTML), které říkají jen co se musí vyřešit, ale ne konkrétní instrukce potřebné k provedení a z toho důvodu často nejsou považovány za programovací jazyky, ale používá se pro ně označení kódovací. Další skupina jsou funkcionální (např. Haskell), které ačkoliv se řadí mezi deklarativní mají znaky obou skupin a je možné jejich přístup použít i v imperativních

jazycích. Na rozdíl od imperativních nevyužívají žádné globální proměnné a vše je prováděno uvnitř funkcí. Na Obr. 1 je porovnání sumy zapsané pomocí imperativního a funkcionálního jazyku.[1, 2]

Obr. 1 imperativní vs. funkcionální jazyk [2]

Z popisu základních paradigmat je vidět, že jazyk bude vybírán z imperativních jazyků, které se dále dělí na dvě podskupiny. Procedurální[3] (např. C) pracují s funkcemi přijímajícími data pouze z parametrů nebo globálních proměnných. Pro svázání více souvisejících hodnot je možné použít strukturu, která je jako pole umožňující ukládat různé datové typy. Objektové (např. Java) mají třídy sloužící jako předlohy pro instance nazývané objekty, které stejně jako struktury mohou ukládat více hodnot různých typů, ale mají vlastní metody, a proto není potřeba všechna data předávat pomocí parametrů, protože si je může načíst z objektu kde se nachází. Objektově orientované programování (OOP) má čtyři základní principy: zapouzdření, abstrakce, dědičnost a polymorfismus. Zapouzdření umožňuje omezit viditelnost proměnných a metod mimo třídu, kontrolovat přístup k jejich hodnotám a ověřit, zda je zapisována platná hodnota. Abstrakce znamená, že pro práci s objektem není nutné znát vnitřní funkci jeho metod a při práci v týmu kolegovi stačí znát název, parametry a výstup metody. Použitím dědičnosti třída, která je potomek získá všechny proměnné a metody rodiče, ale je možné přidat nové, či změnit chování metody. Polymorfismus souvisí s dědičností, kde do proměnné typu rodič je možné vložit potomka, ale při volání metody se zavolá její přetížená verze, která má stejné jméno, typ a parametry, ale jiné tělo. Dále je možné přetěžovat metody změnou parametrů nebo návratové typu.[4] Na Obr. 2 je porovnání počítání obsahu čtverců a obdélníků napsané v procedurálním a objektovém jazyce (kvůli délce vynecháno zadávání hodnot). Je evidentní, že pro hry se nejvíce hodí objektové jazyky, a proto ty nejpoužívanější nyní budou probrány více do hloubky.

C

```c
int pocetCtvercu = 2;
Ctverec ctverce[2];
int pocetObdelniku = 2;
Obdelnik obdelniky[2];


for (int i = 0; i < pocetCtvercu; i++)
{
    printf("obsah ctverce=%d\n", obsah(&ctverce[i]));
}
for (int i = 0; i < pocetObdelniku; i++)
{
    printf("obsah obdelniku=%d\n", obsah2(&obdelniky[i]));
}
```

C++

```cpp
Ctverec *tvary[4];


for (size_t i = 0; i < size(tvary); i++)
{
    cout << tvary[i]->vypis()<<endl;
}
```

Obr. 2 procedurální vs objektový jazyk-vlastní

3.1 C++

C++ je více paradigmatový jazyk[5] rozšiřující jazyk C o objekty, nová klíčová slova a datové typy. Byla snaha zachovat co největší zpětnou kompatibilitu, pro usnadnění přechodu z C na C++ umožňující tvorbu komplexnějších programů[6], ale některé kódy možné napsat v C jsou v C++ neplatné[7]. Se zpětnou kompatibilitou souvisí headery obsahující deklarace proměnných, struktur, tříd a jejich metod, které je potřeba používat i v jiných souborech[8], což sebou ale nese i nevýhodu, že přidání nových tříd a metod, či změny jejich hlaviček je nutno provádět na dvou místech.

3.1.1 Kompilace a hardware

Stejně jako jazyk C je kompilován pro konkrétní architekturu procesoru a operační systém, takže je nutno rozlišovat 32bitovou (označovanou jako x86) a 64bitovou verzi operačního systému (x86 dokáže běžet na x64 obráceně ne)[9, 10], ale existuje také C++/CLI, který je součást Microsoft .NET a je kompilován na bytecode (viz kapitola 3.2.1), což umožňuje mít jednu verzi pro obě architektury a sestavit aplikaci z částí napsaných v různých .NET jazycích (viz Kap. 3.3.1)[11]. Tak jako C je i C++ díky své schopnosti pracovat přímo s pamětí a registry pomocí pointerů vhodný pro psaní ovladačů, operačních systémů a řízení jednočipových počítačů[12–14].

3.1.2 Novinky oproti C

Mezi novinky, které C++ přináší patří *namespace*, které umožňují kód organizovat do menších celků a je tak možné, aby se v projektu vyskytoval stejný název vícekrát. Jakožto objektový jazyk dovoluje přetěžování metod, ale oproti Javě porovnává jen parametry, takže funkce s různým návratovým typem a stejnými parametry považuje za stejné a nepůjdou zkompilovat. Dále přibyli *Exceptions* sloužící jako zpráva o chybě ve volané metodě a umožňují tento problém vyřešit, aniž by došlo k pádu programu. Na rozdíl od Javy a C# se může jednat o libovolný datový typ[8].

3.1.3 Nevýhody

standardy C++ neobsahují Garbage Collector, takže se programátor musí starat o alokování a následné uvolňování paměti sám, ale je možné použít některý vytvořený třetí stranou[15]. C++ neobsahuje vlastní GUI a musíte proto použít některou z knihoven třetí strany[16].

3.2 Java

Java je objektový jazyk, který byl vyvinut s myšlenkou, aby bylo možné jeden program spustit na všech systémech. Architektura vychází z jazyků jako Eiffel, SmallTalk a Objective C. Pro snazší přechod programátorů z C++ byla snaha zachovat co nejpodobnější syntaxi, ale jeho funkcionality použity nebyli.[17] oproti C a C++ se v Javě nenachází funkce, které existují samy o sobě a nenáleží žádné třídě, ale jen metody, které jsou součástí objektu, nebo jsou statické[18, 19].

