

Výuková aplikace v Unity engine

Radim Hudcovský
Bakalářská práce | 2026

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav počítačových a komunikačních systémů

Akademický rok: 2025/2026

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Radim Hudcovský
Osobní číslo: A23025
Studijní program: B0688A140008 Informační technologie v administrativě
Forma studia: Prezenční
Téma práce: Výuková aplikace v Unity engine
Téma práce anglicky: Educational Application in the Unity Engine

Zásady pro vypracování

1. Proveďte rešerši na téma existujících výukových aplikací a jejich funkcí.
2. Seznamte se s možnostmi vhodného vývojového prostředí pro tvorbu aplikací.
3. Navrhněte architekturu výukové aplikace včetně datové struktury pro kurzy a uživatele.
4. Implementujte aplikaci s možností tvorby, ukládání a sdílení kurzů pomocí vhodné technologie.
5. Otestujte aplikaci z hlediska funkčnosti, stability a použitelnosti. Věnujte se zabezpečení aplikace.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. HOLAN, Tomáš. *Unity: První seznámení s tvorbou počítačových her*. Vyd. 1. Praha: CZ.NIC, 2021. ISBN 978-80-88168-57-7.
2. KAMPOURTZIS, George. *Educational Game Design Fundamentals: A Journey to Creating Intrinsically Motivating Learning Experiences*. Boca Raton, FL: CRC Press, Taylor & Francis, 2018. ISBN 978-1138631540.
3. VALVE CORPORATION. *Steamworks Documentation: Steam Workshop Implementation Guide*. [Online]. [S.l.]: Valve Corporation, [cit. 2025-10-31]. Dostupné z: <https://partner.steamgames.com/doc/features/workshop/implementation>
4. KLOPFER, Eric, Jason HAAS, Scot OSTERWEIL a Louisa ROSENHECK. *Resonant Games: Design Principles for Learning Games that Connect Hearts, Minds, and the Everyday*. Cambridge, MA: MIT Press, 2018.
5. MAYER, Richard E. *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. 5. vyd. Hoboken: Wiley, 2023. ISBN 978-1394177370.
6. UNITY. *Unity 6.0 LTS*. [Online]. Unity Technologies, [nedatováno]. [Cit. 2025-10-31]. Dostupné z: <https://unity.com/releases/unity-6>.

Vedoucí bakalářské práce: **doc. Ing. Jiří Vojtěšek, Ph.D.**
Ústav řízení procesů

Datum zadání bakalářské práce: **2. prosince 2025**

Termín odevzdání bakalářské práce: **25. května 2026**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



Ing. Miroslav Matýšek, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 2. prosince 2025

Prohlášení autora závěrečné kvalifikační práce

Beru na vědomí, že

- odevzdáním závěrečné práce souhlasím se zpřístupněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- závěrečná práce bude uložena v elektronické podobě v univerzitním informačním systému;
- jedno vyhotovení závěrečné práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji závěrečnou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – závěrečnou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování závěrečné práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky závěrečné práce využít ke komerčním účelům;
- pokud je výstupem závěrečné práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na závěrečné práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- za účelem vylepšení stylistické stránky práce jsem využil model ChatGPT (<https://chatgpt.com/>) a za výsledek přebírám plnou zodpovědnost;
- odevzdaná verze závěrečné práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....

autor

Abstrakt

Tato bakalářská práce se zabývá návrhem a realizací desktopové výukové aplikace v herním enginu Unity 6. Aplikace umožňuje tvorbu, úpravu, studium a distribuci uživatelsky vytvořených kurzů. Teoretická část práce vymezuje oblast e-learningu, serious games a gamifikovaných aplikací a porovnává vybraná existující řešení a technologie vhodné pro vývoj vzdělávacího softwaru. Praktická část popisuje návrh datového modelu založeného na formátu JSON, implementaci tvořivého a studijního režimu, lokální ukládání kurzů a integraci služeb Steamworks SDK. Součástí řešení je také využití služeb Steam Workshop pro distribuci kurzů a Steam Cloud pro synchronizaci rozpracovaných dat. Výsledkem práce je funkční prototyp aplikace, který poskytuje přehledné prostředí pro tvorbu vlastního vzdělávacího obsahu a jeho interaktivní studium.

Klíčová slova

Unity 6, výuková aplikace, e-learning, serious games, platforma Steam, uživatelský obsah

Abstract

This bachelor's thesis deals with the design and implementation of a desktop educational application developed in the Unity 6 game engine. The application enables users to create, edit and study user-generated courses divided into chapters and educational modules. The theoretical part defines the areas of e-learning, serious games and gamification, and compares selected existing solutions and technologies suitable for the development of educational software. The practical part describes the design of a JSON-based data model, the implementation of the creative and study modes, local course storage and the integration of Steamworks services. The solution also includes the use of Steam Workshop for course distribution and Steam Cloud for the synchronization of work-in-progress data. The result of the thesis is a functional application prototype that provides a clear environment for creating custom educational content and studying it interactively.

Keywords

Unity 6, educational application, e-learning, serious games, Steam platform, user-generated content

Rád bych poděkoval vedoucímu své bakalářské práce, panu děkanovi fakulty doc. Ing. Jiřímu Vojtěškoví, Ph.D., za jeho cenné rady a čas, který mi věnoval.

Dále bych chtěl poděkovat své blízké rodině za velkou podporu, která mi umožňuje vysokou školu studovat.

Obsah

Seznam obrázků	17
Seznam tabulek	18
Seznam použitých symbolů a zkratek	19
Seznam příloh	20
Úvod	21
1 Kontext E-learningu a výukových aplikací	22
1.1 Velikost trhu s výukovými aplikacemi a serious games	23
2 Technologická řešení, e-learningová konkurence a jejich analýza	24
2.1 Analýza e-learningové konkurence	24
2.1.1 Duolingo	24
2.1.2 Autoškola plus	24
2.1.3 Kahoot!	25
2.1.4 MOODLE	25
2.2 Srovnání Unity6 s konkurenčními technologickými řešeními	26
2.2.1 Unity6	26
2.2.2 Unreal engine 5	27
2.2.3 Godot	28
2.2.4 Webové řešení	29
2.3 Distribuce a Steamworks SDK	30
2.3.1 Autentizace a správa uživatelské identity	30
2.3.2 Distribuce obsahu pomocí Steam Workshop	30
2.3.3 Technická realizace v prostředí Unity	31
3 Specifikace požadavků a návrh aplikace	32
3.1 Funkční a nefunkční požadavky	32
3.1.1 Funkční požadavky	32
3.1.2 Nefunkční požadavky	33
3.2 Návrh architektury datového modelu	33
3.2.1 Hierarchická struktura JSON souboru	34
3.2.2 Ukládání a správa příložených souborů	34

4	Tvořivý režim	36
4.1	Úprava kapitol dle funkcí kódu.....	36
4.2	Úprava obsahu kapitol dle funkcí kódu	37
4.3	Vytvoření nového kurzu dle funkcí kódu.....	38
4.4	Výběr rozpracovaného kurzu dle funkcí kódu	39
4.5	Uživatelský pohled na tvořivý režim	40
4.5.1	Stránka CreateMenu.....	41
5	Studijní režim	44
5.1	Stránka pro studium kapitoly dle funkcí kódu	44
5.2	Uživatelský pohled na studijní režim.....	45
5.2.1	Hlavní studijní stránka	46
5.2.2	Modul poznámky	46
5.2.3	Modul kvíz	47
5.2.4	Modul drag and drop.....	48
5.2.5	Modul karta	49
6	Ostatní implementační části aplikace.....	50
6.1	Jednotný vzhled aplikace	50
6.2	Integrace se službami Steamworks	50
6.2.1	Příprava stránky v obchodě.....	51
6.2.2	Příprava buildu aplikace.....	54
7	Testování a optimalizace.....	55
7.1	Grafický výkon v rozhraní	55
7.2	Správa a optimalizace	55
7.3	Synchronizace rozpracovaných kurzů pomocí Steam Cloud.....	56
7.4	Zabezpečení aplikace	57
7.5	Testování aplikace.....	57
7.6	Testování funkčnosti aplikace autorem.....	58
7.6.1	Funkční testování aplikace.....	58
7.6.2	Uživatelské rozhraní	58
7.6.3	Ukládání a načítání dat.....	59
7.6.4	Steam API	59

8	Možnosti budoucího rozšíření aplikace	60
	Závěr	61
	Seznam použité literatury	63

Seznam obrázků

Obr. 1: Hlavní stránka tvořivého režimu	40
Obr. 2: Stránka pro vytváření obsahu kapitoly	42
Obr. 3: Stránka pro výběr kurzu ke studiu	45
Obr. 4: Hlavní stránka studijního režimu	46
Obr. 5: Ukázka poznámkového modulu	47
Obr. 6: Ukázka kvízového modulu	48
Obr. 7: Ukázka drag and drop modulu.....	48
Obr. 8: Ukázka karetního modulu.....	49
Obr. 9: Ukázka úpravy popisu aplikace s požadavky ve službě Steamworks	53

Seznam tabulek

Tab. 1: Funkční požadavky	32
Tab. 2: Nefunkční požadavky	33
Tab. 3: Specifikace testovacích zařízení	58

Seznam použitých symbolů a zkratek

2D	Two-dimensional (Dvourozměrný)
3D	Three-dimensional (Trojrozměrný)
AAA	Triple-A (velkorozpočtový herní projekt)
API	Application Programming Interface (rozhraní pro programování aplikací)
BYOD	Bring Your Own Device (užívání vlastního zařízení v organizaci)
CSS	Cascading Style Sheet (jazyk pro stylování webových stránek)
GB	Gigabyte
GDPR	General Data Protection Regulation (nařízení o ochraně osobních dat)
HTML	Hypertext Markup Language (značkovací jazyk pro webové stránky)
JSON	JavaScript Object Notation (strukturovaný textový formát)
MHz	Megahertz
PIN	Personal Identification Number (osobní identifikační číslo)
SDK	Software Development Kit (sada vývojářských nástrojů)
SSD	Solid State Drive
TB	Terabyte
UI	User Interface (uživatelské rozhraní)
UMG	Unreal Motion Graphics (systém pro tvorbu uživatelského rozhraní v Unreal Engine)
USD	United States Dollar (americký dolar)
UXML	Unity Extensible Markup Language (značkovací jazyk systému UI Toolkit v Unity)

Seznam příloh

Ostatní přílohy

Příloha A: Uživatelský manuál

manual.pdf

Příloha B: Zdrojový Unity projekt

projekt.zip

Úvod

Současné vzdělávání stále častěji využívá digitální nástroje, které rozšiřují klasické formy výuky o interaktivitu, okamžitou zpětnou vazbu a možnost samostatného studia vlastním tempem. Vedle běžných e-learningových systémů se v této oblasti uplatňují také výukové aplikace a gamifikované nástroje, které propojují vzdělávací obsah s prvky motivace, procvičování a postupného zlepšování. Tyto prostředky mohou být přínosné zejména v situacích, kdy samotné pasivní čtení studijních materiálů nebo jednorázové testování nestačí k dlouhodobému udržení pozornosti a zapamatování učiva.

Významnou roli v oblasti digitálního vzdělávání hraje také možnost nejen studovat výukový obsah, ale rovněž jej jednoduše vytvářet, upravovat a dále využívat. Mnoho existujících aplikací je zaměřeno na konkrétní oblast vzdělávání nebo funguje jako uzavřený systém s předem připraveným obsahem. Takový přístup může být pro uživatele pohodlný, zároveň však omezuje možnost přizpůsobit výuku konkrétním potřebám vyučujícího či studenta. Z tohoto důvodu vzniká potřeba nástroje, který umožní nejen práci s hotovým obsahem, ale také tvorbu vlastních kurzů bez nutnosti zásahu do zdrojového kódu aplikace.

Cílem této bakalářské práce je navrhnout a vytvořit výukovou aplikaci v herním enginu Unity 6, která bude sloužit pro tvorbu a studium uživatelsky vytvořených kurzů. Práce vede k aplikaci, koncipované jako desktopové řešení s důrazem na jednoduché ovládání, přehledné uživatelské rozhraní a srozumitelnou strukturu výukového obsahu. Uživatel může vytvářet vlastní kurzy členěné do kapitol a výukových modulů, s následnou možností jejich studia. Součástí návrhu je také lokální ukládání dat ve strukturovaném formátu, aby bylo možné s kurzy pracovat i bez závislosti na trvalém připojení k internetu.

Před samotným vývojem aplikace je řešené téma zasazeno do širšího kontextu digitálního vzdělávání, výukových aplikací a principů, které se v této oblasti využívají. Pozornost je také věnována vybraným existujícím řešením a technologiím, které mohou sloužit pro návrh vlastního softwaru. Na základě tohoto vymezení budou následně stanoveny požadavky na aplikaci, navržena její datová struktura a popsán způsob realizace jednotlivých funkčních částí.

Výsledkem této práce je funkční prototyp výukové aplikace, která propojuje tvorbu vzdělávacího obsahu, jeho studium a následné distribuci. Přínosem navrženého řešení je především možnost pracovat v přehledném a intuitivním prostředí, ukládání vytvořených dat v jednotné struktuře a následně jejich využití ve studijním režimu. Aplikace tak představí praktický nástroj, který bude moci sloužit jako základ pro další rozvoj výukového softwaru zaměřeného na uživatelsky vytvářený obsah.

1 Kontext E-learningu a výukových aplikací

Současné vzdělávání napříč různými sektory stále častěji naráží na limity v oblasti aktivního zapojení studentů a efektivního zprostředkování neustále rostoucího objemu informací. Tradiční formy výuky, jako je práce s tištěnými učebnicemi nebo pasivní poslech výkladu, mnohdy nedokáží dlouhodobě udržet pozornost a motivaci studentů. Jedním z moderních přístupů, které se snaží na tuto situaci reagovat, jsou digitální nástroje označované jako výukové aplikace, určené k předávání znalostí, procvičování dovedností nebo ověřování porozumění učivu. Takový software rozšiřuje a transformuje tradiční vzdělávací postupy tím, že výuku propojuje s interaktivitou, okamžitou zpětnou vazbou a možností individuálního tempa učení.

Zároveň zaznamenávají výrazný rozvoj i počítačové hry, jejichž popularita dlouhodobě roste. Vedle her klasických, které jsou vnímány především jako prostředek zábavy nebo jako interaktivní umělecké dílo, se postupně rozvíjí také oblast serious games, česky označované jako hry vážné nebo účelové. Tento pojem označuje hry, u kterých hlavním cílem není pouze zábava, ale také vzdělávání nebo řešení konkrétního problému. Pojem serious games byl poprvé použit panem Clarkem Abtem, který tak pojmenoval hry s explicitně a pečlivě formulovaným vzdělávacím účelem. Zábavní prvek v nich může být dle něj přítomen, ale slouží především jako motivační složka podporující zapojení studenta. [1]

V širším pojetí lze serious games zařadit do oblasti e-learningu, kategorie vzdělávání zprostředkovaného elektronickým zařízením, například počítačem nebo mobilním telefonem. Jeho cílem je rozvoj znalostí a dovedností pomocí digitálního obsahu, interakcí a výukových aktivit. E-learning může mít podobu asynchronní výuky, při níž student pracuje samostatně vlastním tempem a nemusí být současně přítomen vyučující, nebo synchronní výuky, při níž výuka probíhá ve stejném čase pro více účastníků, například ve virtuální třídě. Mezi běžné formy e-learningu pak patří tutoriály, tedy krátké výukové jednotky s vysvětlením a procvičením, virtuální učebny vedené učitelem, videa, výukové hry či simulace reálných situací. [2]

S pojmem serious games úzce souvisí i pojmy gamifikace či gamifikované aplikace. Těmi se rozumí využití herních prvků v prostředí, které samo o sobě není plnohodnotnou hrou. Gamifikovaná aplikace například nemusí obsahovat příběh, herní svět nebo komplexní herní prvky, ale může do výuky přidávat body, odznaky, žebříčky, úrovně nebo vizualizaci postupu označovanou jako progrese. Cílem gamifikace je psychologicky motivovat uživatele, zvýšit jeho zájem o výuku a podpořit pravidelné plnění úkolů. [3]

Další z pojmů, se kterými se můžeme setkat v rámci tématu výukových her a aplikací, je spojení slov Resonant learning. Používají jej autoři v publikaci Resonant Games [3] a lze ho chápat jako učení propojené s osobní zkušeností, motivací a každodenním světem studenta. Podle tohoto přístupu kvalitní vzdělávací hra nemá pracovat pouze s tím, že hráč odpovídá na otázky nebo sbírá body, ale má respektovat celého studenta, to znamená jeho motivaci, emoce, sociální kontext, předchozí zkušenosti a vztah k učivu. Důležitá je také

sociální povaha učení, což znamená, že učení často neprobíhá izolovaně, ale prostřednictvím komunikace, spolupráce a sdíleného řešení problémů. Tento pohled je významný i pro návrh výukové aplikace, protože upozorňuje, že herní prvky samy o sobě nezaručují vzdělávací efekt. Proto významnou roli v e-learningu hraje i správná práce s multimédií. [3]

