

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

TVORBA VZDĚLÁVACÍCH MATERIÁLŮ PRO VÝUKU
HERNÍHO VÝVOJE S VYUŽITÍM UNITY A JEJICH
INTEGRACE DO MOODLE

Bakalářská práce

Autor práce: Stanislav Čech

Vedoucí práce: Ing. Zdeněk Smiščík

Jihlava 2025

Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	Stanislav Čech
Studijní program:	Aplikovaná informatika
Obor:	Aplikovaná informatika
Garant studijního programu:	Ing. Lenka Kulišová Pavelková, Ph.D.
Název práce:	TVORBA VZDĚLÁVACÍCH MATERIÁLŮ PRO VÝUKU HERNÍHO VÝVOJE S VYUŽITÍM UNITY A JEJICH INTEGRACE DO MOODLE
Vedoucí práce:	Ing. Zdeněk Smištík
Cíl práce:	Cílem této bakalářské práce je vytvořit vzdělávací materiály pro výuku vývoje počítačových her s využitím herního engine Unity a programovacího jazyka C#. Tyto materiály budou pedagogicky strukturovány tak, aby studentům poskytly teoretické znalosti a praktické dovednosti v oblasti herního vývoje. Práce zahrne nejen technické aspekty tvorby her, ale také vytvoření SCORM balíčku pro LMS Moodle, který umožní snadnou integraci výukového obsahu do online vzdělávacího prostředí. Výstupem budou učební texty, videonávody, praktické úkoly a testy, které budou systematicky uspořádány s ohledem na pedagogickou efektivitu.

Abstrakt

Bakalářská práce se zaměřuje na tvorbu pedagogického materiálu pro výuku vývoje počítačových her s využitím platformy Unity a programovacího jazyka C#. Hlavním cílem je poskytnout studentům interaktivní a prakticky orientovaný způsob výuky, který propojuje teoretické základy herního vývoje s jejich aplikací v reálném projektu. Praktická část práce zahrnuje návrh a implementaci jednoduché 2D hry, ve které hráč ovládá vozidlo pohybující se po herní mapě, sbírá zásilky a doručuje je zákazníkům. Hra obsahuje základní herní mechaniky, jako je řízení vozidla, práce s fyzikou, kolizní systém, časový limit, jednoduché uživatelské rozhraní či dynamické generování objektů. Tyto prvky slouží jako modelový příklad, na němž lze demonstrovat typický proces vývoje malé hry v Unity.

Součástí práce je také tvorba výukového kurzu, který kombinuje textové materiály, instruktážní videa, praktické úkoly a testy. Celý kurz je exportován ve formátu SCORM a integrován do vzdělávací platformy LMS Moodle, což umožňuje jeho snadné nasazení v prostředí elektronického vzdělávání. Forma tohoto zpracování podporuje moderní pedagogické přístupy, které kladou důraz na samostatnou práci studenta a propojení výkladu s okamžitou praktickou aplikací.

Cílem práce je vytvořit ucelený a didakticky efektivní výukový modul, který studentům umožní pochopit základní principy vývoje her v Unity a zároveň jim poskytne prvotní zkušenost s kompletním vývojovým procesem od návrhu projektu až po jeho prezentaci ve formě výukového obsahu v LMS.

Klíčová slova

Unity; C#; SCORM; LMS Moodle; vývoj her; výuka programování; 2D hry; herní engine; e-learning; počítačové hry

Abstract

The aim of this bachelor thesis is to create educational materials for teaching game development using the Unity platform and the C# programming language. The primary objective is to provide students with an interactive and practically oriented method of learning that combines theoretical knowledge with its direct application in a real project. The practical part of the thesis includes the design and implementation of a simple 2D game in which the player controls a vehicle moving around a map, collecting packages and delivering them to customers. The game features fundamental mechanics such as movement control, physics handling, collision detection, object spawning, a time limit, and a basic user interface. These elements serve as a model example for demonstrating the typical workflow of developing a small game in Unity.

The thesis also focuses on the creation of a complete educational course that includes textual study materials, instructional videos, practical assignments, and quizzes. The course is exported as a SCORM package and integrated into the LMS Moodle platform, enabling easy deployment within an e-learning environment. This approach supports modern pedagogical methods that

emphasize active learning, self-study, and the connection between theory and hands-on practice.

The overall goal of the thesis is to develop a comprehensive and pedagogically effective learning module that helps students understand the essential principles of game development in Unity while gaining experience with the full development process—from game design to the creation of reusable instructional content.

Keywords

Unity; C#; SCORM; LMS Moodle; game development; programming education; 2D games; game engine; e-learning; video games

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 26. listopadu 2025

.....

Podpis studenta

Poděkování

Rád bych na tomto místě vyjádřil poděkování všem, kteří mě během zpracování této bakalářské práce podporovali. Mé poděkování patří především vedoucímu práce panu Ing. Zdeňkovi Smiščíkovi, který mi poskytl cenné rady, odborné vedení a zpětnou vazbu nezbytnou pro úspěšné dokončení této práce.

Mé díky patří také mé přítelkyni, která mi poskytla významnou podporu a vytvořila mi podmínky a prostor potřebný k realizaci této práce. Její trpělivost, pochopení a ochota přizpůsobit se nárokům spojeným s tvorbou projektu pro mě byly nenahraditelné.

*Zároveň bych rád poděkoval tvůrci výukového obsahu známému jako **Code Bro**, jehož videa představovala důležitý zdroj inspirace při tvorbě vzdělávacích materiálů i samotného kurzu. Jeho práce přispívá k dostupnějšímu a efektivnějšímu vzdělávání začínajících vývojářů.*

Obsah

Seznam obrázků.....	8
Seznam zkratk.....	9
Úvod	10
1 Počítačové hry.....	11
1.1 Definice a typologie	11
1.2 Historie vývoje	11
1.3 Moderní trendy ve vývoji her	12
1.4 Vliv počítačových her	12
1.5 Vývoj her	13
1.6 Základy herního designu.....	13
1.7 Technologie ve vývoji her	14
2 Unity.....	15
2.1 Hlavní výhody Unity:.....	15
2.2 Fáze vývoje her	17
2.3 Budoucnost herního vývoje	18
2.4 Rozdíly mezi 2D a 3D	18
3 LMS Moodle	20
3.1 Klíčové vlastnosti Moodle.....	20
3.2 Význam Moodle ve vzdělávání	20
3.3 Role multimediálního obsahu při výuce programování	20
3.4 SCORM balíčky	21
3.5 Srovnání Moodle s dalšími LMS systémy.....	23
3.6 Kritéria výběru LMS pro vzdělávací účely	23
3.7 Trendy a budoucnost LMS systémů.....	24
4 Realizace výukového kurzu v Unity	25
4.1 Návrh ukázkové hry	25
4.2 Implementace herní logiky v Unity.....	26
4.3 Implementace zvukových prvků	32
4.4 Implementace částicových efektů (Particle System).....	33
4.5 Možnosti dalšího rozvoje herního projektu.....	34
4.6 Tvorba SCORM balíčku v nástroji exeLearning	35
4.7 Tvorba výukových videí	36
4.8 Vytvoření výukových textů	36
4.9 Návrh praktických úkolů	36
4.10 Tvorba testů.....	36
4.11 Hostování výukového kurzu v prostředí Moodle.....	37
4.12 Vizualizace a pedagogická struktura výukového kurzu	37
4.13 Uživatelské testování a zpětná vazba	39
5 Zhodnocení výsledků	40
6 Závěr.....	41
Seznam použité literatury	42

Seznam obrázků

Obr. 1: Tennis for Two (1958)	11
Obr. 2: Vývoj her	13
Obr. 3: Unity logo	15
Obr. 4: Unreal Engine Logo	16
Obr. 5: Godot Engine.....	16
Obr. 6: Fáze vývoje her.....	17
Obr. 7: Super Mario Bros	19
Obr. 8: Doom.....	19
Obr. 9: SCORM	22
Obr. 10: Návrh mapy	25
Obr. 11: Hierarchie projektu	26
Obr. 12: Inspector objektu Mcqueen.....	27
Obr. 13: Ukázka karty Tag	29
Obr. 14: Zvuková komponenta.....	33
Obr. 15: Particle system Component.....	34
Obr. 16: Náhled finálního projektu	35
Obr. 17: Zobrazení kurzu.....	38
Obr. 18: Obsah kurzu	38

Seznam zkratk

LMS	Learning Management System
VR	Virtuální realita (Virtual Reality)
AR	Rozšířená realita (Augmented Reality)
SCORM	Sharable Content Object Reference Model
NPC	Nehratelné postavy (Non-Playable Characters)
AI	Umělá inteligence (Artificial Intelligence)

Úvod

Vývoj počítačových her představuje dynamicky se rozvíjející oblast informačních technologií, která přitahuje pozornost jak profesionálních vývojářů, tak široké veřejnosti. Rostoucí dostupnost moderních herních engineů umožňuje vytvářet funkční a vizuálně atraktivní hry i začátečníkům, kteří tak mají možnost osvojit si základní principy programování prostřednictvím interaktivních úloh. Počítačové hry nabízejí nejen zábavu, ale také efektivní způsob učení, jelikož spojují motivaci, okamžitou zpětnou vazbu a praktickou aplikaci získaných znalostí.

Bakalářská práce se zaměřuje na tvorbu pedagogického materiálu pro výuku vývoje počítačových her s využitím nástroje Unity a programovacího jazyka C#. Cílem práce je vytvořit ucelený výukový modul, který studentům umožní porozumět základním principům herní logiky, skriptování a práce s herním engineem. Součástí práce je také vytvoření multimediálních výukových materiálů zahrnujících texty, instruktážní videa, praktická cvičení a testy. Materiály jsou následně integrovány do vzdělávací platformy Moodle prostřednictvím SCORM balíčku.

V praktické části je navržena a realizována jednoduchá 2D hra, v níž hráč ovládá vozidlo pohybující se po herní mapě, sbírá zásilky a doručuje je zákazníkům. Hra obsahuje základní herní mechaniky, jako je práce s fyzikou, detekce kolizí, dynamické generování objektů, časomíra či uživatelské rozhraní. Tento modelový projekt slouží jako prostředek k demonstraci postupů používaných při vývoji her v Unity a zároveň jako základní platforma pro výklad jednotlivých kapitol výukového kurzu.

1 Počítačové hry

Počítačové hry jsou důležitou součástí digitální kultury a technologického rozvoje. Jsou definovány jako interaktivní média, která hráčům umožňují ovlivňovat průběh herního děje a výsledky prostřednictvím předem definovaných pravidel a mechanik. Díky své variabilitě jsou využívány nejen jako zábava, ale i jako nástroj pro výuku, simulace a odborný trénink.