3.2.1 JIT (Just In Time)

Oproti C++ není kód kompilován přímo do strojového kódu, ale do vysokoúrovňového platformě nezávislého kódu nazývaného bytecode, který je spouštěn ve virtuálním stroji (Java Virtual Machine neboli JVM), což umožňuje, aby stejný program bylo možné spustit na všech operačních systémech v 32bitové i 64bitové verzi[9], ale ke spuštění programu musí být na zařízení naistalována odpovídající verze JVM. Nevýhodou bytecodu je jeho

výpočetní náročnost, neboť je překládán do strojového kódu v momentě, kdy je spouštěn. Díky just in time (JIT) překladu je ovšem možné provést optimalizaci pro konkrétní CPU a tím dosáhnout vyšší rychlosti, než jaké dosahují programy napsané například v C nebo C++ a zkompilované na počítači, který je starší než ten, kde je spouštěn.[20].

3.2.2 Přístup k paměti a ovládání hardware

Java neumožňuje pracovat s pointery, neboť správu paměti zajišťuje run time[19]. Jelikož program nepřistupuje k paměti přímo je možné zajistit, že nebude zasahovat do paměti ostatních programů, což by mohlo způsobit pád systému či neoprávněný přístup k citlivým údajům[21]. Pomocí Java ME Embedded je možné ovládat i jednočipové počítače, ale je podporováno pouze Raspberry Pi Model B a dva čipy od STMicroelectronics[22].

3.2.3 Výhody

Na rozdíl od C++ Java nepoužívá headery a pro použití třídy v jiném souboru stačí, aby se nacházely ve stejném *namespace*, nebo na příslušný namespace přidat referenci. Oproti C++ má Java Garbage Collector, který se stará o uvolňování paměti mazáním objektů bez reference, čímž usnadňuje programátorovi práci, ovšem za cenu občasného zastavení běhu aplikace, což je možné vyřešit přidáním dalšího vlákna[23]. Doba potřebná ke smazání „mrtvých“ objektů zaleží na počtu „živých“ a velikosti paměti[24, 25]. Java má pro GUI dvě knihovny, jimiž jsou *awt* a odlehčený *swing*[26, 27].

3.2.4 Nevýhody

Stejně jako u C++ je zde možné využívat přetěžování metod, ale signaturu tvoří kromě parametrů i návratový typ, avšak oproti C++ a C# Java neumí přetěžovat operátory[19]. Další nevýhoda Javy je, že za generický typ, který se nejčastěji využívá u *Collection* (např. *ArrayList*) není možné dosadit primitivní datový typ, takže například pro přidání *int* do seznamu je třeba vytvořit nový objekt typu *Integer* s jeho hodnotou[28]. Java nemá datový typ pro bezznaménková celá čísla (*uint*)[19], takže je k dispozici pouze polovina rozsahu a pokud je potřeba zapsat hodnotu nad dvě miliardy (2^{31}) musí se použít *long* (64bitový).

3.3 C#

C# je plně objektový jazyk a hlavní zástupce rodiny Microsoft .NET, který spojuje to nejlepší z C++ a Javy. Ačkoliv vznikl původně pro Windows v posledních letech s přibývajícími frameworky postupně nahrazuje Javu ve vývoji mobilních aplikací (Xamarin a MAUI), PHP v back-endu (ASP .NET) a JavaScript na front-endu (Blazor) webových aplikací.

3.3.1 Microsoft .NET

Microsoft .NET je prostředí a rodina jazyků, které ho využívají. Tyto jazyky jsou vzájemně kompatibilní díky požadavkům na CTS (Common Type Specification), CLS (Common Language Specification), CLR (Common Language Runtime) a CLI (Common Language Infrastructure). Hlavní úlohou CLR je správa paměti a vláken. Mimo toho také kontroluje typovou bezpečnost. CTS zajišťují, že všechny jazyky mají stejnou definici datových typů a nemůže se tak stát, aby jednou byl *int* reprezentován třiceti dvěma bity a podruhé pouze šestnácti. Součástí těchto požadavků je, že veškeré referenční i hodnotové datové typy jsou potomky třídy *System.Object* a tím pádem jsou všechny .NET jazyky plně objektové. CLS zajišťuje, aby všechny jazyky byli kompilovatelné do bytecodu označovaného jako MSIL (Microsoft Intermediate Language), což umožňuje v jednom programu kombinovat knihovny napsané v C#, Visual Basic, F#, C++/CLI nebo jiném z více než dvaceti jazyků [11, 29, 30].

MSIL je objektový nízko úrovňový jazyk, který tak jako většinu bytecode je možné kompilovat v režimu JIT (Just In Time), ale navíc také podporuje AOT (Ahead Of Time), kdy se výsledný soubor chová podobně, jako v případě C++, a je tedy nutné ho sestavit pro každý systém a architekturu, kde chceme program spouštět[30]. Výhodou předem zkompilované aplikace je rychlejší start a pro složitější programy i výrazný nárůst výkonu, ovšem za cenu většího souboru, neboť obsahuje také MSIL, který je v některých případech potřeba[31]. při generování AOT jsou využívány nástroje NGen (Native Image Generator) pro .NET Framework a Crossgen2 pro .NET Core. Výstupy těchto nástrojů se nazývají nativní obrazy a jsou instalovány do NIC (Native Image Cache), kam jsou přidávány i závislosti, které je možno používat více obrazy, čímž se eliminuje duplicita. Kompilaci je možné spustit na počítači programátora, nebo až při instalaci programu. Vytvoření obrazu u

uživatele má výhodu, že kód bude optimalizován pro jeho procesor a bude tak dosahovat nejvyššího možného výkonu. [11, 32, 33] Další výhodou AOT je, že není potřeba, aby byl překlad co nejrychlejší, takže má dost času provést optimalizace[34].

Velkou výhodou je, že .NET runtime je od Windows Vista součást operačního systému, takže je aktualizován společně se systémem[35], díky čemuž uživatel nemusí nic instalovat. Prostředí .NET bylo původně určeno pouze pro platformy Microsoftu (Windows a Xbox), což se změnilo až v roce 2014 vydáním .NET Core[30], ovšem s GUI pro ostatní systémy se vývojáři museli spoléhat na třetí strany. V roce 2022 bylo vydáno .NET MAUI umožňující vytvořit jednu aplikaci na Windows, Android, iOS a macOS s minimálními zásahy do kódu.[36]

3.3.2 Přístup k paměti a ovládání hardware

Na rozdíl od Javy je v C# možné využívat i pointery a obcházet tak správce paměti, což může vylepšit výkon, ale současně vést k bezpečnostním problémům a nestabilitě, kvůli čemuž není možné ověřit bezpečnost a takovýto kód musí být umístěn do bloku vyznačeného pomocí preprocesorů *unsafe*. Kód uvnitř toho bloku se podobá tomu, který by se napsal v C++ nebo C[37]. Ačkoliv C# oficiálně neumožňuje ovládání jednočipových počítačů, existují rozšíření třetích stran, jako například nanoFramework nebo placené visualmicro, které podporují čipy založené na ARM architektuře[38, 39].