Výukový software by neměl zahlcovat uživatele nadbytečnými texty, zvuky nebo grafickými prvky, které nepodporují samotný výukový cíl. Takové nadbytečné prvky mohou zvyšovat kognitivní zátěž, tedy mentální náročnost zpracování informací, aniž by zlepšovaly porozumění učivu. Naopak vhodně použitá grafika, text, zvuk a interaktivní prvky mohou studentovi pomoci zaměřit pozornost na podstatné informace, porozumět souvislostem a převést získané znalosti do praxe. Proto se tato práce v praktickém výstupu zaměří na jednoduché a přehledné prostředí pro vytváření a studium kurzů tak, aby uživatel nebyl zbytečně kognitivně zatěžován grafickými a zvukovými prvky. Důležitou součástí efektivního digitálního učení jsou také procvičovací úlohy a postupné zvyšování obtížnosti. Ty budou ve výstupu práce v režii samotných uživatelů a vyučujících vytvářejících kurzy. [2]

Z pohledu návrhu výukové hry je rovněž důležité vycházet z jasného herního jádra, tedy základní návrhové myšlenky, která určuje, co má hráč ve hře dělat, proč to má dělat a jak tato činnost souvisí s učním. Publikace Educational Game Design Fundamentals [4] pracuje s představou, že vzdělávací hra má být navržena jako propojení tří hlavních oblastí: samotné hry, hráče a didaktické stránky. Návrhář by měl již na začátku uvažovat o cílové skupině, vzdělávacích cílech, herních principech, způsobu zpětné vazby, technické náročnosti a dostupných zdrojích. Herní jádro tím pomáhá vymezit, které prvky jsou pro zamýšlenou vzdělávací zkušenost nezbytné, a brání tomu, aby se vývoj hry rozpadl mezi nesouvisející pedagogické, technické a estetické požadavky. [4]

1.1 Velikost trhu s výukovými aplikacemi a serious games

Relevanci zvoleného tématu podtrhuje fakt, že vývoj trhu se serious games, tedy vzdělávacími, tréninkovými a simulačními hrami s jiným než čistě zábavním účelem, vykazuje v posledních letech výrazný růst. Podle analytických zpráv dosáhla hodnota tohoto trhu v roce 2024 přibližně 11,44 miliardy USD [5]. Z pohledu současného roku 2026 se již můžeme opřít o data z roku 2025, kdy trh s tímto softwarem úspěšně dosáhl hodnoty 13,52 miliardy USD [6]. Tím se potvrdilo silné tempo růstu odpovídající složené roční míře přesahující 18 %. Dlouhodobější prognózy navíc předpokládají další masivní rozvoj tohoto trhu, a to až na 31,66 miliardy USD do roku 2029 [5], s výhledem na překročení hranice 71 miliard USD do roku 2034 [5]. Tyto ekonomické ukazatele odrážejí rostoucí uplatnění serious games a gamifikace nejen ve školství, ale také v oblastech zdravotnictví, obrany a podnikového vzdělávání, kde virtuální simulace a interaktivní trénink hrají stále významnější roli.

2 Technologická řešení, e-learningová konkurence a jejich analýza

Před zahájením samotného návrhu a vývoje nové výukové aplikace je nezbytné zhodnotit aktuální stav trhu s konkurenčním vzdělávacím softwarem a různých technologických řešení na jejich vývoj. Tato kapitola se proto nejprve zaměřuje na rozbor vybraných konkurenčních e-learningových platforem z hlediska jejich architektury a výukových metodik. Následně poskytuje technologická srovnání dostupných herních enginů a webových frameworků, čímž teoreticky odůvodňuje volbu enginu Unity 6 pro realizaci tohoto projektu. V závěru kapitoly je pak představena distribuční platforma Steam a její vývojářské rozhraní Steamworks SDK, které bude tvořit hlavní cloudovou infrastrukturu vyvíjené aplikace.

2.1 Analýza e-learningové konkurence

Trh s výukovým softwarem se v poslední době rozrostl o nejrůznější tituly a nadále roste. Na následujících stránkách bude proto rozebrán výčet softwarových nástrojů používaných pro výuku, které v různých směrech mohou konkurovat této bakalářské práci.

2.1.1 Duolingo

Duolingo je celosvětově jednou z nejlépe známých a nejznámějších vzdělávacích platforem. Její primární zaměření spočívá ve výuce cizích jazyků. Z pedagogického hlediska Duolingo exceluje ve využívání silné gamifikace. Učení je rozděleno do velmi krátkých, takzvaných mikro-lekcí, za jejichž plnění je uživatel odměňován virtuální měnou, odznaky a udržení denní série. Tento psychologický prvek udržuje vysokou motivaci k pravidelnému studiu. Aplikace navíc využívá pokročilé algoritmy pro opakování učiva v čase, které dynamicky vyhodnocují, kdy student s největší pravděpodobností dané cizí slovo zapomene, a v ten moment mu ho znovu předloží. Její hlavní limitací v kontextu této bakalářské práce je však striktně uzavřený ekosystém. Kurzy jsou centrálně vytvářeny a schvalovány samotnou společností, což neumožňuje uživatelům tvořit a sdílet vlastní specifická výuková témata. [7]

2.1.2 Autoškola plus

Na dalším spektru e-learningových nástrojů stojí aplikace Autoškola Plus. Jedná se o úzce profilovanou mobilní aplikaci, jejímž jediným a striktním cílem je úspěšná příprava uživatele na složení teoretických zkoušek k získání řidičského průkazu. Oproti hravému Duolingu volí Autoškola Plus mnohem pragmatičtější přístup. Využívá k tomu oficiální a státem schválené sady testových otázek, u kterých je garantována naprostá faktická správnost. Výukový proces je zde realizován prostřednictvím kvízových typů otázek a taktéž využívá metodu přerušovaného učení pro efektivnější zapamatování předpisů a křížovatek. I zde se však projevuje filozofie uzavřeného obsahu. Aplikace plní svůj účel dokonale, ale je zcela vázána

na jednu konkrétní databázi dat. Student ani lektor do ní nemůže nahrát vlastní materiály, modifikovat otázky nebo vytvářet nové moduly nad rámec oficiálních testů Ministerstva dopravy. [8]

2.1.3 Kahoot!

Dalším významným zástupcem na poli e-learningových platforem je norská aplikace Kahoot! Ta se od dříve zmíněných platforem odlišuje především svým důrazem na uživatelsky generovaný obsah. Aplikace svým uživatelům neposkytuje pouze jeden uzavřený a předem definovaný výukový program, ale funguje jako otevřený nástroj, který umožňuje učitelům, lektorům i samotným studentům snadno vytvářet vlastní interaktivní kvízy na naprosto libovolné téma. Z pedagogického hlediska se Kahoot! spoléhá na silné prvky kompetitivní gamifikace, jako jsou přísné časové limity pro odpovědi, dynamické přidělování bodů a zobrazování žebříčků úspěšnosti v reálném čase, což prokazatelně zvyšuje krátkodobou angažovanost a pozornost studentů. [9]

Základní princip fungování aplikace Kahoot! je historicky postaven na konceptu synchronní výuky a zapojení celé skupiny najednou. Typický scénář využití v praxi probíhá tak, že vyučující spustí předpřipravený kvíz na hlavním sdíleném zobrazovacím zařízení, kterým bývá nejčastěji interaktivní tabule nebo projektor ve třídě. Studenti si do hry nemusí složité stahovat žádný software, i když i ten je k dispozici, ale připojují se jednoduše prostřednictvím webového prohlížeče, po zadání unikátního číselného kódu, takzvaného herního PINu. Tento přístup, známý jako BYOD (Bring Your Own Device), výrazně usnadňuje nasazení aplikace v běžném školním prostředí. [9]

2.1.4 MOODLE

Moodle (Modular Object-Oriented Dynamic Learning Environment) představuje jeden z nejrozšířenějších a nejkomplexnějších systémů pro správu výuky na světě. Na rozdíl od úzce zaměřených mobilních aplikací či rychlých gamifikovaných platforem je Moodle koncipován jako robustní softwarové řešení, které primárně slouží plnohodnotnému institucionálnímu vzdělávání od základních škol přes univerzity až po korporátní sektor. Jeho architektonický i pedagogický základ vychází z filozofie sociálního konstruktivismu, která zdůrazňuje, že vzdělávání je nejefektivnější tehdy, když studenti v interakci se svým okolím aktivně vytvářejí obsah a spolupracují v komunitě spíše než aby pouze pasivně přijímali hotové informace. [10]

Z technického a organizačního hlediska je platforma založena na modulární architektuře. Výuka je zde organizována do jednotlivých kurzů, které může vyučující strukturovat tematicky, chronologicky nebo dle jiných specifických formátů. Každý kurz funguje jako relativně nezávislé virtuální prostředí, do kterého vyučující vkládá dva základní typy obsahu: pasivní studijní materiály (tzn. textové dokumenty, PDF soubory, multimediální odkazy) a interaktivní činnosti. Právě široká paleta činností tvoří jádro celého systému. Patří mezi ně strukturované odevzdávání úkolů, diskusní fóra a chaty, společně tvořené slovníky pojmů,

wiki stránky nebo komplexní testy s velkou škálou typů otázek od výběru z možností přes doplňování textu až po tvoření dlouhých esejí. [10]

Obrovskou výhodou a specifikem platformy Moodle je její open-source licence. To znamená, že instituce mohou systém nejen zdarma využívat, ale také jej plně modifikovat, spravovat jeho zdrojové kódy a hostovat jej na svých vlastních zabezpečených serverech, čímž mají plnou kontrolu nad daty uživatelů. Díky masivní celosvětové komunitě vývojářů existují tisíce doplňků a pluginů, které umožňují Moodle integrovat s dalšími podnikovými či školními systémy. Lze jej tak propojit s videokonferenčními nástroji pro synchronní výuku (např. MS Teams, Zoom), se systémy pro odhalování plagiátorství nebo s pokročilými analytickými nástroji. Tato modularita a otevřenost z něj činí vysoce škálovatelný nástroj přizpůsobitelný jakýmkoliv vzdělávacím potřebám, ačkoliv to s sebou přináší daň v podobě vysoké náročnosti na prvotní nastavení, údržbu serverů a nezbytné zaškolení pedagogů. [10]

2.2 Srovnání Unity6 s konkurenčními technologickými řešeními

Herní engine představují komplexní softwarové nástroje navržené k efektivnějšímu vývoji především her, ale i interaktivních aplikací. Obsahují rozsáhlé funkční celky, které zajišťují základní procesy, jako je simulace fyziky, zpracování zvuku či vykreslování grafických objektů. Vývojářům tak zásadně usnadňují práci, neboť eliminují nutnost programovat tyto základní systémy od úplného začátku. Následující podkapitoly podrobněji představují engine Unity 6, který byl zvolen pro realizaci praktické části této práce, a porovnávají jej s vybranými konkurenčními řešeními.

2.2.1 Unity6

Herní engine Unity představuje jeden z nejrozšířenějších nástrojů pro tvorbu interaktivních aplikací. Jak uvádí Tomáš Holan ve své publikaci [11], Unity je komplexní vývojové prostředí, které tvůrcům zásadně usnadňuje práci tím, že poskytuje již hotové funkce pro nasazení na široké spektrum platform. Oproti tradičním webovým aplikacím, které se v oblasti e-learningu běžně využívají, nabízí Unity zásadní výhodu v podobě snadné tvorby plnohodnotné desktopové hry či aplikace. Webové platformy zpravidla vyžadují neustálé připojení k internetu, pronájem serverů a údržbu složitého backendu, tedy skrytou serverovou část aplikace, která se stará o ukládání dat a veškerou aplikační logiku. Zvolené řešení v Unity naproti tomu umožňuje aplikaci bezproblémově fungovat v plně offline režimu, kdy si uživatelé mohou kurzy vytvářet i studovat lokálně, přičemž veškerá výuková data jsou v projektu ukládána do lokálního úložiště.

Pro vývoj moderních výukových aplikací, které jsou založeny primárně na interakci s textem a formuláři, nabízí Unity 6 pokročilý systém tvorby uživatelského rozhraní zvaný UI Toolkit [12]. Tento systém přináší do vývoje aplikací standardy známé z moderního webového vývoje a striktně odděluje vizuální strukturu díky stylování prostřednictvím značkovacích jazyků UXML a USS od samotné programové logiky v jazyce C#. To v projektu umožňuje čistý a snadno škálovatelný návrh, který se stará o dynamické vykreslování výukového obsahu. Jak zmiňuje Holan, v Unity často stačí poměrně málo čistého kódu k vytvoření

robustního základu [11]. V kombinaci s architekturou jediné scény, jak bude použito v této práci, tak engine umožňuje realizovat plynulé a animované přechody mezi jednotlivými kapitolami bez přerušení a zdlouhavého načítání, což výrazně zvyšuje celkovou uživatelskou přívětivost vzdělávacího softwaru.

Další velkou výhodou Unity 6 oproti webovým řešením či menším frameworkům je však jeho připravenost na bezproblémovou integraci knihoven a distribučních systémů třetích stran, jakým bude v tomto projektu platforma Steam a její Steamworks SDK. Vytvoření vlastního komplexního e-learningového systému by znamenalo programování rozsáhlých databází, řešení autentizace uživatelů a zajištění kybernetické bezpečnosti. Díky této integraci vývojář získává přístup k již ověřené globální infrastruktuře. Aplikace díky tomu bezpečně identifikuje uživatele přes jejich stávající SteamID bez nutnosti tvořit další registrační formuláře, ukládání postupu uživatelů zálohuje prostřednictvím Steam Cloud a distribuci uživatelsky vytvořených kurzů řeší komunitní sdílení v síti Steam Workshop. Unity 6 zde tedy neslouží primárně jako nástroj pro vykreslování grafiky, ale jako stabilní spojovací prvek mezi interaktivním frontendem (neboli viditelným uživatelským rozhraním aplikace) pro uživatele a spolehlivou backendovou (serverovou) architekturou Steamu.

2.2.2 Unreal engine 5

Nejvýraznějším přímým konkurentem enginu Unity na současném trhu je Unreal Engine 5 od společnosti Epic Games. Jedná se o průmyslový standard, který dominuje především v oblasti filmové produkce, architektonických vizualizací a takzvaných AAA herních titulů. Pojmem AAA (čteno jako „tří áčkové“) se v herním průmyslu označují projekty s vysokými rozpočty, které představují technologickou a vizuální špičku trhu. Právě vysoká úroveň grafického zpracování je největší předností tohoto enginu, a to zejména díky dvěma technologiím pojmenovaným Nanite a Lumen. Nanite je systém virtualizované geometrie, který umožňuje vývojářům importovat 3D modely s miliony polygonů ve filmové kvalitě bez nutnosti jejich ruční optimalizace, a to díky tomu, že engine jejich detaily dynamicky vykresluje podle vzdálenosti od kamery. Technologie Lumen naproti tomu představuje plně dynamický systém globálního osvětlení a odrazů, který okamžitě a realisticky reaguje na změny ve scéně (například na pohyb slunce nebo rozsvícení baterky) bez nutnosti zdlouhavého přepočítávání takzvaných světelných map. Dále engine nabízí systém vizuálního skriptování zvaný Blueprints, který umožňuje tvořit logiku aplikací bez psaní klasického kódu. [13]

Přestože je Unreal Engine 5 technologickým lídrem v oblasti vykreslování trojrozměrného prostředí, pro vývoj výukové aplikace zaměřené na interakci s textem a formuláři přináší jeho architektura řadu komplikací. Zatímco Unity 6 nabízí pro tyto účely systém UI Toolkit, Unreal Engine využívá k tvorbě rozhraní systém zvaný UMG [14]. UMG¹ je však primárně navrženo jako nadstavba nad trojrozměrným prostorem. Tvorba dynamicky generovaných dvourozměrných seznamů, jako jsou kapitoly a moduly, je v něm často méně

¹ UMG = vizuální nástroj v Unreal Enginu 5 určený k tvorbě uživatelského rozhraní

přímočará a vyžaduje složitější provázání s herní logikou [14]. Chybí zde lehkost webového stylování, kterou nabízí právě UI Toolkit v Unity.

Dalším důležitým faktorem pro e-learningový software je hardwarová a paměťová náročnost. Unreal Engine 5 je ze své podstaty koncipován jako takzvaný „heavyweight²“ engine. I zcela prázdný projekt bez objektů obsahuje velké množství základních knihoven a systémů běžících na pozadí, což se negativně podepisuje na velikosti výsledného programu a hardwarových požadavcích na klientské zařízení. Výukové aplikace jsou často spouštěny na běžných, méně výkonných kancelářských noteboocích nebo školních počítačích. Unity v tomto ohledu nabízí mnohem lépe škálovatelný výkon.