1.1 Definice a typologie

Rozdělit počítačové hry lze do několika kritérií, která odrážejí jejich různé vlastnosti a možnosti využití. Každý z těchto typů her se vyznačuje odlišným využitím ve vzdělávání, zábavě či profesionálním prostředí. Mezi hlavní kategorie patří:

- **Žánry:** Akční hry, strategické hry, simulátory, adventury, RPG (role-playing games), logické hry a mnoho dalších. Každý žánr přináší specifický styl hraní a zaměření na odlišné aspekty, jako je rychlost rozhodování, řešení problémů nebo týmová spolupráce.
- **Platformy:** Hry určené pro herní konzole, osobní počítače, mobilní zařízení či cloudové hraní. Platformy určují technické možnosti hry a její dostupnost pro uživatele.
- **Způsob distribuce:** Single-player hry zaměřené na individuální zážitek, multiplayer hry vyžadující interakci více hráčů nebo online hry, které propojují hráče z celého světa prostřednictvím internetu.

1.2 Historie vývoje

Historie počítačových her začíná v polovině 20. století. První pokusy o vytvoření digitálních her zahrnovaly experimentální simulace na velkých počítačích, které byly tehdy k dispozici pouze na univerzitách. Jednou z prvních her byla *Tennis for Two* (1958), která simulovala tenisovou hru na osciloskopu.



Obr. 1: Tennis for Two (1958)

Zdroj: https://en.wikipedia.org/wiki/Tennis_for_Two

V 70. letech se objevily první komerčně dostupné arkádové hry, jako například *Pong*, které znamenaly průlom v zábavním průmyslu. 80. léta přinesla domácí konzole a počítače, jako Atari nebo Commodore 64, což umožnilo rozšíření her mezi širokou veřejnost. V 90. letech herní průmysl dosáhl nové úrovně díky 3D grafice a komplexním příběhovým hrám.

Současné hry využívají nejmodernější technologie, jako je virtuální realita (VR), rozšířená realita (AR) a umělá inteligence (AI), které výrazně mění způsob, jakým hráči interagují s herním světem. Herní průmysl se stal jedním z největších odvětví zábavy s ročním obratem přesahujícím stovky miliard dolarů.

1.3 Moderní trendy ve vývoji her

Moderní herní průmysl se neustále vyvíjí a reaguje na technologické změny i rostoucí nároky hráčů. Mezi nejvýraznější současné tendence patří zejména zjednodušování vývojových nástrojů a zpřístupňování tvorby her širší veřejnosti. Významnou roli hrají multiplatformní nástroje, jako je například Unity, které umožňují vytvářet komplexní hry bez nutnosti hlubokých znalostí programování. Díky tomu se do popředí dostává také komunitní a nezávislá tvorba, která přináší nové herní nápady a experimentální mechaniky.

Roste také důraz na hratelnost, přístupnost a celkovou uživatelskou zkušenost. Vývojáři stále častěji kombinují tradiční 2D stylizaci s moderními postupy a technikami, protože 2D hry umožňují rychlejší prototypování, přehlednější vizuální styl a snadnější výuku základních principů designu. Tyto trendy mají přímý dopad i na vzdělávání, kde se hry a herní enginy využívají nejen jako nástroje pro výuku programování, ale také jako prostředek k rozvoji logického myšlení a kreativních dovedností.

1.4 Vliv počítačových her

Počítačové hry představují významný fenomén současné společnosti a jejich dopad zasahuje do mnoha oblastí každodenního života. Jedním z nejdůležitějších pozitivních vlivů je podpora kognitivních a technických dovedností. Hráči se při hraní učí rychle reagovat, řešit problémy a analyzovat situace, což posiluje jejich kreativitu i strategické myšlení. Hry se rovněž stávají důležitou součástí vzdělávání, kde slouží jako nástroj umožňující interaktivní a atraktivní formu výuky. Učební hry mohou přispět k lepšímu pochopení například matematických, logických či programovacích konceptů.

Významný je také ekonomický dopad herního průmyslu, který se řadí mezi nejrychleji rostoucí odvětví zábavního průmyslu. Herní vývoj zaměstnává široké spektrum profesí, od programátorů přes grafiky a designéry až po specialisty na hudbu či zvuk.

Vedle pozitivních dopadů jsou však počítačové hry diskutovány také v souvislosti s možnými riziky, mezi něž patří zejména vznik návykového chování či nadměrné trávení času hraním. Aspekty jsou předmětem dlouhodobých výzkumů a jejich hodnocení se různí. V kontextu práce jsou hry nahlíženy především jako nástroj pro rozvoj znalostí a dovedností, zejména v oblasti programování a kreativního myšlení.

1.5 Vývoj her

Vývoj her je komplexní proces, který kombinuje technické dovednosti, kreativitu a týmovou spolupráci. Tento obor zahrnuje několik fází, od plánování a návrhu až po implementaci, testování a distribuci. Vývoj her je nejen technickou disciplínou, ale i uměním, které vyžaduje pochopení herní mechaniky, vizuálního designu a interakce s hráči.



Obr. 2: Vývoj her

Zdroj: <https://www.turing.com/resources/4-reasons-why-you-should-hire-a-video-game-developer>

1.6 Základy herního designu

Herní design představuje klíčovou disciplínu v procesu vývoje počítačových her, jejímž cílem je definovat strukturu, pravidla a celkový charakter herního zážitku. Jeho úkolem je navrhnout takové herní prvky a mechaniky, které budou pro hráče srozumitelné, zábavné a dostatečně motivující. Základem herního designu jsou herní mechaniky, pravidla a akce, které může hráč ve hře vykonávat. Tyto mechaniky určují interakci hráče s herním světem a vytvářejí rámec celého herního zážitku.

Důležitou součástí herního designu je také tzv. core gameplay loop, opakující se cyklus činností, který tvoří jádro hratelnosti. Jedná se o sekvenci akcí, které hráč vykonává opakovaně po celou dobu hraní, a které musí být dostatečně přitažlivé, aby udržely jeho pozornost. Správně navržený cyklus vede hráče k tomu, aby se k hře vracel, protože nabízí rovnováhu mezi výzvou, odměnou a pokrokem.

Zásadní roli hraje také vyváženost obtížnosti. Kvalitní herní design zajišťuje, že hra není ani příliš jednoduchá, ani nadměrně frustrující. Vyváženost se týká nejen obtížnosti jednotlivých úkolů, ale také tempa hry a míry odměn. Důležitá je rovněž srozumitelná a funkční zpětná vazba, která hráče informuje o výsledcích jeho akcí a motivuje ho k dalšímu postupu. Zpětná vazba

zahrnuje vizuální efekty, zvukové signály, zvýraznění objektů či aktualizaci uživatelského rozhraní.

Herní design je interdisciplinární oblastí, která vyžaduje spolupráci programátorů, grafiků, zvukařů a dalších odborníků. Každý prvek musí být navržen tak, aby zapadal do celku a přispíval k ucelenému hernímu zážitku. Dobře sladěné mechaniky, estetika a technické zpracování jsou klíčem k tomu, aby výsledná hra byla nejen funkční, ale především atraktivní pro hráče.

1.7 Technologie ve vývoji her

Použití moderních technologií hraje zásadní roli v celém procesu vývoje počítačových her. Umožňuje vytvářet vizuálně propracované, technicky stabilní a interaktivní projekty, které splňují současné standardy herního průmyslu. Nástroje používané při vývoji her se neustále vyvíjejí a stávají se přístupnějšími, což výrazně snižuje překážky pro začátečníky i nezávislé vývojáře. Mezi nejdůležitější technologie ve vývoji her patří zejména:

- **Herní enginy:** platformy jako Unity, Unreal Engine nebo Godot poskytují komplexní prostředí pro tvorbu 2D i 3D her. Obsahují nástroje pro práci se scénami, fyzikou, animacemi, skriptováním a uživatelským rozhraním. Unity patří mezi nejrozšířenější díky intuitivnímu ovládání a podpoře multiplatformního vývoje.
- **Grafické nástroje:** software, jako je Blender, Adobe Photoshop nebo Affinity Designer, umožňuje vytváření 2D i 3D grafiky, animací a textur. Tyto programy poskytují vývojářům možnost navrhovat vlastní vizuální styl her a optimalizovat grafické podklady pro herní engine.
- **Programovací jazyky:** vývoj herní logiky a mechanismů je tradičně založen na jazycích C#, C++ nebo Python. Unity využívá především C#, který nabízí kombinaci jednoduché syntaxe a dostatečné flexibility pro tvorbu komplexních systémů.
- **Testovací a diagnostické nástroje:** moderní vývojové prostředí umožňuje automatizovat testování kolizí, výkonu nebo logiky hry. Profiler, ladicí nástroje a simulátory zátěže pomáhají identifikovat chyby a optimalizovat hru ještě před jejím dokončením.

Tyto technologie se vzájemně doplňují a společně umožňují efektivní vývoj herních projektů různé složitosti. V kontextu této práce je hlavní pozornost věnována nástroji Unity a programovacímu jazyku C#, které tvoří základ tvorby ukázkové hry i výukového materiálu.

2 Unity

Unity je jedním z nejrozšířenějších herních enginů, používaným pro vývoj 2D i 3D her. Je oblíbený díky své přístupnosti, rozsáhlé dokumentaci a podpoře širokého spektra platforem, včetně PC, konzolí, mobilních zařízení a VR.

2.1 Hlavní výhody Unity:

- **Univerzálnost:** Umožňuje vývoj her pro různé platformy bez nutnosti významných změn v kódu.
- **Programování v C#:** Unity využívá intuitivní a přístupný jazyk C#, což usnadňuje výuku i práci vývojářů.
- **Asset Store:** Poskytuje vývojářům přístup k tisícům modelů, textur a nástrojů, které urychlují vývoj.
- **Podpora VR a AR:** Unity je často preferováno pro projekty zaměřené na virtuální nebo rozšířenou realitu.



Obr. 3: Unity logo

Zdroj: https://en.wikipedia.org/wiki/Unity_%28game_engine%29

2.1.1 Unreal Engine

Unreal Engine, vyvíjený společností Epic Games, je známý svým výkonem a pokročilými grafickými možnostmi. Je často používán pro tvorbu AAA her a filmových vizuálních efektů díky svým výkonným nástrojům pro práci s 3D grafikou.

Hlavní výhody Unreal Engine:

- **Realistická grafika:** Unreal Engine nabízí technologie jako ray tracing (renderovací technika používaná v počítačové grafice k simulaci fyzikálně přesného chování světla) a fotorealistické renderování.
- **Blueprints Visual Scripting:** Umožňuje vytvářet herní mechaniky bez nutnosti psát kód, což usnadňuje práci méně zkušeným vývojářům.
- **Pokročilé nástroje pro fyziku a animace:** Poskytuje realistické simulace pohybu a interakce objektů.
- **Podpora velkých projektů:** Unreal Engine je ideální pro rozsáhlé a komplexní hry s detailní grafikou.



Obr. 4: Unreal Engine Logo

Zdroj: https://rvb.fandom.com/wiki/Unreal_Engine

2.1.2 Godot

Godot je open-source herní engine, který si získává popularitu díky své jednoduchosti, flexibilitě a dostupnosti. Je vhodný pro vývoj 2D i 3D her, přičemž klade důraz na intuitivní uživatelské rozhraní a nízkou náročnost na systémové zdroje.