3.3.3 Porovnání s Javou

Stejně jako u Javy je zde viditelnost tříd řízena pomocí *namespace*. Při přetěžování metod je signatura dána typem a pořadím parametrů, ale oproti Javě umí přetěžovat i operátory[40]. C# dokáže primitivní datové typy (např. *int*) automaticky měnit na objekty[29]. Tak jako Java i C# má Garbage Collector, který za programátora uvolňuje paměť. K jeho spuštění dochází při nedostatku paměti, nebo překročení stanoveného limitu[41, 42]. Ačkoliv C# v některých situacích vyžaduje oproti Javě další klíčová slova, čímž působí jako pomalejší na psaní, snižuje se tím množství chyb a urychluje orientaci v kódu, protože je na první pohled vidět přetěžování při dědičnosti a použité modifikátory.

3.3.4 Podobnosti s C++

Tak jako C++ má i C# struktury, které by se daly označit jako hodnotová verze objektu, ale mají omezené možnosti. Například nemohou mít hodnotu *null*, používat dědičnost a mít proměnné inicializované při deklaraci[37].

Podobně jako má C++ pointery na funkce, v C# jsou využíváni delegáti, kteří slouží k předávání metod v parametru, nebo umožňují dynamicky měnit volanou funkci. Delegáty je možné sloučit do *MulticastDelegate*, který obsahuje jejich seznam a při volání je postupně provádí[43–45]. Další jejich využití jsou eventy (např. kliknutí na tlačítko), kde metody, které na něj reagují, musí být typu *void* a mít parametry typu Object a *EventArgs* nebo jeho potomka. První parametr říká, jaký objekt event vyvolal a druhý obsahuje podrobnosti, jako například jaká je poloha kurzoru[46, 47].

3.3.5 Modifikátory parametrů metod

U parametru metody je možné použít klíčové slovo *out*, které ho změní na výstupní hodnotu, což umožňuje vracet více než jednu hodnotu bez nutnosti použít pole objektů, ze kterého by se poté postupně přiřazovaly do příslušných proměnných, nebo vracet bool, pokud metoda proběhla úspěšně, a tuto hodnotu předávat výstupním parametrem. Dále je možné využít modifikátory *ref*, který mění hodnotovou proměnnou na referenční, a *in*, který brání úpravám hodnoty[48].

3.3.6 nové funkce

Mezi novinky, které C# přináší patří *properties*, umožňující zabalit *get* a *set* pod jeden název, se kterým se při volání pracuje jako by se jednalo o proměnnou[49]. Další nová funkce je modifikátor *partial* umožňující rozdělit definici třídy, struktury nebo interface na více částí, které mohou být i ve více souborech. Pomocí této funkce se dá zvýšit přehlednost velkých tříd rozdělením na menší logické celky a zjednodušuje tak práci u týmových projektů, kde každý programátor může pracovat na své části, aniž by omezoval kolegu. Dále se této možnosti využívá při generování časti třídy, aniž by ovlivnila programátorův soubor. Příkladem je Windows Form, jehož grafická část je generována Visual Studiem[50].

3.4 výběr

V Tab. 1**Chyba! Nenalezen zdroj odkazů.** je přehled vlastností porovnávaných jazyků, kde zeleně jsou označeny výhody, červeně nevýhody a žlutě body, kde záleží na situaci. Například když je potřeba maximální rychlost, může být Garbage Collector nevýhodou, ale zajišťuje že program nebude v paměti nechávat data bez reference a spotřebovávat tak zbytečně více paměti, než potřebuje. Z porovnání je vidět, že C# umožňuje snazší implementaci knihovny, jelikož není potřeba importovat header pro každou použitou třídu a má snazší práci s eventy.

Jelikož jsou všechny 3 porovnávané jazyky objektové nebude mít volba na návrh logiky výrazný vliv a rozdíly budou jen v komunikaci objektů mezi sebou (event). Rozdíly se projeví při realizaci (např. práce s pamětí, přetěžování nebo generika) a implementaci, protože každá technologie podporuje jen některé jazyky.

Jelikož Java není podporována žádným z hlavních engynů, tak je pro univerzální knihovnu nevhodná.

Kvůli kompilaci do MSIL, který umožňuje sestavit program z kódů napsaných ve více jazycích, byl jako jazyk, pro který bude knihovna navrhována C#. Takto bude knihovna univerzálnější, neboť její implementace nebude omezena na jeden jazyk.

	C++	Java	C#
Byte code	Ne (C++/CLI ano)	ano	Ano (může být přeložen)
Headery	ano	ne	ne
Garbage Collector	Ne (možno použít třetí strany)[15]	ano	ano
Přetěžování operátorů	ano	ne	ano
Vlastní eventy	ano[51]	Vlastní řešení	ano
Properties	ne	ne	ano
In/out/ref	pointery	ne	ano

Maximální počet rodičovských tříd	neomezen	1	1
uint	ano	ne	ano
struct	ano	ne	ano
MSIL	jen C++/CLI	ne	ano
Příklad Herních enginů	Unreal, CRYENGINE	Greenfoot, libGDX[52]	Unity, CRYENGINE

Tab. 1 porovnání jazyků

4. Výběr herních žánrů vhodných pro implementaci

Jelikož je cílem praktické části navrhnout knihovnu pro tvorbu her, je třeba se podívat na základní žánry a zhodnotit, jak velkou část knihovny bude možné využít. Hry dělíme na několik žánrů (RPG, akční, strategie, závodní), podle jejich mechanik (např. vylepšování postavy, inventář, střelba, přeskakování mezi plošinami, stavba budov nebo řízení jednotek), což usnadňuje hráčům orientaci při výběru, jelikož mají základní představu, jaký zážitek od titulu očekávat. Často se stává, že hra spojuje více žánrů a je proto těžké ji jednoznačně zařadit. Další komplikací je nejednotné dělení subžánrů (především u RPG a akčních her), kvůli čemuž se můžete setkat s tím, že hra je na různých stránkách označena jinými štítky (viz Tab. 2**Chyba! Nenalezen zdroj odkazů.**). [53]

hra	**steam**	**Epic games**	**Ubisoft**	**Alza**
Assassins Creed Odyssey	S otevřeným světem, RPG, S asasíny, Akční	Akční, RPG	akční, adventura	akční, adventura, dobrodružná
Baldur's Gate II	RPG, Klasické, Fantasy, Dungeons & Dragons	-	-	RPG, Strategie
Heroes of Might & Magic III	Strategické, Klasické, Tahové strategie, Fantasy	RPG, strategie	strategie, RTS	Akční, Strategie