Z hlediska distribuce a integrace služeb třetích stran, jako je Steamworks SDK, nabízí Unreal Engine 5 srovnatelnou, ne-li v některých ohledech lepší podporu prostřednictvím vlastního modulu Online Subsystem Steam [14]. Umožňuje bezproblémové napojení na Steam Cloud i Steam Workshop. Nicméně, po zvážení celkové architektury aplikace, ve které je hlavním prvkem plynulé dvourozměrné uživatelské rozhraní a práce s textovými výukovými moduly, se zvolení Unreal Enginu 5 jeví jako neefektivní využití zdrojů. Pro daný typ projektu představuje Unity 6 výrazně vhodnější vývojové prostředí.

2.2.3 Godot

Stále významnějším a dynamicky rostoucím konkurentem na trhu herních enginů je Godot. Jeho největším specifíkem je skutečnost, že se jedná o plně open-source software distribuovaný pod licencí MIT, což znamená, že je zcela zdarma pro komerční i nekomerční využití bez jakýchkoliv licenčních poplatků či nutnosti odvádět procenta ze zisku, a to z něj činí atraktivní volbu pro nezávislé vývojáře i akademické instituce [15]. Architektura enginu Godot je postavena na uzlovém systému zvaném Node-based architecture, v níž je každý prvek scény, od grafického spritu³ přes zvukový přehrávač až po uživatelské rozhraní, reprezentován uzlem, z nichž se skládají hierarchické stromy [16].

Na rozdíl od Unreal Enginu disponuje Godot dedikovaným a plnohodnotným 2D vykreslovacím jádrem. Pro dvourozměrné hry a aplikace tak nedochází k žádnému simulování třetího rozměru, což přináší lepší optimalizaci. Přestože by Godot mohl být pro tvorbu 2D výukových aplikací hardwarově velmi nenáročný a efektivní, v otázce tvorby komplexního uživatelského rozhraní nabízí odlišný přístup než Unity 6. Godot využívá pro tvorbu UI⁴ takzvané Control nodes, které se vkládají přímo do scény společně s ostatními herními prvky [16]. Ačkoliv je tento systém vizuálně intuitivní, u rozsáhlejších e-learningových aplikací s větším množstvím textu, formulářů a dynamicky generovaného obsahu může vést k nepřehlednému propletení vizuální stránky a aplikační logiky.

² Heavyweight engine = engine koncipovaný tak, aby měl v sobě zabudovaných hodně funkcí pro usnadnění vývoje, výpočtů, optimalizaci atd. výměnou za hardwarové nároky na výkon či kapacitu systému

³ Sprite = základní dvourozměrný prvek

⁴ UI = uživatelské rozhraní

Dalším rozhodujícím faktorem pro volbu vývojového prostředí je programovací jazyk a podpora integrací třetích stran. Primárním skriptovacím jazykem enginu Godot je GDScript, což je jazyk podobný Pythonu, který je vytvořený specificky pro potřeby tohoto enginu. Přestože Godot nabízí i oficiální podporu jazyka C# prostřednictvím architektury .NET, jedná se až o sekundární řešení [15]. Ekosystém Unity je naopak kolem jazyka C# budován od samého počátku, což se odráží ve velkém množství dostupných komunitních i oficiálních knihoven. V kontextu této bakalářské práce je klíčová integrace služby Steam. I když pro Godot existují komunitní řešení jako je GodotSteam, implementace ověřených C# wrapperů v přirozeném .NET prostředí Unity představuje spolehlivější a lépe dokumentovanou cestu pro naplnění funkčních požadavků spojených s distribucí a synchronizací obsahu v cloudu [17].

2.2.4 Webové řešení

Nejběžnější a nejvyužívanější alternativou k herním enginům pro tvorbu vzdělávacího softwaru jsou standardní webové aplikace. Platformy tohoto typu jsou vyvíjeny pomocí tradičních webových jazyků jako jsou HTML, CSS a JavaScript. Webové aplikace přinášejí nespornou výhodu v maximální dostupnosti, protože si uživatel nemusí instalovat žádný dedikovaný software a aplikaci spustí v jakémkoliv moderním prohlížeči. Pro tvorbu čistě dvourozměrných výukových aplikací založených na textu a formulářích představuje webový vývoj dlouhou dobu technologický standard.

Při srovnání s enginem Unity 6 však dochází k zásadnímu posunu. Unity v minulosti pro 2D rozhraní využívalo systém uGUI, který pro čistě textové aplikace nebyl vždy ideální. S příchodem nového systému UI Toolkit se ale rozdíl mezi webovým vývojem a herním enginem zmenšuje. UI Toolkit napodobuje webové standardy a využívá značkovací jazyk UXML a USS, což jsou obdoby HTML a CSS, k oddělení vizuální struktury od programové logiky v jazyce C# [12]. Vývojář tak získává flexibilitu a čistotu návrhu známou z webového vývoje, avšak s přidanou hodnotou nativního výkonu enginu, který umožňuje bezproblémové a hardwarově akcelerované animace plynulých přechodů a komplexní manipulaci s objekty, což by ve webovém prohlížeči vyžadovalo složitější optimalizaci prostřednictvím dodatečných knihoven.

Nejvýraznější rozdíl mezi tradiční webovou aplikací a vyvíjeným řešením v Unity spočívá v architektuře síťového připojení a řešení bezpečnosti. Typická webová platforma funguje na principu klient-server. Aby mohla ukládat uživatelský postup nebo nabízet tvorbu komunitních kurzů, vyžaduje neustálé připojení k internetu, pronájem hostingu a správu databází. Výuková aplikace v Unity naproti tomu funguje jako plnohodnotný desktopový klient. Veškerá data o kurzech, kapitolách a chybných odpovědích jsou ukládána na lokální disk uživatele, což aplikaci umožňuje plně fungovat i v offline režimu.

S provozem vlastního webového backendu se navíc pojí zátěž v podobě kybernetické bezpečnosti správy uživatelských účtů. Vývojář webové aplikace musí navrhnout bezpečný registrační a přihlašovací systém, šifrovat hesla, chránit databázi proti útokům a řešit legislativu ochrany osobních údajů označovanou jako GDPR. Tento projekt tuto komplexní

bezpečnostní vrstvu řeší integrací služby Steam díky nástrojům Steamworks SDK. Autentizace probíhá výhradně na straně Steam klienta pomocí existujícího SteamID bez nutnosti tvořit další registrační systém. Ukládání postupu do cloudu i samotnou distribuci a sdílení komunitou vytvořených kurzů na Steam Workshop zajišťuje globálně ověřená infrastruktura společnosti Valve. Vývojář v Unity se tak může plně soustředit na tvorbu a kvalitu samotné aplikace, zatímco bezpečnost uživatelských účtů přebírá platforma Steam.

2.3 Distribuce a Steamworks SDK

Platforma Steam od společnosti Valve představuje v současnosti největší celosvětovou síť pro digitální distribuci primárně počítačových her, ale i softwaru. Společně s distribuční sítí poskytuje vývojářům také sadu vývojářských nástrojů a rozhraní známou jako Steamworks SDK⁵. V kontextu bakalářské práce představuje využití funkcí Steamworks důležité rozhodnutí. Herní engine Unity zde slouží primárně jako klient pro vykreslování vizuálního obsahu a aplikační logiku, zatímco infrastruktura Steamu plně nahrazuje nutnost vývoje, provozu a zabezpečování vlastního serveru. Integrace služeb Steamworks do výukové aplikace řeší tři důležité pilíře online infrastruktury. Autentizaci uživatelů, zálohování dat a distribuci komunitního obsahu.

2.3.1 Autentizace a správa uživatelské identity

Jedním z problémů při tvorbě moderních e-learningových systémů je správa uživatelských účtů. Vývojář musí řešit bezpečně ukládání a šifrování hesel, obnovu zapomenutých přístupů a striktní dodržování legislativy týkající se ochrany osobních údajů. Implementací nástrojů Steamworks SDK je tento problém přesunut na platformu Steam, kde ho řeší sama společnost Valve.

Po spuštění aplikace se integrované API⁶ automaticky spojí s běžícím klientem Steam na počítači uživatele. Z něj získá bezpečně ověřený, unikátní identifikátor, takzvané SteamID. Tento proces zcela eliminuje nutnost vytvářet v aplikaci registrační a přihlašovací formulář. Pro uživatele to znamená okamžitý přístup ke kurzu ihned po spuštění aplikace bez zdlouhavé registrace či přihlašování a pro vývojáře pak minimalizaci bezpečnostních rizik spojených se správou databází s uživatelskými účty.

2.3.2 Distribuce obsahu pomocí Steam Workshop

Nejdůležitějším prvkem celého ekosystému je využití služby Steam Workshop. Tato platforma původně vznikla k tomu, aby umožnila hráčům sdílet vlastní modifikace do her a obdobný obsah, nicméně dokáže sloužit i jako globální úložna vzdělávacího obsahu. Umožňuje přetvořit uzavřený výukový nástroj na otevřenou komunitní platformu.

⁵ SDK = Software Development Kit

⁶ API = soubor pravidel, protokolů a nástrojů, který umožňuje různým softwarovým aplikacím spolu vzájemně komunikovat a předávat si data. Zkratka vychází ze spojení slov Application Programming Interface

Když tvůrce či vyučující dokončí tvorbu svého kurzu ve tvořivém režimu aplikace, bude mít možnost jej publikovat. Aplikace v tomto kroku využije API Steamu k vytvoření takzvané nové položky takzvané Workshop Items [18]⁷. Vezme strukturu kapitol zapsanou v lokálním JSON souboru, zkompletuje ji se všemi přiloženými studijními materiály do jednotného datového balíčku a odešle jej na servery Steamu společně s názvem kurzu, jeho popisem a náhledovým obrázkem.

Z pohledu koncového studenta je pak distribuce uživatelsky přívětivá. Aniž by musel hledat instalační soubory kurzů na externích webových stránkách, může si přímo přes rozhraní platformy Steam prohlížet katalog komunitně vytvořených kurzů. Kliknutím na tlačítko pro odběr dá klientu Steam pokyn ke stažení daného obsahu.

Služba Steam Workshop se následně plně stará o:

- **Doručování dat:** Spolehlivě stáhne soubory kurzu do vyhrazené složky na lokálním disku uživatele.
- **Verzování a aktualizace:** Pokud autor původního kurzu provede opravu (například opraví chybu v zadání kvízu nebo přidá obsah) a vydá novou verzi, Steam Workshop automaticky stáhne tuto aktualizaci před spuštěním aplikace na pozadí všech studentů, kteří daný kurz odebírají.

2.3.3 Technická realizace v prostředí Unity

Samotné Steam API je nativně napsáno v jazycích C a C++. Pro jeho implementaci do prostředí herního enginu Unity, které je postaveno na architektuře .NET a jazyce C#, je nezbytně nutné využít takzvaný wrapper, zprostředkující knihovnu, jako je například komunitou udržovaný Steamworks.NET [19]. Tato knihovna slouží jako překladatel, který zpřístupňuje C++ struktury skrze C# metody, což vývojáři umožňuje snadno volat funkce jako `SteamAPI.Init()` pro inicializaci komunikace nebo spravovat odesílání dat na Workshop pomocí asynchronního programování právě v C# jazyce.

⁷ Přesný odkaz = <https://partner.steamgames.com/doc/features/workshop/implementation>

3 Specifikace požadavků a návrh aplikace

Před samotným zahájením programování a vývoje softwarového řešení je nezbytné jasně definovat cíle projektu a stanovit hranice budoucího systému. Tato kapitola se proto nejprve věnuje detailní specifikaci požadavků kladených na vyvíjenou výukovou aplikaci, které slouží jako hlavní stavební kámen pro následný vývoj. Na základě těchto požadavků je v druhé části kapitoly představen návrh celkové architektury aplikace a struktura jejího datového modelu pro ukládání výukových kurzů.

3.1 Funkční a nefunkční požadavky

Před vývojem softwaru je důležité stanovit si požadavky, kterých je potřeba se při vývoji držet. „Funkční požadavky“ nám slouží k pojmenování konkrétních funkcí a služeb softwaru, které musí být uživateli k dispozici. „Nefunkční požadavky“ pak slouží k popsání pasivních charakteristik programu.

3.1.1 Funkční požadavky

Funkční požadavky přesně popisují chování budoucího systému. Definují konkrétní operace, procesy a služby, které musí aplikace svým uživatelům poskytovat, aby splnila svůj hlavní účel.

Tab. 1: Funkční požadavky

FP01	Identita uživatele	Musí být použita identita uživatele ze Steam klienta (SteamID) bez nutnosti další registrace
FP02	Inicializace Steam API	Po spuštění aplikace musí být inicializováno Steamworks API
FP03	Volání Steam overlay	Musí být umožněno volání Steam overlay z UI aplikace
FP04	Ukládání kurzů	Uživateli vytvářejícímu kurz musí být umožněno lokální uložení postupu
FP05	Publikování kurzu	Uživateli vytvářejícímu kurz musí být umožněno publikovat kurz prostřednictvím Steam Workshopu
FP06	Stahování kurzů	Studujícímu uživateli musí být umožněno stáhnout si kurz ze Steam Workshopu
FP07	Offline režim	Aplikace musí být schopná běžet offline bez použití Steamu (lokální ukládání vytváření kurzů a progresu kurzů studovaných)

FP08	Obsah lekcí	Data na obsah lekcí budou načítána z JSON souborů a při vytváření kurzů do něj budou ukládána
FP09	Okamžitá zpětná vazba	Po zodpovězení otázky v kurzu program zobrazí okamžitou zpětnou vazbu a správnou odpověď
FP10	Input	Aplikace musí umožnit vstup pomocí myši a klávesnice
FP11	Opakování otázek	Aplikace musí ukládat špatně zodpovězené otázky uživatele a musí umožnit je znovu zodpovědět

3.1.2 Nefunkční požadavky

Na rozdíl od funkčních požadavků nedefinují nefunkční požadavky konkrétní služby a akce, ale zaměřují se na kvalitativní vlastnosti systému a omezující podmínky pro jeho celkový běh.

Tab. 2: Nefunkční požadavky

NP01	Integrita dat	Uložená data musí zůstat neporušená i při náhlém pádu aplikace.
NP02	Podpora OS	Aplikace bude podporována na operačních systémech Windows 10 a 11
NP03	Bezproblémová distribuce	Finální build aplikace musí být připraven tak, aby byl nainstalovatelný a spustitelný z platformy Steam bez dodatečných manuálních kroků uživatele

3.2 Návrh architektury datového modelu

Pro zajištění maximální přenositelnosti, rychlého čtení a možnosti snadné úpravy, byl pro ukládání výukových dat zvolen formát JSON⁸. Tento odlehčený textový formát umožňuje hierarchické a strukturované ukládání dat pomocí párů klíč-hodnota, což odpovídá potřebné struktuře pro kurzy.

⁸ JSON = zkratka pochází ze spojení slov JavaScript Object Notation

3.2.1 Hierarchická struktura JSON souboru

Veškerá data definující obsah a tok kurzu jsou serializována⁹ do jednoho hlavního JSON souboru, který funguje jako databáze pro konkrétní kurz. Data jsou uvnitř logicky členěna do několika do sebe vnořených sekcí:

- **Základní metadata kurzu:** Na nejvyšší úrovni soubor obsahuje obecné informace o kurzu. Jedná se o klíče `courseName` s hlavním názvem kurzu a `description` s krátkým popisem kurzu. Tyto údaje slouží pro prvotní identifikaci kurzu v hlavní nabídce aplikace a při případném nahrávání na Steam Workshop.
- **Sekce kapitol:** Hlavním nositelem struktury je pole objektů nazvané `chapters`. Každý objekt v tomto poli reprezentuje jednu ucelenou lekci. Obsahuje klíče `chapterIndex` pro určení pořadí kapitoly v kurzu, `chapterName` pro uložení názvu kapitoly.
- **Sekce výukových modulů:** Pole modulů obsahuje samotný edukační obsah. Každý modul je specifikován klíčem `type`, který aplikaci sděluje, jak má daný blok vykreslit a jakou logiku na něj aplikovat (např. klíč `card` pro výkladovou kartu, `quiz` pro kvíz, `dragAndDrop` pro seřazování nebo `notes` pro poznámky či výukový text). Dále obsahuje klíč `indexInChapter` pro určení pořadí v rámci dané kapitoly. Moduly pak mají své další klíče pro otázky, odpovědi, možnosti, či pořadí

3.2.2 Ukládání a správa příložených souborů

Aby byla zajištěna integrita kurzu, zejména při jeho přenášení mezi zařízeními nebo nahrávání z cloudu, využívá systém ukládání složek kurzu v adresáři `AppData`¹⁰ ve Windows. Tím je zajištěno, že na operačních systémech Windows 10 a 11 budou kurz a jeho soubory ukládány vždy do stejného umístění na jakémkoliv zařízení.

Při založení nového kurzu se vedle hlavního JSON souboru automaticky vygeneruje podadresář s názvem „files“. Do této složky jsou následně vkládány všechny uživatelem přiložené materiály, ať už se jedná o výukové obrázky, schémata, nebo dodatečné dokumenty ke studiu.