Hlavní výhody Godotu:

- **Otevřený kód:** Engine je zcela zdarma, bez licenčních poplatků, což z něj činí skvělou volbu pro nezávislé vývojáře.
- **Jednoduché skriptování:** Používá vlastní jazyk GDScript, který je snadno pochopitelný a podobný Pythonu.
- **Flexibilita:** Umožňuje vývoj pro různé platformy, včetně mobilních zařízení a konzolí.
- **Lehkost:** Díky nízkým systémovým požadavkům je Godot ideální pro menší projekty a indie hry.



Obr. 5: Godot Engine

Zdroj: <https://godotengine.org/>

2.2 Fáze vývoje her

Vývoj počítačových her probíhá ve více po sobě navazujících etapách, z nichž každá má specifický účel a charakter. Tyto fáze společně tvoří ucelený proces, který umožňuje převést původní myšlenku do funkčního herního produktu. Jednotlivé části vývoje se vzájemně prolínají a často se k nim přistupuje cyklicky, především při testování nebo úpravě herních mechanik. Grafické znázornění typických fází vývoje je uvedeno na obrázku níže.



Obr. 6: Fáze vývoje her

Zdroj: <https://kevurugames.com/blog/6-key-stages-of-game-development-from-concept-to-standing-ovation/>

Vývoj her lze shrnout do následujících základních fází:

- **Plánování a koncept (Pre-production):** V počáteční fázi se formuluje základní myšlenka hry, její cílová skupina, herní mechaniky a celková vize. Součástí je také návrh technologií, volba herního engine a vytvoření úvodní dokumentace, která slouží jako podklad pro další vývoj.
- **Prototypování (Pre-production):** Pro účely ověření herních nápadů se vytvářejí jednoduché prototypy, které umožňují rychle testovat jednotlivé mechaniky, interakce nebo vizuální styl. Tento krok pomáhá odhalit případné problémy ještě před zahájením plné produkce.
- **Produkce (Production):** V hlavní fázi vývoje probíhá implementace všech klíčových prvků hry. Zahrnuje programování, tvorbu grafických podkladů, zvuků a animací, návrh uživatelského rozhraní a celkovou integraci herních systémů. Produkce tvoří časově nejnáročnější část projektu.
- **Testování (Testing):** V této fázi se zaměřuje pozornost na identifikaci a odstranění chyb, ladění hratelnosti a optimalizaci výkonu. Testování zajišťuje stabilitu hry a ověřuje, zda jsou všechny mechanismy funkční a odpovídají původnímu návrhu.
- **Předběžné vydání (Pre-launch):** Zahrnuje finální úpravy a přípravu hry na distribuci. V této fázi se obvykle vyhodnocují výsledky interního testování, dokončuje se marketingová komunikace a provádí se poslední optimalizační kroky.
- **Vydání a následná podpora (Launch a Post-production):** Po dokončení všech předchozích kroků je hra distribuována na zvolené platformy, například Steam nebo

mobilní obchody. Po vydání následuje fáze podpory, která zahrnuje opravy chyb, vydávání aktualizací nebo rozšiřujícího obsahu na základě zpětné vazby hráčů.

2.3 Budoucnost herního vývoje

Budoucnost herního vývoje je úzce spjata s rychlým technologickým pokrokem, který zásadně ovlivňuje podobu herního průmyslu. Virtuální realita (VR) a rozšířená realita (AR) se stávají stále dostupnějšími, což umožňuje hráčům prožívat hry zcela novým způsobem. Tyto technologie jsou využívány nejen ve hrách, ale také v simulacích a tréninkových programech, například pro výcvik pilotů nebo lékařů.

Dalším významným směrem je využití umělé inteligence (AI), která nachází uplatnění při tvorbě realistických NPC (nehratelných postav) a adaptivních herních prostředí. AI také umožňuje personalizaci herního zážitku, kdy se hra přizpůsobuje schopnostem a preferencím hráče.

Herní průmysl se rovněž zaměřuje na nezávislou tvorbu her. Díky dostupnosti moderních nástrojů, jako jsou Unity a Godot, mají i malé týmy možnost vytvářet hry s vysokou kvalitou. Nezávislá produkce přináší inovativní přístupy k designu a příběhům, které často nejsou typické pro velká studia.

2.4 Rozdíly mezi 2D a 3D

Vývoj her zahrnuje různé přístupy v závislosti na tom, zda je hra navržena ve 2D, 3D nebo pro virtuální realitu (VR). Každý z těchto přístupů má své specifické požadavky, nástroje a technologické výzvy.

2.4.1 Hry ve dvourozměrném prostoru

2D hry představují nejjednodušší formu herního vývoje, zaměřenou na dvourozměrný prostor. Obrazy, nazývané sprite, se pohybují v rovině X a Y, což usnadňuje práci s grafikou a mechanikou. Typickými příklady jsou platformové hry, puzzle nebo arkády.

Výhody:

- Menší náročnost na grafické zdroje.
- Jednodušší programování herní fyziky.
- Ideální pro začínající vývojáře.

Nevýhody:

- Omezená vizuální hloubka.
- Menší prostor pro realistické simulace.



Obr. 7: Super Mario Bros

Zdroj: <https://www.cbr.com/hardest-2d-mario-levels-ranked/>

2.4.2 Videohry v třírozměrném prostoru

3D hry umožňují hráčům pohybovat se ve trojrozměrném prostoru (X, Y, Z). Tento typ her využívá 3D modely a komplexní texturování, což přináší realistický vzhled a větší ponoření do herního světa. Typickými příklady jsou RPG, simulátory nebo akční hry.

Výhody:

- Realistická grafika a fyzika.
- Možnost komplexního herního designu (otevřené světy, detailní animace).

Nevýhody:

- Vyšší nároky na hardware a vývojové dovednosti.
- Delší vývojový cyklus.



Obr. 8: Doom

Zdroj: <https://doomhell.com/index.php/2019/07/28/doom-builder-guide-3d-floors/>

3 LMS Moodle

LMS (Learning Management System) Moodle je open-source platforma určená pro tvorbu a správu e-learningových kurzů. Díky své flexibilitě a široké škále funkcí patří mezi nejrozšířenější nástroje používané ve vzdělávání. Moodle umožňuje učitelům vytvářet interaktivní kurzy, spravovat studijní materiály a sledovat pokroky studentů.

3.1 Klíčové vlastnosti Moodle

Moodle nabízí široké spektrum funkcí, které ho činí vhodným nástrojem pro vzdělávací instituce i jednotlivce:

- **Podpora různých typů obsahu:** Texty, prezentace, videa, interaktivní kvízy, SCORM balíčky.
- **Správa uživatelů:** Umožňuje jednoduchou registraci, přiřazování rolí (učitel, student, správce) a správu kurzů.
- **Sledování pokroku:** Učitelé mohou monitorovat účast a výsledky studentů v kurzech.
- **Modularita:** Moodle umožňuje přidávat a upravovat funkce prostřednictvím pluginů.
- **Open-source povaha:** Díky svému volně dostupnému kódu je Moodle přizpůsobitelný specifickým potřebám uživatelů.

3.2 Význam Moodle ve vzdělávání

Moodle představuje široce využívaný nástroj, který umožňuje vzdělávacím institucím efektivně organizovat a poskytovat kurzy v online prostředí. Platforma podporuje vzdělávání na dálku, díky čemuž mohou studenti přistupovat k výukovým materiálům kdykoliv a odkudkoliv. Moodle zároveň nabízí možnosti personalizovaného učení, jelikož umožňuje sledovat individuální pokroky studentů a přizpůsobovat obsah jejich potřebám. Důležitým prvkem je také podpora interakce mezi studenty a vyučujícími prostřednictvím diskusních fór, chatů a dalších komunikačních nástrojů, které usnadňují spolupráci a výměnu informací.

Ve spojení s formátem SCORM získává Moodle ještě širší využití. Umožňuje totiž snadnou integraci interaktivních materiálů, testů a aktivit, jejichž výsledky je možné automaticky zaznamenávat a vyhodnocovat. Díky tomu se Moodle stává flexibilní platformou vhodnou pro moderní e-learningové postupy, které kladou důraz na aktivní učení a okamžitou zpětnou vazbu.

3.3 Role multimediálního obsahu při výuce programování

Multimediální obsah představuje klíčový prvek moderního e-learningu, zejména v oblastech, kde je nutné kombinovat teoretické poznatky s praktickou ukázkou. Výuka programování a vývoje her je jedním z typických příkladů. V těchto oblastech studenti často pracují s abstraktními koncepty, jejichž pochopení je výrazně usnadněno vizuální, zvukovou nebo interaktivní formou prezentace.

Využití multimédií přispívá ke zvýšení didaktické účinnosti výukového procesu hned několika způsoby. Zaprvé umožňuje kombinaci textového výkladu s vizuální demonstrací. Například

základní principy práce s herními komponentami v Unity, jako jsou transformace, kolizní vrstvy nebo události typu trigger, lze sice popsat textově, avšak skutečného porozumění studenti dosahují především tehdy, když vidí jejich fungování přímo v editoru. Krátká instruktážní videa poskytují okamžitý vhled do postupů, které by jinak vyžadovaly zdlouhavé popisy, a pomáhají účinně překlenout rozdíl mezi abstraktní teorií a praktickým provedením.

Dalším významným přínosem multimediálního obsahu je možnost demonstrovat dynamické **jevy**, které nelze zachytit statickým textem. V prostředí Unity se mnohé procesy odehrávají v reálném čase například pohyb objektů, reakce na kolize, aktualizace proměnných v průběhu herního cyklu nebo běh časovače. Instruktážní videa umožňují sledovat tyto jevy přímo během vývoje a poskytují studentovi jasnou představu o tom, jak jsou skripty propojeny s chováním objektů ve scéně.

Multimediální prvky se uplatňují i ve zvyšování motivace studentů. Interaktivní kurz, který kombinuje textové popisy, obrázky, videoukázky, testové otázky a praktické úkoly, podporuje aktivní zapojení studenta výrazně více než jednotvárné textové materiály. Videoobsah navíc umožňuje studentům pracovat vlastním tempem mohou si jej zastavit, zpomalit nebo přehrát znovu, což je důležité zejména při práci s programovacím jazykem C#, kde pochopení každého detailu může zásadně ovlivnit výsledek.