Tab. 2 odlišné značení her[54–57][58, 59][60–63]

4.1 RPG

Tento žánr vychází z deskových her jako jsou Dungeons & Dragons[64] (zkráceně D&D nebo DnD) nebo české Dračí doupě, kde hráč nebo skupina hráčů hraje za postavy, které mají různé rasy (např. člověk, elf nebo trpaslík), třídy (např. válečník, mág nebo zloděj), inventář, statistiky a schopnosti. Většinou plní úkoly zadané NPC, za což dostanou odměnu v podobě zkušeností, peněz a předmětů. Hlavním znakem je získávání zkušenostních bodů (exp) a vylepšování postav. Nejčastěji je hra zasazena do fantasy světa. Většinou mají RPG propracovanější a delší příběh než ostatní žánry. V deskové verzi se hází kostkami, zda se akce podařila, což je v digitální podobě nahrazeno generátorem náhodných čísel, případně je náhoda vynechána a výsledek záleží na schopnostech hráče a statistikách postavy (např. přesnost luku určuje hráčova práce s myší a jak dobře postava umí luk používat) nebo je akce vždy úspěšná. RPG má několik subžánrů, které ne všechny stránky a obchody rozlišují a z RPG samostatně vyčleňují jen některé (viz Tab. 2).

První a nejstarší kategorie je již zmíněná digitalizovaná verze deskových her (např. Baldur's Gate[65]), či jejich slovní podoby, která je na tahy a hráč má čas promyslet si své další kroky. Původně byly pouze textové, kde hráč vybíral z možností a později se začaly objevovat i grafické verze.

Druhá a dnes nejrozšířenější kategorie jsou akční RPG (ARPG) odehrávající se v reálném čase a hráč musí rychle reagovat na akce nepřátel. Hráč hraje za jednu postavu, která v některých titulech může mít společníky, nad kterými hráč nemá kontrolu (např. The Elder Scrolls V: Skyrim[66]) nebo může přepnout ovládání ze své postavy na společníka (např. Star Wars: Knights of the Old Republic[67]). Tyto hry mají často otevřený svět, což umožňuje hráči volně se pohybovat po celé mapě, prozkoumávat ji a plnit hlavní i vedlejší úkoly v libovolném pořadí. Většinou hráč svými rozhodnutími v dialozích ovlivní příběh, nebo jak na něj ostatní postavy reagují. Hry využívající tyto dva prvky se označují jako nelineární, nebo západní RPG, což je opak JRPG (Japan RPG), které jsou lineární a kladou důraz hlavně na vyprávění předem daného příběhu (např. Final Fantasy[68]).

Další kategorie je MMORPG (Massively Multiplayer Online RPG), kde na velkých mapách hraje současně tisíce hráčů, kteří mohou navzájem komunikovat,

pomáhat si, obchodovat, bojovat v PvP (Player vs Player) arénách a spojovat se do aliancí (např. World of Warcraft[69]).

Poslední jsou taktická RPG (TRPG), která kombinují vylepšování postav a strategii (např. XCOM[70]). Tento subžánr je často řazen mezi strategie. [53, 71–74][75] Především na mobilních zařízeních je možné se setkat s TRPG, kde hráč sbírá postavy s různými schopnostmi, které vylepšuje a následně v pětičlenných týmech bojuje proti jiným týmům, kde jich je v kampani v rámci jedné úrovně více za sebou (např. Star Wars: Galaxy of Heroes[76]).

4.2 akční

Akční hry je souhrnné označení bojových her, střílečeк, plošinovek a ostatních her s rychlým tempem, které nemají vlastní žánr. [77] V bojových hrách si hráč typicky volí postavu ze seznamu a následně s ní bojujete proti soupeři (např. Mortal Kombat[78]). Ačkoliv se střílečky řadí mezi akční hry, většinou jsou brány jako samostatný žánr. Střílečky dělíme podle umístění kamery na FPS (First Person Shooter), která je z pohledu postavy (např. Doom[79]) a TPS (Third Person Shooter), kde je v záběru kamery i postava (např. Mafia[80]). Pod pojmem plošinovka si většina lidí představí 2D hry jako Mario[81], kde se hráč pohybuje po platformách a musí se dostat na konec úrovně. Patří sem i akčnější tituly jako Tomb Raider[82] a Prince of Persia[83], které obsahují i soubojový systém se zbraněmi.[84, 85]

4.3 strategie

Podobně jako RPG, mají i strategie předlohu ve stolních hrách (např. Warhammer 40 000[86]). Existují dva způsoby členění. První je podle toho, zda je hra na kola (tahová neboli turn-based) a hráči se střídají (např. Civilization V[87]), nebo vše probíhá v reálném čase (RTS) jako například Age of Empires[88]. Druhé je podle zaměření, jako jsou například válečné či budovatelské. Když se řekne strategie většina lidí si vybaví válečnou, kde hráč těží suroviny na stavbu budov a výrobu jednotek, kterými se snaží porazit protivníka. V budovatelských se hráč stává starostou města či manažerem (např. zábavního parku, přepravní společnosti či nemocnice) a jeho

úkolem je vyřešit logistiku, zajistit zisk na další rozvoj a starat se o spokojenost lidí (např. Cities: Skylines[89] a RollerCoaster Tycoon[90]).[72, 74, 91]