Když uživatel prostřednictvím editačního rozhraní vybere soubor k nahrání, aplikace neponechá jeho původní název, ale před vložením do složky „files“ jej automaticky přejmenuje podle vývojářem určeného způsobu. Tento krok je důležitý z několika důvodů. Zabraňuje nechtěnému přepsání souborů se stejným názvem, usnadňuje automatizované čištění nepoužívaných souborů a na první pohled prozrazuje, ke které konkrétní části výuky soubor náleží.

⁹ Serializace = proces, při kterém jsou data nebo objekty převedeny na text

¹⁰ `AppData` = skrytá systémová složka, která je unikátní pro každého uživatele ve Windows systému

Pojmenování probíhá podle dvou typů souborů:

- **Obrázky pro kvízový modul:** Formát pro pojmenování je „obr_chXXX_iichXXX_název-obrazku.typ-souboru“. „Obr“ značí, že se jedná o obrázek, „XXX“ za písmenem „ch“ značí v kolikáté kapitole se obrázek nachází a „XXX“ za písmeny „iich“ značí, jaké pořadí má modul v kapitole, ve kterém je obrázek přiřazen.
- **Soubory přiložené skrze poznámkový modul:** Formát pro pojmenování je „file_chXXX_iichXXX_název-souboru.typ-souboru“. „File“ znamená, že se jedná o soubor (může být i obrázek, ale je určen jako podklad pro studium), „XXX“ za písmenem „ch“ značí v kolikáté kapitole se obrázek nachází a „XXX“ za písmeny „iich“ značí, jaké pořadí má modul v kapitole, ve kterém je obrázek přiřazen.

Část kódu sestavující kompletní název souboru ve skriptu EditChapter.cs

```
string chPart = chapterDisplayNumber.ToString("D3");
string iichPart = indexInChapter.ToString("D3");
string safeName = Path.GetFileName(originalFileName) ?? "file";
return $"file_ch{chPart}_iich{iichPart}_{safeName}";
```

Toto pojmenování pak slouží ke správnému zobrazování u jednotlivých výukových modulů při studiu a správnému zobrazení toho, ve které kapitole se soubor nachází v kompletním seznamu souborů. Tyto dlouhé předpony se uživateli nikde v aplikaci nezobrazují. Jsou odfiltrovány pomocí regulárních výrazů, které fungují jako filtr a uživatel pak vidí pouze originální názvy souborů tak, jak je vložil.

4 Tvořivý režim

Do tvořivého režimu se uživatel dostane proklikem na tlačítko „vytvořit“ v hlavním menu aplikace a následně může vytvářet a upravovat své vlastní kurzy. V této kapitole bude popsáno, jak fungují jednotlivé stránky tvořivého režimu z pohledu funkcí v kódu i z pohledu uživatele a jaké funkce jsou mu k dispozici.

4.1 Úprava kapitol dle funkcí kódu

Jádro funkční části rozhraní na úpravu obsahu s kapitolami je soustředěno v obsáhlém skriptu `CreateMenu.cs`. Tento skript dědí vlastnosti ze základní třídy `MonoBehaviour`, což je třída v Unity, která umožňuje programovému kódu napojit se na vnitřní životní cyklus samotného vývojového prostředí, například reagovat na spuštění, vypnutí nebo vykreslování snímků.

Jak již bylo zmíněno, využívá aplikace k vykreslování obsahu systém UI Toolkit. V kódu se tak pracuje s objekty typu `VisualElement`, což je základní stavební blok rozhraní. `VisualElement` je ekvivalentem HTML tagu `<div>` ve webovém vývoji. K propojení napsaného kódu s grafickým rozhraním slouží metoda typu `Query`. Kód si díky tomu po načtení stránky vyhledá připravené prázdné kontejnery (například `ScrollView` pro scrollovatelný seznam) a uloží si na ně odkazy do paměti, aby s nimi mohl později pracovat.

Klíčovým aspektem tvorby aplikace je správná správa paměti a událostí (tzv. eventů). Skript využívá systémové metodu `OnEnable`, kterou Unity volá při aktivaci stránky a `OnDisable`, které si zavolá při opuštění stránky. V metodě `OnEnable` skript začne pracovat s událostmi, například reaguje na kliknutí myši (`PointerDownEvent`). Je ale důležité, aby skript v metodě `OnDisable` práci s těmito událostmi ukončil (`UnregisterCallback`) a to i všechny běžící procesy na pozadí. Pokud by tak neučinil, aplikace by se při častém přepínání stránek začala zpomalovat kvůli zahlcení paměti (tzv. `memory leak`).

Rozhraní není statické, ale musí se přizpůsobit tomu, jaký kurz uživatel zrovna upravuje. Data v JSON souboru slouží ke dvěma účelům:

- **Čtení dat:** Stránky plně fungují v offline režimu. Veškerá data o kurzech se ukládají do lokálních souborů ve formátu JSON. Metoda `RenderFromActiveJson` tento soubor na disku najde, načte jej a převede text do strukturovaných C# objektů.
- **Vykreslování:** Na základě načtených dat (např. pole kapitol) kód dynamicky, v takzvaném cyklu `for`, vytváří nové instance `VisualElement`. Těmto prvkům programově přiřazuje texty (třída `Label`), USS třídy pro vzhled (`AddToClassList`) a interaktivitu (např. vázání metody na kliknutí myši). Takto se sestavuje celý centrální seznam kapitol (metoda `RenderChapters`).

Aby nepůsobilo rozhraní trhaně, obsahuje plynulé animace (např. zvětšení kapitoly po kliknutí nebo vysunutí tlačítek "Upravit" a "Smazat" po kliku na kartu kapitoly). K tomu slouží takzvané Corutiny (`StartCoroutine`). Běžný kód se vykoná okamžitě. Corutina je

však speciální funkce, která dokáže své vykonávání pozastavit, počkat na vykreslení dalšího snímku na monitoru a plynule měnit parametry prvku (např. jeho průhlednost nebo velikost) v čase, aniž by zablokovala celou aplikaci.

4.2 Úprava obsahu kapitol dle funkcí kódu

Tato podkapitola je věnována detailnímu rozboru skriptu `EditChapter.cs`, který tvoří jádro rozhraní pro tvorbu obsahu kapitoly. Zatímco nadřazené menu (které bylo popsáno v předchozích kapitolách) řeší strukturu kurzu, tento skript umožňuje uživateli vstupovat do konkrétních kapitol a plnit je interaktivními výukovými moduly, jako jsou výkladové karty, kvízy, drag & drop cvičení nebo poznámky s přílohami.

Skript využívá stejné principy životního cyklu a napojení na rozhraní jako `CreateMenu.cs`. I zde je využívána funkce `Update()` například pro naslouchání stisku klávesy ESC, která zde slouží k zavření vyskakovacích oken nebo k přechodu zpět na stránku pro správu kapitol, pokud nejsou otevřena žádná vyskakovací okna.

Zatímco generování seznamu kapitol je poměrně přímočaré, sestavování obsahu kapitoly je komplexnější. Na základě načtených dat ze souboru JSON skript identifikuje typ modulu (karty, poznámky, kvíz atd.) a zavolá příslušnou metodu. U složitějších modulů kód pouze načte data, ale dokáže dynamicky a vizuálně reagovat na lokální uživatelské vstupy. Pokud například uživatel u kvízu změní počet odpovědí, skript po kliku na tlačítko plus nebo mínus, vytvoří určený počet zadávací polí pro odpovědi.

Aby aplikace mohla fungovat spolehlivě a plynule bez nutnosti používat centrální tlačítko pro uložení změn, využívá techniku okamžité synchronizace. Každý interaktivní prvek (například textové pole `TextField` pro zadání definice pojmu, otázky či odpovědi) má na sobě softwarově zavěšený takzvaný `callback` (neboli funkci zajišťující zpětné volání) reagující na událost změny hodnoty.

Jakmile uživatel napíše či smaže, klidně i jediný znak, událost se zachytí a skript asynchronně přepíše konkrétní vlastnost uvnitř JSON souboru na disku. Tím je zajištěna integrita dat, a to i při náhlém pádu aplikace, a uživatel tak neztratí své poslední zapsané změny.

Jednou z dalších důležitých funkcí tohoto skriptu je vkládání lokálních souborů z počítače (např. vkládání obrázků do kvízů nebo přidávání studijních dokumentů k poznámkám). Unity aplikace běží ve svém vlastním izolovaném prostředí (tzv. `sandboxu`). Pro zobrazení klasického systémového „průzkumníka souborů“ bylo nutné programově vystoupit z tohoto enginu a komunikovat přímo s operačním systémem. K tomu kód využívá speciální jmenný prostor `System.Runtime.InteropServices`. Tento přístup umožňuje aplikaci v jazyce C# vypůjčit si a spustit nativní funkce napsané v jazyce C++, které jsou zabudované přímo v knihovnách systému Windows (konkrétně se inicializuje a volá struktura `OPENFILENAME` z Windows API). Během toho, co aplikace čeká, až uživatel najde a vybere požadovaný soubor ve svých složkách, celý proces běží pomocí dříve představených `corutin`. Díky tomu je hlavní vykreslovací vlákno enginu odlehčeno, systémové okno nezpůsobí „zamrznutí“ aplikace a veškeré animace a interakce v rozhraní na pozadí zůstávají plně funkční. Po

potvrzení výběru pak skript s využitím standardních tříd pro vstup a výstup soubor zkopíruje do předem definované složky v `AppData` a zaznamená jeho název do JSON souboru.

4.3 Vytvoření nového kurzu dle funkcí kódu

Tato kapitola analyzuje skript `CreateNew.cs`, který slouží jako vstupní brána pro tvůrce vzdělávacího obsahu. Úkolem této části aplikace je zpracovat počáteční vstupy od uživatele, jako jsou název kurzu a jeho stručný popis, vytvořit pro ně odpovídající adresářovou strukturu na pevném disku a založit výchozí datový soubor.

Stejně jako u předchozích skriptů je i zde využita architektura dědění od třídy `MonoBehaviour` a správa životního cyklu prostřednictvím metod `OnEnable` a `OnDisable`, které zajišťují bezpečné přihlašování k událostem nadřazeného správce obrazovek (tedy skript `SingleSceneMenuFlow.cs`).

Po kliku na tlačítko „nový“ na stránce `CreateOption`, která slouží k výběru toho, zda uživatel chce vytvořit nový kurz či pokračovat na úpravu již existujícího kurzu, se spustí metoda `Bind`. Ta má za úkol propojit napsaný C# kód s grafickými prvky UI Toolkitu.

Samotný proces založení kurzu se spouští po kliknutí na tlačítko "Vytvořit", které vyvolá metodu `TryCreateJson`. Celá tato metoda je obalena v takzvaném `try-catch` bloku. Jedná se o standardní programovací techniku pro zachytávání chyb. Pokud se kdekoliv v procesu tvorby složek nebo zápisu dat stane neočekávaná chyba (například uživatel nemá práva k zápisu na disk, nebo nevyplní jméno), kód v bloku `try` se okamžitě přeruší a řízení převezme blok `catch`. Ten chybu zachytí a místo toho, aby celá aplikace takzvaně spadla, vypíše text chyby uživateli do připraveného červeného popisu na obrazovce.

Jakmile je název validní, skript vyhledá hlavní kořenovou složku pro kurzy (standardně v dříve vysvětleném systémovém úložišti `AppData`) a přistoupí k vytvoření nové podsložky pro konkrétní kurz. Zde nastává další důležitý bod: co se stane, když uživatel vytvoří dva různé kurzy se stejným názvem (například „Matematika“)? Kdyby program bezmyšlenkovitě vytvořil složku, přepsal by původní kurz. Skript `CreateNew.cs` to řeší pomocí dedikované metody `MakeUniqueDirectoryPath`. Tato metoda funguje na principu nekonečného cyklu `while(true)`. Nejprve se pokusí vytvořit složku s původním názvem. Pokud zjistí, že složka `Matematika` již existuje, vytvoří si vnitřní počítadlo začínající číslem 2 a zkusí název `Matematika (2)`. Pokud i ta existuje, cyklus se zopakuje s číslem 3, dokud nenajde volný a zcela unikátní název na disku. To však nepřejmenuje název kurzu, ale pouze složku kurzu, ve kterém se nachází. V aplikaci uživatel uvidí pouze dva kurzy s názvem, který zadal. To je potřeba zachovat z důvodu, aby bylo možné vytvářet kurzy i různými uživateli bez nutnosti vymýšlet unikátní název kurzu.

V posledním kroku, když je na disku připravena unikátní složka, vygeneruje skript úplnou cestu k JSON souboru. Pomocí třídy `CourseJson.CreateNew` vytvoří v paměti výchozí prázdnou strukturu kurzu, do které vloží název a popis kurzu. Tato struktura je následně zapsána na disk uživatele.

4.4 Výběr rozpracovaného kurzu dle funkcí kódu

Výběr rozpracovaného kurzu je proveden skrze skript `CourseChecker.cs`, který tvoří softwarové jádro pro obrazovky výběru kurzů. Jeho hlavním úkolem je dynamicky prohledat lokální úložiště v `AppData`, načíst metadata uložených kurzů a vygenerovat z nich interaktivní seznam pro uživatele. Na rozdíl od dříve rozebíraných skriptů se v tomto případě nejedná o komponentu typu `MonoBehaviour`, která by byla fyzicky umístěna na nějakém herním objektu ve scéně. `CourseChecker` je navržen jako takzvaná statická třída (`public static class`). To znamená, že funguje jako globální knihovna pomocných funkcí, kterou může jakákoliv jiná část aplikace zavolat bez nutnosti vytvářet její instanci. Tento přístup výrazně usnadňuje znovupoužitelnost kódu. Stejná část kódu se díky tomu stará jak o načítání kurzů rozpracovaných tvůrcem, tak o načítání hotových kurzů určených ke studiu (rozebráno v pozdější kapitole).

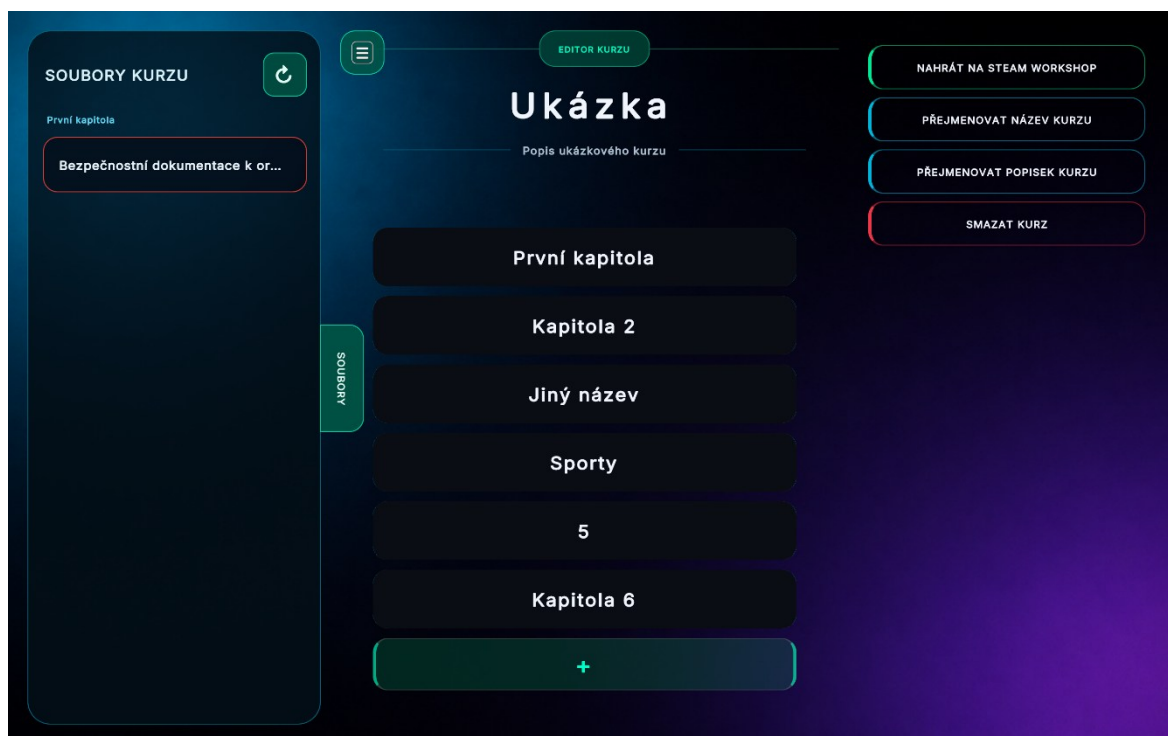
Fungování skriptu lze rozdělit do tří hlavních kroků:

- **Prohledávání lokálního disku a chytré řazení:** Jakmile rozhraní požádá o vykreslení seznamu, zavolá se hlavní metoda `RefreshFromDirectory`. Tato metoda nejprve pro jistotu ověří existenci cílové složky a následně pomocí systémové třídy vyhledá všechny soubory s koncovkou „.json“ nacházející se ve všech adresářích kurzů. Pokud skript nějaké kurzy najde, přistoupí k jejich řazení (metoda `Sort`). Aby bylo uživatelské prostředí co nejpřívětivější, neřadí se kurzy primárně podle abecedy, ale podle času posledního vytvoření kurzu. Tím je zajištěno, že kurz, který tvůrce vytvořil naposledy, se vždy zobrazí na vrcholu seznamu.
- **Čtení JSON dat:** Kurzy mohou obsahovat obrovské množství textu a dat. Kdyby skript při vykreslování seznamu načítal kompletní paměťovou strukturu každého kurzu, aplikace by se při desítkách kurzů výrazně zpomalila. Z tohoto důvodu skript implementuje optimalizaci pomocí vlastní vnitřní třídy `CoursePreviewFile` opatřené atributem `[Serializable]`. Když systémová funkce `JsonUtility.FromJson` čte lokální soubory, nehledá v nich složité definice kapitol nebo testových otázek. Zaměří se výhradně na základní pole pro název a popis. Tento přístup funguje jako rychlý filtr. Získaná data navíc skript vizuálně čistí (metoda `NormalizeText`), odstraňuje přebytečné mezery či prázdné řádky, a pokud je textový popis příliš dlouhý, automaticky jej ořízne na maximálně 50 znaků (`BuildDescriptionPreview`), aby nedošlo k rozbití vizuálního rozhraní.
- **Dynamické sestavení uživatelského rozhraní (UI Toolkit):** Jakmile má skript připravena rozříděná a očištěná metadata, vykreslí vizuální karty (metoda `BuildCourseRow`). Kód opět využívá systém UI Toolkit. Pro každý nalezený kurz se dynamicky vytvoří nový obdélníkový kontejner, do kterého se vloží textové popisky s názvem a popisem kurzu. Aby autor ušetřil složité stylování zásahem do C# kódu, přiřazuje prvkům pouze USS třídy. Kód se stará primárně o

oživení rozhraní, registruje takzvané události. Pomocí `PointerEnterEvent` a `PointerLeaveEvent` skript detekuje, kdy uživatel na kartu kurzu najede myší, a dynamicky jí přidá třídu `is-hover`, což vizuálně aktivuje podkreslovací grafiku. Pomocí `ClickEvent` je na kartu navázána funkce, která po kliknutí odešle systému přesnou cestu ke zvolenému JSON souboru, čímž se aplikace plynule přepne do editačního režimu konkrétního kurzu. Dále je použita vlastnost `PickingMode.Ignore` u textových vrstev uvnitř vygenerovaného tlačítka. Tím je zajištěno, že samotné texty nebudou fyzicky blokovat kliknutí myši a celá karta bude reagovat jako jeden souvislý interaktivní prvek.

4.5 Uživatelský pohled na tvořivý režim

Cílem výukové aplikace není pouze zprostředkovat informace studentům, ale také poskytnout lektorům a tvůrcům intuitivní prostředí pro tvorbu obsahu. Uživatelské rozhraní tvořivého režimu je proto navrženo s důrazem na minimalizaci rušivých elementů. Celkový vizuální styl je laděn do tmavých odstínů, které neoslňují a snižují únavu očí při delší práci. Tmavý základ je doplněn o výrazné neonové prvky, které vizuálně oddělují jednotlivé sekce a jasně indikují interaktivní části.



Obr. 1: Hlavní stránka tvořivého režimu
(Zdroj: Vlastní)

4.5.1 Stránka CreateMenu

Stránka pojmenovaná jako CreateMenu slouží jako centrální bod pro organizaci celého kurzu. Důležitým prvkem stránky je vertikálně řazený seznam vytvořených kapitol. Uživatel může velmi snadno přidávat další kapitoly kliknutím na tlačítko se symbolem plus na konci seznamu. Při kliknutí myši na konkrétní kapitolu se plynule vysunou možnosti pro její úpravu či smazání. Smazat lze ale jen poslední kapitolu v seznamu.

V pravé horní části obrazovky, jak je možné vidět na Obr. 1, jsou tvůrci neustále k dispozici nástroje pro úpravu názvu kurzu či popisu kurzu, jeho vydání na Steam Workshop a případně jeho smazání. Po kliknutí na ně se zobrazí vyskakovací menu s možností daného tlačítka.

Zásadním prvkem pro udržení přehlednosti je panel viditelný na levé straně Obr. 1, který obsahuje veškeré soubory, které kurz obsahuje. Aby se uživatel neztratil ani u rozsáhlých kurzů s velkým množstvím příloh, nabízí panel chytrou organizaci podle názvu kapitoly. Aplikace navíc uživateli výrazně usnadňuje orientaci tím, že jednotlivé soubory barevně odlišuje. Rámečky okolo souboru změni barvu podle jeho typu, například oranžově se zbarví soubory PPTX (neboli powerpointové prezentace), zeleně excelové soubory, modře soubory z Wordu, červeně PDF a další. Po kliknutí na kterýkoliv soubor ho uživatel může otevřít, pokud má k dispozici potřebný software.

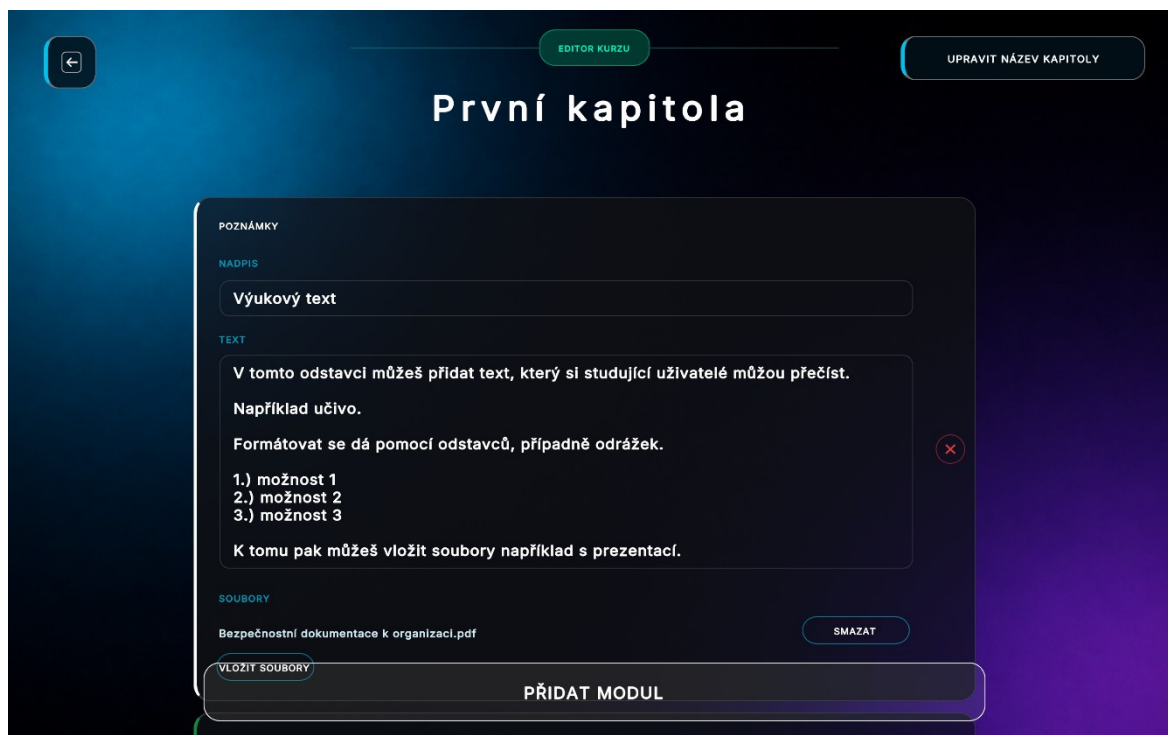
Kliknutím na konkrétní kapitolu a následně na tlačítko upravit se uživatel přesune do detailnějšího zobrazení (viz Obr. 2), kde probíhá samotné plnění kapitoly výukovými informacemi. Rozhraní dává uživateli na výběr z několika typů modulů, které lze libovolně kombinovat. Patří sem například běžné výkladové poznámky, kvízy, interaktivní seřazování (modul drag&drop) nebo výuka cizích slov, zkratk a definic pomocí karetního modulu. Tyto moduly lze přidávat po kliku na tlačítko „přidat modul“, které zobrazí výběrové menu a po výběru jednoho z modulů se zařadí na konec vytvářeného seznamu.

Tvůrce je pak může plnit daty. Poznámkový modul umožňuje zadat nadpis, text pro předání informací studujícímu, a to i za použití psaní odstavců i odrážek. Dále může uživatel přiložit i různé typy souborů, které se pak následně propíší i do obdélníku se soubory na předchozí stránce. Tam se automaticky přiřadí nadpis kapitoly, do které patří. Tyto soubory může uživatel tvořící kurz kdykoliv odstranit po kliknutí na tlačítko „smazat“ vedle daného souboru.

Karetní modul nabízí dvě zadávací pole s textem, jedno pro pojem, zkratku nebo termín a druhé pro vysvětlivku. Pomocí tohoto modulu se dají třeba vytvářet karty pro výuku cizích slov, zkratk, ale i celých vět a pojmů.

Dalším modulem, který lze přidat do kurzu, je kvíz. Zde tvůrce může zadat dva typy kvízových otázek. První z nich je klasická otázka s odpověďmi, druhým typem je pak možnost dotázat se na obrázek, který může přiložit jednoduše tlačítkem. Následně musí zvolit počet odpovědí a ty naplnit textem. Aplikace uživateli dovolí vybrat minimálně dvě možnosti, a to může provést tlačítka plus a mínus. Po stisku těchto tlačítek se změny okamžitě projeví, bez nutnosti dalšího potvrzování volby. V poslední řadě by uživatel měl zakliknout

čtverec vedle některé z odpovědí (může i více z nich), který indikuje, která odpověď je správná, pokud to neudělá, všechny pak pro studujícího budou špatné.



Obr. 2: Stránka pro vytváření obsahu kapitoly
(Zdroj: Vlastní)

Další modul ve výběru je modul „drag and drop“. Ten slouží k řazení jednotlivých odpovědí tak, jak jdou za sebou. Proto je tvůrce musí zadat vždy ve správném pořadí shora. Vytvořit se tak dají například úlohy typu „Seřaď čísla vzestupně“, „Seřaď státy od nejmenších dle počtu obyvatel“, „Seřaď vrstvy ISO/OSI od nejnižších“ a další. Opět je uživateli k dispozici úprava počtu možností, tentokrát však v rozsahu od dvou do osmi.

Všechny moduly mají několik společných funkcí, které platí i pro ty, které by mohly v budoucnu přibýt. První z nich je možnost kdykoliv a jakýkoli modul smazat pomocí červeného křížku, který je zobrazen ve středu modulu na jeho pravé straně vždy, když je modul aktivní (tedy rozkliknut uživatelem). S aktivními moduly se pojí funkce sloužící pro zvýšení přehlednosti ve chvíli, kdy tvůrce vytvoří více výukových modulů. Tato funkce vždy zmenší každý modul na nezbytnou velikost, ve které je vidět pouze otázka daného modulu a zbytek se uzavře. Pokud uživatel chce upravit jiný modul, stačí na něj pouze kliknout. Po tomto kliku se nový modul roztáhne dle obsahu a předchozí modul se zmenší. Aby zmenšené moduly bylo možné okamžitě rozeznat podle typu, ne pouze podle zdlouhavého čtení otázek, má každý typ svou barvu levé strany (například poznámky jsou bílé, kvíz zelený atd.).

Další funkcí této stránky je možnost přejmenování kapitoly. Kapitoly se automaticky pojmenovávají „Kapitola + pořadí v kurzu“. To je ale možné právě změnit. Po kliku na

tlačítko „upravit název kapitoly“ na stránce pro její úpravu, zobrazí se uživateli stejně fungující okno, jako pro změnu názvu kurzu. V případě, že uživatel opravdu změní název kapitoly, je název propsán i na stránce pro správu kapitol.

5 Studijní režim

Kliknutím na tlačítko „studovat“ v hlavním menu se uživatel dostane do části aplikace věnované studijním účelům. V následujících podkapitolách je popsáno, jak tato část vypadá, funguje a jsou v ní představeny výstupní formy tvořivého režimu.

5.1 Stránka pro studium kapitoly dle funkcí kódu

Skript `StudyChapter.cs` zajišťuje funkčnost studijní části aplikace. Je zodpovědný za transformaci statických dat z JSON souborů, které představují obsahová data z tvořivého režimu, do interaktivního studijního prostředí, kde student prochází moduly, získává okamžitou zpětnou vazbu a ukládají se mu špatné odpovědi pro případné zopakování.

Základní architektura, dědičnost od třídy `MonoBehaviour`, správa životního cyklu `OnEnable`, `OnDisable` a napojení na vizuální prvky UI Toolkitu pomocí metody `.Q<T>()` jsou realizovány obdobným způsobem jako v kapitole 4.1.

Zatímco editační skripty jako například `EditChapter.cs` generovaly rozhraní s textovými poli pro vstup, `StudyChapter.cs` vytváří uzly určené pro interakci a čtení. Metoda `Render()` vyčistí kontejner a vykresluje pole modulů v JSON souboru. Zásadním rozdílem oproti editačnímu režimu je zohlednění stavu postupu. Každý modul je vykreslován s ohledem na to, zda je již studentem zodpovězen či nikoliv. Tento stav se resetuje po vypnutí kurzu, aby mohl studující uživatel procházet kapitoly s učivem opakovaně. Tato logika je řízena porovnáním indexu modulu s hodnotou `lastUnlockedModuleIndex` uloženou v datech kurzu. Přechod na další modul však není podmíněn předchozí odpovědí. Pokud uživatel chce, může odpověď klidně přeskočit.

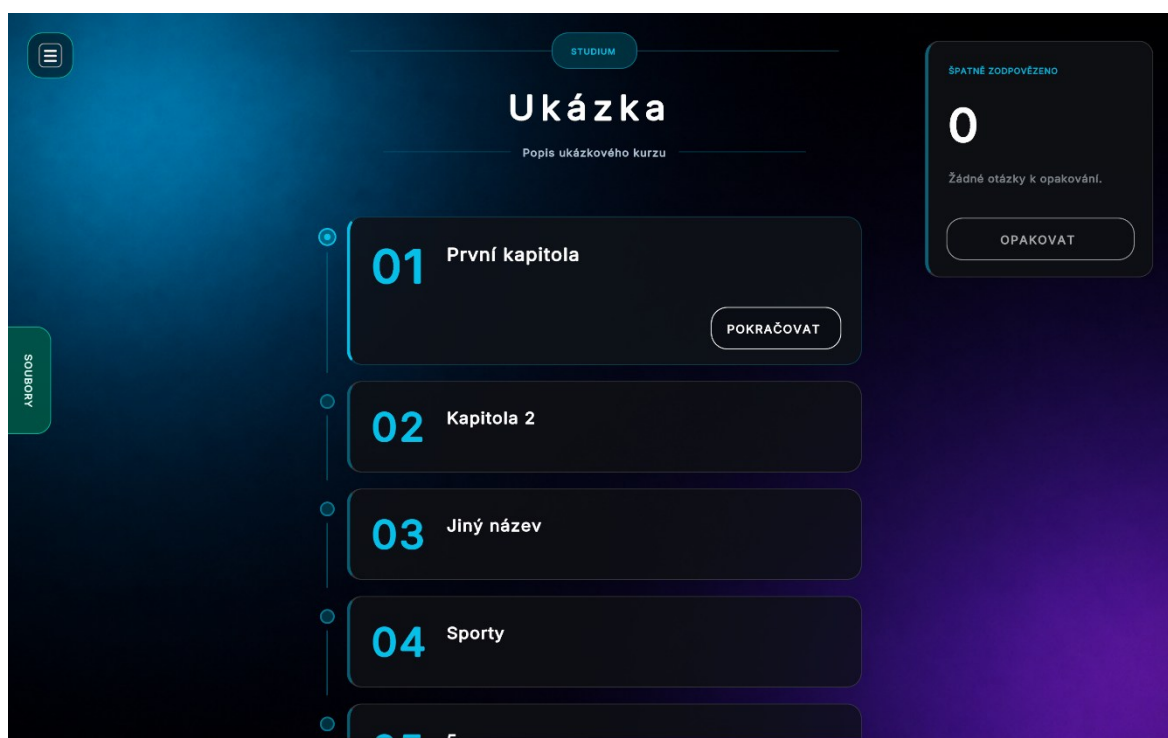
Jádrem skriptu jsou specializované metody pro tvorbu interaktivních bloků (např. `BuildQuizNode`, `BuildDragDropNode`). Postup tvorby těchto prvků je strukturálně podobný jako v kapitole 4.1, avšak namísto ukládání textu se zde na prvky vážou `callbacky` pro vyhodnocování odpovědí.