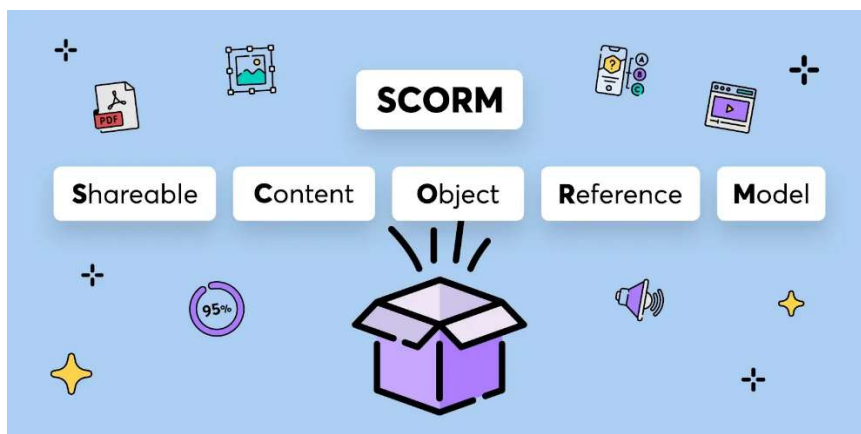
Další výhodou multimédií je možnost simultánního využití více forem zpětné vazby. Studenti mohou sledovat nejen výsledné chování objektů, ale také dění v Inspectoru, konzoli nebo hierarchii objektů. To poskytuje komplexní pohled na práci v Unity a umožňuje studentům rychle pochopit význam jednotlivých komponent a jejich parametrů.

Multimediální obsah má zásadní význam také v kontextu e-learningových platform. V kombinaci se SCORM balíčky lze videoobsah integrovat do struktury lekcí a navázat jej na testové otázky nebo praktické úkoly. To vede k vytvoření uzavřeného a logicky provázaného celku, který studentovi poskytuje soudržnou a ucelenou výukovou zkušenost.

Díky těmto vlastnostem se multimediální prvky stávají nepostradatelnou součástí výuky programování a vývoje her. Jejich zařazení do výukového modulu umožňuje efektivní prezentaci složitých technických postupů a přispívá ke zlepšení porozumění i celkové úspěšnosti studentů.

3.4 SCORM balíčky

SCORM (Sharable Content Object Reference Model) je standard, který umožňuje vytvářet e-learningové materiály kompatibilní s různými LMS systémy, včetně Moodle. SCORM balíčky obsahují obsah a logiku pro sledování aktivit uživatelů, což umožňuje monitorovat jejich pokroky v kurzu.



Obr. 9: SCORM

Zdroj: <https://www.learnworlds.com/scorm-guide/>

3.4.1 Vytváření SCORM balíčků

Kromě komerčních autorware nástrojů, jako jsou Articulate Storyline nebo Adobe Captivate, existují také otevřené a bezplatné platformy vhodné pro tvorbu SCORM balíčků. Jednou z nich je **exeLearning**, open-source editor určený pro vytváření interaktivních výukových modulů s podporou standardů **SCORM 1.2**, **SCORM 2004**, xAPI (TinCan) a dalších.

Nástroj exeLearning umožňuje:

- **Tvorbu navigační struktury kurzu**, která se automaticky přepisuje do SCORM organizačních souborů.
- **Vkládání textových, multimediálních a interaktivních prvků**, jako jsou kvízy, úkoly, obrázky či vložená videa.
- **Jednoduchý export do formátu ZIP**, který je ihned kompatibilní s většinou LMS systémů.
- **Úpravu metadat** SCORM balíčku včetně názvu, popisu, identifikátoru a jazykových informací.

Výhodou exeLearningu je jeho dostupnost, nízké nároky na systém a jednoduché ovládání, díky čemuž je vhodný pro vzdělávací instituce i pro individuální autory pedagogického obsahu. Program tak představuje efektivní nástroj pro tvorbu výukových materiálů, které lze snadno integrovat do prostředí LMS, jako je Moodle.

3.4.2 Integrace SCORM balíčků

Integrace SCORM balíčků do Moodle je poměrně jednoduchý proces, který je plně podporován nativními nástroji této platformy. Většinou probíhá následujícím postupem:

- **Nahrání SCORM balíčku:** Učitel nahraje .zip soubor obsahující SCORM modul přímo do kurzu prostřednictvím nabídky pro přidání nové aktivity.
- **Konfigurace nastavení:** V rámci Moodle lze určit počet povolených pokusů, způsob vyhodnocování, podmínky dokončení nebo viditelnost pro studenty.

- **Sledování a analýza:** Moodle automaticky zaznamenává výsledky studentů, jejich průchod aktivitami, čas strávený učením a úspěšnost v jednotlivých úkolech.

Díky těmto možnostem se SCORM stává efektivním způsobem distribuce interaktivního vzdělávacího obsahu, který je snadno spravovatelný, přenosný a vhodný pro široké spektrum vzdělávacích scénářů.

3.4.3 Alternativní e-learningové standardy (xAPI, H5P) a jejich porovnání se SCORM

Standard xAPI, známý také jako TinCan API, umožňuje sledovat širokou škálu aktivit probíhajících mimo klasické LMS prostředí. Na rozdíl od SCORM, který je omezen na interakce uvnitř kurzu, dokáže xAPI zaznamenávat i činnosti realizované v mobilních aplikacích, webových stránkách, VR simulacích nebo fyzickém prostředí. Díky tomu je vhodný pro komplexnější vzdělávací systémy, kde je nutné evidovat nestandardní formy učení. Nevýhodou je vyšší technická náročnost integrace a chybějící nativní podpora ve starších LMS.

Další alternativou je **H5P**, open-source nástroj pro tvorbu interaktivních aktivit, jako jsou videa, prezentace, simulace či interaktivní kvízy. Výhodou H5P je jeho jednoduché ovládání, dostupný editor ve webovém prohlížeči a široká podpora v moderních LMS systémech. Na rozdíl od SCORM však nenabízí tak detailní sledování průběhu studia a není primárně určen pro komplexní strukturování celých kurzů.

Ve srovnání s těmito nástroji zůstává SCORM vhodnou volbou pro situace, kdy je potřeba centrálně spravovat výukové moduly, sledovat výsledky studentů a zajistit kompatibilitu napříč různými platformami. Alternativní standardy rozšiřují možnosti e-learningu v oblastech, kde SCORM naráží na své limity, nicméně jejich použití závisí na konkrétních pedagogických cílech a technickém zázemí vzdělávací instituce.

3.5 Srovnání Moodle s dalšími LMS systémy

Moodle je jedním z nejpoblárnějších open-source LMS systémů, ale na trhu existují i další významní hráči, jako Google Classroom, Blackboard a Canvas. Moodle vyniká svou modularitou a možností přizpůsobení, což je důležité pro vzdělávací instituce s unikátními potřebami. Na druhou stranu systémy jako Google Classroom nabízejí jednodušší uživatelské rozhraní, které může být vhodnější pro menší školy nebo individuální učitele.

Canvas a Blackboard jsou často preferovány většími institucemi díky robustním funkcím pro správu obsahu a integraci s dalšími systémy. Tyto platformy však bývají licencované a finančně nákladnější ve srovnání s Moodle.

3.6 Kritéria výběru LMS pro vzdělávací účely

Při výběru LMS platformy hrají roli různé faktory, jako jsou náklady na implementaci, dostupné funkce, uživatelská přívětivost a podpora. Moodle je díky své bezplatné licenci oblíbený zejména ve vzdělávacích institucích, které hledají ekonomicky výhodné řešení. Na druhou stranu placené platformy mohou nabízet větší spolehlivost a profesionální technickou podporu.

3.7 Trendy a budoucnost LMS systémů

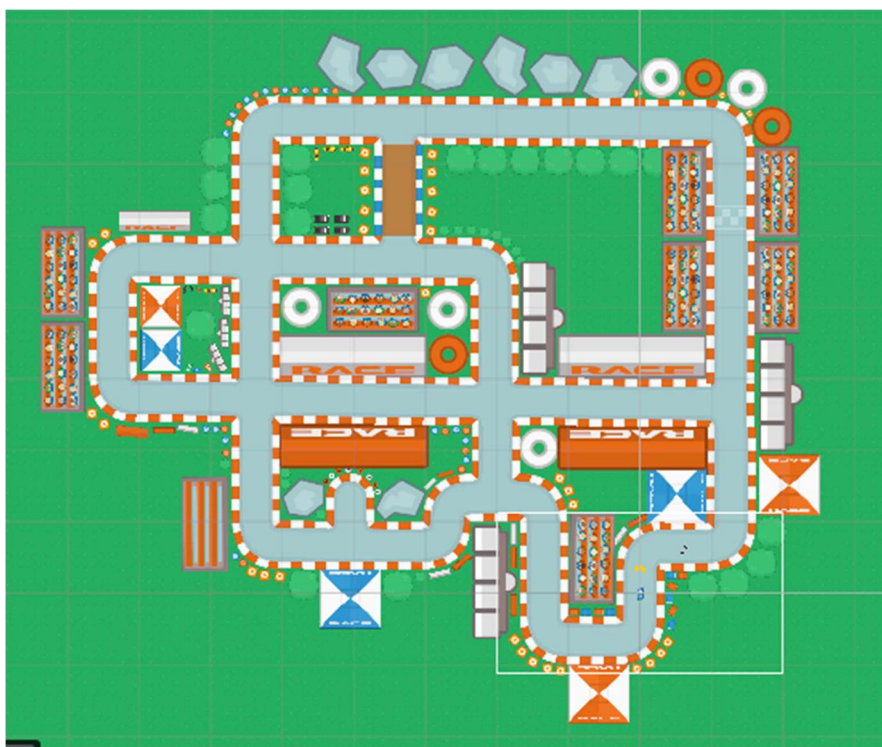
Budoucnost LMS systémů směřuje k větší integraci s pokročilými technologiemi, jako je umělá inteligence. AI může hrát klíčovou roli při personalizaci výuky, analýze dat o pokroku studentů a vytváření adaptivních kurzů. Další potenciál spočívá v integraci s VR a AR, které mohou studentům přinést zcela nové formy interaktivního vzdělávání.

4 Realizace výukového kurzu v Unity

Část se zaměřuje na návrh, tvorbu a implementaci výukových materiálů určených pro výuku základů vývoje her v prostředí Unity. Součástí realizace je návrh a vývoj demonstrační hry, tvorba doprovodných videí, výukových textů, praktických úkolů a testů. Tyto materiály jsou následně integrovány do LMS Moodle pomocí SCORM balíčku.

4.1 Návrh ukázkové hry

Vývoj ukázkové hry představoval úvodní krok praktické části projektu. Hra byla navržena jako jednoduchý 2D titul, který umožňuje demonstrovat základní principy herního vývoje v Unity pohyb hráče, fyziku objektů, interakce, sbírání předmětů, práci s uživatelským rozhraním a základní herní logiku. Hra byla navržena tak, aby odpovídala znalostem začínajících vývojářů, ale současně zahrnovala množství dílčích mechanik, na nichž lze ilustrovat fungování enginu i skriptování v C#. Základní princip hry spočívá v tom, že hráč ovládá malé vozidlo pohybující se po dvourozměrné mapě stylizované jako městská oblast. Úkolem hráče je sbírat zásilky rozmístěné na mapě a doručovat je náhodně generovaným zákazníkům. Hra obsahuje časový limit, během jehož trvání se hráč snaží doručit co nejvíce zásilek. Jednotlivé prvky prostředí byly vytvořeny jako jednoduché 2D sprity doplněné o kolizní komponenty. Návrh mapy i herní mechaniky vycházely z principů, které začátečníci zvládnou bez nutnosti pokročilých znalostí programování nebo grafiky, ale zároveň umožňují přirozeně přidávat další prvky.