4.4 závodní

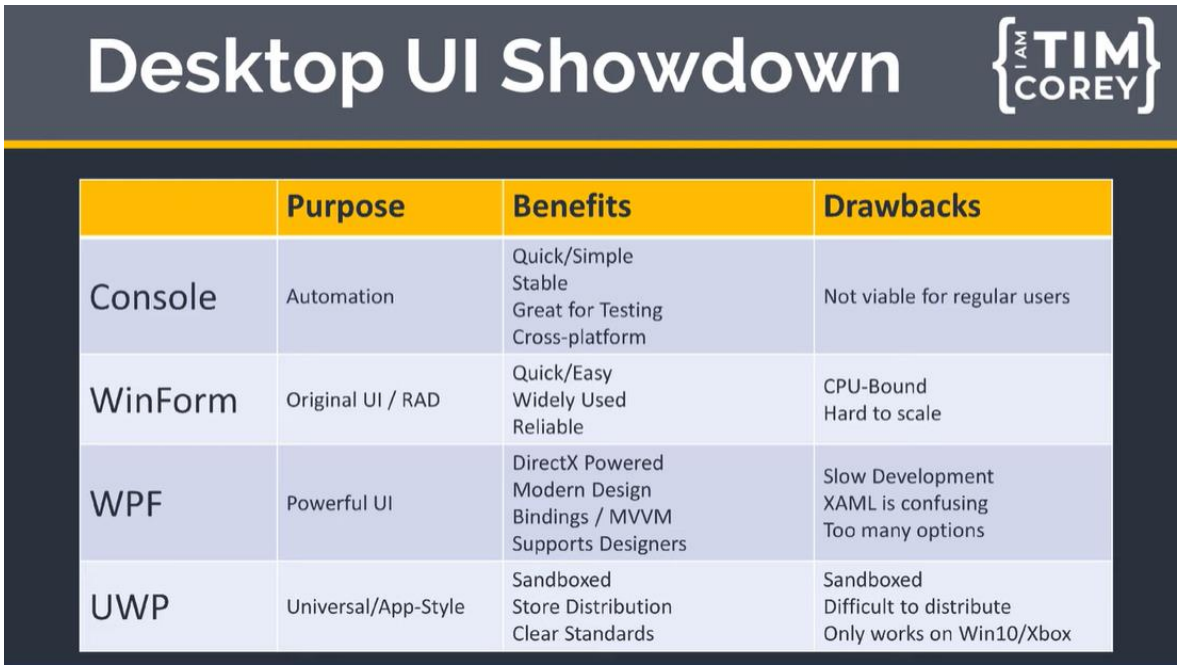
Jedním z možných dělení závodních her je podle jízdního modelu. Realistický (např. DiRT Rally[92]) se snaží co nejvíce napodobit chování skutečných vozů. Arkádový (např. Need For Speed[93]) ubírá realističnost ve prospěch jednodušší hratelnosti. Dalším dělením je, zda se závodí v autech či jiných dopravních prostředcích (např. vesmírné lodě v Redout 2[94]). Ačkoliv většina her hráče za narážení do soupeřů penalizuje, existují také série jako například FlatOut[95] a Asphalt[96], které za to naopak hráče odměňují a umožňují soupeře zpomalit či úplně vyřadit ze závodu [74, 97].

4.5 shrnutí

Toto bylo jen stručné seznámení s nejvíce zastoupenými žánry, ale existuje mnoho dalších, které vnikají například kombinací, či rozšiřováním těchto základních. Jelikož je tato práce zaměřena na RPG, je knihovna plně implementovatelná a neměla by být žádná část, která není pokryta a programátor musí dělat pouze grafickou část, reakce na Eventy a vstupy od hráče, nastavení hodnot a umělou inteligenci NPC. Jelikož většina RPG je současně akční hrou je i pro tento žánr možné využít téměř celou knihovnu bez výrazných úprav logiky. Výjimkou jsou akční hry mezi jejich mechaniky nepatří souboj (např. Blackhole[98]) nebo neobsahují předměty, protože v takovém případě zůstane velká část knihovny nevyužita. Pro strategie je z knihovny možné využít třídu *Postava* na jednotky a budovy nebo *GameManager* a *Chunk* na generování náhodné mapy. U závodních her lze z knihovny využít *StatList* na vlastnosti vozidla (rychlost, akcelerace, …) a *Predmet* na vylepšení. V přídě her, kde se dá poškodit soupeřův vůz lze použít také *Postava*. Jelikož třídy nejsou určeny na toto využití, obsahují velké množství dat na víc, která budou zbytečně zabírat paměť a je proto lepší vytvořit si vlastní třídy.

5. Grafické výstupy aplikací

Programy ke komunikaci s uživateli potřebují uživatelské rozhraní. U nejjednodušších her se může jednat o konzoli, ale pro většinu her pouhé psaní nestačí a potřebují grafické prostředí neboli GUI. Na Obr. 3**Chyba! Nenalezen zdroj odkazů.** jsou v několika bodech shrnuté rozdíly UI pro .NET, které jsou níže rozvinuty.



	Purpose	Benefits	Drawbacks
Console	Automation	Quick/Simple Stable Great for Testing Cross-platform	Not viable for regular users
WinForm	Original UI / RAD	Quick/Easy Widely Used Reliable	CPU-Bound Hard to scale
WPF	Powerful UI	DirectX Powered Modern Design Bindings / MVVM Supports Designers	Slow Development XAML is confusing Too many options
UWP	Universal/App-Style	Sandboxed Store Distribution Clear Standards	Sandboxed Difficult to distribute Only works on Win10/Xbox

Obr. 3 srovnání Windows UI[99]

5.1 konzolová aplikace

Nejjednodušší UI je konzole, která jako vstup a výstup používá příkazový řádek. Dnes se s ní běžný uživatel obvykle nesetká, protože kvůli textovým vstupům není tak příjemná na ovládání jako GUI. Využívá se především pro automatizované úkoly a na programy s důrazem na co nejnižší zátěž hardwaru využívané administrátory. Je možné mít aplikaci s GUI, která spouští konzolové aplikace a následně zobrazuje jejich návratovou hodnotu.[99]

5.2 okenní aplikace

Te

5.2.1 WinForm

Te

5.2.2 WPF

Te

6. Návrh aplikačního modelu

Te

7. Návrh vzorového řešení

Te

8. Zhodnocení realizace aplikace

Te

9. Závěr

Text…

Seznam použitých zdrojů

[1] *imperative programming* [online]. [vid. 2021-03-16]. Dostupné z: https://whatis.techtarget.com/definition/imperative-programming

[2] COMPUTERPHILE. *Programming Paradigms - Computerphile* [online]. 2013 [vid. 2021-03-29]. Dostupné z: https://www.youtube.com/watch?v=sqV3pL5x8PI

[3] *procedural and object oriented programming* [online]. [vid. 2021-03-29]. Dostupné z: https://www.geeksforgeeks.org/differences-between-procedural-and-object-oriented-programming/

[4] FREECODECAMP.ORG. *Intro to Object Oriented Programming - Crash Course - YouTube* [online]. 2020 [vid. 2021-07-04]. Dostupné z: https://www.youtube.com/watch?v=SiBw7os-_zI

[5] STROUSTRUP, Bjarne. *Stroustrup: FAQ-multiparadigm* [online]. [vid. 2021-07-22]. Dostupné z: https://www.stroustrup.com/bs_faq.html#multiparadigm

[6] STROUSTRUP, Bjarne. From The Handbook of Object Technology (Editor: Saba Zamir) [online]. 1999 [vid. 2021-07-18]. Dostupné z: https://www.stroustrup.com/crc.pdf

[7] STROUSTRUP, Bjarne. *Stroustrup: FAQ-C subset of C++* [online]. [vid. 2021-07-20]. Dostupné z: https://www.stroustrup.com/bs_faq.html#C-is-subset

[8] PRATA, Stephen. *Mistrovství v C++*. 1. vyd. Praha: Computer Press, 2001. ISBN 80-7226-339-0.