Například u kvízového modulu skript přiřadí každému tlačítku odpovědi `listener` volající metodu `HandleAnswer()`. Ta porovná volbu studenta s pravdivostní hodnotou v JSON souboru a okamžitě vizuálně změní styl prvku, zelenou barvou u správné odpovědi a červenou barvou u odpovědi špatné, čímž naplňuje funkční požadavek.

Skript dále implementuje logiku pro sledování špatně zodpovězených otázek. Pokud student odpoví chybně, skript vyvolá metodu, která zapíše identifikátor modulu do sekce `wrongAnswered` v JSON souboru. Tato data jsou následně využívána pro možnost opakování problematických částí učiva.

5.2.1 Hlavní studijní stránka

Po rozkliknutí kurzu uživatelem se zobrazí stránka, která je velmi podobná, jako hlavní stránka pro tvorbu obsahu (viz. Obr. 4). Má jen pár rozdílných prvků, jako jsou tlačítka v pravém horním rohu. Ta byla nahrazena kartou s počítadlem špatně zodpovězených otázek, pod níž je tlačítko, které z nich vygeneruje samostatnou opakovací kapitolu. Další vizuální rozdíl je v zobrazení jednotlivých kapitol. Ty jsou tady vykresleny v číslované verzi, s možností rozkliknutí dané kapitoly. Po kliku na konkrétní kapitolu se její karta zvětší a objeví se na ní tlačítko s možností studia jejího obsahu.



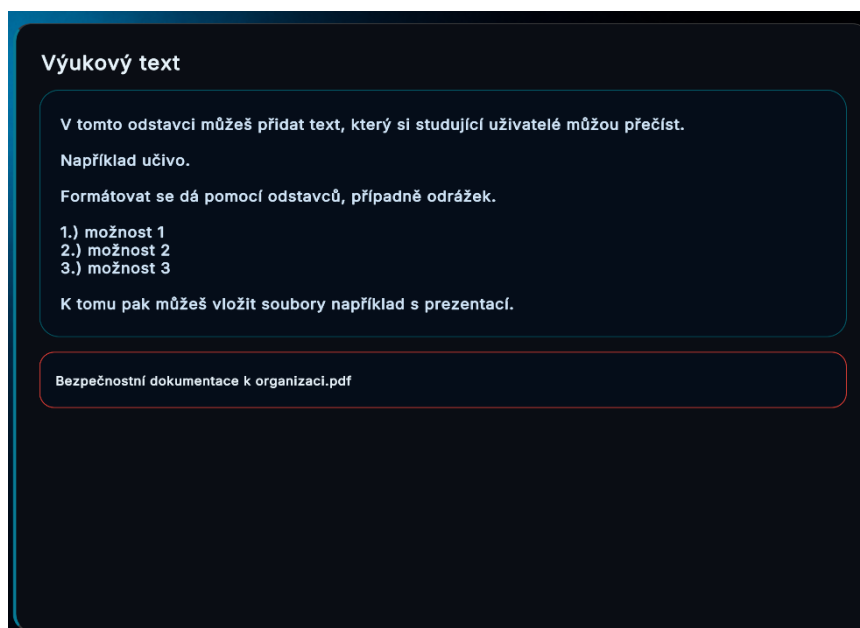
Obr. 4: Hlavní stránka studijního režimu

(Zdroj: Vlastní)

5.2.2 Modul poznámky

Pro studium kapitoly uživatel musí kliknout na tlačítko „pokračovat“ nacházející se na její kartě. Následně je zobrazena stránka vygenerována z obsahu v JSON souboru. Každý modul má svoje dané rozložení podle toho, jaký je jeho typ dle sekce type v JSON.

První je zobrazení modulu, který je pojmenován jako „poznámky“ (viz Obr. 5). Ten nabízí zobrazení nadpisu, dále textu naformátovaného pomocí odstavců, případně odrážek a příložených souborů pod textem. V případě, že je přiloženo mnoho souborů nebo modul obsahuje velké množství řádků textu a nevejde se na stránku, může uživatel posunout obsah otáčením kolečka myši tak, aby viděl vše, co je skryto níže.



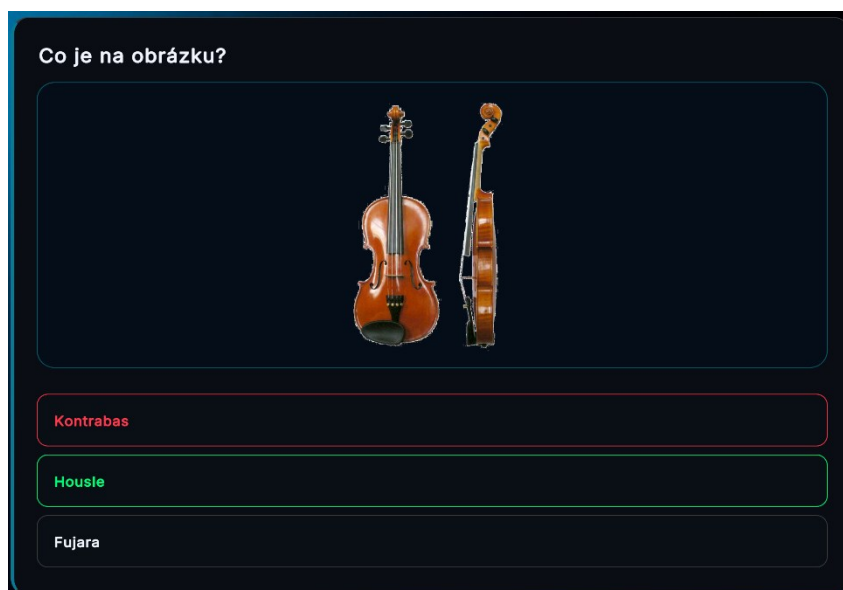
Obr. 5: Ukázka poznámkového modulu

(Zdroj: *Vlastní*)

5.2.3 Modul kvíz

Dalším modulem v pořadí je „kvíz“. Ten slouží ke zobrazení úkolu typu otázka a dvě až čtyři odpovědi. Bonusem je i možnost kombinovat tento modul s obrázkem a je tak možné vytvořit kvízovou otázku například „Co je na obrázku?“, „Které auto projede křižovatkou jako první?“ a mnoho dalších.

Studující uživatel pak klikne na odpověď, kterou považuje za správnou a aplikace mu okamžitě ukáže, zda odpověděl správně pomocí zvýraznění jeho odpovědi zeleně, či nikoliv pomocí zvýraznění jeho odpovědi červeně a správné barvou zelenou tak, jak lze vidět na Obr. 6.

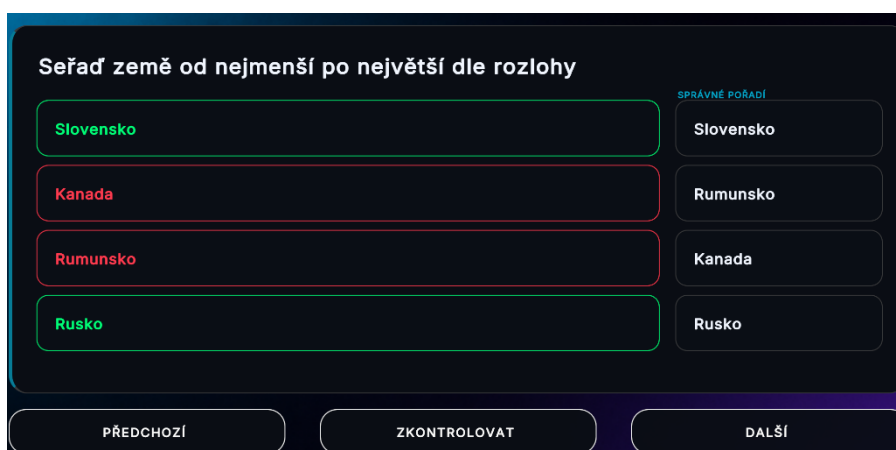


Obr. 6: Ukázka kvízového modulu

(Zdroj: Vlastní)

5.2.4 Modul drag and drop

Vhodným nástrojem pro studium je i modul „drag and drop“. Ten slouží k seřazování možností dle zadání a dají se díky němu vytvořit úlohy typu „Seřaď země od nejmenší po největší dle rozlohy“, „Seřaď vrstvy ISO/OSI“ a další. Uživatel pak pomocí podržení levého tlačítka myši a tažením může přesouvat pořadí jednotlivých možností a jakmile si bude myslet, že má vše správně seřazené, klikne na tlačítko „zkontrolovat“ pod modulem. Aplikace mu vyhodnotí správné pozice odpovědi zelenou barvou a špatné barvou červenou (viz Obr. 7). Dále se zobrazí na pravé straně sloupec, ve kterém je uvedeno správné řešení úlohy.



Obr. 7: Ukázka drag and drop modulu

(Zdroj: Vlastní)

5.2.5 Modul karta

Posledním dostupným modulem je karta. Ta umožní jednoduché studium cizích slov, definic nebo čehokoliv, co je třeba si zapamatovat nazpaměť. To je provedeno pomocí strany karty se zadáním, právě třeba s cizím slovem, větou nebo definicí (viz Obr. 8), a po kliknutí uživatele se karta plynule otočí na stranu se správnou odpovědí. Na straně s odpovědí jsou umístěna i dvě tlačítka, která slouží k určení toho, zda studující uživatel správnou odpověď věděl nebo ne (viz Obr. 8). Pokud klikne na tlačítko „nevím“, uloží se tento modul do špatných odpovědí pro budoucí zopakování.



Obr. 8: Ukázka karetního modulu
(Zdroj: *Vlastní*)

6 Ostatní implementační části aplikace

V aplikaci se nachází mnoho dalších částí mimo výukový a tvořivý režim. K jejich popisu slouží následující podkapitoly věnované barevné paletě softwaru, popisu integrace a splnění požadavků platformy Steam a společnosti Valve a další.

6.1 Jednotný vzhled aplikace

Aplikace je navržena v jednotném vizuálním stylu, který se projevuje zejména konzistentním zpracováním tlačítek, barev, ikon a dalších grafických prvků. Pro uživatelské rozhraní byla zvolena barevná paleta založená na třech hlavních barvách: světle modré, modrozelené a fialové. Světle modrá barva je využívána především ve studijní části aplikace, modrozelená barva reprezentuje část určenou pro tvorbu obsahu a fialová slouží jako doplňková barva, která je současně spojena s nastavením aplikace.

Kromě hlavních barev jsou v rozhraní použity také barvy se specifickým významem. Červená barva označuje negativní nebo rizikové akce, například chybné odpovědi, mazání kurzů, návratová tlačítka nebo ukončení aplikace. Zelená barva je naopak vyhrazena pouze pro správné odpovědi. Takto zvolená barevná paleta podporuje intuitivní orientaci uživatele v aplikaci a napomáhá jednoznačně rozlišit aktuální část aplikace i význam provedené akce.

6.2 Integrace se službami Steamworks

Pro zveřejnění softwaru na platformě Steam je potřeba udělat několik kroků. Ty se dělí na tři části.

První z nich je příprava stránky obchodu a obsahu v něm k zaujmutí potenciálního uživatele a informování jej o obsahu a charakteristice aplikace samotné. Společnost Valve pro tuto část dodává seznam úkolů, který je třeba splnit před schvalováním stránky. Po jeho splnění a následném zažádání o schválení má Valve 3-5 pracovních dnů na validaci.

Seznam povinných úkolů pro prezentaci v obchodě [18]¹¹:

1. Základní informace
2. Popisy
3. Dotazník ohledně obsahu
4. Plánované datum vydání
5. Systémové požadavky
6. Popis podpory ovladačů
7. 5 nebo více snímků obrazovky
8. Obrázky pro výběry
9. Prvky pro knihovnu

¹¹ Přesný odkaz = <https://partner.steamgames.com/doc/store/releasing>

10. Informace o podpoře
11. Vývojáři a vydavatelé
12. Popisné značky v obchodě
13. Ikona aplikace
14. Ikona zástupce
15. Kvóta pro soubory
16. Limit pro soubory

Druhým krokem je naprogramování důležitých částí kódu přímo v softwaru. V tomto projektu jsou to funkce pro nahrání kurzu na workshop a automatické stahování či případné aktualizace odebíraných kurzů studujícím uživatelem.

Třetím a posledním krokem před zveřejněním aplikace na platformě Steam je splnění druhého seznamu společnosti Valve, na jehož schvalování má také 3-5 pracovních dnů. Po jeho schválení je aplikace dostupná ke stažení veřejně, pokud vývojář nenastaví jinak.

Seznam povinných úkolů s názvem „Build Vaší aplikace“ [18]¹²:

1. Shoda podporovaných platform
2. Upoutávka
3. Konfigurace aplikace
4. Alespoň jedno nakonfigurované úložiště
5. Alespoň jeden nakonfigurovaný build
6. Definované možnosti spuštění
7. Nastavený instalační adresář
8. Nakonfigurované jazyky úložišť
9. Shoda balíku v obchodu a balíku devcomp
10. Balík s úložištěm platformy Windows
11. Všechna úložiště zahrnuta
12. Název workshopu
13. Reprezentující obrázek

6.2.1 Příprava stránky v obchodě

Postup přípravy stránky v obchodě lze rozdělit do několika navazujících kroků. V první fázi je nutné vyplnit základní informace o produktu. Tato část zahrnuje především identifikaci aplikace v obchodě, tedy její název, základní zařazení a další údaje, které slouží k jednoznačné prezentaci produktu uživatelům. Současně je třeba doplnit informace o vývojáři a vydavateli, které jsou následně zobrazeny na veřejné stránce aplikace. Pro tento projekt byla v obou případech zvolena herní přezdívka autora.

Dalším krokem je vytvoření textového obsahu stránky. Steamworks vyžaduje vyplnění stručného i rozšířeného popisu produktu povinně v anglickém jazyce a v případě této práce i v jazyce českém. Tyto texty mají informovat potenciální uživatele o účelu aplikace, jejich

¹² Přesný odkaz = <https://partner.steamgames.com/doc/store/releasing>

hlavních funkcích a způsobu použití. Součástí této fáze je také přiřazení popisných značek v obchodě, které napomáhají kategorizaci produktu a jeho dohledatelnosti v rámci vyhledávání.

Dále je třeba zpracovat obsahovou a technickou specifikaci produktu. Steamworks vyžaduje vyplnění dotazníku ohledně obsahu, který slouží k určení charakteru zobrazovaného materiálu a k posouzení vhodnosti obsahu pro uživatele. Po vyplnění dotazníku je automaticky určen doporučený minimální věk uživatele. Této aplikaci vyšla věková kategorie od tří let, protože aplikace samotná neobsahuje žádnou formu explicitního obsahu. Současně je nutné uvést systémové požadavky aplikace, aby si uživatel mohl předem ověřit, zda jeho zařízení splňuje minimální podmínky pro spuštění. Pokud aplikace podporuje ovladače, je třeba vyplnit také popis podpory ovladačů. Tato část má význam zejména pro správné informování uživatelů o způsobu ovládání a kompatibilitě programu.

Samostatnou část přípravy tvoří vizuální podklady. Pro prezentaci produktu v obchodě je nezbytné nahrát nejméně pět snímků obrazovky zachycujících skutečné rozhraní a funkce aplikace. Kromě toho Steamworks vyžaduje také speciální grafické výstupy určené pro různé části platformy v přesném rozlišení, zejména obrázky pro výběry a prvky pro knihovnu. Tyto grafické materiály slouží k tomu, aby byl produkt správně zobrazován v katalogu obchodu, ve vyhledávání i v knihovně uživatele. V rámci komunitní části je dále nutné doplnit ikonu aplikace a ikonu zástupce, které reprezentují software v klientovi Steam a v souvisejících uživatelských prvcích.

Dalším krokem je doplnění informací o podpoře. Vývojář musí uvést kontaktní nebo informační údaje, jejichž prostřednictvím mohou uživatelé řešit technické problémy nebo získat doplňující pomoc. Tato část může zahrnovat například e-mailový kontakt, webovou stránku projektu nebo odkaz na dokumentaci. Autor projektu zvolil e-mailovou adresu info.learne@gmail.com.

V případě využití služby Steam Cloud je třeba současně nakonfigurovat parametry ukládání dat do cloudu. Seznam úkolů v této souvislosti vyžaduje nastavení kvóty pro soubory a limitu pro soubory. Tím se určuje, jaké množství dat může aplikace v cloudovém úložišti využívat a jaká omezení budou pro ukládané soubory platit.

Po doplnění všech obsahových, technických a vizuálních náležitostí následuje nastavení plánovaného data vydání. Tím vývojář určí, kdy má být produkt v obchodě dostupný veřejnosti. Tento údaj je důležitý nejen z organizačního hlediska, ale také pro samotný proces schválení a zveřejnění stránky. V případě výukové aplikace LEARN-E byl zvolen formát 2. kvartál 2026, přičemž interní datum vydání je nastaveno na 25. května 2026. Datum bylo zvoleno s ohledem na termín odevzdání této práce.