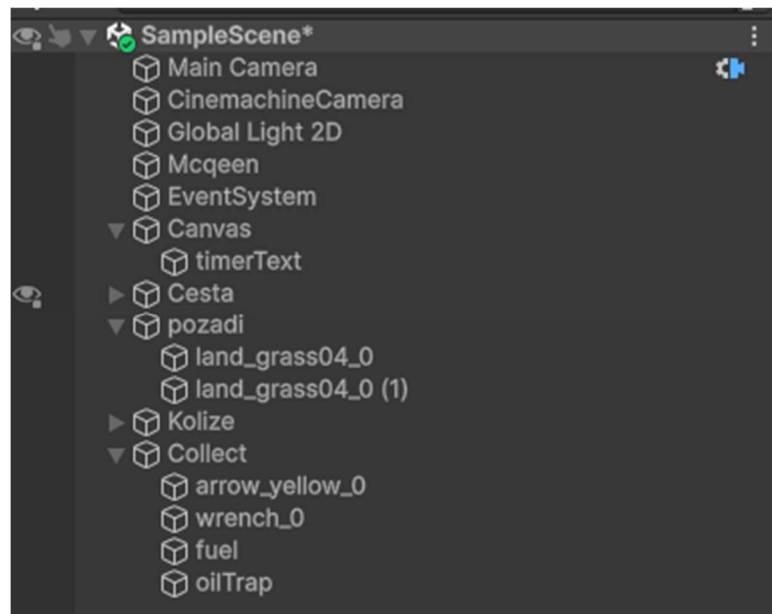


Obr. 10: Návrh mapy

Zdroj: Vlastní zpracování

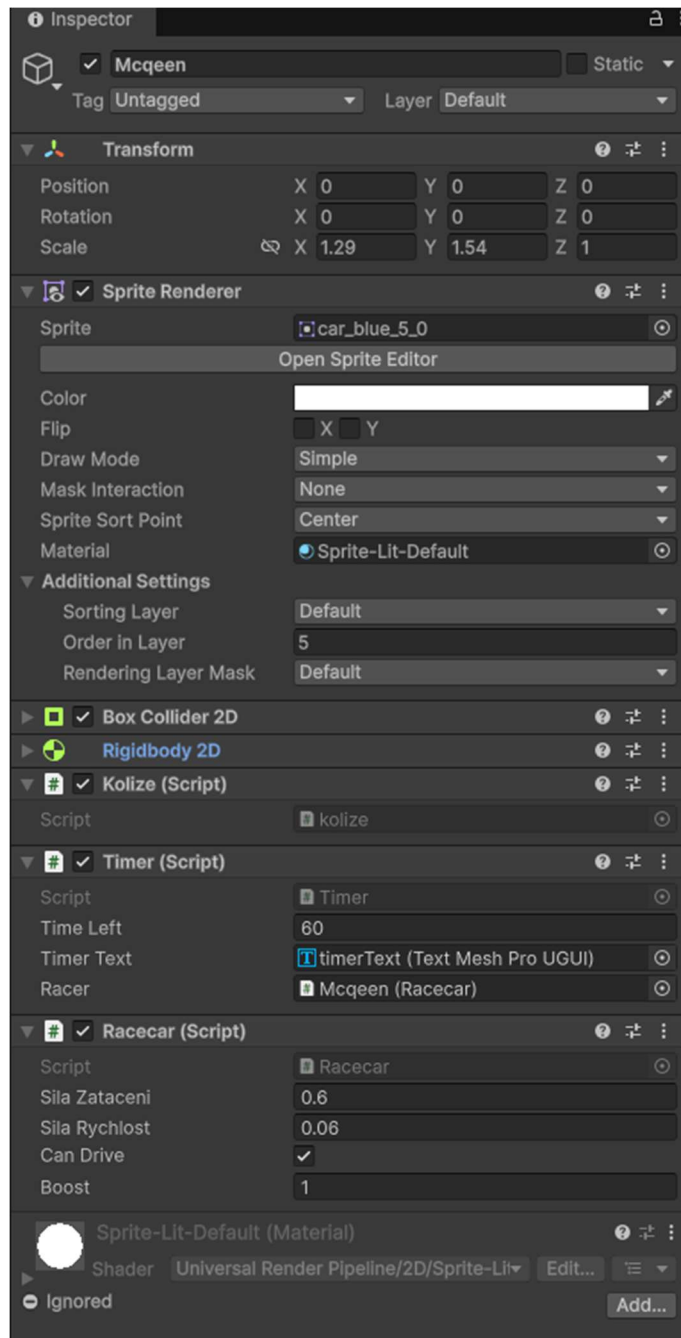
4.2 Implementace herní logiky v Unity

Implementace herního prostředí probíhala ve více etapách. Nejprve byla vytvořena základní scéna, která obsahovala pozadí, kolizní objekty představující hranice mapy a statické dekorace. Následně byly přidány interaktivní prvky, jako jsou zásilky, zákazníci a posilující objekty. Celé prostředí funguje na principu 2D fyziky Unity, která poskytuje dostatečný základ pro simulaci jednoduchých kolizí a pohybů. Jednotlivé objekty jsou rozděleny podle funkcí a označeny tagy, což usnadňuje jejich identifikaci ve skriptech a umožňuje přehledné řízení herní logiky. Struktura scény byla navržena tak, aby byla pro studenty snadno pochopitelná. Hierarchie obsahuje hlavní objekt hráče, prefaby sběrných objektů, prvek generátoru a samostatnou vrstvu pro uživatelské rozhraní. Každý objekt má přiřazen odpovídající skript, který zajišťuje jeho funkčnost. Tento modularizovaný přístup umožňuje začínajícím vývojářům dobře pochopit, jak jednotlivé části hry spolupracují a jak lze pomocí komponent a skriptů vytvářet interaktivní chování.



Obr. 11: Hierarchie projektu

Zdroj: Vlastní zpracování



Obr. 12: Inspector objektu Mcqeen

Zdroj: Vlastní zpracování

4.2.1 Pohyb hráče

Implementace pohybu hráče je jedním z klíčových prvků celé hry, protože interakce hráče s prostředím tvoří hlavní dynamiku projektu. Pohyb je realizován skriptem Racecar.cs, který pracuje s uživatelským vstupem z klávesnice. Skript vyhodnocuje dvě základní akce pohyb vpřed/vzad a otáčení vozidla kolem svislé osy. Pohyb i rotace jsou propojeny s proměnnými určujícími rychlost a citlivost řízení, které lze snadno upravit v Inspectoru díky použití atributu SerializeField. Součástí mechaniky pohybu je i proměnná boost (dočasné zrychlení), která dočasně zvyšuje rychlost vozidla. Pokud hráč narazí do překážky, vozidlo se zpomalí

na předdefinovanou hodnotu a hráč musí najít opravářský klíč, aby se rychlost navrátila do normálu.

Mechanika ukazuje propojení řízení vstupu, fyziky a herního stavu. Logika je pro studenty snadno rozšiřitelná například o další stavy hráče nebo různé typy poškození.

```
void Update()
{
    if(!canDrive) return;
    float zatacení = 0f;
    float jet = 0f;

    if (Keyboard.current.wKey.isPressed)
    {
        jet = 1f;
    }
    else if (Keyboard.current.sKey.isPressed)
    {
        jet = -1f;
    }
    if (Keyboard.current.aKey.isPressed)
    {
        zatacení = 1f;
    }
    else if (Keyboard.current.dKey.isPressed)
    {
        zatacení = -1f;
    }

    transform.Rotate(0,0, zatacení * silaZatacení);
    transform.Translate(0, jet * silaRychlost * boost, 0);
}
```

4.2.2 Kolizní systém: náraz a oprava vozidla

Kolizní mechaniky tvoří další důležitou část projektu. V případě nárazu vozidla do překážky se modifikuje rychlost pohybu a hráč je penalizován dočasným zpomalením. Tím se simuluje jednoduchá mechanika poškození vozidla. Součástí systému je také objekt reprezentující nástroj pro opravu (wrench), který umožní vrátit vozidlo zpět do původního stavu.

Část projektu je vhodná pro demonstraci využití kolizních událostí v Unity, práce s komponentou Collider2D a podmíněného vyhodnocení stavu objektů. Studenti si na tomto příkladu mohou snadno osvojit způsob, jakým herní logika reaguje na události, a jak je možné upravovat atributy hráče na základě interakcí ve scéně.

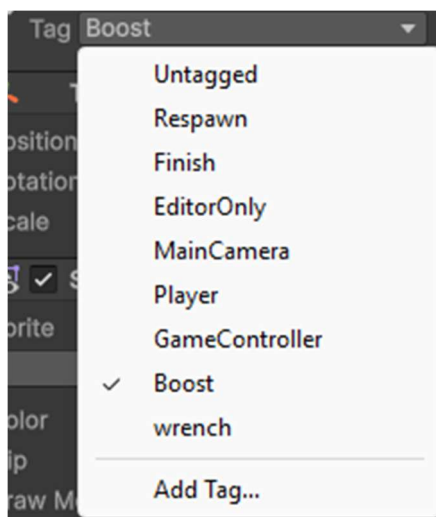
```
if (collision.CompareTag("Wrench") && boost < 1)
{
    Debug.Log("OPRAVENO!");
    boost = 1f;
}
```

```

        Destroy(collision.gameObject, 0.15f);
        AudioSource.PlayOneShot(repairUp);
    }
    else if (collision.CompareTag("Wrench"))
    {
        Debug.Log("Nepotřebuješ opravu.");
    }
}

void OnCollisionEnter2D(Collision2D collision)
{
    Debug.Log("Naboural jsi!");
    boost = 0.3f;
    AudioSource.PlayOneShot(crash);
}

```



Obr. 13: Ukázka karty Tag

Zdroj: Vlastní zpracování

4.2.3 Mechanika dočasného zrychlení (Boost)

Další implementovanou funkcí je systém dočasného zrychlení, který reaguje na sbírání speciálního herního objektu. Po jeho aktivaci se na určitou dobu zvýší rychlost pohybu vozidla, čímž hráč získá taktické zvýhodnění. Mechanika je vhodná pro práce s proměnnými, které ovlivňují herní chování, a také umožňuje studentům pochopit princip podmíněného zpracování událostí.