[9] *Why Java is Platform Independent? | by Neil Wilston | Medium* [online]. [vid. 2021-07-21]. Dostupné z: https://medium.com/@neil.wilston123/why-java-is-platform-independent-1d82c2249a69

[10] *What is x86 Architecture and its difference between x64? - Latest open tech from seeed studio* [online]. [vid. 2021-07-21]. Dostupné z: https://www.seeedstudio.com/blog/2020/02/24/what-is-x86-architecture-and-its-difference-between-x64/

[11] HANÁK, Ján. *C++/CLI začínáme programovat*. Brno: artax a.s., 2009. ISBN 978-80-87017-04-3.

[12] STROUSTRUP, Bjarne. *C++ Applications* [online]. [vid. 2021-07-20]. Dostupné z: https://www.stroustrup.com/applications.html

[13] STROUSTRUP, Bjarne. *Stroustrup: FAQ* [online]. [vid. 2021-07-20]. Dostupné z: https://www.stroustrup.com/bs_faq.html#true

[14] STROUSTRUP, Bjarne. *Stroustrup: FAQ-unsafe* [online]. [vid. 2021-07-20]. Dostupné z: https://www.stroustrup.com/bs_faq.html#unsafe

[15] STROUSTRUP, Bjarne. *Stroustrup: FAQ-garbage-collection* [online]. [vid. 2021-07-20]. Dostupné z: https://www.stroustrup.com/bs_faq.html#garbage-collection

[16] STROUSTRUP, Bjarne. *Stroustrup: FAQ-GUI* [online]. [vid. 2021-07-20]. Dostupné z: https://www.stroustrup.com/bs_faq.html#gui

[17] ORACLE. *The Java Language Environment* [online]. [vid. 2023-02-10]. Dostupné z: https://www.oracle.com/java/technologies/introduction-to-java.html

[18] *Difference between Methods and Functions in JavaScript - GeeksforGeeks* [online]. [vid. 2021-07-22]. Dostupné z: https://www.geeksforgeeks.org/difference-between-methods-and-functions-in-javascript/

[19] ORACLE. *The Java Language Environment* [online]. [vid. 2021-07-22]. Dostupné z: https://www.oracle.com/java/technologies/simple-familiar.html

[20] *Is Java slow? Compared to C++, it's faster than you think* [online]. [vid. 2021-07-22]. Dostupné z: https://www.theserverside.com/opinion/Is-Java-slow-Compared-to-C-its-faster-than-you-think

[21] EGGES, Arjan, Jeroen D. FOKKER a Mark H. OVERMARS. *Learning C# by Programming Games* [online]. 2013. ISBN 3642365795. Dostupné z: doi:10.1007/978-3-642-36580-5

[22] ORACLE. *Oracle Java ME Embedded Getting Started* [online]. [vid. 2021-07-22]. Dostupné z: https://www.oracle.com/java/technologies/javame-embedded/javame-embedded-getstarted.html

[23] FREECODECAMP.ORG. *Garbage Collection in Java – What is GC and How it Works in the JVM* [online]. [vid. 2021-07-21]. Dostupné z: https://www.freecodecamp.org/news/garbage-collection-in-java-what-is-gc-and-how-it-works-in-the-jvm/

[24] IBM. *Garbage collection impacts to Java performance - IBM Documentation* [online]. [vid. 2021-07-21]. Dostupné

z: https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-garbage-collection-impacts-java-performance

[25] ORACLE. *Java SE 6 HotSpot[tm] Virtual Machine Garbage Collection Tuning* [online]. [vid. 2021-07-22]. Dostupné z: https://www.oracle.com/java/technologies/javase/gc-tuning-6.html

[26] DOCS.ORACLE.COM. *java.awt (Java Platform SE 7)* [online]. [vid. 2021-07-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html

[27] DOCS.ORACLE.COM. *javax.swing (Java Platform SE 7)* [online]. [vid. 2021-07-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html

[28] ITNETWORK.CZ. *Lekce 3 - Seznam (List) pomocí pole v Javě* [online]. [vid. 2021-07-22]. Dostupné z: https://www.itnetwork.cz/java/kolekce-a-proudy/java-tutorial-seznamy-kolekce-list

[29] DOCS.MICROSOFT.COM. *A Tour of C# - C# Guide | Microsoft Docs* [online]. [vid. 2021-07-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

[30] . *NET (and .NET Core) - introduction and overview | Microsoft Learn* [online]. [vid. 2023-02-05]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/core/introduction

[31] *ReadyToRun deployment overview - .NET | Microsoft Learn* [online]. [vid. 2023-02-07]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/core/deploying/ready-to-run

[32] RICHARD LANDER. Conversation about crossgen2 - .NET Blog. *.NET Blog* [online]. 2021 [vid. 2023-02-07]. Dostupné z: https://devblogs.microsoft.com/dotnet/conversation-about-crossgen2/

[33] DOCS.MICROSOFT.COM. *Ngen.exe (Native Image Generator) | Microsoft Docs* [online]. [vid. 2021-08-06]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/framework/tools/ngen-exe-native-image-generator

[34] MICROSOFT. *.NET Glossary | Microsoft Learn* [online]. [vid. 2023-01-29]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/standard/glossary#implementation-of-net

[35] DOCS.MICROSOFT.COM. *.NET Framework versions and dependencies* [online]. [vid. 2021-06-05]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies?redirectedfrom=MSDN#net-framework-30

[36] DOCS.MICROSOFT.COM. *What is .NET MAUI? - .NET MAUI | Microsoft Docs* [online]. [vid. 2021-09-23]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/maui/what-is-maui