Závěrečnou fází je kontrola úplnosti celé produktové prezentace a její předložení ke schválení v rámci systému Steamworks. Teprve po splnění všech položek kontrolního seznamu úkolů může být stránka považována za připravenou k publikaci. Tento postup zajišťuje, že produkt bude před uvedením do distribuce prezentován jednotně, informačně úplně a v souladu s požadavky platformy Steam.

Informace o produktu

Přehled: Tento prostor využijte pro popsaní svého produktu. Zpravidla je dobré začít stručným přehledem – pokud si zákazníci nepřečtou nic jiného, měl by jim takovýto přehled dát celkem dobrou představu, o čem produkt je. Poté můžete zajít do větších detailů, přidat seznam jeho funkcí a vlastností a další věci, které Vám přijdou důležité zmínit.

Poznámka: V této sekci byste neměli zahrnovat odkazy na jiné webové stránky, QR kódy ani jiné obrázky propagující Vaše účty na sociálních sítích. Takové odkazy můžete uvést v záložce „Základní informace“ nebo je zveřejnit v komunitě.

Lokalizace: Pro přidání lokalizované verze textu vyberte požadovaný jazyk z rozbalovacího seznamu napravo a do pole vložte text lokalizovaný do daného jazyka. Případně můžete lokalizovaný text nahrát hromadně v záložce „Lokalizace“ výše.

VLASTNÍ OBRÁZKY

Přehled: Prostor níže využijte pro nahrání a správu vlastních obrázků, které budou použitelné v sekcích „Informace o produktu“, „Ocenění“ a v sekcích pro speciální oznámení. Obrázky zde nahrané lze do textu vložit pomocí tlačítka z lišty možností formátování v editoru.

Design: Dbejte na velikost obrázků, které nahráváte. Čím více obrázků nahrajete, tím déle se bude stránka načítat zákazníkům a někteří mohou vlivem toho dokonce svoji návštěvu úplně zrušit.

Formát: Statické prvky: Formáty PNG, JPG, GIF a WEBP.
Animované prvky: V současnosti jsou podporovány formáty MP4 a WebM, nicméně animace se přehrají pouze v sekci „Informace o produktu“ a v sekcích pro speciální oznámení na stránce v obchodu. Pro animované prvky s průhledností použijte formát WebM (s kódováním vp9), APNG nebo GIF.

Rozměry: Na desktopových rozlišeních má oblast pro popis šířku 780 pixelů, nicméně doporučujeme nahrát prvky s většími rozměry (případně budou zmenšeny na maximální rozměry 4096x8192 pixelů). Délka animovaných prvků může být maximálně 12 sekund.

Lokalizace: Když do názvu prvku zahrnete kód jazyka (např. „mujobrazek_czech.jpg“), bude automaticky vytvořena skupina obrázků, kterou vložíte do textového popisu a služba Steam poté vybere pro každého uživatele ten správný obrázek (resp. jeho lokalizovanou verzi).

[Pro více informací navštivte dokumentaci pro správu doplňkových grafických prvků.](#)

Použít vizuální editor

⌂
↶
↷
B
I
U
S
↶
H2
☰
🖼️
📄
☰
☰
☰
Čeština

Režim náhledu: Desktopový prohlížeč / klient Mobilní prohlížeč / aplikace Steam Deck / Big Picture

INFORMACE

Aplikace Vám umožní vytvářet kurzy na jakékoliv téma. Funguje jako intuitivní nástroj, který umožní Vytvořit kurz pomocí různých typů modulů.

Poznámky: Tento modul Vám umožní vložit nadpis, výukový text s informacemi, které chcete předat a soubory k potvrzení daného tématu. Příložené soubory se zobrazují i mimo jednotlivé kapitoly a pro lepší přehlednost jsou nadepsány, do které kapitoly patří

Drag&Drop: Tento modul umožňuje vytvořit úlohy typu "Seřad čísla vzestupně", "Seřad státy od nejmenších po největší" a další. Toho je dosaženo pomocí tahacích dlaždic s kontrolním systémem

Karty: Modul, který funguje jako klasická papírová karta. Umožňuje výuku cizích slov, pojmů či definic

Kvíz: Kvízový modul Vám dovolí vytvořit si různé typy otázek a to i za pomoci obrázků. Vytvořit se pak dají úlohy typu "Otázka + 2-5 možností" nebo "Otázka + obrázek + 2-5 možností"

Kurz můžete i přejmenovávat, měnit jeho popisek, měnit názvy kapitol a měnit náhledový obrázek kurzu.

Výsledný kurz je možné nahrát na Steam workshop pro sdílení mezi studenty a další studující. nahraný kurz můžete i nadále upravovat, přidávat mu další obsah a následně ho aktualizovat opětovným nahráním na workshop bez nutnosti studujících, kteří již odeberají Váš kurz, stahovat další. Automaticky se jim změna kurzu stáhne při spuštění aplikace do jejich zařízení.

Celá aplikace funguje lokálně na zařízení bez nutnosti připojení k internetu. Výjimkou je nahrávání nebo stahování kurzů z workshopu. Tam je připojení k síti potřeba.

Aplikace vznikla jako bakalářská práce na Fakultě Aplikované Informatiky University Tomáše Bati

Vlastní obrázky

Sem přetáhněte prvky, které mají být nahrány pro použití na stránce v obchodu.

Nebo je vyhledejte: Vybrat soubor

Informace ohledně podporovaných formátů a jazyků naleznete výše.

Obr. 9: Ukázka úpravy popisu aplikace s požadavky ve službě Steamworks
(Zdroj: Vlastní)

6.2.2 Příprava buildu aplikace

Po dokončení přípravy produktové stránky v obchodě Steam následuje další fáze procesu, kterou je příprava samotného buildu neboli konečného sestavení aplikace. Ve Steamworks tato část zahrnuje technické nastavení, nahrání finálních souborů aplikace a jejich následné odeslání ke kontrole společnosti Valve. Správná konfigurace buildu je nezbytná pro to, aby bylo možné aplikaci distribuovat koncovým uživatelům prostřednictvím platformy Steam.

Prvním krokem je nahrání funkčního sestavení aplikace. Build by měl být v této fázi použitelný v celém zamýšleném rozsahu a měl by obsahovat všechny podstatné funkce, které jsou uvedeny na stránce produktu v obchodě.

Druhým krokem je splnění technického seznamu úkolů potřebných pro distribuci aplikace. Tato část vyžaduje podrobné nastavení aplikace v administračním rozhraní Steamworks. Součástí konfigurace je zejména příprava alespoň jednoho úložiště, označovaného jako depot, a vytvoření odpovídajícího buildu, který toto úložiště obsahuje. U jednotlivých depotů je zároveň nutné nastavit podporované jazyky a zajistit, aby byly do buildu zahrnuty všechny potřebné soubory aplikace.

Další část konfigurace se týká nastavení instalace a spouštění aplikace. Vývojář musí definovat instalační adresář a určit způsob spuštění aplikace pro cílovou platformu. V případě platformy Windows se typicky jedná o určení příslušného spustitelného souboru, který bude po instalaci aplikace spuštěn. Toto nastavení je důležité zejména proto, aby Steam dokázal aplikaci po stažení správně nainstalovat a následně spustit z klientského rozhraní.

Součástí technické přípravy je také přiřazení balíků a kontrola jejich souladu s podporovanými platformami. Je nutné ověřit, zda balík uvedený v obchodě odpovídá balíku obsahujícímu příslušné úložiště aplikace. Zvláštní pozornost je vhodné věnovat primární cílové platformě, kterou je v tomto případě systém Windows. Nesprávné přiřazení balíků nebo chybějící vazba mezi obchodem a depotem by mohly způsobit, že uživatel po zakoupení nebo aktivaci aplikace nezíská přístup ke správným instalačním souborům.

Vzhledem k tomu, že aplikace využívá funkce spojené se sdílením obsahu prostřednictvím Steam Workshopu, je nutné nakonfigurovat také tuto část služby. Konfigurace zahrnuje vyplnění názvu workshopu a nahrání reprezentativního obrázku, který bude použit při jeho prezentaci v rámci platformy Steam. V návaznosti na předchozí implementaci funkcí pro nahrávání a stahování kurzů se jedná o nezbytný krok pro správnou integraci uživatelsky vytvářeného obsahu. Součástí přípravy buildu je rovněž doplnění mediálních prvků, přičemž pro splnění požadavků je vyžadována alespoň jedna upoutávka prezentující aplikaci.

Posledním krokem této fáze je odeslání kompletního buildu ke schválení. Po odeslání probíhá kontrola ze strany společnosti Valve, jejímž cílem je ověřit, zda aplikace splňuje technické, obsahové a distribuční požadavky platformy Steam. Pokud jsou během kontroly zjištěny nedostatky, je vývojář povinen je odstranit a následně build znovu předložit k posouzení. Teprve po úspěšném dokončení kontroly lze build považovat za připravený k vydání prostřednictvím platformy Steam.

7 Testování a optimalizace

V rámci této kapitoly bylo provedeno ověření funkčnosti hlavních částí výukové aplikace, uživatelského rozhraní a správného chování vybraných technických prvků. Dále se v kapitole nachází krátký výčet optimalizačních řešení a řešení připomínky společnosti Valve při prvním schvalování aplikace před vydáním.

7.1 Grafický výkon v rozhraní

Při návrhu uživatelského rozhraní v Unity byl původně použit efekt rozmazaného pozadí vytvořený kombinací stylování v USS a post-processingu¹³. Toto řešení sice vizuálně odpovídalo požadovanému modernímu vzhledu aplikace, ale zároveň zvyšovalo nároky na výkon, protože efekt rozmazání byl zpracováván v reálném čase.

Z důvodu optimalizace výkonu bylo dynamické rozmazání nahrazeno statickým obrázkem pozadí, který byl předem graficky upraven tak, aby vizuálně připomínal rozmazaný efekt. Výsledkem bylo zachování požadovaného vzhledu rozhraní při výrazně nižší zátěži grafického systému.

Na autorově počítači, níže představeného jako „Počítač“ v tabulce i s jeho konfigurací, před touto změnou v optimalizaci běžela aplikace na 100 % výkonu grafické karty při uzamknutých 60 snímcích za vteřinu. Po změně se výkon karty drží okolo 25 % jejího využití při stejné snímkové frekvenci. Tyto hodnoty byly naměřeny pomocí překrytí aplikace NVIDIA APP.

7.2 Správa a optimalizace

V aplikaci je ovládání tlačítek řešeno pomocí registrace událostí na prvky uživatelského rozhraní. Tlačítka a interaktivní části rozhraní jsou po vytvoření nebo zobrazení stránky napojeny na příslušné akce, například otevření editace kapitoly, smazání kapitoly, obnovení seznamu souborů nebo otevření dialogového okna. K tomu jsou využívány mechanismy UI Toolkit, zejména `RegisterCallback`, `Clickable` a pomocná metoda `BindClickOnce`, která zajišťuje, aby se stejná akce na jeden prvek nepřipojila opakovaně.

Důležitou částí řešení je také odhlašování událostí. Při deaktivaci skriptu se například odpojuje událost `PageShown`, ruší se `callback` pro kliknutí mimo vybraný uzel, odhlašují se události pro změnu geometrie rozhraní a zastavují se běžící `coroutiny`. Skript zároveň kontroluje, zda již není daný kořenový prvek stránky zaregistrován, a při změně stránky nejprve odpojí původní `callbacky`, než připojí nové.

¹³ Post-processing = soubor grafických efektů a úprav, které se aplikují na finální obraz scény po vykreslení 3D geometrie

Tento postup je důležitý z hlediska výkonu i stability aplikace. Pokud by se události neodhlašovaly nebo by se při opakovaném zobrazení stránky registrovaly znovu, mohlo by dojít k vícenásobnému spuštění stejné akce po jednom kliknutí, zbytečnému zpracování událostí na neaktivních prvcích nebo držení starých referencí v paměti. To by mohlo vést ke zpomalení rozhraní, vyšší spotřebě paměti a obtížně dohledatelným chybám v ovládání aplikace.

Správné přihlašování a odhlašování tlačítek proto zajišťuje, že aktivní jsou pouze ty události, které odpovídají aktuálně zobrazené části uživatelského rozhraní. Díky tomu zůstává ovládání předvídatelné, nedochází k duplicitnímu volání metod a aplikace si zachovává lepší výkon i při opakovaném přepínání mezi stránkami.

7.3 Synchronizace rozpracovaných kurzů pomocí Steam Cloud

Verze aplikace, která byla předložena k prvnímu schválení společnosti Valve, umožňovala ukládání rozpracovaných kurzů pouze lokálně. Kurzy, které uživatel vytvořil, ale nenahrál je na Steam Workshop, tak zůstávaly dostupné jen na zařízení, na kterém vznikly. Steam Workshop je v aplikaci určen primárně pro publikaci a sdílení dokončených kurzů mezi uživateli, nikoliv pro soukromou synchronizaci pracovních dat autora.

Tento nedostatek se projevil při první kontrole aplikace společností Valve, kdy aplikace nebyla schválena s požadavkem na doplnění funkční synchronizace uživatelských dat. Na základě této zpětné vazby byla do projektu doplněna podpora Steam Auto-Cloud¹⁴. V nastavení Steamworks byl následně nakonfigurován adresář, do kterého aplikace ukládá rozpracované kurzy, aby mohl být automaticky synchronizován mezi zařízeními uživatele přihlášeného ke stejnému Steam účtu.

Rozpracované kurzy jsou v aplikaci ukládány jako lokální soubory, například ve formátu JSON, společně s případnými přidruženými daty. Steam Auto-Cloud následně zajišťuje jejich automatické nahrání do cloudového úložiště po vypnutí aplikace a obnovení na jiném počítači před jejím spuštěním. Výhodou tohoto řešení je, že aplikace nemusí implementovat vlastní systém pro ruční ukládání a stahování pracovních souborů, protože synchronizaci zajišťuje přímo platforma Steam.

Doplněním této funkce bylo odděleno veřejné sdílení kurzů od soukromé synchronizace rozpracovaných dat. Steam Workshop tak slouží k publikaci hotových kurzů pro ostatní uživatele, zatímco Steam Cloud umožňuje autorovi pokračovat v práci na vlastních rozpracovaných kurzech i na jiném zařízení. Tím byl splněn požadavek společnosti Valve a zároveň se zvýšily uživatelský komfort i praktičnost celé aplikace.

¹⁴ Steam auto-cloud = funkce služby Steam Cloud, která umožňuje vývojářům automaticky zálohovat uložené pozice a nastavení her na servery Steamu bez nutnosti upravovat kód hry nebo aplikace

7.4 Zabezpečení aplikace

Zabezpečení aplikace je řešeno především využitím služeb platformy Steam. V aplikaci není vytvořen vlastní registrační ani přihlašovací systém, a proto neukládá uživatelská jména, hesla ani jiné citlivé přihlašovací údaje. Identita uživatele je získávána prostřednictvím platformy Steam, s níž po spuštění aplikace komunikuje, která poskytne aplikaci jednoznačný identifikátor přihlášeného uživatele, takzvané SteamID. Tento přístup navazuje na již popsanou integraci služeb Steamworks v práci.

SteamID slouží jako hlavní identifikační údaj uživatele v rámci aplikace. Výhodou tohoto řešení je, že autor aplikace nemusí řešit vlastní správu účtů, šifrování hesel, obnovu přístupů ani databázi uživatelských údajů. To vše je ponecháno na platformě Steam. Aplikace tak pracuje pouze s ověřenou identitou získanou ze Steam klienta a nezatěžuje uživatele další registrací.

Z hlediska komunitního obsahu je důležitá také bezpečnost kurzů publikovaných přes Steam Workshop. Kurzy mohou obsahovat nejen hlavní JSON soubor s výukovou strukturou, ale také příložené soubory, například obrázky, dokumenty nebo jiné studijní materiály. Autor si je vědom rizika, že studující uživatel může otevřít soubor přiložený v publikovaném kurzu, který by mohl obsahovat škodlivý obsah nebo malware. Tento druh rizika obecně vzniká u každé aplikace s možností sdílení uživatelského obsahu.

Z tohoto důvodu je aplikace naprogramována tak, aby příložené soubory nikdy automaticky nespouštěla. Vhodným opatřením je zobrazení upozornění uživateli před otevřením přílohy, omezení povolených typů souborů nebo zakázání rizikových formátů, například spustitelné soubory či skripty. Takové zabezpečení ale zatím aplikace nenabízí. Koncový uživatel tak musí přemýšlet, které soubory v kurzu neotevírat podle jejich typu, podobně, jako když mu přijde email. V tomto směru je zde prostor pro vylepšení aplikace, a to už at' z pohledu zabezpečení samotného, tak i pro pohodlí uživatelů.

7.5 Testování aplikace

Testování aplikace představuje důležitou část vývoje, jejímž cílem je ověřit funkčnost, stabilitu a použitelnost výsledného řešení. V případě e-learningové aplikace vytvořené v herním enginu Unity se testování běžně zaměřuje nejen na správnou funkci jednotlivých prvků aplikace, ale také na celkový uživatelský zážitek. U tohoto typu aplikace je vhodné ověřovat zejména správné načítání a ukládání dat, funkčnost uživatelského rozhraní, reakce aplikace na vstupy uživatele, průchod jednotlivými částmi výukového obsahu a stabilitu aplikace při běhu na různých zařízeních.