Boost je koncipován tak, aby nebyl dostupný v momentě, kdy je vozidlo poškozené. Tím se vytváří jednoduchá herní rovnováha — hráč si musí nejprve opravit vozidlo, než může využít další posílení rychlosti. Implementace této funkce představuje základní ukádku propojení herních objektů, kolizí a logiky řízení stavu hráče.

```

if (collision.CompareTag("Boost"))
{
    if (boost == 1 || boost > 1)

```

```

    {
        Debug.Log("BOOST!");
        boost = 1.5f;
        Destroy(collision.gameObject, 0.15f);
        audioSource.PlayOneShot(speedUp);
    }
    else
    {
        Debug.Log("Musíš prvně opravit auto!");
    }
}
}

```

4.2.4 Systém generace objektů

Systém generace objektů patří mezi vizuálně nenápadné, ale logicky důležité části projektu. Jeho úkolem je zajistit, aby se ve hře vyskytovala vždy pouze jedna zásilka a jeden zákazník. Skript Spawner.cs (generátor) pracuje s poli bodů ve scéně, na kterých je možné nové objekty generovat. Po doručení zásilky zákazníkovi generátor vytvoří nový pár objektů na náhodně vybraných pozicích. Tím vzniká cyklus, který udržuje hru neustále aktivní a hráče motivuje k dalším akcím.

Systém je navržen modulárně, počet generačních míst lze snadno rozšířit pouhým přidáním nového prázdného objektu do scény. Stejně jednoduché je i přidání nových typů předmětů.

```

public Transform[] packageSpawnPoints;
public Transform[] customerSpawnPoints;

GameObject currentPackage;
GameObject currentCustomer;

void Start()
{
    SpawnNewPair(); // Spawn první dvojice
}

void SpawnNewPair()
{
    SpawnNewPackage();
    SpawnNewCustomer();
}

public void PackageCollected()
{
    // Zásilka byla sebrána → smažeme ji
    if (currentPackage != null)
        Destroy(currentPackage);
}

public void PackageDelivered()

```

```

{
    // Po doručení smažeme zákazníka a spawnujeme novou dvojici
    if (currentCustomer != null)
        Destroy(currentCustomer);

    SpawnNewPair();
}

void SpawnNewPackage()
{
    int index = Random.Range(0, packageSpawnPoints.Length);
    currentPackage = Instantiate(packagePrefab,
packageSpawnPoints[index].position, Quaternion.identity);
}

void SpawnNewCustomer()
{
    int index = Random.Range(0, customerSpawnPoints.Length);
    currentCustomer = Instantiate(customerPrefab,
customerSpawnPoints[index].position, Quaternion.identity);
}
}

```

4.2.5 Uživatelské rozhraní (UI)

Uživatelské rozhraní tvoří důležitou součást celkové prezentace hry a poskytuje hráči informace o aktuálním stavu herního průběhu. V projektu je UI realizováno pomocí TMP - TextMeshPro prvků, které zajišťují moderní vzhled textu a jeho dobrou čitelnost. Základními prvky jsou časovač, zobrazení skóre a panel indikující konec hry.

Časovač je řízen skriptem Timer.cs, který v každém snímku snižuje hodnotu zbývajících času. Jakmile čas dosáhne nuly, skript deaktivuje ovládání hráče, zobrazí finální skóre a aktivuje grafický panel s informací o ukončení hry. Část projektu představuje ideální ukázkou propojení UI, skriptování a herního stavu. Studenti zde mohou snadno pochopit, jak se UI aktualizuje v reálném čase a jak lze propojit grafické prvky s logikou hry.

```

public class Timer : MonoBehaviour
{
    public float timeLeft = 60f; // Počet vteřin
    public TMP_Text timerText;
    public Racecar racer;
    // Update is called once per frame
    void Update()
    {
        timeLeft -= Time.deltaTime;

        if(timeLeft <= 0)
        {
            timeLeft = 0;

```

```

        if(racer != null) racer.canDrive = false;
    }
    if(timerText != null)
    {
        timerText.text = "Time: " + Mathf.Ceil(timeLeft).ToString();
    }
}
}

```

4.3 Implementace zvukových prvků

Zvuková složka představuje důležitou část herního zážitku, která podporuje orientaci hráče a poskytuje mu okamžitou zpětnou vazbu na jeho akce. V rámci praktické části byla realizována jednoduchá, ale funkční audio implementace využívající komponentu AudioSource, která je

v Unity určena k přehrávání zvukových souborů typu AudioClip. Přestože se projekt zaměřuje primárně na výuku základních principů herního vývoje, zvukové prvky zde slouží jako ukázka toho, jak lze propojit herní logiku se zvukovou odezvou.

Základ audio implementace je připojen ke skriptu časovače, který řídí konec hry. Jakmile časomíra dosáhne nuly, hra hráči signalizuje ukončení partie zvukovým efektem, který je přehrán prostřednictvím příkazu PlayOneShot. Tento způsob je vhodný pro jednorázové zvuky, jelikož neovlivňuje případnou hudbu na pozadí a umožňuje nezávislé přehrávání efektů bez nutnosti jejich slučování.

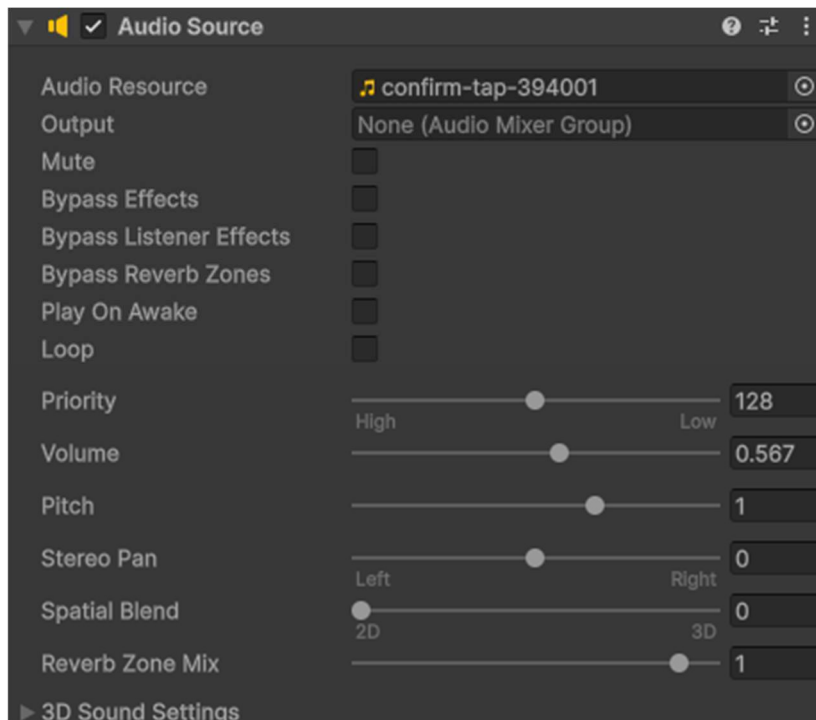
V rámci projektu bylo použito především 2D audio, které není ovlivněno pozicí objektů ve scéně. To zajišťuje konzistentní hlasitost zvuků bez ohledu na to, kde se hráč na mapě aktuálně nachází. Volba je vhodná zejména pro 2D hry, kde není klíčové prostorové rozmístění objektů ani realističnost zvukového prostředí. Připojení zvukových efektů tak slouží primárně jako didaktická demonstrace principu, nikoli jako plnohodnotný sound design.

```

public AudioSource audioSource;           // Audio Source na autě
public AudioClip pickUp;                  // Zvuk při sebrání zásilky
public AudioClip dropIn;                  // Zvuk při doručení

void OnTriggerEnter2D(Collider2D collision)
{
    audioSource.PlayOneShot(pickUp);
    audioSource.PlayOneShot(dropIn);
}

```



Obr. 14: Zvuková komponenta

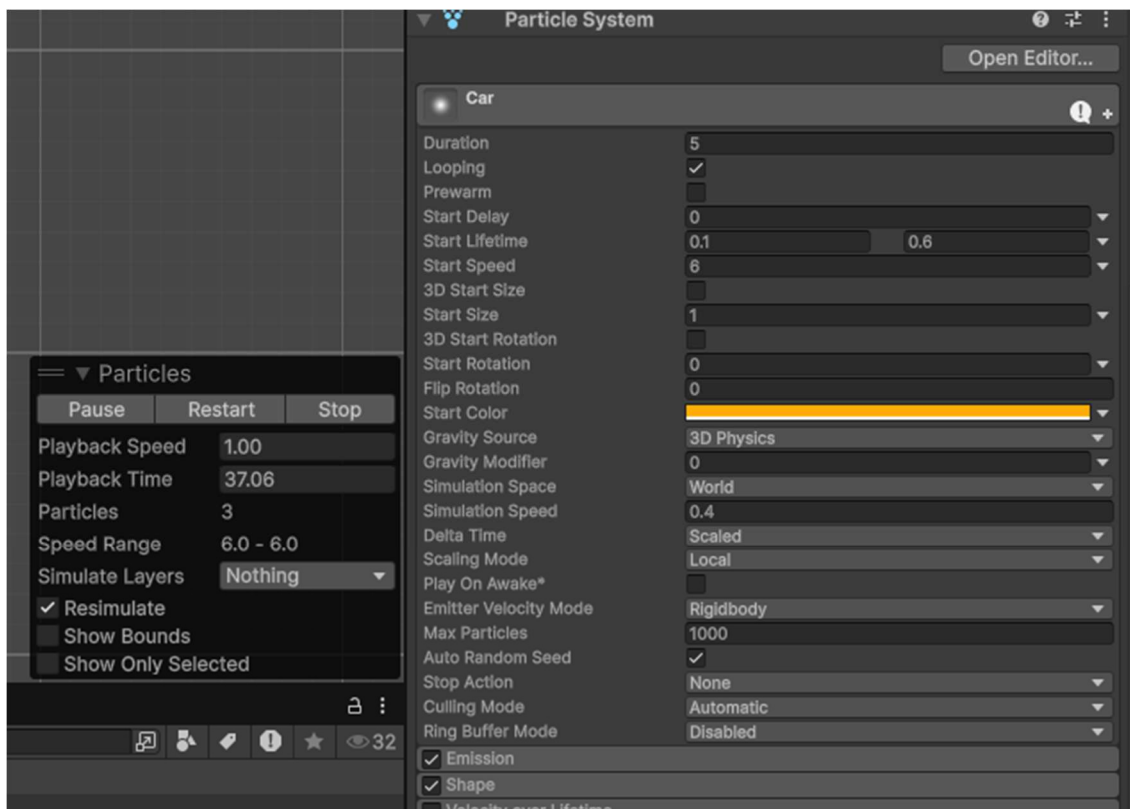
Zdroj: Vlastní zpracování

4.4 Implementace částicových efektů (Particle System)

Částicové efekty (particle effects) představují vizuální prvek, který dokáže výrazně oživit herní prostředí a podpořit vizuální odezvu na různé hráčské akce. Unity obsahuje vlastní systém částic s názvem Particle System, který umožňuje vytvářet efekty, jako je prach, kouř, jiskry, magické animace nebo signální zvýraznění herních událostí.