[37] DOCS.MICROSOFT.COM. *Unsafe code, pointers to data, and function pointers | Microsoft Docs* [online]. [vid. 2021-07-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/unsafe-code

[38] *VisualMicro - Arduino IDE For Visual Studio* [online]. [vid. 2021-07-23]. Dostupné z: https://www.visualmicro.com/#

[39] . *NET nanoFramework VS2019 Extension - Visual Studio Marketplace* [online]. [vid. 2021-07-23]. Dostupné z: https://marketplace.visualstudio.com/items?itemName=nanoframework.nanoFramework-VS2019-Extension

[40] *C# | Method Overloading - GeeksforGeeks* [online]. [vid. 2021-07-23]. Dostupné z: https://www.geeksforgeeks.org/c-sharp-method-overloading/

[41] DOCS.MICROSOFT.COM. *Fundamentals of garbage collection | Microsoft Docs* [online]. [vid. 2021-07-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/standard/garbage-collection/fundamentals

[42] DOCS.MICROSOFT.COM. *.NET garbage collection | Microsoft Docs* [online]. [vid. 2021-07-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/standard/garbage-collection/

[43] DOCS.MICROSOFT.COM. *MulticastDelegate Class (System) | Microsoft Docs* [online]. [vid. 2021-07-25]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/api/system.multicastdelegate?view=net-5.0#code-try-3

[44] ŽIVĚ.CZ. *Poznáváme C# a Microsoft.NET 15. díl – delegáty – Živě.cz* [online]. [vid. 2021-07-25]. Dostupné z: https://www.zive.cz/clanky/poznavame-c-a-microsoftnet-15-dil--delegaty/sc-3-a-123479/default.aspx

[45] DOCS.MICROSOFT.COM. *Delegates - C# Programming Guide | Microsoft Docs* [online]. [vid. 2021-07-25]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/delegates/

[46] DOCS.MICROSOFT.COM. *Handling and Raising Events | Microsoft Docs* [online]. [vid. 2021-07-25]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/standard/events/

[47] DOCS.MICROSOFT.COM. *EventHandler Delegate (System) | Microsoft Docs* [online]. [vid. 2021-07-25]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/api/system.eventhandler?view=net-5.0

[48] DOCS.MICROSOFT.COM. *out parameter modifier - C# Reference | Microsoft Docs* [online]. [vid. 2021-07-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/out-parameter-modifier

[49] DOCS.MICROSOFT.COM. *Properties - C# Programming Guide | Microsoft Docs* [online]. [vid. 2021-07-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/properties

[50] DOCS.MICROSOFT.COM. *Partial Classes and Methods - C# Programming Guide | Microsoft Docs* [online]. [vid. 2021-07-24]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/partial-classes-and-methods

[51] DOCS.MICROSOFT.COM. *Event handling in native C++* [online]. [vid. 2021-06-06]. Dostupné z: https://docs.microsoft.com/en-us/cpp/cpp/event-handling-in-native-cpp?view=msvc-160#:~:text= Event handling in native C%2B%2B 1,you use the intrinsic function __hook... More

[52] LIBGDX. *libGDX features* [online]. [vid. 2021-06-05]. Dostupné z: https://libgdx.com/features/

[53] SLÁMA, David. Průvodce herními žánry – dungeony a rpg – Doupě.cz. *Computer* [online]. 2010 [vid. 2021-08-14]. Dostupné z: https://doupe.zive.cz/clanek/pruvodce-hernimi-zanry--dungeony-a-rpg

[54] *Hra na PC Assassins Creed Odyssey - PC DIGITAL | Hra na PC na Alza.cz* [online]. [vid. 2021-08-24]. Dostupné z: https://www.alza.cz/media/assassins-creed-odyssey-pc-digital-d6222907.htm?o=3

[55] *Ušetřete 75% na produktu Assassin's Creed® Odyssey ve službě Steam* [online]. [vid. 2021-08-24]. Dostupné z: https://store.steampowered.com/app/812140/Assassins_Creed_Odyssey/

[56] *Assassin's Creed Odyssey | Download and Buy Today - Epic Games Store* [online]. [vid. 2021-08-24]. Dostupné z: https://www.epicgames.com/store/en-US/p/assassins-creed-odyssey

[57] *Assassin's Creed Odyssey on PS4, Xbox One, PC | Ubisoft (UK)* [online]. [vid. 2021-08-24]. Dostupné z: https://www.ubisoft.com/en-gb/game/assassins-creed/odyssey

[58] *Baldur's Gate II: Enhanced Edition ve službě Steam* [online]. [vid. 2021-08-26]. Dostupné z: https://store.steampowered.com/app/257350/Baldurs_Gate_II_Enhanced_Edition/

[59] *Hra na PC Baldur's Gate II Enhanced Edition - PC DIGITAL | Hra na PC na Alza.cz* [online]. [vid. 2021-08-26]. Dostupné z: https://www.alza.cz/media/baldurs-gate-ii-enhanced-edition-pc-digital-d5866684.htm

[60] *Hra na PC Heroes of Might & Magic III - HD Edtion (PC) DIGITAL | Hra na PC na Alza.cz* [online]. [vid. 2021-08-26]. Dostupné z: https://www.alza.cz/media/heroes-of-might-magic-iii-hd-edtion-pc-digital-d5346604.htm

[61] *Might & Magic Heroes 3 | Download and Buy Today - Epic Games Store* [online]. [vid. 2021-08-26]. Dostupné z: https://www.epicgames.com/store/en-US/p/might-and-magic-heroes-3

[62] *Heroes® of Might & Magic® III - HD Edition ve službě Steam* [online]. [vid. 2021-08-26]. Dostupné z: https://store.steampowered.com/app/297000/Heroes_of_Might__Magic_III__HD_Edition/

[63] *Buy Heroes of Might and Magic III: Complete PC (Download)* [online]. [vid. 2021-08-26]. Dostupné z: https://store.ubi.com/uk/game?pid=575ffd9ba3be1633568b4d8c&dwvar_575ffd9ba3be1633568b4d8c_Platform=pcdl&edition=Complete%20Edition&source=detail

[64] *D&D Official Homepage | Dungeons & Dragons* [online]. [vid. 2021-11-12]. Dostupné z: https://dnd.wizards.com/

[65] *Baldur's Gate: Enhanced Edition* [online]. [vid. 2021-11-12]. Dostupné z: https://www.baldursgate.com/

[66] *The Elder Scrolls | Skyrim* [online]. [vid. 2021-11-12]. Dostupné z: https://elderscrolls.bethesda.net/en/skyrim