Standardní proces testování podobné aplikace by obvykle zahrnoval několik úrovní. Nejprve by probíhalo průběžné vývojářské testování, při kterém autor ověřuje funkčnost jednotlivých částí aplikace bezprostředně po jejich implementaci. To autor prováděl průběžně tak, jak bylo potřeba.

Další významnou součástí testování e-learningové aplikace by mělo být uživatelské testování. To slouží k ověření, zda je aplikace pro cílové uživatele srozumitelná, intuitivní a

zda její ovládání odpovídá očekávání uživatele. Jeho výsledky by následně mohly sloužit jako podklad pro úpravy uživatelského rozhraní, změny v ovládání nebo doplnění funkcí, které by zvýšily použitelnost výsledného programu.

Původně bylo uživatelské testování plánováno, nicméně se však do data odevzdání práce nepodařilo uskutečnit. Hlavním důvodem bylo pozdní dokončení aplikace, které bylo způsobeno problémy v průběhu vývoje. Na časový harmonogram měl vliv také schvalovací proces platformy Steam. Ten standardně trvá přibližně 3 až 5 pracovních dnů, avšak v případě neschválení aplikace, tak jak se stalo i při vývoji této práce, musí produkt absolvovat další schvalovací proces, čímž se celková doba validace může prodloužit až na 12 pracovních dnů.

S přihlédnutím k těmto komplikacím nebylo tedy možné uživatelské testování v plánovaném rozsahu realizovat před odevzdáním práce. Autor proto provedl alespoň vlastní testování na dvou jemu dostupných zařízeních. Toto testování bylo zaměřeno zejména na ověření základní funkčnosti aplikace, správné spuštění, stabilitu, použitelnost ovládacích prvků a kontrolu chování aplikace v běžných situacích. Průběh a výsledky tohoto svépomocného testování jsou popsány v následující kapitole.

7.6 Testování funkčnosti aplikace autorem

Testování aplikace autorem bylo provedeno na dvou zařízeních, a to výkonově silném osobním počítači a průměrném kancelářském notebooku. Jejich konfigurace jsou v tabulce níže:

Tab. 3: Specifikace testovacích zařízení

Komponenty	Počítač	Notebook
Operační systém	Windows 11	Windows 11
Procesor	Intel Core i5-12600K	Intel Core i7-1260p
Operační paměť	32GB 5200MHz	16GB 4800MHz
Grafická karta	NVIDIA RTX 4070 12GB	Intel Iris Xe Graphics
Pevný disk	2TB SSD	512GB SSD

7.6.1 Funkční testování aplikace

V rámci funkčního testování byly ověřeny hlavní části aplikace, zejména spuštění aplikace, práce s hlavním menu, vytvoření nového kurzu, načtení existujícího kurzu, přidávání kapitol a úprava výukového obsahu. Dále bylo testováno ukládání a opětovné načítání dat ve formátu JSON.

Všechny testované funkce pracovaly správně a během testování nebyly zjištěny chyby, které by bránily běžnému používání aplikace.

7.6.2 Uživatelské rozhraní

Testování uživatelského rozhraní bylo zaměřeno na přehlednost jednotlivých obrazovek, funkčnost tlačítek, reakce na uživatelské vstupy a orientaci v aplikaci. Bylo ověřeno, že

jednotlivé prvky rozhraní správně reagují na kliknutí a že přechody mezi obrazovkami probíhají podle očekávání.

Uživatelské rozhraní bylo při testování vyhodnoceno jako funkční a srozumitelné. Nebyly zjištěny žádné zásadní problémy v ovládání aplikace.

7.6.3 Ukládání a načítání dat

Při testování ukládání dat bylo ověřeno, že aplikace správně vytváří soubory kurzů, ukládá do nich zadané informace a dokáže je při dalším spuštění opět načíst. Testována byla také práce s kapitolami a výukovými moduly v rámci vytvořeného kurzu.

Ukládání i načítání dat probíhalo správně a vytvořené kurzy zůstaly po restartu aplikace zachovány. Stejně výsledky byly i při testování správného zobrazování ve studijním režimu.

7.6.4 Steam API

V rámci testování Steam API bylo ověřeno automatické zálohování kurzů, které dosud nebyly vydány na Steam Workshop. Cílem tohoto testu bylo ověřit, zda může uživatel pokračovat v práci na rozpracovaných kurzech také na jiném zařízení. Test potvrdil, že při zavření aplikace se správně vytvářené kurzy, jejich změny se zálohují a před spuštěním aplikace na jiném zařízení se správně zkontrolují změny na cloudu a případně se stáhnou, aby uživatel mohl pokračovat na rozpracovaných kurzech.

Dále bylo testováno nahrání vytvořeného kurzu na Steam Workshop a automatické stahování kurzů, které uživatel na Workshopu odebírá. Součástí testování bylo také ověření aktualizace již staženého kurzu v případě, že došlo ke změně jeho obsahu na Workshopu. Stahování odebíraných kurzů funguje správně a vydání kurzu na workshop také. Pouze pro aktualizaci změněného obsahu již staženého kurzu vyžaduje restartování platformy Steam pokud tvůrce kurzu provede aktualizaci ve chvíli, kdy má studující uživatel aplikaci spuštěnou.

Všechny testované funkce napojené na Steam API jinak pracovaly správně. Automatické zálohování, nahrávání kurzů na Workshop i stahování proběhly bez chyb.

8 Možnosti budoucího rozšíření aplikace

Aplikace vytvořená v rámci této bakalářské práce představuje základ výukového systému, který by měl být intuitivní, jednoduchý na používání a v budoucnu je možné jej dále rozšiřovat. Díky tomu, že je aplikace vytvořena v herním enginu Unity a zároveň propojena se službou Steam, nabízí hned několik zajímavých možností dalšího vývoje.

Jednou z hlavních možností rozšíření je doplnění dalších typů výukových modulů. Kromě základních kartiček nebo jednoduchých otázek by bylo možné přidat například časově omezené testy, doplňování chybějících výrazů nebo přiřazování pojmů. Díky Unity by mohly být tyto úlohy doplněny o animace, zvukové efekty, krátká videa nebo interaktivní vizuální prvky. Aplikace by tak mohla působit atraktivněji a více využívat možnosti herního enginu.

Významný potenciál má také využití Steam API. Jednou z nejvhodnějších funkcí jsou *achievements*¹⁵, které by mohly být udělovány například za počet dokončených kurzů, dosažení vysoké úspěšnosti, pravidelné učení nebo vytvoření vlastního výukového materiálu. *Achievements* by v tomto případě nesloužily pouze jako herní odměna, ale také jako motivační prvek a ukazatel dlouhodobého pokroku. Kromě toho by bylo možné využít také Steam statistiky nebo žebříčky. Ty by mohly zobrazovat například žebříček vybraných přátel uživatele v úspěšnosti daného kurzu.

Velmi praktickým rozšířením by bylo také využití Steam Cloudu o funkce ukládání postupu uživatele ve studijním režimu, výsledky testů nebo nastavení aplikace. Uživatel by tak mohl pokračovat ve studiu i na jiném počítači bez nutnosti ručního přenášení souborů.

V neposlední řadě je možnost přidání vícejazyčné lokalizace a postupná úprava uživatelského rozhraní na základě zpětné vazby od uživatelů.

Celkově lze říct, že propojení Unity a Steamu dává aplikaci dobrý základ pro další rozvoj. Největší potenciál mají zejména nové interaktivní typy úloh, sledování pokroku, *achievements* a rozšíření cloudového ukládání. Díky těmto rozšířením by se aplikace mohla postupně změnit z jednoduchého výukového nástroje na komplexnější platformu pro tvorbu, procvičování a sdílení vzdělávacího obsahu.

¹⁵ Achievement = Ohodnocení uživatele, v případě platformy Steam prostřednictvím speciální integrované funkce

Závěr

Cílem této bakalářské práce bylo navrhnout a vytvořit výukovou aplikaci v herním engine Unity 6, která umožní tvorbu, úpravu a následné studium uživatelsky vytvořených kurzů. Práce vycházela z potřeby vytvořit nástroj, který nebude omezen pouze na předem připravený obsah, ale umožní uživatelům vytvářet vlastní výukové materiály bez nutnosti zásahu do zdrojového kódu aplikace a bude jednoduchý na používání. Tento cíl byl v rámci práce naplněn vytvořením funkčního prototypu aplikace, který propojuje tvořivý režim, studijní režim, lokální ukládání dat, jejich zálohování a distribuci prostřednictvím platformy Steam.

V teoretické části práce byl nejprve popsán širší kontext e-learningu, výukových aplikací, serious games a gamifikace. Bylo ukázáno, že digitální vzdělávací nástroje mají význam zejména tam, kde je potřeba zvýšit aktivitu studenta, umožnit individuální tempo učení a nabídnout okamžitou zpětnou vazbu. Dále byla provedena analýza vybraných existujících řešení, například aplikací Duolingo, Autoškola Plus, Kahoot! a systému Moodle. Tato analýza ukázala, že dostupné platformy často nabízejí buď silnou gamifikaci, nebo rozsáhlé možnosti správy výuky, ale zároveň mohou být omezeny konkrétním obsahem, složitostí ovládání nebo závislostí na online infrastruktuře. Na základě technologického srovnání byl pro praktickou realizaci zvolen engine Unity 6, především kvůli možnosti tvorby desktopové aplikace, využití systému UI Toolkit a dobré připravenosti pro integraci se službami třetích stran.

Praktická část práce se zaměřila na návrh požadavků, datového modelu a samotnou implementaci aplikace. Pro ukládání kurzů byl zvolen formát JSON, který umožňuje přehledně reprezentovat strukturu kurzu, jeho kapitoly a jednotlivé výukové moduly. Výhodou tohoto řešení je jednoduchá čitelnost, přenositelnost a možnost ukládání dat lokálně na zařízení uživatele. Součástí návrhu byla také správa příložených souborů, které jsou ukládány do samostatné složky kurzu a automaticky pojmenovávány tak, aby bylo možné je jednoznačně přiřadit ke konkrétní kapitole a modulu.

Výsledná aplikace obsahuje tvořivý režim, ve kterém může uživatel založit nový kurz, upravovat jeho kapitoly a vkládat do nich různé typy výukových modulů. Implementovány byly například poznámkové moduly, kvízové otázky, karty pro procvičování pojmů a modul typu drag and drop pro řazení odpovědí. Tyto prvky umožňují vytvářet různorodý vzdělávací obsah a přizpůsobit kurz konkrétním potřebám vyučujícího nebo studenta. Důraz byl kladen také na přehlednost rozhraní, barevné odlišení modulů pro rychlou orientaci uživatele v prostředí a jednoduché ovládání pomocí myši a klávesnice.

Druhou hlavní částí aplikace je studijní režim. V něm uživatel prochází vytvořené nebo stažené kurzy a jednotlivé moduly jsou mu zobrazovány v podobě vhodné pro učení. Aplikace poskytuje okamžitou zpětnou vazbu u interaktivních úloh, čímž podporuje aktivní procvičování učiva. Součástí studijní části je také ukládání špatně zodpovězených otázek, které lze následně využít pro opakování problematických částí kurzu. Tím aplikace nepředstavuje

pouze pasivní prohlížeč studijních materiálů, ale nabízí i základní mechanismy podporující efektivnější učení.

Významnou částí práce byla také příprava aplikace na distribuci a integraci se službami Steamworks. V práci byly popsány možnosti využití SteamID pro identifikaci uživatele, Steam Workshopu pro sdílení kurzů a Steam Cloudu pro práci s daty a jejich automatické zálohování. Součástí řešení byla také příprava stránky aplikace v obchodě Steam a příprava buildu pro distribuci. Díky tomu má výsledný prototyp potenciál fungovat nejen jako lokální výukový nástroj, ale také jako základ komunitní platformy, ve které mohou uživatelé vytvářet a sdílet vlastní kurzy.

Za hlavní přínos práce lze považovat vytvoření funkčního základu výukové aplikace, který spojuje jednoduchou tvorbu obsahu, lokální práci s daty, studijní režim a možnost sdílení kurzů přes platformu Steam. Aplikace ukazuje, že herní engine Unity 6 lze efektivně využít nejen pro tvorbu klasických her, ale také pro vývoj vzdělávacího softwaru zaměřeného na uživatelsky vytvářený obsah. Zvolené řešení zároveň ponechává prostor pro další rozšiřování bez nutnosti zásadní změny architektury.

Do budoucna je možné aplikaci rozšířit o další typy výukových modulů, pokročilejší systém statistik nebo detailnější vyhodnocování úspěšnosti studenta. Dalším vhodným směrem rozvoje je hlubší integrace se službou Steam Workshop, dlouhodobé uživatelské testování, doplnění lokalizací a optimalizace uživatelského rozhraní podle zpětné vazby reálných uživatelů. Výsledná práce tak nepředstavuje pouze uzavřený prototyp, ale především základ pro další vývoj výukové aplikace, která může být v budoucnu rozšířena do podoby plnohodnotného nástroje pro tvorbu, sdílení a studium vzdělávacích kurzů.

Seznam použité literatury

- [1] ABT, Clark. Serious Games. The Viking Press, 1970. ISBN 9780670634903.
- [2] CLARK, Ruth a MAYER, Richard. E-Learning and the Science of Instruction Proven Guidelines for Consumers and Designers of Multimedia Learning. 5. John Wiley, 2023. ISBN 978-1-394-17737-0.
- [3] KLOPFER, Eric; HAAS, Jason; OSTERWEIL, Scot a ROSENHECK, Louisa. Resonant Games Design Principles for Learning Games that Connect Hearts, Minds, and the Everyday. MIT Press, 2018. ISBN 978-0-262-03780-8.
- [4] KALMPOURTZIS, George. Educational Game Design Fundamentals A Journey to Creating Intrinsically Motivating Learning Experiences. CRC Press, 2019. ISBN 978-1-138-63154-0.
- [5] Serious Games Global Market Report 2025: Serious Game Market Overview. Online. In: . Prosinec 2025, s. 200. Dostupné z: <https://www.thebusinessresearchcompany.com/report/serious-games-global-market-report>. [cit. 2026-01-23].
- [6] Online. In: Serious Games Market Report 2026. The Business Research Company, 2026, s. 250. Dostupné z: <https://www.researchandmarkets.com/reports/5767439/serious-games-market-report>. [cit. 2026-03-28].
- [7] Duolingo 101: How to learn a language on Duolingo. Online. In: Blog Duolingo. 2024. Dostupné z: <https://blog.duolingo.com/duolingo-101-how-to-learn-a-language-on-duolingo/>. [cit. 2026-03-16].
- [8] Google Play. Online. 2026. Dostupné z: <https://play.google.com/store/apps/details?id=com.metastream.autoskolaplus&hl=cs>. [cit. 2026-03-16].
- [9] Kahoot!. Online. Kahoot!. 2013. Dostupné z: <https://kahoot.it/>. [cit. 2026-03-28].
- [10] MOODLE documentation. Online. 2018, 5.12.2018. Dostupné z: <https://docs.moodle.org/501/en/Philosophy>. [cit. 2026-03-28].
- [11] HOLAN, Tomáš. Unity: První seznámení s tvorbou počítačových her. Milešovská 5, 130 00 Praha 3: CZ.NIC, 2020. ISBN 978-80-88168-60-7.
- [12] Unity 6 UI Toolkit: News and updates. Online. In: Unity. 2024, s. 1. Dostupné z: <https://unity.com/blog/unity-6-ui-toolkit-updates>. [cit. 2026-03-15].
- [13] EPIC GAMES. Unreal Engine - Features. Online. 2022. Dostupné z: <https://www.unrealengine.com/en-US/features>. [cit. 2026-03-28].
- [14] EPIC GAMES. Epic Games Product Documentation. Online. 2026. Dostupné z: <https://dev.epicgames.com/documentation/>. [cit. 2026-03-28].
- [15] GODOT - About. Online. © 2026. Dostupné z: <https://godot.foundation/>. [cit. 2026-03-28].
- [16] GODOT DOCS - Nodes and Scenes. Online. © 2014. Dostupné z: https://docs.godotengine.org/en/stable/getting_started/step_by_step/nodes_and_scenes.html. [cit. 2026-03-28].
- [17] GodotSteam. Online. © 2015. Dostupné z: <https://godotsteam.com/>. [cit. 2026-03-28].
- [18] VALVE. Dokumentace systému Steamworks. Online. 2022. Dostupné z: <https://partner.steamgames.com/doc/home>. [cit. 2026-04-11].

[19] MAYER, Jiří. Installing Steamworks in Unity. Online. In: Unisave. 2025. Dostupné z: <https://unisave.cloud/guides/installing-steamworks-in-unity>. [cit. 2026-04-11].