V rámci ukázkového projektu byl částicový efekt využit zejména pro vizuální zvýraznění okamžiku, kdy hráč sebere zásilku. Po interakci s příslušným objektem je aktivován krátký efekt, který slouží jako vizuální potvrzení úspěšného pickupu. Integrace má primárně ilustrační význam a demonstruje studentům způsob, jakým lze v Unity kombinovat vizuální efekty s herní logikou.

Particle System funguje jako samostatná komponenta připojená k hernímu objektu. Je možné jej spouštět skrze skript pomocí metod jako `Play()` nebo `Stop()`, což umožňuje přesné řízení efektů podle stavu hry. Díky tomu lze efekty navázat nejen na sbírání objektů, ale také na další události, jako je náraz, oprava vozidla či dokončení úkolu.



Obr. 15: Particle system Component

Zdroj: Vlastní zpracování

4.5 Možnosti dalšího rozvoje herního projektu

Přestože výsledná hra splňuje všechny pedagogické cíle stanovené na začátku práce, nabízí řadu možností dalšího rozšíření. Rozšíření mohou posloužit jednak jako inspirace pro studenty, kteří chtějí projekt zdokonalit, a jednak jako základ pro výuku pokročilejších témat v oblasti herního vývoje.

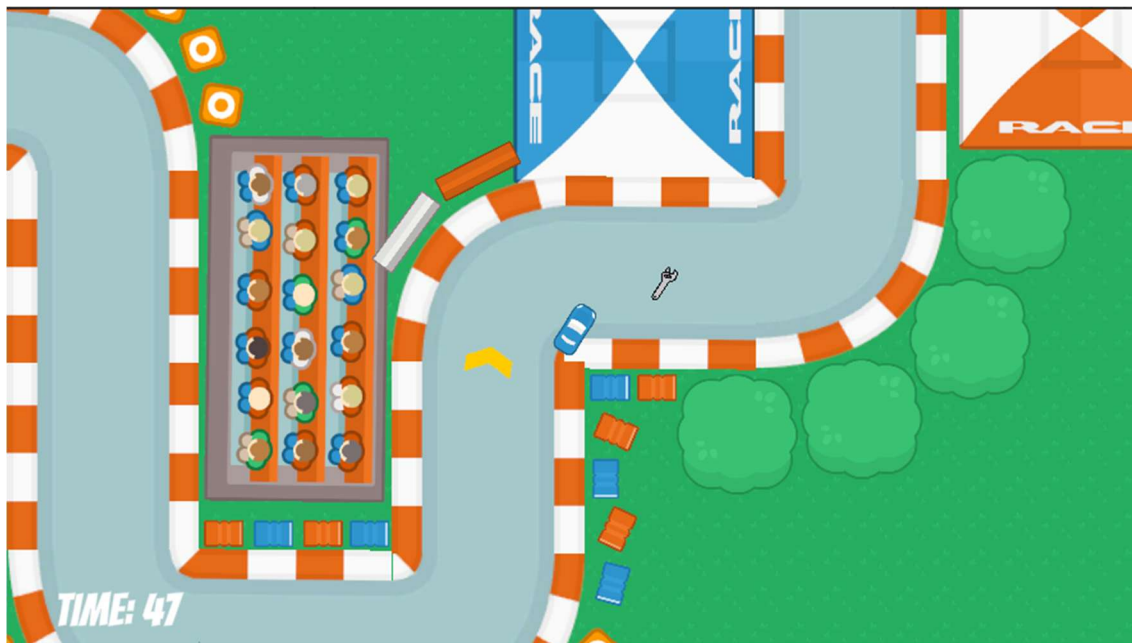
Za přirozené rozšíření lze považovat doplnění animací pro pohyb vozidla, zákazníků nebo interaktivních objektů. Unity nabízí širokou podporu pro animátory, kombinaci animací či použití přechodových stavů, což by studentům umožnilo pochopit další důležitou oblast vývoje her.

Další možností je implementace **více paralelních úkolů**, kdy hráč může současně nést více zásilek nebo se rozhodovat mezi několika zákazníky. Tím by vznikla komplexnější herní logika, která by zahrnovala plánování trasy, časový management či systém priorit.

Za zmínku také stojí rozšíření UI prvků o **mini-mapu**, ukazatele vzdálenosti, upozornění nebo grafické efekty reagující na stav vozidla. Pro výuku práce s uživatelským rozhraním jde o velmi vhodnou příležitost.

Hru lze dále doplnit o systém **bodování s penalizací**, různě hodnocené typy zásilek, dynamické překážky nebo časové výzvy. Tyto mechaniky by umožnily aplikaci principů herního designu a zvýšily variabilitu výsledného herního zážitku.

V neposlední řadě lze projekt rozšířit o základní ukládání dat pomocí **PlayerPrefs**, serializace nebo jednoduché databáze, čímž by se studenti seznámili s persistencí dat ve hrách.



Obr. 16: Náhled finálního projektu

Zdroj: Vlastní zpracování

4.6 Tvorba SCORM balíčku v nástroji exeLearning

Pro finalizaci výukových materiálů a jejich následné vložení do LMS Moodle byl použit nástroj **exeLearning**, který umožňuje vytvářet plně strukturovaný SCORM balíček bez nutnosti využívat komerční software. Proces tvorby modulu probíhal v několika etapách.

Nejprve byla vytvořena **hierarchická struktura kurzu**, odpovídající jednotlivým kapitolám a tematickým celkům výukového modulu. Každá kapitola získala vlastní obsahovou stránku, do které byly postupně vloženy texty, obrázky, kvízové prvky a další doplňkový materiál. Díky modulárnímu přístupu bylo možné flexibilně rozšiřovat a upravovat obsah podle potřeby.

Dále byly přidány **interaktivní komponenty**, zejména kvízy znalostí a praktické úkoly související s vývojem hry v Unity. exeLearning poskytuje integrované prvky pro tvorbu otázek typu výběr z možností, krátké odpovědi či reflexivní otázky. Tyto prvky jsou plně kompatibilní se SCORM režimem sledování výsledků.

Po dokončení obsahu byl kurz exportován ve formátu **SCORM 1.2**, který zajišťuje kompatibilitu s většinou vzdělávacích platform. Export vygeneroval komprimovaný ZIP soubor obsahující všechny potřebné soubory, metadata a organizační strukturu definovanou standardem SCORM.

Takto vytvořený SCORM balíček byl následně nahrán do prostředí Moodle, kde byl otestován a ověřena jeho kompatibilita. Balíček byl schopen správně předávat údaje o pokroku studenta, úspěšnosti i dokončení jednotlivých aktivit, čímž splnil požadavky na plně funkční e-learningový modul.

4.7 Tvorba výukových videí

Součástí výukového modulu je sada instruktážních videí, která doplňují textové materiály a poskytují vizuální demonstraci jednotlivých kroků vývoje hry. Každé video se zaměřuje na konkrétní část procesu od práce se scénou přes skriptování až po závěrečné ladění hry. Videá byla vytvořena s důrazem na postupné tempo, jasný výklad a názornost, aby byla snadno pochopitelná i pro studenty bez předchozí zkušenosti s Unity.

Při tvorbě videí bylo dbáno na to, aby jednotlivé kapitoly navazovaly na obsah výukových textů a SCORM balíčku. Vícekanálová forma výuky zvyšuje srozumitelnost a umožňuje studentovi orientovat se podle toho, zda preferuje vizuální nebo textový styl učení. Každé video končí krátkým shrnutím, které zdůrazňuje nejdůležitější poznatky z dané části vývoje.

4.8 Vytvoření výukových textů

Výukové texty tvoří základní strukturu celého SCORM balíčku. Jednotlivé kapitoly jsou psány formálním jazykem a obsahují teoretické vysvětlení, ukázky kódu, ilustrace prostředí Unity a krátká cvičení. Texty byly koncipovány tak, aby byla zachována maximální srozumitelnost, a zároveň aby reflektovaly logický postup vývoje hry. Každá kapitola odpovídá konkrétnímu tématu z instruktážního videa, což studentovi umožňuje přecházet mezi textem, videem a praktickým úkolem.

Výukové texty také obsahují kontrolní otázky, které slouží jako rychlá verifikace pochopení látky. Student si tak může okamžitě ověřit, zda porozuměl základním pojmům a principům Unity, aniž by musel přecházet do závěrečného testu.

4.9 Návrh praktických úkolů

Praktické úkoly byly navrženy tak, aby studentům umožnily aktivně procvičit probírané principy. Úkoly mají stoupající obtížnost a odrážejí témata jednotlivých kapitol kurzu. Zpočátku jsou zaměřeny na jednoduché akce, jako je vytvoření objektu či přidání komponenty, později se orientují na skriptování, interakci objektů a práci s uživatelským rozhraním. Tento systematický přístup umožňuje studentům posilovat praktické dovednosti krok za krokem a zároveň imituje reálný postup vývoje menší hry.

Úkoly byly optimalizovány tak, aby podporovaly samostatnost studenta zadání neobsahuje přímé řešení, ale pouze popis cílového stavu. Pokud student neví, jak úkol provést, může se vrátit k příslušnému videu nebo textu, čímž se podporuje typické iterativní učení.

4.10 Tvorba testů

Testy umístěné na závěr jednotlivých kapitol slouží k rychlému ověření znalostí. Obsahují otázky zaměřené na pochopení základních pojmů, jako jsou komponenty, tagy, prefaby, skriptovací metody nebo průběh herního cyklu. V testech jsou použity různé typy otázek: otevřené, výběrové i doplňovací. Jejich cílem není hodnotit studenta formálně, ale poskytnout mu okamžitou zpětnou vazbu a pomoci identifikovat oblasti, ve kterých má nedostatky.

Testy byly koncipovány tak, aby byly stručné a přehledné. Každý z nich je úzce navázán na obsah konkrétní kapitoly, což umožňuje studentovi ověřit své znalosti postupně, nikoliv až na konci celého kurzu. Kurz zahrnuje závěrečný test, který slouží k rekapitulaci a ověření, zda student opravdu nabral znalosti užitečné k vývoji herních mechanik.

4.11 Hostování výukového kurzu v prostředí Moodle

Pro zpřístupnění vytvořeného výukového obsahu studentům byl využit cloudový hosting platformy **Moodle**, který umožňuje kompletní správu kurzů bez nutnosti lokální instalace. Tento přístup poskytl rychlé, flexibilní a infrastrukturně nenáročné řešení umožňující plnohodnotné nasazení výukového modulu.

V rámci prostředí Moodle byl vytvořen nový kurz, do kterého byly nahrány jednotlivé výukové prvky včetně SCORM balíčku vytvořeného v exeLearningu, externích souborů, výukových PDF a dalších podpůrných materiálů. Následně byla nastavena struktura kurzu, viditelnost jednotlivých částí, přístupy uživatelů a způsob vyhodnocování.