[67] *Knights of the Old Republic | StarWars.com* [online]. [vid. 2021-12-20]. Dostupné z: https://www.starwars.com/games-apps/knights-of-the-old-republic

[68] *FINAL FANTASY PORTAL SITE | SQUARE ENIX* [online]. [vid. 2021-11-12]. Dostupné z: https://na.finalfantasy.com/

[69] *World of Warcraft* [online]. [vid. 2021-11-12]. Dostupné z: https://worldofwarcraft.com/en-gb/

[70] *XCOM 2* [online]. [vid. 2021-12-20]. Dostupné z: https://xcom.com/

[71] Fantasy světy – historie počítačových her na hrdiny díl I. *Fantasymag.cz* [online]. 2017. Dostupné z: https://www.fantasymag.cz/fantasy-svety-historie-pocitacovych-her-hrdiny-dil-i/

[72] GAMEDESIGNING.ORG. *34 Popular Types of Video Games, Explained (With Examples and Fun Graphics)* [online]. [vid. 2021-07-27]. Dostupné z: https://www.gamedesigning.org/gaming/video-game-genres/

[73] Fantasy světy díl II. - čtverečkové dungeony na PC | Fantasymag.cz. *Fantasymag.cz* [online]. 2018 [vid. 2021-07-29]. Dostupné z: https://www.fantasymag.cz/fantasy-svety-dil-ii-ctvereckove-dungeony-pc/

[74] *Herní žánry na Databázi her – Nápověda – Databáze-her.cz* [online]. [vid. 2021-08-14]. Dostupné z: https://www.databaze-her.cz/napoveda/herni-zanry-na-databazi-her/

[75] TECHRAPTOR.NET. *Playing Roles: On Tactical-RPGs | TechRaptor* [online]. [vid. 2021-08-16]. Dostupné z: https://techraptor.net/originals/playing-roles-on-tactical-rpgs

[76] *Star Wars™ Galaxy of Heroes - Free Mobile Game - EA Official Site* [online]. [vid. 2021-11-12]. Dostupné z: https://www.ea.com/games/starwars/galaxy-of-heroes

[77] What is an Action/Adventure Game? - Gameranx. *Gameranx* [online]. 2011 [vid. 2021-08-30]. Dostupné z: https://gameranx.com/features/id/3350/article/what-is-an-action-adventure-game/

[78] *Mortal Kombat 11 Ultimate* [online]. [vid. 2021-11-12]. Dostupné z: https://www.mortalkombat.com/

[79] *DOOM Eternal | Bethesda.net* [online]. [vid. 2021-11-12]. Dostupné z: https://bethesda.net/en/game/doom

[80] *Mafia: Trilogy - Home* [online]. [vid. 2021-12-20]. Dostupné z: https://mafiagame.com/cs-CZ/

[81] *The official home of Super Mario™ – History* [online]. [vid. 2021-12-20]. Dostupné z: https://mario.nintendo.com/history/

[82] *Shadow Of The Tomb Raider | SQUARE ENIX* [online]. [vid. 2021-12-20]. Dostupné z: https://tombraider.square-enix-games.com/en-us

[83] *Prince of Persia | Ubisoft (US)* [online]. [vid. 2021-12-20]. Dostupné z: https://www.ubisoft.com/en-us/game/prince-of-persia/prince-of-persia

[84] IDTECH.COM. *Ultimate List of Different Types of Video Games | 49 Genres & Subcategories* [online]. [vid. 2021-08-31]. Dostupné z: https://www.idtech.com/blog/different-types-of-video-game-genres

[85] DESPAIN, Wendy. *Writing for Video Game Genres* [online]. B.m.: A K Peters/CRC Press, 2009. ISBN 9780429063343. Dostupné z: doi:10.1201/b10641

[86] *Warhammer 40,000 - Warhammer 40,000* [online]. [vid. 2021-11-12]. Dostupné z: https://warhammer40000.com/

[87] *Civilization V | Homepage* [online]. [vid. 2023-03-17]. Dostupné z: https://civilization.com/civilization-5/

[88] *Age of Empires Franchise - Official Web Site* [online]. [vid. 2023-03-17]. Dostupné z: https://www.ageofempires.com/

[89] *Cities: Skylines - Paradox Interactive* [online]. [vid. 2023-03-17]. Dostupné z: https://www.paradoxinteractive.com/games/cities-skylines/about

[90] *RollerCoaster Tycoon: Deluxe - RollerCoaster Tycoon - The Ultimate Theme park Sim* [online]. [vid. 2023-03-17]. Dostupné z: https://www.rollercoastertycoon.com/rollercoaster-tycoon-deluxe/

[91] KOŠŤÁL, Filip. Průvodce herními žánry - válečné strategie – Doupě.cz. *Computer* [online]. 2011 [vid. 2021-08-23]. Dostupné z: https://doupe.zive.cz/clanek/pruvodce-hernimi-zanry---valecne-strategie

[92] *DiRT Rally - The official game site* [online]. [vid. 2023-03-18]. Dostupné z: https://dirtgame.com/dirtrally/us/home

[93] *Need for Speed Video Games - Official EA Site* [online]. [vid. 2023-03-18]. Dostupné z: https://www.ea.com/games/need-for-speed

[94] *Home - Redout 2 - The Fastest Racing Game in the Universe* [online]. [vid. 2023-03-18]. Dostupné z: https://redout.games/redout2/

[95] *Bugbear Entertainment | Drive hard* [online]. [vid. 2023-03-18]. Dostupné z: https://bugbeargames.com/

[96] *Asphalt 9: Legends - Arcade Racing | Asphalt Legends* [online]. [vid. 2023-03-18]. Dostupné z: https://asphaltlegends.com/

[97] INDIAN. *Wreckfest - Recenze - YouTube* [online]. [vid. 2023-03-18]. Dostupné z: https://www.youtube.com/watch?v=3_3nvi1vsZ4

[98] *O HŘE | BLACKHOLE :: PC, MAC, LINUX :: 2D Platfomer* [online]. [vid. 2021-12-03]. Dostupné z: https://blackhole-game.com/cs/o-hre

[99] IAMTIMCOREY. *WinForm vs WPF vs UWP vs Console - The C# Desktop UI Showdown (and the future with .NET 5)* [online]. 2019 [vid. 2021-05-03]. Dostupné z: https://www.youtube.com/watch?v=yq0dSkA1vpM

Přílohy

Odkazovaný seznam příloh