Hostované prostředí umožnilo také provést testovací provoz, během něhož byl ověřen správný chod SCORM balíčku, zda se záznamy o aktivitách studentů korektně ukládají a zda jsou videa a další dokumenty plně přístupné. Výsledkem bylo stabilní a funkční nasazení výukového kurzu, které splňuje požadavky na moderní e-learningové prostředí.

4.12 Vizualizace a pedagogická struktura výukového kurzu

Výukový kurz vytvořený v rámci této práce je navržen tak, aby odpovídal principům postupného a srozumitelného učení. Obsah je uspořádán do kapitol podle logického sledu vývoje hry v Unity od základního seznámení s prostředím až po implementaci jednotlivých herních mechanik. Jednotlivé části vycházejí z reálného vývojového procesu a na sebe přirozeně navazují, což studentům umožňuje porozumět nejen jednotlivým prvkům, ale i jejich vzájemným souvislostem.

Kurz je koncipován tak, aby podporoval tzv. postupné budování znalostí. Úvodní kapitoly mají za úkol studenty uvést do problematiky, zatímco pozdější části se věnují složitějším principům skriptování, uživatelskému rozhraní či správě objektů ve scéně. Díky této struktuře lze nové informace lépe pochopit a ukotvit v kontextu dříve probraného obsahu.

Důležitou roli v kurzu hrají také vizuální prvky, které doprovázejí jednotlivé výkladové pasáže. Obrázky prostředí Unity, ukázky kódu či zachycení herních objektů pomáhají studentům snáze pochopit prezentované postupy, jelikož propojují text s praktickými ukázkami. Tyto vizuální pomůcky podporují lepší orientaci v prostředí herního engine a zvyšují názornost celého kurzu.

Vývoj 2D her v Unity (praktický kurz pro začátečníky)

Kurz Nastavení Účastníci Známky Činnosti Další ▾

Úvod Sbalit vše

Announcements

Všechny podklady | SCROM

Kapitola 2: Instalace a prostředí Unity

Kapitola 2

Kapitola 3: Základy programování v Unity

Kapitola 3

Kapitola 4: Ovládání objektů a pohyb v 2D

Kapitola 5: Kolize a fyzikální interakce

Obr. 17: Zobrazení kurzu

Zdroj: Vlastní zpracování

UnityBalíček

- Kapitola 1: Úvod
- Kapitola 2: Instalace a prostředí
 - Test - Kapitola 2
- Kapitola 3: Základy programování v C# v Unity
- Kapitola 4: Ovládání objektů a pohybu v 2D
- Kapitola 5: Kolize a fyzikální interakce v 2D
- Kapitola 6: Prefaby a opakovatelně použitel...
 - Test - Kapitola 6
- Kapitola 7: Fyzika, Collider a RigidBody v U...
 - Test - Kapitola 7
- Kapitola 8: Tags, Layer a organizace objektů ve...
- Kapitola 9: Cinemachine a práce s kamerou
- Kapitola 10: Assety a práce s texturami
- Kapitola 11: Tvorba Mapy, kolozní bloky a leve...
- Kapitola 12: Spawner a dynamické vytváření o...
- Kapitola 13: UI, Canvas, Texty a herní informace
- Kapitola 14: Audiosystém, zvukové efekty a hu...
- Kapitola 15: Dokončení herní logiky, Game ove...
- Závěrečný test

UnityBalíček

Kapitola 1: Úvod

Kapitola 1 – Úvod do vývoje her

Cíl kapitoly
Student se seznámí se základními pojmy a principy vývoje počt Unity.

1.1 Co je vývoj počítačových her
Vývoj počítačových her představuje interdisciplinární oblast, kte inteligence a herní dramaturgie. Moderní vývoj her probíhá zpr vytvářet 2D i 3D scény, řídit chování obíektů.

Obr. 18: Obsah kurzu

Zdroj: Vlastní zpracování

4.13 Uživatelské testování a zpětná vazba

Součástí vývoje výukového modulu bylo také jeho pilotní otestování, jehož cílem bylo ověřit funkčnost, přehlednost a pedagogickou účinnost vytvořených materiálů. Testování proběhlo s několika uživateli, kteří představovali typickou cílovou skupinu osoby se základními znalostmi programování, ale omezenou zkušeností s vývojem her v Unity. Tento výběr testujících umožnil získat zpětnou vazbu relevantní pro vzdělávací účely.

Testování se zaměřovalo na více oblastí současně:

- **Technickou funkčnost hry**
- **Jasnost a srozumitelnost výukových materiálů**
- **Funkčnost SCORM balíčku v prostředí Moodle**
- **Logickou návaznost kapitol**
- **Pochopitelnost videí a textových materiálů**

Testující potvrdili, že samotná hra je intuitivní a jednotlivé herní mechaniky jsou pochopitelné již po krátkém seznámení. Pozitivně byla hodnocena také přehlednost uživatelského rozhraní, jednoduchá vizuální stylizace a jednoznačné cíle jednotlivých úkolů. Studenti uvedli, že hra dobře demonstrovuje vyučované koncepty například práci s komponentami, kolizemi, generování objektů nebo načasováním událostí.

V rámci Moodle byl testován především průběh SCORM modulu, přehrávání videí a fungování testových otázek. Testování potvrdilo, že balíček je kompatibilní s LMS, korektně ukládá výsledky studenta a umožňuje návrat k jednotlivým kapitolám. Uživatelé ocenili také to, že každá kapitola obsahuje jasně vymezený cíl i závěrečné shrnutí, což usnadňuje orientaci v obsahu.

Zpětná vazba přinesla také několik podnětů pro možné rozšíření. Mezi návrhy se objevila možnost doplnění rozšířených kvízových otázek, přidání příkladů chybových stavů ve skriptech a jejich řešení nebo rozšíření videí o část, která by ukazovala testování vytvořené hry v různých scénářích.

Pilotní testování potvrdilo, že výukový modul je funkční, srozumitelný a didakticky účinný. Získané připomínky byly využity k dílčím úpravám a zároveň představují cennou inspiraci pro budoucí verze kurzu.

5 Zhodnocení výsledků

Výsledkem praktické části je kompletní výukový modul, který propojuje teoretické poznatky o herním vývoji se skutečnou tvorbou funkční 2D hry. Vytvořené výukové materiály zahrnují texty, videa, praktické úkoly, testy a finální SCORM balíček, který lze snadno integrovat do LMS Moodle. Hra samotná slouží jako ukázkový projekt, na němž lze demonstrovat základní principy práce v Unity. Její vývoj poskytuje studentům příležitost pozorovat reálný proces tvorby hry od návrhu až po finální implementaci.

Výukový kurz podporuje propojení teorie a praxe, což je zásadní pro efektivní osvojení nových dovedností. Výsledný materiál je ucelený, konzistentní a flexibilní v budoucnu jej lze rozšiřovat o pokročilejší techniky, nové kapitoly nebo další herní mechaniky.

Vytvořený výukový kurz byl pilotně otestován malou skupinou uživatelů z cílové skupiny, kteří měli za úkol projít jednotlivé části kurzu, zhlédnout multimediální obsah a otestovat interaktivní prvky. Pilotní testování bylo zaměřeno na ověření srozumitelnosti, technické funkčnosti SCORM balíčku a plynulosti struktury kurzu. Získaná zpětná vazba potvrdila funkčnost všech aktivit a zároveň poskytla podněty pro drobná vylepšení, zejména v oblasti vizuální orientace v jednotlivých lekcích.

6 Závěr

Bakalářská práce se zabývala tvorbou pedagogického materiálu pro výuku vývoje počítačových her, jeho implementací do LMS Moodle a využitím SCORM balíčků. Vývoj her byl představen jako disciplína, která spojuje technické a kreativní aspekty. Důraz byl kladen na praktické využití nástrojů, jako je Unity, a na základní principy herního designu. Práce ukázala přínosy e-learningových platforem, jako je Moodle, a jejich schopnost zlepšit správu vzdělávacích materiálů a sledování pokroků studentů. Další výzkum by mohl rozšířit možnosti využití technologií, jako je virtuální nebo rozšířená realita, a jejich propojení s herním vývojem, čímž by se zvýšila interaktivita a atraktivita vzdělávacích kurzů.

Seznam použité literatury

- ADVANCED DISTRIBUTED LEARNING INITIATIVE. SCORM 101: A Guide for Beginners. Online. 2020. Dostupné z: <https://adlnet.gov/scorm/>. [cit. 2025-01-10].
- BISHOP, Richard. Introduction to Computer Graphics with Unity. CRC Press. Online. Dostupné z: <https://www.crcpress.com/>. [cit. 2025-01-10].
- BRO CODE, C# Full Course for free (YouTube video). 2024. Online. Dostupné z: <https://www.youtube.com/watch?v=wxznTygnRfQ>. [cit. 2025-01-10].
- CHANDLER, Heather Maxwell. The Game Production Handbook. 3rd Edition. Jones & Bartlett Learning. Online. Dostupné z: <https://jblearning.com/>. [cit. 2025-01-10].
- CLARK, Ruth Colvin. Developing Technical Training: A Structured Approach. Wiley. Online. Dostupné z: <https://www.wiley.com/>. [cit. 2025-01-10].
- COMPUTERS & EDUCATION JOURNAL. Case Studies in Game-Based Learning for Programming Education. Online. 2021. Dostupné z: <https://www.journals.elsevier.com/computers-and-education>. [cit. 2025-01-10].
- DOUGIAMAS, Martin. Moodle: A Case Study in Open-Source Learning Management. EdTech Press. Online. Dostupné z: <https://moodle.org/>. [cit. 2025-01-10].
- HOLM, Jacob. Practical C# Scripting for Unity Game Development: A Didactic Approach. CRC Press. Online. 2023. Dostupné z: <https://www.crcpress.com/>. [cit. 2025-01-10].
- O'NEILL, Michael. E-Learning in the 21st Century: Frameworks and Applications. Routledge. Online. Dostupné z: <https://www.routledge.com/>. [cit. 2025-01-10].
- RESNICK, Mitchel. Scratch Programming for Young People. MIT Press. Online. Dostupné z: <https://www.scratch.mit.edu/>. [cit. 2025-01-10].
- SHELL, Jesse. The Art of Game Design: A Book of Lenses. 3rd Edition. CRC Press. Online. Dostupné z: <https://artofgamedesign.com/>. [cit. 2025-01-10].
- SMART, John. Using SCORM and Interactive Multimedia Content to Enhance Remote Learning Effectiveness. *Journal of Educational Technology Systems*. Online. 2020. Dostupné z: <https://journals.sagepub.com/home/ets>. [cit. 2025-01-10].
- UNITY TECHNOLOGIES. Unity Documentation. Online. Dostupné z: <https://docs.unity3d.com/>. [cit. 2025-01-10].
- UNITY TECHNOLOGIES. Unity Learn. Online vzdělávací platforma. Online. Dostupné z: <https://learn.unity.com/>. [cit. 2025-01-10].
- UNITY TECHNOLOGIES. Unity Learn Tutorials. Online. Dostupné z: <https://learn.unity.com/tutorials>. [cit. 2025-01-10].